Biometric Cryptosystems:

Authentication, Encryption and Signature for Biometric Identities

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Neyire Deniz Sarıer

aus

Istanbul, Türkei

Bonn 2011

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Erstgutachter-Betreuer: Prof. Dr. Joachim von zur Gathen, (b-it, Universität Bonn)
Zweitgutachter: Prof. Dr. Preda Mihailescu, (Georg-August-Universität Göttingen)
Tag der Promotion: 08.04.2013
Erscheinungsjahr: 2013

Abstract (Zusammenfassung)

Biometrics have been used for secure identification and authentication for more than two decades since biometric data is unique, non-transferable, unforgettable, and always with us. Recently, biometrics has pervaded other aspects of security applications that can be listed under the topic of "Biometric Cryptosystems". Although the security of some of these systems is questionable when they are utilized alone, integration with other technologies such as digital signatures or Identity Based Encryption (IBE) schemes results in cryptographically secure applications of biometrics. It is exactly this field of biometric cryptosystems that we focused in this thesis. In particular, our goal is to design cryptographic protocols for biometrics in the framework of a realistic security model with a security reduction. Our protocols are designed for biometric based encryption, signature and remote authentication.

We first analyze the recently introduced biometric remote authentication schemes designed according to the security model of Bringer et al.. In this model, we show that one can improve the database storage cost significantly by designing a new architecture, which is a two-factor authentication protocol. This construction is also secure against the new attacks we present, which disprove the claimed security of remote authentication schemes, in particular the ones requiring a secure sketch. Thus, we introduce a new notion called "Weak-identity Privacy" and propose a new construction by combining cancelable biometrics and distributed remote authentication in order to obtain a highly secure biometric authentication system. We continue our research on biometric remote authentication by analyzing the security issues of multi-factor biometric authentication (MFBA). We formally describe the security model for MFBA that captures simultaneous attacks against these systems and define the notion of user privacy, where the goal of the adversary is to impersonate a client to the server. We design a new protocol by combining bipartite biotokens, homomorphic encryption and zero-knowledge proofs and provide a security reduction to achieve user privacy. The main difference of this MFBA protocol is that the server-side computations are performed in the encrypted domain but without requiring a decryption key for the authentication decision of the server. Thus, leakage of the secret key of any system component does not affect the security of the scheme as opposed to the current biometric systems involving crypto-

graphic techniques. We also show that there is a tradeoff between the security level the scheme achieves and the requirement for making the authentication decision without using any secret key.

In the second part of the thesis, we delve into biometric-based signature and encryption schemes. We start by designing a new biometric IBS system that is based on the currently most efficient pairing based signature scheme in the literature. We prove the security of our new scheme in the framework of a stronger model compared to existing adversarial models for fuzzy IBS, which basically simulates the leakage of partial secret key components of the challenge identity. In accordance with the novel features of this scheme, we describe a new biometric IBE system called as BIO-IBE. BIO-IBE differs from the current fuzzy systems with its key generation method that not only allows for a larger set of encryption systems to function for biometric identities, but also provides a better accuracy/identification of the users in the system. In this context, BIO-IBE is the first scheme that allows for the use of multi-modal biometrics to avoid collision attacks. Finally, BIO-IBE outperforms the current schemes and for small-universe of attributes, it is secure in the standard model with a better efficiency compared to its counterpart.

Another contribution of this thesis is the design of biometric IBE systems without using pairings. In fact, current fuzzy IBE schemes are secure under (stronger) bilinear assumptions and the decryption of each message requires pairing computations almost equal to the number of attributes defining the user. Thus, fuzzy IBE makes error-tolerant encryption possible at the expense of efficiency and security. Hence, we design a completely new construction for biometric IBE based on error-correcting codes, generic conversion schemes and weakly secure anonymous IBE schemes that encrypt a message bit by bit. The resulting scheme is anonymous, highly secure and more efficient compared to pairing-based biometric IBE, especially for the decryption phase. The security of our generic construction is reduced to the security of the anonymous IBE scheme, which is based on the Quadratic Residuosity assumption. The binding of biometric features to the user's identity is achieved similar to BIO-IBE, thus, preserving the advantages of its key generation procedure.

Acknowledgments

My sincere gratitute and thanks go to my supervisor Prof. Dr. Joachim von zur Gathen for giving me the opportunity to pursue this PhD study under his guidance. His support, advice and suggestions has opened new perspectives for my research on biometric security. I am also grateful to the B-IT Research School for supporting me through a PhD scholorship and for providing a splendid research environment with various seminars and invited lectures. In particular, I cannot forget the oberseminars, summer schools and invited lectures organized by Cosec, where I had the chance to meet experts on security and cryptography. Cosec enabled me to gain teaching experience on biometric security and to present my work to people interested in my research area. I have also benefited from conversations and correspondence with many researchers that I met at different events. I wish to thank particularly to the organizers of WISTP'10 due to the best student paper award for my paper and Prof. Massimo Tistarelli, Marc Joye, Pascal Paillier, Julien Bringer, Prof. Terrance E. Boult, and lecturers of Biometrics Summer School for discussions which sharpened my knowledge in security and biometrics. Finally, I would like to thank Prof. Joachim von zur Gathen, Prof. Preda Mihailescu and Ass. Prof. Alexander Markowetz for their encouragement and support that has broadened my perspective on my future work.
This thesis is dedicated to my beloved mother.

Bonn 2011

Notations

Bit Strings

$\bar{a}$	bit complement of the string a
$a\|\|b$	concatenation of the strings a and b
$\{0,1\}^n$	set of n-bit strings
$\{0,1\}^*$	set of all finite binary strings

Sets

$\emptyset$	empty set
$\|A\|$	cardinality of the set A
$a \in A$	a is an element of the set A
$a \notin A$	a is not an element of the set A
$a \in [1,n]$	$a \in \{1,...,n\}$
$A \subset B$	set A is contained in set B
$A \subseteq B$	set A is contained in or equal to set B
$A \cup B$	union of sets A and B
$A \cap B$	intersection of sets A and B
$A \setminus B$	difference of sets A and B
$A \times B$	Cartesian product of sets A and B
$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integers
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$\mathbb{Z}_N$	set of integers modulo N (denoted also the set $\mathbb{Z}/N\mathbb{Z}$)
$\mathbb{Z}_N^*$	group of units in $\mathbb{Z}_N$
$\mathbb{Z}_N^*[+1]$	the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol $+1$
$QR(N)$	the set of quadratic residues (or squares) in $\mathbb{Z}_N^*$
$\mathbb{F}_q$	finite field of cardinality q
$\mathbb{F}_q^*$	multiplicative group of $\mathbb{F}_q$

Groups

$(\mathbb{G}, +)$	group $\mathbb{G}$ is denoted additively
$(\mathbb{G}, \cdot)$	group $\mathbb{G}$ is denoted multiplicatively
$0_{\mathbb{G}}$	identity in $(\mathbb{G}, +)$
$1_{\mathbb{G}}$	identity in $(\mathbb{G}, \cdot)$
$< g >$	group generated by the element g

Integers

$x \leftarrow y$	(for variables x and y) assigning the value of y to x
$a << b$	a is strictly of smaller order than b
$a >> b$	a is strictly of larger order than b
$\gcd(a, b)$	greatest common divisor of integers a and b
$a = b \bmod n$	a is congruent to b modulo n
$\varphi(N)$	Euler's totient function.
$(\frac{a}{n})$	Jacobi symbol of integer a and any positive odd integer n

Functions

$f : A \to B$	f is function from set A to set B
$a \mapsto b$	a is mapped to b (by some function)
$negl$	negligible function

Events, probabilities, and statistics

$\neg\, E$	complement of event E
$E_1 \wedge E_2$	intersection of event E_1 and event E_2
$E_1 \vee E_2$	union of event E_1 and event E_2
$\Pr[E]$	probability of event E
$\Pr[E_1 \mid E_2]$	probability of event E_1 given event E_2
$a \leftarrow D$	(for a distribution D) a is sampled from distribution D
$a \xleftarrow{R} S$	(for a finite set S) a is selected uniformly at random from set S

Acronyms

ABE	attribute based encryption
ABS	attribute based signature
ANO	anonymity
BDH	bilinear Diffie-Hellman
BDHI	bilinear Diffie-Hellman inversion
CCA	chosen ciphertext attack
CDH	computational Diffie-Hellman
CMA	chosen message attack
CPA	chosen plaintext attack
DDH	decisional Diffie-Hellman
DEM	data encapsulation mechanism
DHI	Diffie-Hellman inversion
DBRA	distributed biometric remote authentication
ECC	error correcting codes
EUF	existential unforgeability
FAR	false acceptance rate
FDH	full domain hash
FRR	false reject rate
GDH	gap Diffie-Hellman
HVZK	honest verifier zero-knowledge
IBE	identity based encryption
IBS	identity based signature
IND	indistinguishability
KEM	key encapsulation mechanism
MFBA	multi factor biometric authentication
MOC	match-on-card
NIZK	non-interactive zero knowledge
NM	non-malleability
OW	one-wayness
PCA	plaintext checking attack
PIR	private information retrieval
PKE	public key encryption
PPTM	probabilistic polynomial Turing machine
SS	secure sketch
ROM	random oracle model
ZKP	zero knowledge proof

Contents

1 Introduction **1**

1.1 Motivations 3

1.2 Contributions 6

2 Background **11**

2.1 Biometric Modalities 11

2.1.1 Fingerprint 12

2.1.2 Face 13

2.1.3 Iris 14

2.1.4 Multi-Modal Biometrics 15

2.2 Biometric Cryptosystems 15

2.2.1 Metric Spaces 17

2.2.2 Error-Correcting Codes 18

2.2.3 Secure Sketches 19

2.2.4 Biometric Key Generation/Locking/Release 20

2.2.5 Cancelable Biometrics 26

2.2.6 Distributed Biometric Remote Authentication 28

2.2.7 Fuzzy IBE/IBS 29

2.3 Public Key Encryption 30

2.4 Security Notions of PKE 31

2.4.1 Means of the Adversary 31

2.4.2 Goals of the Adversary 32

2.4.3 One-Wayness . . . 32
2.4.4 Indistinguishability . . . 33
2.4.5 Non-Malleability . . . 34
2.5 Hash Functions . . . 34
2.6 Security Reduction . . . 35
2.6.1 Random Oracle Model . . . 36
2.6.2 Decisional and Computational Assumptions . . . 37
2.7 Example PKE Schemes . . . 39
2.7.1 RSA Encryption Scheme: . . . 39
2.7.2 Goldwasser-Micali Scheme . . . 39
2.7.3 Paillier Encryption Scheme . . . 40
2.7.4 ElGamal Encryption Scheme . . . 41
2.7.5 Homomorphic encryption . . . 41
2.8 Tools for CCA security . . . 42
2.8.1 Generic Transforms . . . 42
2.8.2 Double Encryption . . . 43
2.8.3 Zero Knowledge Proofs . . . 43
2.8.4 Plaintext-Awareness . . . 46
2.9 Identity Based Cryptography . . . 47
2.9.1 Identity Based Signature . . . 48
2.9.2 Identity Based Encryption . . . 51
2.9.3 Bilinear Groups and Maps . . . 54
2.9.4 Assumptions based on Bilinear Pairings . . . 55
2.9.5 Security Notions of IBE . . . 58
2.10 Fuzzy IBE . . . 60
2.10.1 Shamir's secret sharing . . . 60
2.10.2 Based on Boneh-Boyen IBE . . . 62
2.10.3 Based on Boneh-Franklin IBE . . . 63
2.10.4 Based on Baek et al.'s IBE . . . 64
2.10.5 Based on Sakai-Kasahara IBE . . . 66

3 Distributed Biometric Remote Authentication Systems 68
3.1 Introduction . . . 69
3.1.1 Motivation and Contributions . . . 70
3.1.2 Related Work . . . 73
3.2 Architecture of the System . . . 74
3.2.1 Authentication Workflow . . . 74
3.3 Overview of the required cryptographic techniques . . . 75
3.3.1 Homomorphic encryption . . . 76
3.3.2 Secure Sketches . . . 76
3.3.3 Private Information Retrieval . . . 77
3.4 Security Model . . . 79
3.4.1 Trust Relationships . . . 79
3.4.2 Security Notions . . . 79
3.5 The first protocol . . . 81
3.5.1 Attacks against Bringer et al.'s scheme . . . 82
3.5.2 Analysis of the Attack and its Extension . . . 83
3.5.3 Modified Scheme . . . 83
3.6 Schemes based on ElGamal Encryption . . . 84
3.7 Schemes based on Paillier Encryption Scheme . . . 86
3.8 A first attempt for an efficient Biometric Remote Authentication . . . 88
3.8.1 An efficient storage mechanism for biometrics . . . 88
3.8.2 A concrete scheme based on ElGamal . . . 90
3.8.3 Security Analysis . . . 92
3.8.4 Efficiency Analysis . . . 94
3.8.5 Improving the accuracy . . . 94
3.8.6 The concrete protocol with improved accuracy . . . 97
3.9 New Attacks . . . 97
3.10 Preventing the Attacks . . . 101
3.10.1 A New Protocol for Cancelable Biometric Setting . . . 103
3.10.2 Identity Privacy for Cancelable Biometrics . . . 107

3.11 Comparison 108
3.12 Conclusion 109

4 Practical Multi-factor Biometric Remote Authentication 110
4.1 Introduction 111
4.1.1 Related Work 113
4.1.2 Motivation and Contributions 114
4.2 Preliminaries and Definitions 117
4.2.1 Forking Lemma 118
4.2.2 Zero Knowledge Proof 118
4.2.3 Plaintext Awareness 118
4.3 A New Design for MFBA 119
4.3.1 Choosing the cryptographic method 119
4.3.2 Biometric Template Generation 120
4.4 Security Model 121
4.4.1 Adversarial Capabilities and Goals 122
4.4.2 User Privacy 123
4.4.3 The concrete scheme 124
4.4.4 Biometrics as an Unordered Set 130
4.5 Discussion 132
4.6 Conclusion 134

5 Efficient Biometric Identity Based Signature 135
5.1 Introduction 136
5.1.1 Related Work 137
5.1.2 Our Contributions 138
5.2 Definitions and Building Blocks 139
5.2.1 Forking Lemma 140
5.2.2 Fuzzy Identity Based Signature 140
5.2.3 Security Model 141
5.2.4 Signer-Attribute Privacy 142

5.3 A New Efficient Biometric IBS Scheme 142
5.3.1 Modified t-ABS 143
5.3.2 Our Efficient Biometric IBS Scheme 144
5.3.3 A Stronger Security Model 150
5.3.4 Weak Signer-Attribute Privacy 152
5.3.5 Some arguments against the architecture of t-ABS 153
5.4 Efficiency Discussions and Comparison 153
5.5 Conclusion 154

6 Biometric Identity Based Encryption 155
6.1 Introduction 156
6.1.1 Motivation and Contributions 157
6.1.2 Related Work 162
6.1.3 Organization 163
6.2 Definitions and Building Blocks 163
6.2.1 Fuzzy Identity Based Encryption 164
6.2.2 Security Model 164
6.2.3 The small universe construction of Sahai and Waters 165
6.2.4 A first Attempt for an efficient biometric IBE 168
6.2.5 Error Correcting Codes and Fuzzy Extractors 174
6.2.6 Our New Method for Biometric Identities 175
6.3 BIO-IBE 176
6.4 A Stronger Security Model 178
6.5 Improving the reduction cost of BIO-IBE 182
6.6 BIO-IBE in the standard model for small universe 186
6.7 A New Denial of Service Attack 189
6.7.1 The modified BIO-IBE 191
6.8 Comparison 193
6.9 Conclusion 195

7 Anonymous Biometric IBE without Pairings 196

7.1 Introduction . . . 197
7.1.1 Motivation and Contributions . . . 197
7.1.2 Related Work . . . 199
7.1.3 Organization . . . 200
7.2 Definitions and Building Blocks . . . 200
7.2.1 Robust Sketch and Robust Fuzzy Extractors . . . 201
7.3 A weakly secure Generic Construction . . . 201
7.3.1 Entropic Security vs. Indistinguishability . . . 203
7.4 Security Properties . . . 203
7.4.1 Anonymity . . . 203
7.4.2 Identity Privacy . . . 205
7.5 Generic Constructions for Biometric IBE . . . 206
7.5.1 Based on the Fujisaki-Okamoto Conversion . . . 207
7.5.2 Collision Attacks . . . 213
7.5.3 Based on REACT . . . 215
7.6 A Generic Biometric ID-KEM Construction . . . 218
7.7 Applications . . . 221
7.7.1 Based on the scheme of Boneh et al. . . . 221
7.7.2 Based on the scheme of Ateniese et al. . . . 224
7.8 Comparison . . . 226
7.9 Conclusion . . . 227

8 Conclusion **229**

List of Figures

1.1 Biometric Modalities 2

2.1 Fingerprint Minutia 12
2.2 Facial features 13
2.3 Feature vector of face 14
2.4 fusion in a bimodal biometric system 16
2.5 multimodal biometric system 17
2.6 Systems for Biometric Template Protection 18
2.7 Fuzzy Commitment Scheme 21

3.1 Overview of DBRA 75
3.2 Overview of Bringer et al.'s protocol 82
3.3 Overview of the new protocol 91
3.4 Sketch Generation and Reconstruction 95

4.1 Security and Privacy Issues of MFBA 122
4.2 Registration Phase 125
4.3 Verification Phase 126

5.1 Comparison of error tolerant IBS schemes 154

6.1 Modified BIO-IBE for single-biometric trait 190
6.2 Modified BIO-IBE for two biometric traits 192
6.3 Comparison to various fuzzy IBE schemes 194

7.1 A weakly secure generic construction 202

List of Tables

2.1 Implementation of Brute Force Attack 25

3.1 Comparison of PIR systems 78
3.2 The number of common features 90
3.3 Comparison of DBRA schemes 95
3.4 Properties of various DBRA schemes 108

4.1 Attacks against MFBA systems 133

6.1 Fuzzy IBE Schemes in the Standard Model 193
6.2 Fuzzy IBE Schemes in ROM 194
6.3 Abbreviations 194

7.1 Computational Cost of biometric IBE systems 227
7.2 Ciphertext size of biometric IBE systems 227

Chapter 1

Introduction

To prove our identity, we can use three ways:

- Something we have (e.g. a smartcard)
- Something we know (e.g. a PIN code, a password)
- Something we are (biometrics, e.g. fingerprint, face, iris)

In everyday life, we usually give our trust to a combination of something-we-have and something-we-know (e.g. banking cards, SIM card in mobile phones) but a password can be communicated or guessed and a personal device can be lost or borrowed. Building a three-factor authentication with the addition of one or several biometric techniques brings high confidence in our authenticated interlocutor. The advantage of using biometrics in a single or multi-factor authentication setting is that biometric data is always handy, we cannot forget or loose it, we do not need to remember or keep it secret for secure authentication (as in the case of a long password). Biometrics uniquely defines a user and is direct evidence of personal participation in authentication, especially when two different biometric traits are combined as it is called as multi-modal biometrics or biometric fusion. Currently, many countries collect at least two different biometric traits (fingerprint and face) from each traveller in border control applications, for instance the US-visit program. Finally, under supervision or in controlled environments, it is very difficult to forge biometrics and impersonate a user, although it is much easier to forge documents.

ISO/IEC JTC1 SC37 Standing Document defines biometrics as: "Automated recognition of individuals based on their behavioral and biological characteristics". A behavioral aspect of a biometric measures data pertaining to a personal trait, learned over time, or to a learned action. Biometrics with stronger behavioral aspects (e.g.,

keystroke, sign/signature, voice) utilize acoustics, pressure, and speed whereas those with stronger biological aspects (e.g., fingerprint, iris, hand geometry, vein) measure characteristics residing on or near the surface of the human body [Tilton, 2007]. The classification of biometric modalities are given as below.

Figure 1.1: Biometric Modalities

A traditional biometric system has four important modules:

1. The *sensor module* which captures the trait in the form of raw biometric data
2. The *feature extraction* module which processes the data to extract a feature set that is a compact representation of the trait. This set may consist of biometric features that can be either ordered/grouped or not, depending on the biometric trait.
3. The *matching module* which employs a classifier to compare the extracted feature set with the templates residing in the database to generate matching scores. The comparison is accomplished using a distance function that can be Hamming distance, set difference, edit distance or Euclidean distance as described in the following section for metric spaces.
4. The *decision module* which uses the matching scores to either determine an identity or validate a claimed identity.

The accuracy of a biometric system is measured by the following performance criteria:

- False accept rate or false match rate (FAR or FMR): the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs which are incorrectly accepted.

- False reject rate or false non-match rate (FRR or FNMR): the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs which are incorrectly rejected.

Traditionally, biometric technology is used for identification or authentication purposes. In the identification mode, the biometric system identifies a person from the entire enrolled users in the system by searching a database for a match. This is sometimes called "one-to-many" matching. A system can also be used in authentication (i.e. verification) mode, where the biometric system authenticates a person's claimed identity from their previously enrolled pattern. This is also called "one-to-one" matching. When the matching is performed at the remote server, then the system is called as a remote biometric authentication. Alternatively, biometrics can be used for local authentication - for example, to control access to a private key on a smart card.

Biometric-based authentication applications include workstation and network access, single sign-on, application logon, data protection, remote access to resources, transaction security, and Web security. The promises of e-commerce and e-government can be achieved through the utilization of strong personal authentication procedures. Secure electronic banking, investing and other financial transactions, retail sales, law enforcement, and health and social services are already benefiting from these technologies. Biometric technologies are expected to play a key role in personal authentication for large-scale enterprise network authentication environments, Point-of-Sale and for the protection of all types of digital content such as in Digital Rights Management and Health Care applications [Podio and Dunn, 2001].

1.1 Motivations

Recently, biometrics has pervaded other aspects of security applications that are listed under the topic of *biometric cryptosystems*: Biometric encryption, biometric key generation/key locking/key release systems output a secret key such as a 128-bit AES key that can be used for encryption or authentication purposes. Although the security of some of these systems is questionable when they are utilized alone, integration with other technologies such as digital signatures or Identity Based Encryption (IBE) schemes results in cryptographicly secure applications of biometrics. It is exactly this

field of biometric cryptosystems that we focused in this thesis. In particular, our goal is to design cryptographic protocols for biometrics in the framework of a realistic and strong security model and provide a security reduction to a hard problem. The majority of the biometric cryptosystems in the literature lacks a security analysis from a cryptographic point of view, which have led us to work in this field. In fact, the first biometric authentication system that assumes biometrics as public data and that is evaluated in a reductionist sense of security is described by Bringer et al. in 2007.

Despite the fact that biometric technologies are implemented for a long period of time, the security concerns about the storage of the biometric templates (i.e. biometric features of each user enrolled in the authentication system) and solutions for template protection have emerged recently. Also, there is skeptical view about the secrecy of biometrics, which invalidates the assumptions that many biometric cryptosystems rely on. In fact, if we assume biometrics as public data as in distributed biometric authentication and biometric-based IBE/IBS systems, the security provided by the majority of biometric cryptosystems for biometric key generation/template protection is questionable. Thus, we base our designs for biometric-based IBE/IBS and for remote authentication on valid assumptions such as public biometric data, whose distribution is known to the adversary.

The first biometric-based encryption system with a security reduction is introduced in Eurocrypt 2005 by Sahai and Waters. The authors define their system as "Fuzzy Identity-Based Encryption (IBE): Privacy for the Unprepared". As the name suggests, fuzzy IBE does not require either the sender or the receiver to possess/store an RSA public key and the related online certification authority that binds this RSA key to its owner. Certification in fuzzy IBE is natural since the biometric data used as the public key is bound to the user. The system is even less complicated than a standard IBE system, which requires a unique e-mail address as the identity that should be proven to be different for the people sharing the same name. Thus, an unprepared user can present his biometrics as his encryption key to someone when they are physically present or when a user is traveling and another party encrypts an ad-hoc meeting between them. A biometric identity is an inherent trait and will always with the person, namely, the person will always have their public key handy. Besides, the process of obtaining a secret key from an offline authority is very natural and straightforward. In standard IBE schemes, a user with a an identity such as an e-mail address "proves" to the trusted authority that he is indeed entitled to this identity. This will typically involve presenting supplementary documents or credentials. The type of authentication that is necessary is not always clear and robustness of this process is questionable (the supplementary documents themselves could be subject to forgery). However, in fuzzy/biometric IBE, the user only presents his biometrics to the trusted authority under the supervision of a well trained operator. The operator is able to detect imitation attacks, for example playing the recording of a voice. We emphasize that the biometric measurement for

an individual need not be kept secret since it is used as a public key. This assumption is also accepted by the biometrics community who consider the biometrics as public data, whereas the biometric template that is stored in a database for authentication purposes should be kept private.

Finally, using biometric as the identity has the advantage that identities are unique if the underlying biometric is of a good quality. Some types of standard identities, such as the name "Bob Smith" will clearly not be unique or change owners over time.

In parallel to the introduction of fuzzy IBE, the first biometric-based signature was already implemented based on a standard IBS scheme and a special primitive called fuzzy extractor. Since then, other schemes for biometric-based encryption and signature have been designed to improve the efficiency of the previous schemes. Despite these improvements, new constructions are necessary in order to significantly reduce the computational cost resulting from the high number of pairing operations at each decryption.

The second line of work that assumes biometrics as public data and includes a security reduction focuses on remote biometric authentication. In 2007, distributed biometric remote authentication was introduced by Bringer et al. with a new security model that evaluates security against insider adversaries from a cryptographic point of view. In this system, the server-side functionalities are performed in encrypted domain and in a distributed fashion using a detached biometric database and non-colluding system components. This leads to a new security notion called identity privacy that guarantees the privacy of the link between the identity (name) and the (reference) biometrics of the user although biometrics is assumed as public data. Moreover, the database is prevented from tracking the user that authenticates to the system, thus transaction anonymity against a (malicious) database is satisfied which is the second notion for biometric remote authentication. In this security model, the authors describe a concrete protocol that achieves these two notions by reducing the security to breaking the underlying homomorphic encryption scheme. With this property, this system is the first (pure) biometric authentication protocol with a security reduction in the framework of a strong and realistic security model. Other papers followed the work of Bringer et al. to improve the security, accuracy and efficiency of this protocol. Recently, attacks have been proposed to the first system in this model, which lead us to question the security of the following schemes and the correctness of the security notions.

We note that the research on fuzzy IBE and distributed remote authentication does not include implementation of the protocols since this requires additional sources of information on biometric traits, biometric sampling, image processing, feature extraction, feature selection, etc. Thus, as in the case of fuzzy commitment and fuzzy vault, the implementation of these biometric cryptosystems for a particular biometric trait is of interest to the biometrics community.

1.2 Contributions

This thesis presents the ensemble of my PhD results [Sarier, 2008, 2009a,b, 2010a,b,c,d, 2011a,b] obtained in the area of biometric cryptosystems. The thesis can be split into two parts. The first part with chapters 3 and 4 is concentrated on biometric remote authentication, whereas the second part consists of the biometric-based IBE/IBS constructions that are described in chapters 5 through 7.

Chapter 3 starts with the analysis of the distributed biometric remote authentication (DBRA) by presenting the recently introduced security model and the current schemes for this particular type of authentication. Next, we present our contributions by describing a new and efficient biometric storage mechanism, where the biometric features of the users in the system are stored as a random pool of features instead of storing each complete reference template. This way, common features belonging to different users in the system are not stored multiple times, which results in a reduced storage cost. Secondly, we prove that identity privacy cannot be achieved for DBRA, if biometrics is assumed as public data and a publicly stored sketch is employed for improved accuracy. Besides, a statistical attack is shown that is effective even if the sketch is stored as encrypted. For DBRA with encrypted sketches, we define a weaker notion of security called "Weak Identity Privacy" in order to eliminate such statistical attacks. The remaining schemes are vulnerable to our attacks if they are not implemented as a two-factor solution, where a tamper-proof smartcard is required as the second factor to store some parameters of the scheme secretly. In view of this result, we describe a new two-factor biometric remote authentication system by combining distributed biometric authentication and cancelable biometrics, which is also applicable for biometrics represented as a set of features. Next, we define "identity privacy for cancelable biometrics" as a new notion and show that existing schemes vulnerable to our attacks are secure in the cancelable biometric setting. The security of our protocols are reduced to the security of well-defined problems such as Gap Diffie-Hellman or security of ElGamal Encryption scheme.

This chapter is based on our work presented in [Sarier, 2009a,b, 2010b, 2011a].

In **chapter 4**, we evaluate the security properties of Multi-Factor Biometric Authentication (MFBA), where biometrics is assumed as a set of features that can be either ordered or unordered depending on the biometric modality. We propose efficient schemes for MFBA that are suitable for a different template extraction method used in bipartite biotokens that separates the stable and non-stable parts of each feature, thus, a cryptographic protocol can be applied to encrypt the stable parts and the matching score is computed in the encryption domain at the remote server, whereas another matching can be performed at the client-side by checking whether each non-stable part is within its predefined range. We formally describe the security model for

MFBA, where the server-side computations are performed in the encrypted domain but without requiring a decryption key for the authentication decision of the server. Thus, leakage of the secret key of any system component does not affect the security of the scheme as opposed to the current biometric systems involving cryptographic techniques. Finally, we reduce the security of our design to the unforgeability of the Schnorr Signature Scheme according to our new security model that captures simultaneous attacks against a MFBA. In this context, we define the notion of user privacy, where the goal of the adversary is to impersonate a client to the server. The adversary has access to different oracles that model the adversaries capabilities such as eavesdropping on the communication channel -even in the case of a compromised session key that is used to build a secure communication link before the start of the protocol execution-, and compromise of *either* the sensor (namely biometrics of the user) *or* the smart card of the user that stores the secret parameters for the stable/non-stable part separation. Our system is based on the signed ElGamal encryption scheme, which is IND-CCA secure and plaintext aware in Random Oracle Model (ROM). Due to the combination of a (weakly secure) encryption scheme and non-malleable proof of knowledge of the randomness used, the adversary proves his knowledge of the (stable) biometric features, thus a decryption oracle would be useless and security against Chosen Ciphertext Attacks (CCA) is provided. Hence, our design is the first biometric system based on a CCA secure encryption system. Furthermote, when implemented on a pairing friendly elliptic curve, the server can make the authentication decision without any decryption operation through the use of bilinear pairings. In particular, elliptic curve signed ElGamal achieves at most OW-CCA security in ROM if bilinear pairings are used to test for equality of biometric data in the encrypted domain. We also show that there is a tradeoff between the security the scheme achieves (OW-CCA instead of IND-CCA) and the requirement for making the authentication decision without using any secret key. Clearly, if the final decision is made by decrypting the resulting computation as in current biometric authentication systems, our construction achieves highest (i.e. IND-CCA) security level. For unordered biometrics such as fingerprint minutia, we employ RSA encryption combined with a zero knowledge proof of knowledge of plaintext for RSA. The results described in this chapter were published in [Sarier, 2010a].

In **chapter 5**, we present a new biometric Identity Based Signature (IBS) scheme that is more efficient compared to the current fuzzy IBS and threshold Attribute Based Signature (t-ABS) schemes in all aspects, since it is based on the currently most efficient pairing-based IBS scheme. Moreover, our scheme could function as a fuzzy IBS or t-ABS scheme if the biometric features are replaced by attributes defining the identity of the signer.

We prove the security of our new scheme first in the framework of the security model of fuzzy IBS. Next, we define a stronger security model, which basically simulates the leakage of partial secret key components of the challenge identity. This property is not

considered in the model of fuzzy IBS (and t-ABS), which return to the adversary only the private key components belonging to any identity other than (i.e. not similar to) the challenge identity. However, in our security reduction, we allow the adversary to query for some of the private key components belonging to the challenge identity. The contributions of this chapter are based on the paper [Sarier, 2010c].

In **chapter 6**, we present two efficient biometric IBE schemes based on pairings. The schemes are based on the Sakai Kasahara IBE, which includes the most efficient key generation method among pairing-based IBE schemes. As a starting point, we present a new construction for identities represented as an ordered set of features/attributes and provide a security reduction following the security model described for fuzzy IBE systems. Next, we introduce BIO-IBE, which has a different structure compared to current fuzzy IBE systems. The security reduction is presented for large universe of biometric attributes in ROM and for small universe, in the standard model. We will show that for the large universe of attributes, BIO-IBE is more efficient compared to the fuzzy IBE schemes and for the small universe, it is more efficient than the small universe construction of [Sahai and Waters, 2005]. Moreover, we describe a stronger security model and prove the security of BIO-IBE based on this stronger model that basically simulates the leakage of partial secret key components of the challenge identity similar to the model for our biometric IBS. As different from the fuzzy IBE scheme of [Sahai and Waters, 2005], the ciphertexts can contain a variable number of attributes but the error tolerance parameter is a fixed threshold value as in current systems.

Our new scheme BIO-IBE is the first biometric IBE scheme that allows for the use of multi-modal biometrics for defining the identity of the user. In particular, the key generation algorithm of BIO-IBE fuzzy-extracts a unique biometric identity string ID from the biometrics of the user instead of picking a different polynomial for each user as in other fuzzy IBE schemes. For instance, the private key components of each user is computed by using a biometric trait such as fingerprint, face, palmprint, etc. combined with the unique biometric identity string fuzzy-extracted from the same or a different biometric trait such as the Iris scan of the user. This combination is used to bind the private key components to that user and to avoid collision attacks, thus different users sharing common biometric attributes with the receiver of the ciphertext cannot decrypt this ciphertext by combining their secret key components associated to these common attributes. From the efficiency point of view, the fuzzy extraction is performed only by the sender, is independent of the message, and hence can be done once and for all. In current fuzzy IBE schemes, collision attacks are prevented by picking a unique polynomial for each user that is evaluated at each biometric feature of the user and that is combined with the master secret key to generate the private key components. In BIO-IBE, the final computed private key components of each user can be thought as a biometric fusion at the feature level. This new combination does not only prevent collision attacks, but also has the advantage of better accuracy/identification compared

to the use of uni-modal biometrics as in current fuzzy IBE. Impersonation/spoofing attacks are prevented as forging two different biometric traits is more difficult compared to the use of one modality. Multimodal biometric systems are shown to be more reliable, which also relax the requirement for a well trained supervisor since a challenge response type of mechanism by requesting the user to present a random subset of biometric traits thereby ensuring that a "live" user is indeed present at the point of data acquisition.

Secondly, the new design eliminates the requirement of a special hash function called MapToPoint hash function that maps a user's identity to a point on the underlying elliptic curve in IBE schemes. Currently, efficient fuzzy IBE schemes [Pirretti et al., 2006, Baek et al., 2007, van Liesdonk, 2007] employ this special function, which is usually implemented as a probabilistic algorithm and is more expensive than a point scalar multiplication in terms of computation time. Besides, it is difficult to find groups as the range of the MapToPoint hash function and to define an efficient isomorphism at the same time. Apart from the efficiency gain resulting from the replacement of the MapToPoint hash function with an ordinary one, BIO-IBE has a structurally simpler key generation algorithm and provides better efficiency in terms of the key generation and decryption algorithms compared to the existing fuzzy IBE schemes despite the additional fuzzy extraction process.

Next, we improve the reduction cost of BIO-IBE by reducing its security to the decisional k-BDHI problem instead of computational k-BDHI problem. We see a tradeoff between the tightness of the reduction cost and the hardness of the underlying problem. By a small modification, BIO-IBE becomes resistant against an Denial of Service (DoS) attack resulting from the use of the fuzzy extraction process. We should note that the use of multi-biometric based encryption using a fuzzy extractor is claimed to be introduced in 2011 by [Zhang et al., 2011], although the first use of this approach is presented for BIO-IBE in 2008. Finally, our new method for preventing collision attacks can be applied in other IBE systems that are not based on pairing based cryptography, as we will see in chapter 7. The contributions of this chapter are based on the papers [Sarier, 2008] and [Sarier, 2011b].

In **chapter 7**, we present a novel framework for the generic construction of biometric Identity Based Encryption (IBE) schemes, which do not require bilinear pairings and result in more efficient schemes than existing fuzzy IBE systems implemented for biometric identities. Currently, fuzzy IBE schemes are based on stronger (bilinear) assumptions and guarantee a weak level of security, but they could be combined with well-known generic conversion systems to obtain a high level of security. Besides, biometrics is considered as public information, thus, biometrics w' of the receiver is sent together with the corresponding ciphertext so that the receiver with biometrics w can determine the common features between w and w' in order to apply the correct secret key components. Clearly, this could effect the privacy of the user's actions if we

consider anonymity notion for IBE systems. Thus, we analyze the security properties that are specific to biometric IBE, i.e. anonymity, and introduce a new notion for biometric IBE called as identity privacy. We note that current fuzzy IBE and biometric IBE systems do not consider anonymity and privacy of user biometrics at the same time, hence, it is vital to describe an efficient and anonymous error-tolerant encryption system for biometric identities in order to avoid traceability of the user's actions. Considering these notions, we present generic constructions for biometric IBE and ID-KEM based on weakly secure anonymous IBE schemes, error correcting codes and generic conversion schemes in order to obtain highly secure anonymous biometric IBE schemes. Our generic constructions for biometric IBE and ID-KEMs convert any weakly secure anonymous IBE scheme encrypting a message bit by bit to a highly secure biometric IBE scheme. Finally, we describe concrete applications of our framework and compare them to the existing fuzzy IBE systems in terms of time complexity and bandwidth. We design our new constructions for any type of biometrics that can be represented as an ordered set of features (i.e. a sequence of n feature points) such as face, online handwritten signatures, iris, voice etc. However, if anonymity property is not required, then our system is applicable for any type of biometrics since we can allow for the attachment of the biometrics to the ciphertext as in fuzzy IBE schemes. To avoid collusion attacks and to guarantee the security notions that we present, the anonymous IBE schemes are implemented for biometric identities using our new method that combines each feature with a unique biometric string obtained via a fuzzy extractor as described in chapter 6. Thus, we achieve more efficient and anonymous biometric IBE schemes based on weaker assumptions (such as Quadratic Residuosity) with higher security/accuracy due to the multi-modal identities. The contributions of this chapter are based on the paper [Sarier, 2010d] that received the best student paper award.

Chapter 2

Background

The definitions of the primitives and methods used in the following chapters are given in two parts. The first part of the chapter summarizes the necessary definitions of biometric concepts and describes biometric template protection methods including key locking/hiding/extraction, multi-modal biometrics and different applications/representations of biometrics. The second part summarizes the definitions and properties of the cryptographic primitives and reviews different Identity Based Encryption (IBE) schemes that form the basis for the recently introduced fuzzy IBE/IBS systems.

Part I : Background in Biometrics

2.1 Biometric Modalities

As shown in figure 1.1, various biometric traits can be used for various applications. However, each of these biometrics has its strengths and weaknesses and therefore the choice of the biometric depends on the application. Apart from the classification of biometric modalities in figure 1.1, biometrics can be categorized based on its representation. A large class of biometric traits can be represented as a set of ordered/grouped features, namely a sequence of n ordered feature points [Ballard et al., 2008, Teoh et al., 2008], such as Iris [Kanade et al., 2009, Bringer et al., 2007b], fingercode [Jain et al., 2000, Tong et al., 2007], face [Li et al., 2006, Ekenel and Stiefelhagen, 2009, Gao et al., 2009, Boehnen et al., 2009, Moreno et al., 2005], online signatures [Igarza et al., 2004]. For instance, the face recognition system of [Moreno et al., 2005] extracts each feature from particular regions of face such as mouth, nose,eyes, etc., where a

feature is zero valued if it cannot be computed because of the non-existence of a region from which it is derived. Besides, the fuzzy commitment construction of [Juels and Wattenberg, 1999] is based on ordered biometric feature vectors as we discuss in the following section. However, there exists also biometric representations, which cannot be ordered as in the case of fingerprint minutia. In this section, the properties of the most commonly used biometrics are outlined. Subsequently, the representation of these biometric characteristics and which features of these characteristics can be extracted, is explained.

2.1.1 Fingerprint

Fingerprints are the oldest traits that are mostly represented as friction minutiae-based features while systems are rarely designed to use an entire image of a fingerprint. Thus, a fingerprint of a user consists of a set of unordered features called as fingerprint minutia. Each minutia is assumed to be represented by its 2D spatial location and its local orientation. Discontinuities in the flow of ridges (ridge bifurcations and endings) constitute the minutiae. The following figure shows a sample fingerprint image with overlaid minutiae.

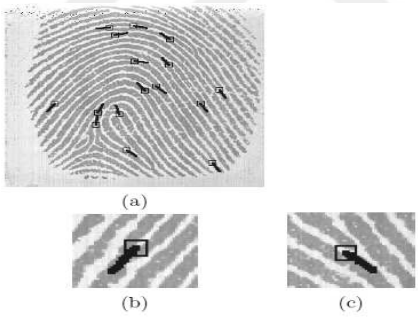

Figure 2.1: Fingerprint Minutia

The main difficulty within fingerprint biometrics is in finding specific fingerprint orientation and its center. Otherwise, all calculations resulting out of minutiae are destined to be orientation/position-dependent. Thus, the matching algorithm has to deal with transformations of fingerprint data by aligning fingerprint images using high curvature points and orientation lines.

In the systems we consider, fingerprints are represented by the cartesian coordinates of the fingerprint minutia features, where the couples of coordinates are concatenated to single numbers that are mapped to the elements of a finite field by some convention [Nandakumar et al., 2007a, Mihailescu, 2007].

2.1.2 Face

In a face recognition system, images of the whole face of a person are captured out of which unique key features are extracted to identify persons reliably. The acquired set of key features include relative distances between characteristics such as eyes, the nose, the mouth, cheekbones and the jaw. Using all of this information, a unique template is created by applying dimension reduction. In [Moreno et al., 2005], the ordered set of face features are obtained through the segmentation of regions and lines of interest, feature extraction from the segmented regions and lines, and classification of the feature vectors that model the faces. In figure 2.2, the segmentation of regions and lines of interest of a face model is shown.

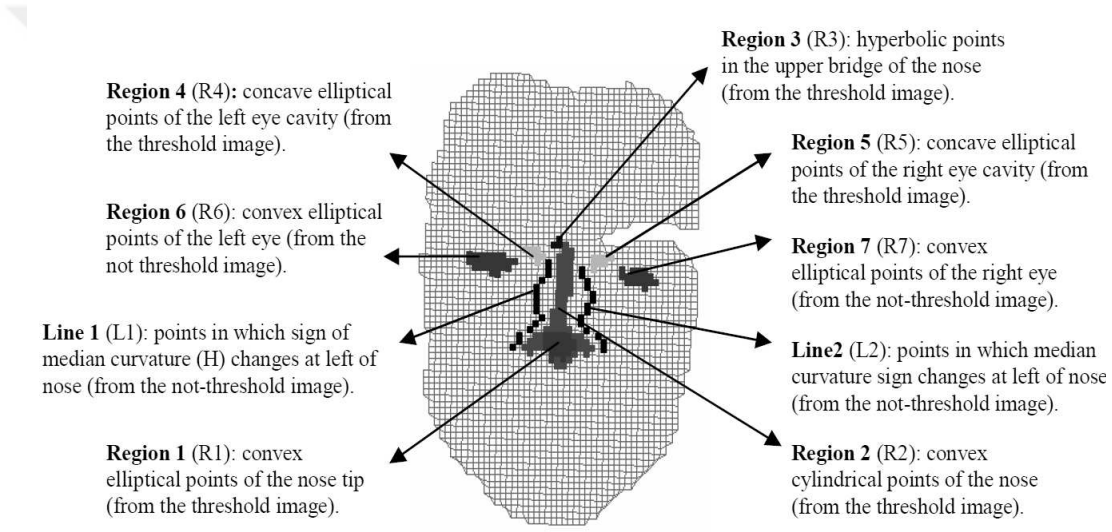

Figure 2.2: Segmented regions and lines of a face from which facial features are extracted [Moreno et al., 2005]

In figure 2.3, 30 more discriminating features and their position in an ordered list according to their discriminating power are shown. As a result, the feature set of a user is represented by 30-37 ordered features. The used acronyms are: Ri = region i; Li = line i; A_Ri = area of region i; C_Ri = centroid of region i; d(P1, P2) = Euclidean distance between 3D points P1 and P2; ang(P1, P2, P3) = angle defined by the 3D points P1, P2 and P3, being P2 the intermediate vertex; H_Ri and K_Ri are the respective averages of the Mean and Gaussian curvatures, evaluated in points belonging to the region i; VH_Ri and VK_Ri are the respective variances of the Mean and Gaussian curvatures evaluated in points belonging to the region i.

Rank	Feature description	Rank	Feature description
1	ang(C_R4, C_R3, C_R5)	16	H_R3
2	ang(C_R4, C_R3, C_R1)	17	H_R4
3	1/2[ang(C_R4,C_R3,C_R1)+ang(C_R5, C_R3,C_R1)]	18	d(C_R4, C_R5)
4	ang(C_R5, C_R3, C_R1)	19	d(C_R5, C_R1)
5	1/2[d(C_R4, C_R3)+d(C_R5,C_R3)]	20	K_R4
6	d(C_R1, C_R3)	21	H_R5
7	ang(upper point of L1,C_R3, upper end of L2)	22	K_R5
8	d(C_R4, C_R3)	23	ang(C_R6, C_R3, C_R1)
9	d(C_R5, C_R3)	24	H_R2
10	d(C_R4, C_R5)/ d(C_R1, C_R3)	25	1/2[ang(C_R6,C_R3,C_R1) +ang(C_R7, C_R3,C_R1)]
11	K_R3	26	ang(C_R7, C_R3, C_R1)
12	H_(R4 U R5)	27	ang(C_R6, C_R3, C_R7)
13	K_(R4 U R5)	28	H_R1
14	1/2[d(C_R4, C_R1)+d(C_R5,C_R1)]	29	1/2[d(C_R6, C_R3)+d(C_R7,C_R3)]
15	d(C_R4, C_R1)	30	A_R2

Figure 2.3: Face biometrics represented as grouped/ordered feature vector [Moreno et al., 2005]

2.1.3 Iris

Breakthrough work to create iris recognition algorithms required for image acquisition and matching were developed by J. G. Daugman [Daugman, 2004]. Daugman's algorithms are the basis of all today's commercially used iris recognition systems. In his system, first the iris has to be extracted out of the whole image of the person's eye. Therefore the center of the iris and the inner and outer boundaries have to be detected. In the next step, so-called *analysis bands* are defined for the extracted iris (in form of a ring). These bands are used to position points which are then explored using 2D Gabor filters to denoise the acquired signal. Then iris ring is unwrapped by mapping polar coordinates to cartesian-coordinates which results in a rectangular image. In this image, the radii of the previously defined analysis bands is fixed and every explored point is a center of a 2D Gabor wavelet. For this wavelet the coefficients are generated out of which two bits are extracted. This method is applied again until enough bits are extracted. The rectangular image of the iris mostly includes some part of the eyelid and eyelashes. Eyelashes are seen as noise which have to be detected during the unwrapping of the iris. This is done by calculating a bit-mask where one bit represents a region of the iris and is set to 0 if any noise is detected and otherwise to 1. The result of the whole procedure is a so-called **"iris-code"** (for example 2048-bit long in J. G. Daugman's approach). This iris code serves as a biometric template which can be stored in a database together with the bit-mask. After the extraction of the iris-code the matching process can be performed using several metrices, for example, the Ham-

ming distance. This could be easily done by just bitwise XORing two iris-codes and comparing the number of mismatching bits to a specific threshold [Rathgeb, 2008].

2.1.4 Multi-Modal Biometrics

In order to address some of the limitations of unimodal systems such as noisy data, intra-class variations, restricted degrees of freedom, non-universality, spoof attacks, and unacceptable error rates, multimodal biometric systems are deployed [Ross and Jain, 2004].

Firstly, multiple traits ensure sufficient population coverage and deter spoofing since it would be difficult for an impostor to spoof multiple biometric traits of a genuine user simultaneously. Besides, they ensure that liveliness assumption holds by facilitating a challenge-response type of mechanism that requires the user to present a random subset of biometric traits during each authentication. Multi-modal biometric systems can be classified in three levels. (a) *Fusion at the data or feature level*: Either the data itself or the feature sets originating from multiple sensors/sources are fused. (b)*Fusion at the match score level*: The scores generated by multiple classifiers pertaining to different modalities are combined. (c) *Fusion at the decision level*: The final output of multiple classifiers are consolidated via techniques such as majority voting [Ross and Jain, 2004].

In figure 2.4 and 2.5, we present example applications of multi-modal biometric systems. The abbreviations denotes FU: Fusion Module, MM: Matching Module, DM: Decision Module.

In our theses, we focus on the fusion at the feature level and employ the multi-modal biometric scenario for multiple biometric traits denoted with number 2) of figure 2.5.

2.2 Biometric Cryptosystems

Combination of biometrics and cryptography is achieved under different names: Untraceable Biometrics, Biometric Encryption (BE), Fuzzy Extractor, Secure sketch, Helper Data Systems, Biometric Locking, Biometric Key Generation, etc. Despite the different names, the goal of these systems is the same: Biometric Template Protection. In figure 2.6, we see an overview of these systems.

One approach focuses on the BE technologies that securely bind a digital key to a biometric, or extract a key from the biometric so that neither the key nor the biometric can be retrieved from the stored BE template, also called “helper data” [Li and Jain, 2009]; The key is re-created only if a correct biometric sample is presented on verifica-

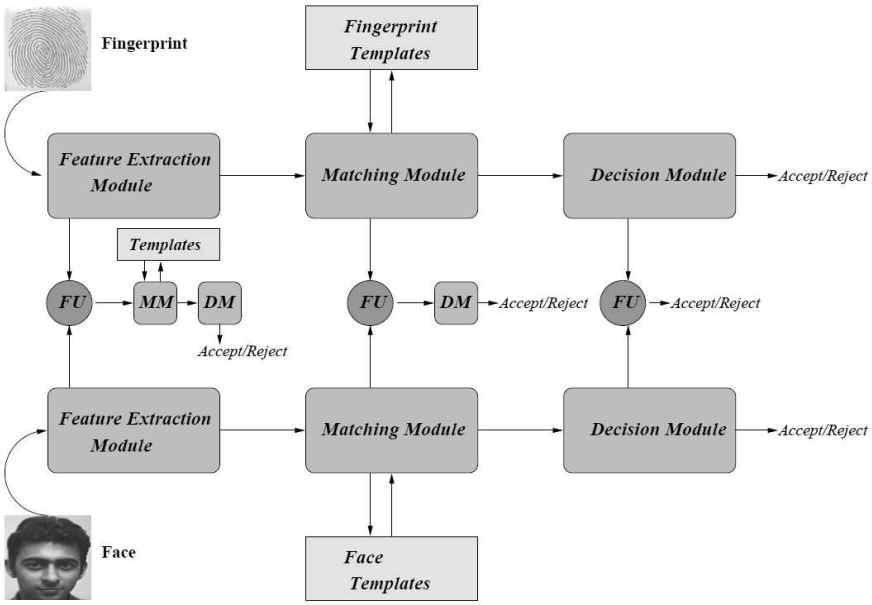

Figure 2.4: Levels of fusion in a bimodal biometric system [Ross and Jain, 2004]

tion; and the output of BE verification is either a key (correct or incorrect) or a failure message. There is always a biometric dependent helper data stored in the system, but the cryptographic key is not kept at all. In practice, BE, like any biometric system, has both false rejection and false acceptance rates (FRR and FAR). We note that BE does not use any matching score; instead, the FRR/FAR tradeoff may be achieved in some cases by varying the parameters of the BE scheme.

A different line of research is Cancelable Biometrics (CB), which is closer to a conventional biometric system. CB technologies apply a secret transform to the biometric. The transform can be invertible or not, and both the transformed template and the secret transform are stored. On verification, the same transform is applied to a fresh biometric sample, and two transformed templates are matched, where the output of CB verification is a Yes/No response.

Recently, biometrics is combined with traditional encryption schemes such as ElGamal-type encryption schemes to obtain secure authentication/fuzzy IBE systems that assume biometrics as public data. These systems provide provable security guarantees since they are designed according to a formal security model with a security reduction to a hard problem.

Before we describe various biometric cryptosystems, we review the definitions of the primitives required in these systems.

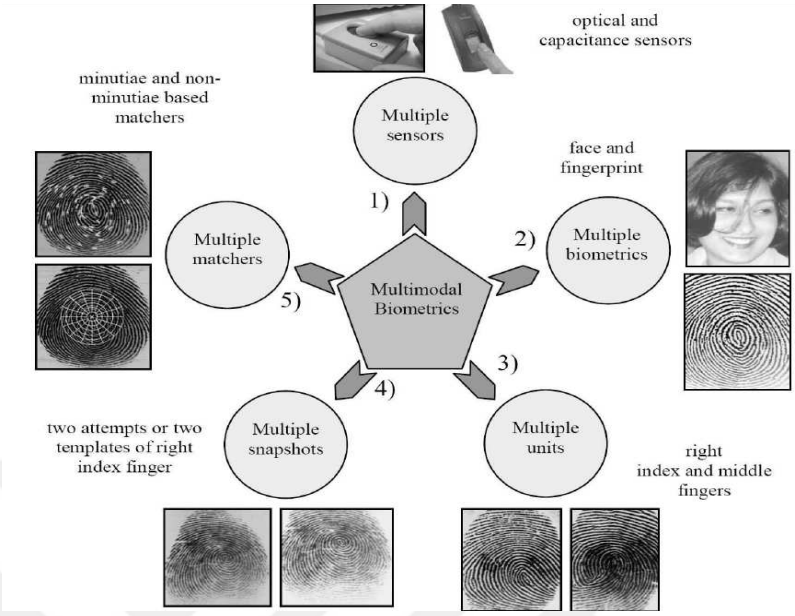

Figure 2.5: Scenarios in a multimodal biometric system [Ross and Jain, 2004]

2.2.1 Metric Spaces

Every biometric trait is assumed to have a metric space. A metric space is a set $\mathcal{M}$ with a distance function dis: $\mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$. $\mathcal{M}$ will always be a finite set, and the distance function will only take on integer values with $\mathsf{dis}\ (x, y) = 0$ if and only if $x = y$, and will obey symmetry $\mathsf{dis}\ (x, y) = \mathsf{dis}\ (y, x)$ and the triangle inequality $\mathsf{dis}\ (x, z) \leq \mathsf{dis}(x, y) + \mathsf{dis}(y, z)$. The similarity measures used in different biometric traits are given in [Dodis et al., 2004] as follows.

- *Hamming distance* denotes the number of symbol positions that differ between two biometrics w and w'. Here, $\mathcal{M} = \mathcal{F}^n$ for some alphabet $\mathcal{F}$, and $\mathsf{dis}(w, w')$ is the number of positions in which the strings w and w' differ. Hamming distance is a suitable similarity measure for Iris.

- *Set difference* denotes the size of the symmetric difference of two input biometric sets and is appropriate whenever the noisy input is represented as a subset of features from a universe of possible features. Here $\mathcal{M}$ consists of all subsets of a universe $\mathcal{U}$. For two sets (w, w') their symmetric difference is $w \triangle w' = \{x \in w \cup w' | x \notin w \cap w'\}$. The distance between two sets (w, w') is $|w \triangle w'|$. We will sometimes restrict $\mathcal{M}$ to contain only k-element subsets for some k. Set difference or set intersection is a suitable similarity measure for fingerprint minutia.

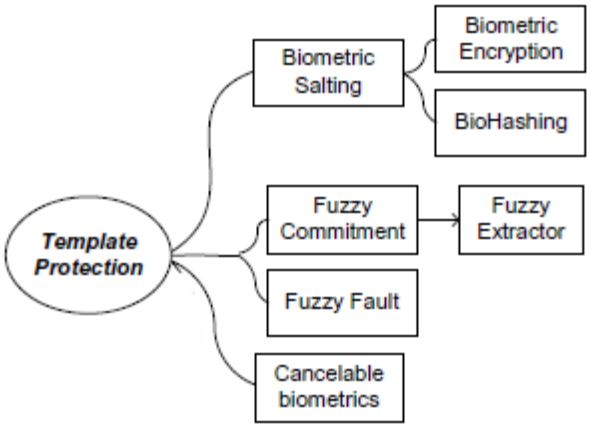

Figure 2.6: Systems for Biometric Template Protection

- *Euclidean distance*: Euclidean distance between points p and q is the length of the line segment connecting them ($\overline{\mathbf{pq}}$).

  In Cartesian coordinates, if $p = (p_1, p_2)$ and $q = (q_1, q_2)$ are two points, then the distance from p to q, or from q to p is given by:

  $$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} = \sqrt{\sum_{i=1}^{2} (q_i - p_i)^2}.$$

  Again, fingerprint minutia can be evaluated in this metric.

- *Edit distance* denotes the number of insertions and deletions needed to convert one string into the other, for instance, when the password is entered as a string, due to typing errors or mistakes made in handwriting recognition. Here, the distance between (w, w') is defined to be the smallest number of character insertions and deletions needed to transform w into w'. This is different from the Hamming metric because insertions and deletions shift the characters that are to the right of the insertion/deletion point. This metric is applicable for handwriting recognition.

For our constructions, we will focus on the first two distance functions.

2.2.2 Error-Correcting Codes

Error correcting codes (ECC) are used in communications, data storage, and in other systems where errors can occur. Biometric Encryption is a new area for the application

of ECC since some noise will be inevitably introduced into biometric samples during acquisition and processing. In general, two types of noise can be corrected in a biometric system. Under noise, each biometric feature can be perturbed by some distance and additionally, some of the features can be replaced. The first noise is denoted as white noise, and the second replacement noise [Li et al., 2006]. Some recent techniques attempt to correct the noise in the data using an ECC.

A code $\mathcal{C}$ is a subset $\{w_0, ..., w_{K-1}\}$ of K elements of $\mathcal{M}$. The map from i to w_i, which we will also sometimes denote by $\mathcal{C}$, is called encoding. For example, a binary block ECC, which is denoted (n, k, d), encodes k bits with $n > k$ bits by adding some redundancy. Those n-bit strings are called codewords; there are $K = 2^k$ of them in total, where k is the key length. The minimum distance (usually a Hamming distance is implied) between the codewords is d. If, at a later stage (in case of BE, on verification), the errors occur, the ECC is guaranteed to correct up to $(d-1)/2$ bit errors among n bits since for every $w \in \mathcal{M}$, there exists at most one codeword c in the ball of radius t around w. Namely, $\mathsf{dis}(w, c) \leq t$ for at most one $c \in \mathcal{C}$. We will use the term decoding for the map that finds, given w, the $c \in \mathcal{C}$ such that $\mathsf{dis}(w, c) \leq t$ (note that for some w, such c may not exist, but if it exists, it will be unique). Ideally, the legitimate users will have a number of errors within the ECC bound so that the ECC will decode the original codeword, and hence, the digital key. On the other hand, the impostors will produce an uncorrectable number of errors, in which case the ECC (and the BE algorithm as a whole) will declare a failure. Some ECCs may work in a soft decoding mode, that is, the decoder always outputs the nearest codeword, even if it is beyond the ECC bound. This allows achieving better error-correcting capabilities.

2.2.3 Secure Sketches

Introduced in [Juels and Wattenberg, 1999] and further refined in [Dodis et al., 2004], a secure sketch allows generating public data about its biometric input which does not reveal (much) information about the original data - but allows regenerating the exact input data if other similar data (such as another biometric reading at enrollment time) is passed to it.

Let $\mathcal{H}$ be a metric space with distance function dis. As in [Dodis et al., 2004], we define the predictability of a random variable A is $\max_a \Pr[A = a]$, and, correspondingly, min-entropy $\mathbf{H}_\infty(A)$ is -log($\max_a \Pr[A = a]$) (min-entropy can thus be viewed as the "worst-case" entropy). The min-entropy of a distribution tells us how many nearly uniform random bits can be extracted from it. The notion of "nearly" is measured by the statistical distance between two probability distributions.

A secure sketch scheme allows recovery of a hidden value $w \in \mathcal{H}$ from any value $w' \in \mathcal{H}$ close to this hidden value with the help of some public value PAR, which does not leak

too much information about w. A $(\mathcal{H}, m, m', t)$- sketch is a pair of functions (SS, Rec):

- The sketching function SS takes $w \in \mathcal{H}$ as input and returns the public parameter PAR in $\{0,1\}^*$ such that for all random variables W over $\mathcal{H}$ with min-entropy $\mathbf{H}_\infty(W) \geq m$, the conditional min-entropy is $\tilde{\mathbf{H}}_\infty(W|\mathsf{SS}(W)) \geq m'$.
- The reconstruction function Rec takes a vector w' and PAR as input and computes w if and only if $\mathsf{dis}(b, b') \leq t$ for any $\mathsf{PAR} = \mathsf{SS}(w)$.

Fuzzy Commitment

The fuzzy commitment scheme [Juels and Wattenberg, 1999] is an error tolerant authentication scheme which follows the above method with the use of a committed value. We observe that the fuzzy-commitment construction of Juels and Wattenberg based on error-correcting codes can be viewed as a (nearly optimal) secure sketch. The main goal is to protect the storage of biometric data involved in an authentication biometric system [Bringer et al., 2007a]. A biometric template must be in the form of an ordered bit string of a fixed length. A key is mapped to an Error Correcting Code (ECC) codeword of the same length as the biometric template. The codeword and the template are xored, and the resulting string is stored into helper data along with the hashed value of the key. On verification, a fresh biometric template is xored with the stored string, and the result is decoded by the ECC. If the codeword obtained coincides with the enrolled one (this is checked by comparing the hashed values), the k-bit key is released. If not, a failure is declared. An overview of this scheme is shown in figure 2.7.

Let C be an $(n, k, 2t+1)$ binary linear error correcting code in Hamming space. Let h be a cryptographic one-way function and $z = c \oplus b$, where c is a random codeword in C. Store $h(c)$, in the enrollment phase, together with z. From the corrupted codeword $c' = z \oplus b' = c \oplus (b \oplus b')$, one can recover c if the hamming distance $\mathsf{dis}_\mathcal{H}$ between b and b' is $\mathsf{dis}_\mathcal{H}(b, b') < t$. The authentication will be a success if the verification returns a codeword c' such that $h(c') = h(c)$.

An important requirement for such a scheme is that the value PAR should not reveal too much information about the biometric template b [Bringer et al., 2007a].

2.2.4 Biometric Key Generation/Locking/Release

Juels and Wattenberg [Juels and Wattenberg, 1999] introduce the fuzzy commitment scheme as a cryptographic primitive, which is specific for biometrics that can be represented as an ordered set of features. However, biometrics can be affected from two

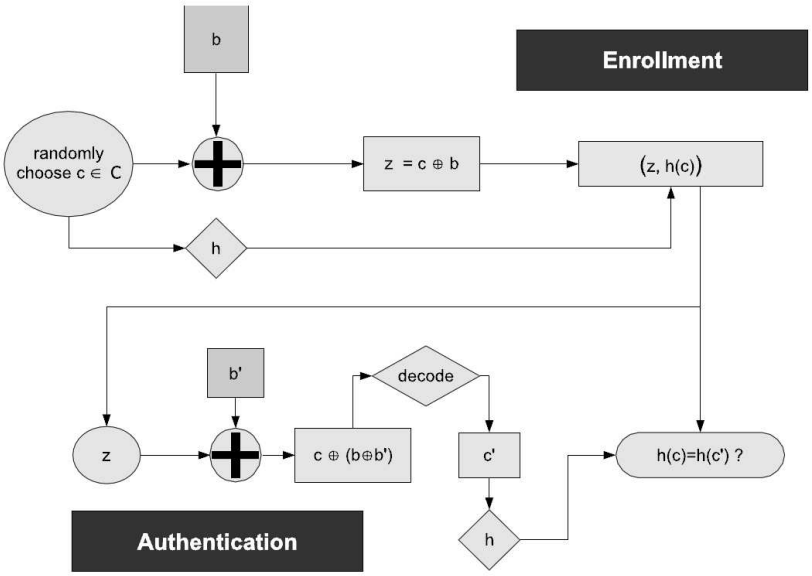

Figure 2.7: Fuzzy Commitment Scheme [Juels and Wattenberg, 1999]

types of noise, i.e. white noise that represents the slight perturbation of each feature and the replacement noise caused by the replacement of some features. Thus, Juels and Sudan have developed the *fuzzy vault* [Juels and Sudan, 2006], which assumes that biometrics consists of an unordered set of features and is designed for the set difference metric. Hence, fuzzy vault construction of Juels and Sudan can be viewed as a secure sketch in the set difference metric. Specifically, fuzzy vault is a key locking system that hides an encoded secret among some chaff points, where the secret key is encoded as the coefficients of a polynomial that is evaluated at the biometric feature locations such as fingerprint minutia locations. Implementation of fuzzy vault for fingerprints are given in [Clancy et al., 2003] and [Uludag et al., 2005, Uludag and Jain, 2006], the latter two use CRC codes for error correction in order to have increased tolerance to biometric intra-class variations.

Fuzzy Vault

Fuzzy vault is a cryptographic primitive used to lock a secret data (i.e. cryptographic key) using an unordered set of locking elements (i.e. biometric features) such that a user who possesses a substantial amount of the locking elements will recover the secret. A fuzzy vault scheme $\mathcal{V} = \mathcal{V}(k, t, r, \mathbb{F}_q)$ consists of three phases:

- Setup: Choose a finite field $\mathbb{F}_q$ of order q and set the parameters of t and r,

where t is the number of genuine points $\mathcal{G}$ hidden in the vault and r denotes the cardinality of the vault points $\mathcal{V} \subset \mathbb{F}_q^2$.

- Lock: A secret data S such as an 128-bit AES key is hidden in the vault as follows.

  1. A secret $S = \{S_i\}_{i=1}^{k}$ is identified with a polynomial $f \in \mathbb{F}_q[X]$ of degree $k-1$ by encoding S as the coefficients of f.
  2. To hide f using $t > k$ biometric features (i.e. locking set $\mathcal{L}$) identified with elements of $\mathbb{F}_q$, the locking set $\mathcal{L}$ (i.e. minutia locations) is evaluated at f resulting in the genuine set $\mathcal{G}$.
  3. Generate chaff points $\mathcal{C}$ uniformly at random such that $\mathcal{C} \subset \mathbb{F}_q/\mathcal{G}$.
  4. The vault is $\mathcal{V} = \mathcal{V}(k, t, r, \mathbb{F}_q) = \mathcal{G} \cup \mathcal{C}$

  Thus, the axis of abscissae determines information relevant to minutiae while the ordinate associates them with the secret. The template used to extract points from the vault must have neglible nonlinear distortion and be aligned modulo affine transform [Nandakumar et al., 2007a]. Chaff points lie with minutiae locations **and** their association to the secret. All data lie in the finite field $\mathbb{F}_q$. The degree k of the polynomial and the number of genuine points t are hidden in the vault. $\mathcal{G}$ uniquely determines f and $\mathcal{L}$. Since using false points yields to a complete different secret, to give extractors a chance, minutiae locations of chaff points must have sufficient distance to genuine minutiae locations, depending on the image quality. Thus, one can not add arbitrary many chaff points to the vault. Otherwise, genuine minutiae locations appear as isolated points. Thus, we generate chaff points $\mathcal{C}$ uniformly at random with the above considerations.

  As different from [Juels and Sudan, 2006], the authors of [Uludag et al., 2005, Uludag and Jain, 2006] append a 16-bit CRC checksum to the secret data and encode the coefficients of the polynomial using a 144-bit data block. This way, increased tolerance to biometric intra-class variation is obtained.

- Unlock: To open the vault one needs to know $k+1$ genuine points of the vault and compute its interpolating polynomial. Genuine and false minutiae locations are determined by the axis of abscissae. Given an unlocking set $\mathcal{L}'$ and a vault $\mathcal{V}$, the algorithm returns the secret data as follows. Since vault points $\mathcal{G}'$ are extracted using a fresh biometrics, this set may contain points which are not genuine points (i.e. randomly matching chaff points). Thus, one may choose wrong extracted points to interpolate f. If not too many points are erroneously extracted most ways to interpolate f will lead to the correct f by extracting k genuine vault points using a second impression of a similar biometrics $\mathcal{L}'$ such that $|\mathcal{L}' \cap \mathcal{L}| \geq k$ and computing its interpolating polynomial to decode S.

In [Uludag and Jain, 2006], the CRC based error detection is used to identify the correct polynomial and thus the correct secret.

Theorem 2.1. *(Coding theoretical interlude). Let* $(x_1, y_1), \ldots, (x_t, y_t) \in \mathbb{F}_q^2$ *be distinct points where* $(k+t)/2$ *lie on a polynomial* $f \in \mathbb{F}_q[X]$ *of degree* k*. There is an* $O(t^3)$ *time algorithm which discovers* f*.*

Corollary 2.1. *Let* $\mathcal{G}'$ *consists of* t *extracted vault points comprising at least* $(k+t)/2$ *genuine points. Then the secret* S *can be efficiently and safely discovered.*

The first implementation of fuzzy vault for fingerprints together with a security analysis is presented in [Clancy et al., 2003], where the authors claim that an attack requires 2^{69} operations. Next, Uludag et al. [Uludag et al., 2005, Uludag and Jain, 2006] present two implementations of fuzzy vault for fingerprints, where the system also includes helper data constructed from the high curvature points of the fingerprint minutia, which does not leak any information about the minutia locations and is used for easing the alignment of the query fingerprint to the original template. Finally, Yang et al. [Yang and Verbauwhede, 2005] describe another implementation with variations on the vault parameters depending on the template quality.

Attacks against the Fuzzy Vault

The first attacks on the fuzzy vault are described in [Adler, 2005, Chang et al., 2006] by identifying the chaff points based on the non-randomness of the fuzzy vault scheme. Apart from that, the broad categorization of the attacks consists of known-plaintext and ciphertext-only attacks, where the former assumes that an attacker can gain access to the secret key hidden in the fuzzy vault, which leads to the biometric template by verifying the secret polynomial on the points in the vault. The second group of attacks, namely ciphertext-only attacks do not require any insider knowledge. For fuzzy vault, brute force attack [Mihailescu, 2007, Mihăilescu et al., 2009] or collusion attacks [Scheirer and Boult, 2007] (i.e. different instances of the vault encoded with the biometrics of the same user is enough to obtain the biometric template of that user), are the two main attacks. Also, Scheirer and Boult present three different attacks (including the collusion attacks) against the fuzzy vault and biometric encryption [Scheirer and Boult, 2007] and in view of that, they proposed revocable biotokens [Boult et al., 2007] and its implementation as bipartite biotokens in [Scheirer and Boult, 2009]. Further implementations of the collusion attack with different parameters is given in [Poon and Miri, 2009].

Collusion Attacks: Collusion attacks (also defined as attacks via record multiplicity [Scheirer and Boult, 2007]) can be applied if an adversary has access to two or more vault instances generated from the same user for different applications, which may be

implemented for hiding the same secret or different secrets. Additionally, [Poon and Miri, 2009] analyzed the case that an adversary has access to n different vaults that lock the same secret using different biometrics. The main idea of this attack is that availability of two vaults locked with the same key enables an attacker to identify the genuine points in the two vaults and decode the vault, which is the most likely situation to occur in practice. Thus, the fuzzy vault scheme does not provide diversity and revocability. A detailed analysis on the security of the fuzzy vault against collusion attacks is presented in [Poon and Miri, 2009], where the authors also suggest some countermeasures against this attack by applying a keyed one-way transform of the locking set $\mathcal{L}$, or generating the chaff points depending on the locking set or by using the maximal vault size, namely the vault will contain the whole $\mathbb{F}_q \backslash \mathcal{L}$ as the abscissae values of the chaff points.

Stolen key inversion attack: In this attack, an adversary somehow recovers the key embedded in the vault through means other than an attack against the biometric template, thus he can decode the vault to obtain the biometric template. It is noted that even if an encrypted version of the secret is hidden in the vault, insider attacks are still possible [Scheirer and Boult, 2007].

Blended substitution attack: Since the vault contains a large number of chaff points, it is possible for an adversary to substitute a few points in the vault using his own biometric features resulting in a successful authentication of the genuine user and the adversary. This attack can be prevented using digitally signed templates [Scheirer and Boult, 2007].

Brute force attack: In [Mihailescu, 2007], the author presents a brute force attack against the fuzzy vault implementation of Uludag et al. and this attack is further investigated in [Mihăilescu et al., 2009] with a proposal to improve the security of fingerprint fuzzy vault using additional minutia information, in particular the use of larger parameters for the vault with increased biometric data, for instance more fingers for identification. The complexity of the attack is found much below cryptographic security, i.e. 2^{55} for Clancy et al. [Clancy et al., 2003] and 2^{37} for Uludag et al. [Uludag et al., 2005], where the additional CRC checksum allows the attacker to verify whether he has found the correct secret and the unlocking set [Mihailescu, 2007].

The attack for a fingerprint vault $\mathcal{V} = \mathcal{V}(k, t, r, \mathbb{F}_q)$ is as follows:

1. Choose k distinct points from the vault $\mathcal{V}$ uniformly at random;
2. Compute the unique degree $k-1$ polynomial $f \in \mathbb{F}_q[X]$ interpolating them;
3. If the graph of f contains t vault points, output f; otherwise go to step 1;

Let $\mathcal{V} = \mathcal{V}(k, t, r, \mathbb{F}_q)$ be a fuzzy fingerprint vault. Clancy, Kiyavash, and Lin determined optimal security parameters for fingerprints image having 251×251 pixels as

Table 2.1: Implementation of Brute Force Attack

System	Parameters	Complexity
Clancy et al. [Clancy et al., 2003]	$\mathcal{V}(14, 38, 313, \mathbb{F}_{251^2})$	2^{55}
Uludag et al. [Uludag et al., 2005]	$\mathcal{V}(8, 24, 224, \mathbb{F}_{2^{25}})$	2^{37}
Uludag et al. [Uludag et al., 2005]	$\mathcal{V}(8, 24, 224, \mathbb{F}_{2^{108}})$	2^{37}

$k = 14$, $t = 38$, and $r = 313$ where $q = 251^2$. Later, Yang and Verbaudwhede describe an implementation where t and k vary depending on the template quality. Finally, Uludag and Jain describe the usage of helper data for easing image alignment. They use CRC codes instead of RS codes and describe experiments with $k = 8$, $t = 24$, and $r = 224$. Table 2.1 presents an overview of the attack against these implementations of fuzzy vault.

Countermeasures: To counter the first three attacks, Nandakumar et al. [Nandakumar et al., 2007b] proposed a hybrid approach where (i) biometric features are first salted based on a user password to prevent collusion attacks, (ii) vault is constructed using the salted template and (iii) the vault is encrypted using a key derived from the password to prevent blended substitution and stolen key inversion attacks. In order to prevent brute force attack, the authors of [Mihăilescu et al., 2009] propose a new primitive called fuzzy vault with quiz, where the security is improved using additional minutiae information i.e. its orientation. Let $(X, Y) \in \mathbb{F}_q^2$ genuine vault point and α its orientation in a granularity of π/n, where n is small. The quiz function uses α to change the Y-coordinate of genuine vault points by choosing a j that encodes a transformation to shift Y, where $j\frac{\pi}{n} = \alpha - \beta \bmod \pi$. Thus, genuine vault points are now of the shape (X, Y', β) where $Y' = T(Y)$ and chaff points are also equipped with a random β. The security against this attack increases by a factor 2^{kb} where k is the secret polynomial degree and $b = \lceil \log_2(n) \rceil$ [Mihăilescu et al., 2009].

However, fuzzy vault with quiz is still vulnerable to collusion attack since the transformation is only on the Y coordinate of the vault. Thus, additional one-way transformation on the locking set $\mathcal{L}$ or the use of maximal vault size is required to avoid these attacks.

Fuzzy Extractors

Linnartz and Tuyls [Linnartz and Tuyls, 2003] defined and constructed a primitive very similar to a fuzzy extractor, which focuses on the continuous space $\mathbb{R}^n$ and assumes a particular input distribution (typically a known, multivariate Gaussian). In the same paper for secure sketch, [Dodis et al., 2004] introduce the concept of a fuzzy extractor, which can be viewed as a generalization of the notion of a "shielding function" of

[Linnartz and Tuyls, 2003] on discrete metric spaces. A fuzzy extractor allows deriving a cryptographic key with uniform randomness from a biometric input that stays the same even if the input changes slightly. In this way a biometric system could be seen as key-generating instead of key-binding, as the biometric data itself can be used as source for a key, instead of binding a key as in the case of the fuzzy commitment scheme. For fuzzy extractors, a similar attack based on the reusability of the same (or a noisy variant) of the biometrics for multiple extractions of independent public strings is described due to improper fuzzy sketch constructions or wrongly chosen error correcting codes. From these public strings, an attacker can exactly regenerate the corresponding secret keys that are output by the fuzzy extractor. In particular, [Boyen, 2004] presented the first attack on the fuzzy extractors based on the reusability of the same noisy secret (biometrics) in the presence of both outsider and insider attackers. The author describes how to improve security of fuzzy extractors when they are used multiple times for the same biometrics, by adding an additional permutation.

2.2.5 Cancelable Biometrics

Another way to secure biometric templates besides helper data methods is to use some kind of transformation which does not allow retrieving the original biometric features from the transformed template. Because the transformation involves additional data such as a seed value to a pseudo random number generator, new templates can be issued for the same biometric data and therefore the templates are cancelable. At enrollment time, such a transformation is applied either directly to the biometric data obtained from the sensor or to the extracted features. In either case the resulting biometric template does not allow restoring the original biometric data as long as the transformation is not invertible. At authentication time, the same transformation is applied to the supplied biometric data - and matching against the stored template is performed with the transformed data.

Informally, the idea of cancelable biometrics (CB) is to transform biometric data with an irreversible transformation and to perform the matching directly on the transformed data allowing the use of existing feature extraction and matching algorithms. Formally, given two biometric data w and w', the matching score will be computed directly on transformed data by $m(f(w), f(w'))$, where m denotes the similarity measure and f be a transformation that does not degrade the matching performances too much. The three properties of f are: (1) w and $f(w)$ do not match together; (2) For two different transformations f_1 and f_2, $f_1(w)$ and $f_2(w)$ do not match together; (3) A pre-image of $f(w)$ is hard to compute.

Besides, [Hirata and Takahashi, 2009, Cambier et al., 2002] proposes another method for cancelable biometrics, where the biometric information is masked by a random

number, and then, the masked information is stored in the server as a template. The random number used for masking is needed to have a certain level of entropy, and to be stored in a smart card carried by the authorized user. Biometric information presented at the authentication phase is also masked by the same random number, and compared with the template (i.e. biometric information masked by the random number) [Sakashita et al., 2009]. This way, biometric data stored at the server is protected through this transformation and biometrics can be updated by changing the transformation function or the randomness. This system also prevents the user's traceability across different biometric databases. Example systems employing a high entropy randomness stored in a smart card for cancelable biometrics are given in [Hirata and Takahashi, 2009, Cambier et al., 2002, Sakashita et al., 2009]. Even if the (masked) templates are compromised, no biometric information will leak out. Also, in this method, no information except for the random number is stored in a smart card, which is assumed as a tamper proof smart card.

BioHash

An ordered biometric feature set is transformed into a new space of a lower dimension by generating a random set of orthogonal vectors and obtaining an inner product between each vector and the biometric feature set [Li and Jain, 2009]. The result (called Biohash) is binarized to produce a bit string. The random feature vectors are generated from a random seed that is kept secret, for example, by storing it in a token. The key is bound to the Biohash via Shamir secret sharing with linear interpolation, or by using a standard Fuzzy Commitment scheme. Very good FRR/FAR numbers were obtained, however, in an unrealistic non-stolen token scenario. Biohashing is referred more often as a CB scheme where Biohashes are matched directly, that is, without the key binding.

Bipartite Biotokens

Another CB approach that is claimed to be secure against collusion and brute force attack is bipartite biotoken [Scheirer and Boult, 2009], which is implemented for fingerprints and works in the finite field $\mathbb{F}_{2^8}$ to encode the secret and the locking set. First, the raw biometric data v'' (the distance d and angles a_1, a_2 of each feature) is transformed via a translation t and scaling s. Next, the transformed biometrics v' is separated to fraction (residual part) r and integer (stable) part g using a reflected modulus $rmod$ that does not increase the distance between points. The amount of stable/unstable data is a function of the biometric modality. In [Boult et al., 2007], the separation parameter E that depends on the expected variations on v'' separates the stable (integer) part $g = int(v'/E)$ and non stable part $r = rmod(v', E)$ using a

simple mod operation:

$$r = rmod(v', E) = v'\%2E \text{ if } (v'\%2E) < E \text{ and}$$
$$r = rmod(v', E) = 2E - (v'\%2E), \text{ otherwise}$$

The stable part g can be hashed or encrypted using public key cryptography to obtain w, whereas the residual r is left in clear. The authors use the term hash as a general concept since any checksum such as MD5, SHA1 or for many embodiments a traditional CRC is sufficient. Finally, multiple embedded polynomials (i.e. four key columns) allows for encoding of larger secret keys.

1. Store the secret S together with E bytes of error correction as B bytes
2. The RS polynomial f encodes B.
3. Compute the distance d and angles a_1, a_2 per row.
4. Transfrom d, a_1, a_2 and split into the stable parts sd, sa_1, sa_2 and the residual parts rd, ra_1, ra_2.
5. Hash the 24 bits of sd, sa_1, sa_2 into an 8-bit value i, hash i to an 8-bit value h.
6. Evaluate h for different polynomials resulting in values rs_1, rs_2, rs_3, rs_4 to support larger keys.
7. Encode the stable part to w, leave the residual part r in clear.
8. Randomize the location of w i.e. rs_1, rs_2, rs_3, rs_4 follow w via circular map.
9. Store the protected parts $w, rs_1, rs_2, rs_3, rs_4, i$ and residuals in the gallery except for h.

Lock Algorithm

1. To unlock the secret key, the server sends the biotoken over an insecure channel.
2. To match the gallery to the probe, create all the stable and non-stable fields and compute h for each row.
3. Find a matching w and check whether the residuals are within threshold.
4. With w identified, extract the polynomial values rs_1, rs_2, rs_3, rs_4.
5. Select a set of correct matched rows from the potentially matching rows.
6. Recover the secret data by Reed Solomon decoding of the polynomial values.
7. Since four different polynomials are evaluated at h at each row, larger keys can be encoded by concatenation.

Unlock Algorithm

2.2.6 Distributed Biometric Remote Authentication

Biometric authentication systems are classified as remote, local or match-on-card (MOC) systems, where the latter two are closed self-contained systems performing the necessary operations and storage of the biometric profile information in a controlled and trusted environment. Recently, distributed biometric remote authentication (DBRA) systems are designed in the framework of cryptographic security, where data acquisition and feature recognition are performed by separate sub-systems, which communicate over an insecure channel. This type of scenario may occur, for instance, if one intends to use biometric authentication to access privileged resources over the Internet [Bar-

bosa et al., 2008]. The main difference of these systems compared to other biometric cryptosystems is that they assume biometrics as public data. In the following chapter, we present recent results in this area in detail.

2.2.7 Fuzzy IBE/IBS

Another application of biometrics in the context of public key cryptography is integration of biometrics into an identity based encryption/signature scheme. In Eurocrypt'05, Sahai and Waters [Sahai and Waters, 2005] proposed a new Identity Based Encryption (IBE) system called fuzzy IBE that uses biometric attributes as the identity instead of an arbitrary string like an email address. Since biometrics can identify a person uniquely, it makes sense to use them as the public key in an identity-based encryption scheme. Encryption using biometric inputs as identities is provided with fuzzy IBE, since the error-tolerance property of a fuzzy IBE scheme is precisely what allows for the use of biometric identities, which inherently will have some noise each time they are sampled. Biometrics usually consist of noisy data, i.e. two measures w and w' of the same biometric are not completely the same. However, the main feature of fuzzy IBE is the construction of the secret key based on the biometric data of the user which can decrypt a ciphertext encrypted with a slightly different measurement of the same biometrics. Specifically, fuzzy IBE allows for error tolerance in the decryption stage, where a ciphertext encrypted with the biometrics w could be decrypted by the receiver using the private key corresponding to the biometrics w', provided that w and w' are within a certain distance of each other according to the "set overlap" (i.e. set intersection) distance metric. This is actually the same metric used in fuzzy vault, where Bob, using an unordered biometric set w, can unlock the vault (and access the hidden secret data) only if w overlaps with w' to a large extent. This is in contrast to regular IBE schemes, which view the identity of a person as a unique string like an e-mail address, thus they are not suitable for error-prone identities. Thus, fuzzy IBE combines the advantages of IBE with using biometrics as an identity, where IBE avoids the need for an online Public Key Infrastructure (PKI), which is the most inefficient and costly part of public key encryption. The use of biometrics as the identity in the framework of IBE simplifies the process of key generation at the Private Key Generator (PKG). Since biometric information is unique, unforgettable and non-transferable, the user only needs to provide his biometrics at the PKG to obtain his secret key instead of presenting special documents and credentials to convince the PKG about his identity. Similar to the distributed biometric remote authentication, biometrics is assumed as public data, thus the compromise of the biometrics does not affect the security of the system as opposed to the fuzzy vault. Also, biometrics is attached to the user, hence the public key of the user is always with him to be used for encryption during an ad hoc meeting. Finally, biometric data could be easily integrated with fuzzy IBE due

to its error tolerance property, which is required for the noisy nature of biometrics. Besides, fuzzy IBE could be applied in the context of Attribute-Based Encryption [Pirretti et al., 2006, Sahai and Waters, 2005], where the sender encrypts data using a set of attributes such as {university, faculty, department} and the ciphertext could only be decrypted if the receiver has the secret key associated to all of these attributes or sufficient number of them.

In current fuzzy IBE schemes, the private key components are generated by combining the values of a unique polynomial evaluated on each attribute with the master secret key. This way, different users, each having some portion of the secret keys associated to the attributes of a given ciphertext c cannot collude to decrypt c, which is defined as collusion resistance. Clearly, this construction is not possible for any IBE system, but only for the ones that provide some error-tolerant encryption through the use of specific primitives such as bilinear pairings and Shamir's secret sharing scheme. In the second part of this chapter, we will review the base schemes of the recently described fuzzy IBE systems.

Part II : Background in Cryptographic Tools

2.3 Public Key Encryption

In 1976, the authors of [Diffie and Hellman, 1976] invented public key encryption (PKE), also called as asymmetric encryption, where the receiver generates a pair of keys pk and sk that will be used for encryption and decryption, respectively. The receiver will publish the encryption key pk and store privately the decryption key sk. With this mechanism, it is obvious that anyone can encrypt a message m using pk and generate the ciphertext c, whilst only the receiver can decrypt this c using his private key sk. The formal definition is as follows.

Definition 2.1. (PKE). *A Public Key Encryption scheme* Π = (Keygen, Encrypt, Decrypt) *with message space* M *consists of three algorithms.*

- Keygen: The key generation algorithm samples a keypair (pk, sk), which is denoted by $(pk, sk) \leftarrow$Keygen(k). Here, $k \in \mathbb{N}$ (also written as 1^k) is called as the security parameter, that measures the degree of security we want to achieve. k denotes the key length, (i.e. the bit-length of the RSA modulus or the order of the group for group-based cryptosystems).
- Encrypt: This algorithm encrypts a message $m \in M$ using the pk of the receiver and outputs the ciphertext c, which is denoted by $c \leftarrow$Encrypt(pk, m).

- Decrypt: The recipient can recover the message m by decrypting c using his secret key sk as $m \leftarrow \mathsf{Decrypt}(sk, c)$.

We require perfect correctness of the scheme, i.e. $\mathsf{Decrypt}(sk, \mathsf{Encrypt}(pk, m)) = m$ for all $m \in M$ and all possible $(pk, sk) \leftarrow \mathsf{Keygen}(k)$.

An asymmetric encryption scheme could be either deterministic or probabilistic. A deterministic scheme means that for a fixed encryption key, a given plaintext will always be encrypted in the same ciphertext. When using a deterministic encryption scheme, it is easy to detect when the same message is sent twice while processed with the same key. So, in practice, we prefer encryption schemes to be probabilistic, namely, encryption of the same message results in different ciphertexts. In other words, for a plaintext we require the existence of several possible ciphertexts, the number of ciphertexts is greater than the number of possible plaintexts. Probabilistic encryption was introduced in order to properly define different security levels, in particular, semantic security. We describe these different security levels as follows.

2.4 Security Notions of PKE

A PKE scheme is secure based on a security notion if an adversary A has negligible probability in a specific game between a challenger and an adversary A, where A is modeled as a probabilistic polynomial-time (PPT) Turing machine with possibly access to some oracles. What we mean with the term "negligible" is defined as follows:

Definition 2.2. (Negligible Function). *A function* $negl : \mathbb{N} \to \mathbb{R}$ *is defined as negligible if for any constant* c*, there exists* $k_0 \in \mathbb{N}$ *with* $k > k_0$ *such that* $negl(k) < (1/k)^c$*. In other words,* $negl$ *eventually vanishes faster than any given polynomial.*

As a result, all algorithms (i.e. any potential adversary) are PPT in the security parameter k and the success of the adversary should be negligible in k. The security notion, i.e. the adversary model, is defined by combining the means and the goals of the adversary A.

2.4.1 Means of the Adversary

Informally, means of the adversary defines the access of A to some oracles, which are listed as below.

- Chosen-plaintext attacks (CPA): A only knows the public key of the user, which implies that he may encrypt any plaintext of his choice.

- Plaintext-checking attacks (PCA): A can check whether a given ciphertext would be decrypted as a given plaintext.
- Chosen-ciphertext attacks (CCA): A may ask for decryption of ciphertexts of its choice except that A cannot ask for the decryption of the challenge ciphertext.

2.4.2 Goals of the Adversary

Briefly, the goals of the adversary A is listed as below.

- Unbreakability: A wants to recover the secret key of the user.
- One-wayness (OW): A wants to be able to decrypt any ciphertext. The encryption scheme is said to be one-way if no efficient attacker is able to decrypt a random ciphertext with non-negligible probability. By a random ciphertext, we mean the ciphertext of a plaintext chosen uniformly at random over the plaintext space.
- Non-malleability (NM): Given a list of ciphertexts, A wants to build a new ciphertext whose plaintext is related to the plaintexts of the input ciphertexts.
- Indistinguishability (IND): A wants to output two distinct messages m_0 and m_1 such that if a challenger encrypts either m_0 or m_1, A would be able to tell which message was encrypted. Clearly, if the encryption scheme is deterministic, there is always a trivial distinguisher: one could select any pair of distinct messages m_0 and m_1, and by encrypting both m_0 and m_1, one could tell which one corresponds to the challenge ciphertext. This implies that probabilistic encryption is necessary to satisfy strong security notions.

If we combine the means and the goals of the adversary, we obtain the following security notions that are considered as the standard notions for the security of PKE schemes. Here, $\Pi = (\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ with security parameter 1^k denotes a PKE scheme.

2.4.3 One-Wayness

Experiment $Exp_{\Pi,A}^{OW-ATK}(k)$
$(pk, sk) \leftarrow \mathsf{Keygen}(1^k)$
$s \leftarrow A_1^O(pk)$
$m \leftarrow M$
$c = \mathsf{Encrypt}(pk, m)$
$m' \leftarrow A_2^O(s, c)$

Here, depending on the attack model ATK, the oracle O represents a decryption oracle or a plaintext checking oracle.

Definition 2.3. *A PKE scheme* Π *is said to be secure in the sense of OW-ATK, ATK* $\in \{CPA, PCA, CCA\}$ *if for all PPT algorithms* $A = (A_1, A_2)$

$$Adv_{\Pi,A}(k) = Pr[m = m'] < negl(k)$$

2.4.4 Indistinguishability

For some encryption schemes that are only one-way secure, it may be easy to compute partial information about the plaintext: For instance, an RSA ciphertext c leaks one bit of information about the plaintext m, namely, the socalled Jacobi symbol. As a higher security level, semantic security was introduced in [Goldwasser and Micali, 1982], which is unavailable without probabilistic encryption. A probabilistic encryption is semantically secure if the knowledge of a ciphertext does not provide any useful information on the plaintext to some adversary with only polynomial computational power. More formally, for any function $\mathbf{f}$ and any plaintext m, the probability to guess $\mathbf{f}(m)$ (knowing $\mathbf{f}$ but not m) does not increase if the adversary knows a ciphertext corresponding to m. This might be thought of as a kind of perfect secrecy in the case when we only have polynomial resources [Fontaine and F.Galand, 2007]. Due to the difficulty to formalize semantic security, the equivalent notion of indistinguishability is used in practice.

Experiment $Exp^{IND-ATK}_{\Pi,A}(k)$
$(pk, sk) \leftarrow \mathsf{Keygen}(1^k)$
$(m_0, m_1, s) \leftarrow A_1^O(pk)$
$b \overset{R}{\leftarrow} \{0, 1\}$
$c = \mathsf{Encrypt}(pk, m_b)$
$d \leftarrow A_2^O(m_0, m_1, s, c)$
if $d = b$ return 1 else return 0

Here, depending on the attack model, A has access a plaintext checking oracle or a decryption oracle except for the query on the challenge c. $|m_0| = |m_1|$ is compulsory to prevent the attacker from breaking the scheme trivially.

Definition 2.4. *A PKE scheme* Π *with security parameter* 1^k *is said to be secure in the sense of IND-ATK with ATK* $\in \{CPA, PCA, CCA\}$ *if for all PPT algorithms* $A = (A_1, A_2)$

$$Adv_{\Pi,A}(k) = Pr[Exp_{\Pi,A}(k) = 1|b = 1] - Pr[Exp_{\Pi,A}(k) = 1|b = 0] < negl(k)$$

A deterministic asymmetric encryption scheme cannot be semantically secure since it cannot be indistinguishable: the adversary knows the encryption function, and thus can compute the single ciphertext corresponding to each plaintext.

2.4.5 Non-Malleability

Consider the following experiment.

Experiment $Exp_{\Pi,A}^{NM-ATK}(k)$
$(pk, sk) \leftarrow \mathsf{Keygen}(1^k)$
$(s, M) \leftarrow A_1^O(pk)$
$x_0, x_1 \leftarrow M$
$y^* = \mathsf{Encrypt}(pk, x_1)$
$R, \mathbf{y} \leftarrow A_2^O(y^*, s)$
$\mathbf{x} = \mathsf{Decrypt}(\mathbf{y}, pk)$
if $y^* \notin \mathbf{y} \wedge R(x_b, \mathbf{x})$
return 1
else return 0

Here, $\mathbf{y}$ denotes a vector of ciphertexts and no component of $\mathbf{y}$ should be equal to y^*. The adversary A hopes that the relation $R(x_1, \mathbf{x})$ holds with a probability significantly more than the probability that the relation $R(x_0, \mathbf{x})$ holds. Here, x_1 with $|x_1| = |x_0|$ is also a plaintext chosen uniformly at random from M and independent of x_0.

Definition 2.5. *A PKE scheme* Π *with security parameter* 1^k *is said to be secure in the sense of NM-ATK with* $ATK \in \{CPA, PCA, CCA\}$ *if for all PPT algorithms* $A = (A_1, A_2)$*, every relation* R *computable in polynomial time such that*

$$Adv_{\Pi,A}(k) = Pr[Exp_{\Pi,A}(k) = 1|b = 1] - Pr[Exp_{\Pi,A}(k) = 1|b = 0] < negl(k)$$

2.5 Hash Functions

Cryptographic hash functions play a role in data integrity and message authentication. A hash function is a function from strings of arbitrary finite bit length to strings of n bits for some fixed integer n. A hash function is necessarily many-to-one, but for cryptographic applications n will be in the range of 128 to 256 bits and should satisfy much stronger conditions than are required for typical hashing purposes. These conditions are classified by the difficulty of solving certain problems, as presented below.

Preimage resistance:	Given $H(x)$ find y such that $H(y) = H(x)$.
Second preimage resistance:	Given x, find y such that $H(y) = H(x)$.
Collision resistance:	Find x, y such that $H(x) = H(y)$

The basic idea of the design of cryptographic hash functions consists in splitting the message to be hashed into blocks of fixed length, and hashing them block by block with a compression function. Moreover, this structure is the origin of the hash functions massively used in cryptography, i.e. SHA-1 and MD5.

2.6 Security Reduction

In order to guarantee the security of a PKE scheme, the precise definition of the security notion for the cryptographic scheme must be stated. To prove that a scheme can achieve the given security notion, one needs to describe a polynomial reduction which is an algorithm that uses an adversary A on a given scheme, in order to solve a hard cryptographic problem with almost the same success and time as the adversary has against the scheme. Basically, if any efficient algorithm solves the problem only with negligible probability, we say the problem is hard. For instance, the unbreakability, i.e. "public key inversion problem" should be hard on average, but not only hard in the worst case, where the latter means there exists hard instances. There is a small number of candidates for computational problems which are hard on average such as factoring $n = pq$, where p and q are large primes. For instance, using a reduction one is able to prove that if an adversary can break a system, then this yields a fast algorithm for factoring. As a result, security proof of a cryptographic scheme is constructed using an attacker running against the scheme, as a sub-part of an algorithm that breaks the underlying assumption. Secure schemes are defined to be those in which the advantage of this computationally bounded attacker is negligible. We denote this reduction as follows. Let P_1 and P_2 be two computational problems. $P_1 \leq_P P_2$ if we have an oracle (or efficient algorithm) to solve problem P_2. We then use this oracle to give an efficient algorithm for problem P_1.

The assumption on the primitive (or the problem P) states that no polynomial time algorithm exists to solve the underlying primitive (or problem P). An assumption is basically the condition to guarantee the security notion. Thus, security relies on computational assumptions such as one way functions (integer factorization, discrete logarithms in a finite field), trapdoor one way function or permutations (RSA), where the inversion of the one way function is hard without knowing the trapdoor. However, assuming this intractability of some computational problem is not enough for efficient cryptographic primitives, which require hard decisional problems. Moreover, the assumption based on the problem P is weaker than the assumption on P' if the problem P is stronger than problem P'. Hence, to obtain more secure cryptographic schemes, weaker assumptions are required to be used. If the probability of breaking the scheme by the adversary is closer to the probability of the problem being solved by an algorithm, the reduction is tighter, which is a measure of provable security. In case of a

non-tight reduction, larger key sizes provide the same security as a scheme with a tight security reduction. It can be argued that a proof relying on a stronger assumption and resulting in a tight reduction can be preferred to a loose reduction from a potentially weaker assumption.

2.6.1 Random Oracle Model

In a security reduction, Random Oracle Model (ROM) is used when certain parts of the cipher such as the hash functions are modeled as random functions. This model requires some additional assumptions to be made, and the security reduction is valid only if these assumptions are valid. In this model, a hash function $h : D \rightarrow R$ is chosen uniformly at random from the set of functions from D to R. Moreover, h is not given by a formula or algorithm to compute its outputs. Thus, the only way to compute the value $h(x)$ of some $x \in D$ is through a call to the function oracle. This can be assimilated to looking up a huge codebook consisting of values in D and corresponding values in R such that for each possible $x \in D$, there exists a completely random value $h(x) \in R$. The only necessary thing in using a security reduction in the ROM is to replace the random function by a particular hash function.

However, when an encryption or signature scheme is actually used in practice, a particular hash function must be specified, and so the assumption used in the random oracle model is not valid. Besides, ROM does not model real life, where there are no random functions instead a single fixed hash function such as MD5 or SHA-1 is used. Therefore, even if an adversary designed with the knowledge of the fixed hash function in the scheme may have no success against the scheme with a random function, it may be successful for this fixed function. Also, due to the fact that MD5 and SHA-1 are polynomial-time computable functions, it is theoretically possible that the adversary would fail for a random function, but not for a polynomial-time one. It is possible that an algorithm can break the scheme for some particular hash functions using the information of the way the hash function is computed. In [Stinson, 2006], it is stated that no practical protocol proven secure in the random oracle model has been broken when used with a "good" hash function, such as SHA-1. On the other hand, Goldreich and Halevi showed that there exists a theoretical signature scheme which has been proved secure in the random oracle model, which becomes insecure whenever the hash function used in the protocol is specified as a polynomial time computable function. Hence, the reduction considers adversaries against certain types of attacks on the protocol instead of attacks on the hash function since any attack which treats the hash function as a random function will not be successful regardless of whether the hash function is actually a random function. In other words, the assumption is made about the attacking algorithm instead of the hash function. Unfortunately, many schemes that seem secure in ROM, could not be proven to be secure in the standard model.

2.6.2 Decisional and Computational Assumptions

Public key cryptography rests heavily upon two one way functions related to two number theoretic hard problems, namely factoring integers and computing discrete logarithms. There exists also other problems that are assumed to be hard, but easier than factoring and discrete logarithm. In this section, we review the assumptions which form the basis of the security of the schemes presented in the next section.

Assumption 2.1. *(RSA). Let N be a product of two equally sized primes p and q (p and q are k-bit integers). Let further y be an integer in $\mathbb{Z}_N^*$ and $e > 1$ be an integer co-prime with $\varphi(N)$. Computing the unique integer $x \in \mathbb{Z}_N^*$ such that $x^e = y \bmod N$ is hard.*

Assumption 2.2. *(Quadratic Residuosity (QR)). Given n as a security parameter and $\mathsf{Keygen}(1^n)$ that generates a RSA-type n-bit Blum modulus N and its two prime factors p, q, distinguishing between the distributions $DQR(n) = \{(c, N) : (N, p, q) \leftarrow \mathsf{Keygen}(1^n); c \leftarrow QR(N)\}$ and $DQRN(n) = \{(c, N) : (N, p, q) \leftarrow \mathsf{Keygen}(1^n); c \leftarrow \mathbb{Z}_N^*[+1] \setminus QR(N)\}$ is hard.*

For this assumption, we denote with $\mathbb{Z}_N^*[+1]$ the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol $+1$ and with $QR(N)$ the set of quadratic residues (or squares) in $\mathbb{Z}_N^*$. For any integer a and any positive odd integer N, the Jacobi symbol is defined as the product of the Legendre symbols corresponding to the prime factors of N:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \cdots \left(\frac{a}{p_k}\right)^{\alpha_k} \text{ where } N = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$$

$(\frac{a}{p})$ represents the Legendre symbol, defined for all integers a and all odd primes p by

$$\left(\frac{a}{p}\right) = \begin{cases} 0 \text{ if } a \equiv 0 \pmod{p} \\ +1 \text{ if } a \not\equiv 0 \pmod{p} \text{ and for some integer } x,\ a \equiv x^2 \pmod{p} \\ -1 \text{ if there is no such } x. \end{cases}$$

Finally, an integer y is called a quadratic residue modulo N if there exists an integer x such that $x^2 \equiv y \pmod{N}$. Otherwise, y is called a quadratic nonresidue modulo N.

A stronger QR assumption is introduced in [Boneh et al., 2007], which is called as interactive QR (IQR) assumption. Basically, the assumption says that the QR assumption holds relative to a square root oracle, which is actually a Rabin signature oracle with a collision-resistant hash function. If this hash is assumed as a random oracle, QR implies IQR.

Assumption 2.3. *(Decisional Composite Residuosity (DCR)). Given a composite N and an integer $z \in \mathbb{Z}_{N^2}^*$, it is hard to decide whether there exists $y \in \mathbb{Z}_{N^2}^*$ such that $z \equiv y^N \pmod{N^2}$.*

For the following assumptions, we have the following notation. $\mathbb{G}^* = \mathbb{G} \setminus \{1_{\mathbb{G}}\}$, where $1_{\mathbb{G}}$ is the identity element of the multiplicative group $\mathbb{G}$ of prime order and g denotes the generator of $\mathbb{G}$.

Assumption 2.4. *(Discrete Log (DL)). For $x \xleftarrow{R} \mathbb{Z}_q^*$ and $g \in \mathbb{G}^*$ be a generator of $\mathbb{G}$, given (g, g^x) computing x is hard.*

Assumption 2.5. *(Computational Diffie-Hellman (CDH)). Let $x, y \xleftarrow{R} \mathbb{Z}_q^*$ and $g \in \mathbb{G}^*$ be a random generator of $\mathbb{G}$. Given (g, g^x, g^y) computing g^{xy} is hard.*

Assumption 2.6. *(Decisional Diffie-Hellman (DDH)). Let $x, y, z \xleftarrow{R} \mathbb{Z}_q^*$ and $g \in \mathbb{G}^*$ be a random generator of $\mathbb{G}$. Given (g, g^x, g^y) distinguishing between the distributions (g, g^x, g^y, g^{xy}) and (g, g^x, g^y, g^z) is hard.*

Assumption 2.7. *(Gap Diffie-Hellman (GDH)). Given a randomly chosen generator g of $\mathbb{G}$, and g^a, g^b for unknown $a, b \in \mathbb{Z}_q^*$, computing g^{ab} with the help of a DDH oracle is hard.*

A prime order group $\mathbb{G}$ is a Gap Diffie-Hellman (GDH) group if there exists an efficient polynomial-time algorithm that solves the DDH problem in $\mathbb{G}$ and there is no probabilistic polynomial-time algorithm that solves the CDH problem with non-negligible probability. The Diffie-Hellman problem on such a group is called Gap Diffie-Hellman Problem, that states given a randomly chosen generator g, and g^a, g^b for unknown $a, b \in \mathbb{Z}_q^*$, compute g^{ab} with the help of the DDH oracle. The Gap-problems are based on the fact that the computational problems such as DL problem and CDH are much harder than the DDH, in other words, $DDH \leq CDH \leq DL$ is satisfied.

The CDH problem can be reduced to the DL problem, namely given $(g, g^x) \in \mathbb{G}$, the oracle for DL in group $(g, \mathbb{G})$ outputs x, which is used to compute the CDH problem of $(g^y)^x$.

The DDH problem can be reduced to the CDH problem using the CDH oracle for group $(g, \mathbb{G})$ with the input g^x, g^y resulting in the value g^{xy}. If this value is compared to the DDH instance g^z then DDH is solved.

When we focus to the DDH problem, it is clear that DDH is a weaker variant of the CDH and thus the DDH can be no harder than the CDH. Nevertheless, for the general case, it is not clear whether the DDH problem is always easier than the CDH problem. In summary, when the security of a protocol is proven depending on the GDH problem (or any Gap-{.}), it is assumed that there is a big gap between the DDH (or Decisional-{.}) and the problem underlying the protocol.

2.7 Example PKE Schemes

The most famous public key encryption scheme called as RSA dates back to 1978 [Rivest et al., 1978], which is a deterministic scheme, thus cannot achieve IND-CPA security.

2.7.1 RSA Encryption Scheme:

- Keygen: Based on the parameter $k = |n|$, the user chooses two large primes p and q and publishes $n = pq$. A random exponent e is chosen that is relatively prime to $\varphi(n) = (p-1)(q-1)$. The decryptor keeps p and q secret, or the invert exponent $d = e^{-1} \text{mod } \varphi(n)$, where d is the secret key sk. The public key is $pk = (n, e)$.
- Encrypt: To encrypt a message $m \in \mathbb{Z}_n^*$, one just has to compute $c = m^e \text{ mod } n$.
- Decrypt: The recipient can recover the message by computing $m = c^d \text{ mod } n$ using his secret key d.

One-wayness of RSA relies on the problem of factoring large numbers and RSA problem.

2.7.2 Goldwasser-Micali Scheme

As different from RSA, Goldwasser-Micali encryption scheme [Goldwasser and Micali, 1982] is a probabilistic encryption scheme, namely, encryption of the same message results in different ciphertexts. Several other schemes were obtained as generalizations of this one. Here, as for RSA, we use computations modulo $n = pq$, a product of two large primes.

- Keygen: Let $k = |n|$ be the security parameter and n be an RSA modulus $n = pq$ and p, q are two primes of equal length. Let x be a non-residue for which the Jacobi symbol is 1. The private key is $sk = (p, q)$ corresponding public key is $pk = (n, x)$.
- Encrypt: To encrypt a message $m \in \{0, 1\}$, one randomly selects $y \leftarrow \mathbb{Z}_n^*$ and computes $c = \text{Encrypt}(m, pk) = y^2 x^m \text{ mod } n$. The ciphertext c is a quadratic residue if and only if $m = 0$. The scheme encrypts 1 bit of information, while its output is usually 1024 bits long.
- Decrypt: To decrypt a ciphertext c, check whether c is a quadratic residue. To do so, we use the property that the Legendre symbol $(\frac{c}{p})$ is equal to $(-1)^m$. If so, then $m = 0$, else $m = 1$.

The semantic security of Goldwasser-Micali scheme is based on the quadratic residuosity problem. Regarding its efficiency, a single bit encryption requires a product and a square, whereas decryption requires an exponentiation. The two properties of this scheme is; its input consists of a single bit. This is not very efficient even if it is considered as practical. Secondly, a single bit of plaintext is encrypted in an integer modulo n. Thus, the expansion is huge for a k bit message. The basic principle of this scheme is to partition a well-chosen subset of integers modulo n into two secret parts: M_0 and M_1. Then, encryption selects a random element of M_b to encrypt b, and decryption allows to know in which part the randomly selected element lies. The core point lies in the way to choose the subset, and to partition it into M_0 and M_1. This scheme uses group theory to achieve the following: the subset is the group G of invertible integers modulo n with a Jacobi symbol, with respect to n, equal to 1. The partition is generated by another group $H \subset G$, composed of the elements that are invertible modulo n with a Jacobi symbol, with respect to a fixed factor of n, equal to 1; with these settings, it is possible to split G into two parts: H and $G \backslash H$. The generalizations of Goldwasser-Micali play with these two groups; they try to find two groups G and H such that G can be split into more than $k = 2$ parts [Fontaine and F.Galand, 2007].

2.7.3 Paillier Encryption Scheme

One of the most well-known homomorphic encryption schemes is due to Paillier [Paillier, 1999], which can be considered as a generalization of Goldwasser-Micali encryption. Again, $n = pq$, with $\gcd(n, \varphi(n)) = 1$, but Paillier considered the group $G = \mathbb{Z}_{n^2}$, and a proper choice of H leads to $k = |n|$.

- Keygen: Let $l = |n|$ be the security parameter and n be an RSA modulus $n = pq$ and p, q are two primes of equal length. Let g be an integer of order a multiple of n mod n^2. Define $\lambda(n) = \text{lcm}(p-1)(q-1)$ and $L(u) = \frac{u-1}{n}$, where L-function takes inputs from the set $S_n = \{u < n^2 | u = 1 \bmod n\}$. The private key is $sk = \lambda(n)$ and the corresponding public key is $pk = (n, g)$.
- Encrypt: To encrypt a message $m \in \mathbb{Z}_n$, one randomly selects $x \leftarrow \mathbb{Z}_n^*$ and computes $\text{Encrypt}(m, pk) = c = g^m x^n \bmod n^2$.
- Decrypt: To decrypt a ciphertext c, one computes $m = L(c^{\lambda(n)} \bmod n^2)/L(g^{\lambda(n)} \bmod n^2) \bmod n$.

The semantic security of Paillier scheme is equivalent to decisional composite residuosity assumption (DCRA), which denotes the problem of n^{th} residuosity [Paillier, 1999].

The encryption cost is not too high. Decryption needs one exponentiation modulo n^2 to the power $\lambda(n)$, and a multiplication modulo n. Paillier showed in his paper how to manage decryption efficiently through the Chinese Remainder Theorem. With smaller expansion and lower cost compared to Goldwasser-Micali encryption, this scheme is really attractive.

2.7.4 ElGamal Encryption Scheme

- Keygen: An authority chooses and publishes a cyclic group $\mathbb{G}$ of prime order q together with a generator g of the group. Here, $k = |q|$ denotes the security parameter. The secret key is $sk = x \leftarrow \mathbb{Z}_q$ and the corresponding public key $pk = y = g^x$.
- Encrypt: To encrypt a message $m \in \mathbb{G}$, one randomly selects $r \leftarrow \mathbb{Z}_q$ and computes $(u, v) = (g^r, y^r m)$. The ciphertext is $c = (u, v) \in C$.
- Decrypt: To decrypt $c = (u, v)$, one computes $m = vu^{-x}$.

ElGamal cryptosystem [Gamal, 1984] is one-way secure based on the CDH problem, IND-CPA secure based on the DDH problem and OW-PCA secure if the GDH problem is hard. In many practical protocols $\mathbb{G}$ would be the group of multiples of a point P on an elliptic curve defined over a finite field. In [Tsiounis and Yung, 1998, Katz, 2002], non-malleable elgamal encryption secure against CCA attacks is presented by combining non-malleable zero knowledge proof of plaintext knowledge with ElGamal encryption.

2.7.5 Homomorphic encryption

All the above listed example PKE schemes have a common property. They are classified as homomorphic encryption, which is defined as follows.

For a given cryptosystem with $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, the message space M and the ciphertext space C that are groups, $\mathsf{Decrypt}(\mathsf{Encrypt}(a) \star \mathsf{Encrypt}(b)) = a * b$, where $a, b \in M$, and $*, \star$ represent the group operations of M, C respectively. We say a scheme is additively homomorphic if we consider addition operators, and multiplicatively homomorphic if we consider multiplication operators.

It should be emphasized that a homomorphic encryption cannot have the non-malleability property. By knowing c, we can compute $c' = c \star c$ and deduce, by the homomorphic property, that c' is a ciphertext of $m' = m * m$. Thus, the highest security level it can reach is IND-CPA.

ElGamal encryption scheme is multiplicatively homomorphic property

$$\mathsf{Encrypt}(a) \times \mathsf{Encrypt}(b) = \mathsf{Encrypt}(a \times b)$$

The modified ElGamal encryption presented in [Cramer et al., 1997] generates the ciphertext $c = \mathsf{Encrypt}_{pk}(m) = (g^r, pk^r G^m)$ instead of $c = (g^r, pk^r m)$, where G is a fixed generator of $\mathbb{G}$ and m is the message. Thus, the homomorphic property is additive as

$$\mathsf{Encrypt}(a) \times \mathsf{Encrypt}(b) = \mathsf{Encrypt}(a + b).$$

The homomorphic property of Goldwasser-Micali Scheme is as

$$\mathsf{Encrypt}(a) \times \mathsf{Encrypt}(b) = \mathsf{Encrypt}(a \oplus b)$$

Paillier Cryptosystem is additively homomorphic. Hence, we have

$$\mathsf{Encrypt}(a) \times \mathsf{Encrypt}(b) = \mathsf{Encrypt}(a + b).$$

Additionally, for Paillier Cryptosystem, we have

$$\mathsf{Encrypt}(a) \times g^b = \mathsf{Encrypt}(a + b).$$

2.8 Tools for CCA security

It is conceptually much harder to achieve IND-CCA security than to achieve IND-CPA security. The technical difficulty in reducing an IND-CCA attack is that the adversary has access to a decryption oracle, which cannot be easily implemented without knowing the secret key. However, there are several strategies to make a reduction possible.

2.8.1 Generic Transforms

Fujisaki and Okamoto proposed a simple conversion scheme called as a hybrid scheme ε^{hy} from weak asymmetric-key encryption (AE) and symmetric-key encryption (SE) schemes into a public-key encryption scheme which is secure in the sense of IND-CCA. Basically, ε^{hy} is defined in [Fujisaki and Okamoto, 1999] as follows.

$$\varepsilon^{hy}(m; \sigma) = < AE_{pk}(\sigma; H(\sigma, m) || SE_{G(\sigma)}(m) >$$

In ε^{hy}, σ is generated at random, H and G are two cryptographic hash functions with H: AKMS $\times$ SKMS $\rightarrow$ COINS and G: AKMS $\rightarrow$ SKS, where AKMS denotes asymmetric-key message space, SKMS denotes symmetric-key message space, and SKS

is the symmetric-key space. The idea is, first encrypt the redundancy σ with the random coin $H(\sigma, m)$ under public key pk using the weakly secure probabilistic scheme AE and then encrypt the message under the symmetric key $G(\sigma)$ using the weakly secure scheme SE. In [Fujisaki and Okamoto, 1999], it is proven that if AE is an one-way encryption scheme, then ε^{hy} is IND-CCA secure in ROM. However, it is shown that if AE scheme satisfies IND-CPA security, then there is a significant improvement in the security reduction. The same hybrid scheme is defined in [Boneh and Franklin, 2003] as $\varepsilon^{hy}(m) = < \varepsilon_{pk}(\sigma; H(\sigma, m), G(\sigma) \oplus m >$ and is applicable to the Boneh-Franklin IBE since it is IND-ID-CPA secure which implies also one-way encryption.

2.8.2 Double Encryption

This paradigm starts from an IND-CPA secure encryption scheme, which is used to encrypt the same message under two different public keys pk_1 and pk_2 that are contained in the public key pk of the double encryption system. Next, the encryptor attaches a proof that the two encryptions really contain the same message. This proof is denoted as a non-interactive zero knowledge (NIZK) proof.

2.8.3 Zero Knowledge Proofs

A proof of knowledge is an interactive proof in which the prover P succeeds "convincing" a verifier V that it knows something. Specifically, a zero knowledge proof (ZKP) allows a user to have a private data, and prove its possession without releasing it. P is modeled by a probabilistic Turing machine whereas V is modeled by a polynomial probabilistic Turing machine. During a ZKP, the parties exchange a sequence of messages called the proof transcript. A zero-knowledge proof must satisfy three properties:

- Completeness: If the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.
- Soundness: If the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.
- Zero-knowledge: If the statement is true, no cheating verifier learns anything other than this fact. This is formalized by showing that every cheating verifier has some simulator that, given only the statement to be proven (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the cheating verifier.

Example of a ZKP

The Schnorr identification protocol was proposed by Schnorr in [Schnorr, 1991] for a real-world (smart card-based) application. This protocol operates in a cyclic group $\mathbb{G}$ of prime order d which is generated by some element g. The common input of the prover P and verifier V is an element y of unknown discrete logarithm in base g, and the private input of the prover is this very discrete logarithm, say x. That is, P proves to V that he knows x. This scheme can be applied for identification w.r.t passive adversaries and using random oracle model, the protocol can be made non-interactive as in any Σ protocol.

- Statement: P knows discrete log of y w.r.t. g, where these are members of some group $\mathbb{G}$ of order d, and g is a generator.
- Public input: g, y
- Prover's private input: x such that $y = g^x$.
- P $\rightarrow$ V: P chooses random $k \in \mathbb{Z}_q$, and sends $t = g^k$.
- P $\leftarrow$ V: V chooses $c \xleftarrow{\text{R}} \{0,1\}^l$ and sends c to Alice.
- P $\rightarrow$ V: P sends $r = k + cx \pmod{d}$ to Bob.
- Verification: V verifies that $ty^c = g^r$.

The completeness of the protocol is trivially achieved with probability 1. For the soundness, we have the Σ condition: if $c \neq c'$ then given t and $c \neq c'$ and $r \neq r'$ such that $ty^c = g^r$ and $ty^{c'} = g^{r'}$ we divide the two equations by each other to get $y^{c-c'} = g^{r-r'}$ but since we know c, c' we can take this to the power $(c - c')^{-1} \pmod{d}$ to get an equation of the form $y = g^x$. In particular, suppose that the cheating prover P' is able to successfully carry out the above protocol without knowing x. That is, P', after having committed to a t, is able to answer the challenge c with a response r satisfying $g^r = ty^c$. Note that, for a fixed t, the last equation corresponds each challenge c to a unique response r. Thus, provided the discrete logarithm problem is hard in $\mathbb{G}$, P' needs to guess c correctly beforehand in order to provide an accepting answer; P' will first choose $r \xleftarrow{\text{R}} \mathbb{Z}_d$, then computes $t = g^r y^{-c}$ and sends it as a commitment in the first step of the protocol. In this way, when P' receives the correctly guessed c, he will simply answer with r. This results in a soundness error equal to 2^{-l}, which corresponds to the probability of correctly guessing the challenge c. As a consequence, the higher the parameter l, the better for the soundness of the protocol. However, we cannot increase this parameter indefinitely since this would compromise the zero

knowledgeness of the protocol. Finally, for the zero knowledge property, we want now to prohibit the verifier from learning anything from the prover apart from the validity of the statement. For this, we provide the following simulator:

1. Generate uniformly a random challenge $c' \overset{R}{\leftarrow} \{0,1\}^l$. Choose a random $r \overset{R}{\leftarrow} \mathbb{Z}_d$, compute $t = g^r y^{-c'}$, then sends it to the verifier.
2. Get c from the verifier.
3. If $c = c'$, the simulator sends back r. Otherwise, it goes to Step 2 (rewinds the verifier).

The prover's first message in the protocol is a random value t in $\mathbb{G}$, and so is the simulator's. Moreover, the distributions of the responses of the prover and of the simulator are identical. Finally, we observe that the simulator runs in expected time 2^l and adjusting l to a factor logarithmic in the security parameter ensures that the simulator will run in expected polynomial time.

Non-interactive zero knowledge (NIZK)

This notion consists of a prover who tries to convince a verifier of the validity of some assertion in one move, i.e. without interaction with the verifier. The basic zero knowledge requirement for such proofs consists in exhibiting an efficient simulator outputting messages indistinguishable from the prover's. It is worth noting here that the definition of the zero knowledge requirement for these proofs is simplified because the verifier cannot affect the prover's actions. The most famous technique to obtain NIZK from their interactive variants is known as the Fiat-Shamir paradigm [Fiat and Shamir, 1986]. It consists of letting the prover compute the verifier's challenge himself as a hash of the statement to be proved and of the first message. The security of this construction is provided only in the random oracle model.

CCA vs. ZKP

In an encryption scheme, the adversary may be given access to a decryption oracle which takes as input any ciphertext C and returns the underlying plaintext. In particular, the adversary receives the ciphertext C. Then, the adversary may interact with the decryption oracle, obtaining the plaintext corresponding to any ciphertext(s) C' of the adversary's choosing. The encryption scheme is said to be "chosen-ciphertext secure" if the contents of C remain hidden from the adversary even after interaction with the decryption oracle.

In order to achieve chosen-ciphertext-secure (interactive) public-key encryption, we can use interactive (zero-knowledge) proofs of knowledge. A message m is encrypted using public key pk via $C = \mathsf{Encrypt}(m, r)$ for random r, and then executing an interactive proof-of-knowledge (with the receiver) of m and r. Unfortunately, while this construction is sufficient to achieve non-adaptive chosen-ciphertext security, it does not guarantee adaptive chosen-ciphertext security when the proof of knowledge is malleable [Katz, 2002] since there is nothing in the definition of a zero knowledge proof that prevents an adversary from mutating a proof of knowledge of m into a (zero knowledge, but still valid) proof of knowledge for some other message m', for which a decryption of the corresponding ciphertext yields information about m. As shown by [Katz, 2002, Sahai, 1999], it is possible to construct zero knowledge proofs having the property that a proof of one statement cannot be adapted or mutated into a proof of another statement (a property known as non-malleability). Using non-malleable zero knowledge proofs, it is then possible to construct cryptosystems which avoid the above flaw and hence achieve CCA security [Jao, 2009].

2.8.4 Plaintext-Awareness

In [Bellare and Rogaway, 1994], a new notion for encryption schemes is defined as plaintext-awareness (PA), which is further refined in [Bellare et al., 1998]. The idea is that an adversary is aware of the decryption of the messages which she encrypts in the sense that she cannot produce a ciphertext without knowing the corresponding plaintext. The notion requires that some (universal) algorithm K (the knowledge extractor) can usually decrypt whatever ciphertext an adversary B may output, just by watching the hash function queries which B makes.

An adversary B for plaintext awareness is given a public key pk and access to the random oracle H and an encryption oracle ε_{pk}^{H}. The adversary outputs a ciphertext y, which is not equal to the output of the ε_{pk}^{H} oracle. To be plaintext aware the adversary B should necessarily know the decryption x of its output y. To formalize this, it is demanded there exist some (universal) algorithm K (the plaintext extractor) that could have output x just by looking at the public key, B's H-queries and the answers to them, and the answers to B's queries to ε_{pk}^{H} [Bellare et al., 1998].

Let $\Pi = (\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, be an encryption scheme, let B be an adversary, and let K be an algorithm (the knowledge extractor). For any $l \in \mathbb{N}$ define

$$\mathsf{Succ}_{K,B,\Pi}(l) = Pr[H \leftarrow \mathsf{Hash}; (pk, sk) \leftarrow \mathsf{Keygen}(1^l); (hH, C, y) \leftarrow \mathsf{run}B^{H,\varepsilon_{pk}^{H}}(pk) : K(hH, C, y, pk) = \mathsf{Decrypt}^{H}(sk, y)]$$

Here, $\mathsf{run}B^{H,\varepsilon_{pk}^{H}}(pk)$ means: Run B on input pk and oracles H, ε_{pk}^{H}, and record B's interaction with its oracles, form it into a list hH all of B's H-oracle queries and

the corresponding answers, form it into a list C all of B's ε_{pk}^{H}-oracle answers (i.e. the ciphertexts received, but the messages that formed the actual queries are not recorded). Finally, record B's output, y.

We say that K is a $\lambda(l)$-extractor if K has running time polynomial in the length of its inputs and for every adversary B, $\mathsf{Succ}_{K,B,\Pi}(l) \geq \lambda(l)$. We say that Π is secure in the sense of PA if Π is secure in the sense of IND-CPA and there exists a $\lambda(l)$-extractor K where $1 - \lambda(l)$ is negligible.

We comment that the above stated definition of plaintext awareness is only achievable in the random oracle model. However, in [Bellare and Palacio, 2004], the authors defined plaintext awareness in the standard model and in [Teranishi and Ogata, 2006], it is shown that combining a one-way secure encryption and plaintext awareness implies IND-CCA security. In the random oracle model, we can show that a plaintext-aware scheme is non-malleable and also secure against chosen-ciphertext attack (CCA). Thus, CCA will not help because the adversary already knows the plaintext of any ciphertext, whose decryption she might request from an available decryption box.

2.9 Identity Based Cryptography

Identity Based Cryptography (IBC) is invented by Adi Shamir [Shamir, 1984]. In his paper, the author was only able to present an application of Identity Based Signature (IBS), although it was an open problem until 2001, when the first practical and secure IBE scheme is constructed in the pioneering work of Sakai, Ohgishi, and Kasahara [Sakai et al., 2000], where they presented a non-interactive key agreement scheme in identity-based setting using bilinear pairings.

Identity based cryptography can be specified as a special form of public key cryptography with a difference: There is no need for the binding of the peer identity and its public key as in public key cryptography, where this binding is provided through the certification authority (CA) that is part of a Public-Key Infrastructure (PKI). In identity-based setting, the public key of a user is simply his identity, simplifying the PKI requirements. The corresponding secret key is issued by a trusted Private Key Generator (PKG), who derives it from a master secret that only the PKG knows, and who is assumed to have an out-of-band way to verify the identity of the user. This eliminates some of the costs associated to PKIs and certificates, and opens the way to more efficient schemes. Since the public key of an entity can be his email address, IP address or his identity, there is no need for a PKI, a CA and/or CA hierarchy, key directory, centralized online authority and pair wise pre-shared secrets among all involved parties. Only the offline authority PKG is necessary for keying and for adding a timestamp or a sequence number to the identity proposed by the entity when joining

the system, to avoid collisions in name space. Since the public key of an entity is equal to its identity, there is no need for a trusted third party to certify its public key or an online PKI to generate, verify and broadcast the public key. After the private-key is extracted, an entity has no need to communicate with the PKG, so the PKG is kept offline.

2.9.1 Identity Based Signature

Digital signatures are among the most basic primitives in cryptography, providing authenticity, integrity, and non-repudiation in an asymmetric setting. In their most basic form, each user in the system generates his own key pair consisting of a public key and a corresponding secret key, and the user is assumed to be uniquely identified by his public key. In the real world however, users are generally not identified by randomly generated keys, but by more meaningful identities like their names or email addresses. To map public keys to real-world identities, the PKI needs to be set up, for example involving a hierarchy of trusted certification authorities (CAs) that can certify public keys as belonging to a certain user [Kiltz and Neven, 2009].

From a security point of view, the major drawback of identity-based cryptography is the inherent key escrow property: the PKG can derive the secret keys of all users in the system, and must therefore be trusted not to abuse this power. This is unlike a traditional PKI, where the CA only issues certificates on user-generated public keys, but does not know the corresponding secret keys. While most people find it a discomforting thought that a malafide PKG can sign any message on their behalf, one should be aware that the same type of fraud is possible in the public-key setting as well. Namely, since the certificate is usually sent along with the signature, a cheating CA can always generate a fake certificate for a public key of which it knows the corresponding secret key, and thereby create valid signatures. The victim could try to prove his innocence by showing his real certificate to a judge, but nothing prevents the CA from claiming that the user registered two different public keys. The escrow property is therefore not so much an issue for signatures as it is for encryption, where a malafide PKG can actually decrypt ciphertexts intended for any of its users. So even though there is no legitimate use for escrow of signing keys, a limited form of key escrow is inherently present in both PKI-based and ID-based signature schemes [Kiltz and Neven, 2009].

An IBS scheme consists of four algorithms: Setup, Extract, Sign and Verify. The first three may be randomized but the last is not.

- Setup: The trusted key distribution center runs the setup algorithm Setup on input 1^k to obtain a master public key M_{pk} and the master secret key M_{sk}, where M_{sk} is only known to PKG. (Here, 1^k is the unary notation of the security parameter k.)

- Extract: Given an arbitrary identifier string $ID = \{0,1\}^*$ and M_{sk}, the algorithm returns the signing key d_{ID} associated to the given identity. The signing key is assumed to be securely communicated to the user in question.
- Sign: Given the signing key of identity ID, and a message $m \in M$, the algorithm returns the signature σ on the message m.
- Verify: Given σ, the message m, identity ID and M_{pk}, the algorithm returns either 1 if σ is valid for ID and m, or 0.

To be consistent, an IBS scheme must satisfy Verify $(\sigma, ID, m, M_{pk}) = 1$ with probability one for all messages $m \in M$ and $k \in \mathbb{N}, ID, M$ whenever the keys M_{pk}, M_{sk}, d_{ID} are generated as indicated above.

The securit notion for an IBS scheme is defined as existential unforgeability under chosen message and chosen-identity attack (EUF-CMA). This notion is described through an experiment with a forger F parameterized with the security parameter k. The experiment begins with the generation of a fresh master key pair $(M_{pk}, M_{sk}) \leftarrow \mathsf{Setup}(1^k)$. The forger F is run on input the master public key M_{pk}, and has access to the oracles:

- Extract: On input identity ID, this oracle returns a secret signing key d_{ID}.
- Sign: On input ID and message $m \in M$, this oracle returns a signature σ.

At the end of its execution, the forger outputs identity ID^*, message m^* and a forged signature σ^*. The forger is said to win the game if Verify $(\sigma^*, ID^*, m^*, M_{pk}) = 1$ and F never queried Extract (ID^*) or Sign(m^*, ID^*).

The advantage $Adv_{IBS,F}(k)$ is defined as the probability that F wins the game, and IBS is said to be EUF-CMA secure if $Adv_{IBS,F}(k)$ is negligible in k for all polynomial-time forgers F. As noted before, the first IBS scheme is described by Shamir [Shamir, 1984] that is summarized as below.

- Setup: This algorithm returns an RSA key pair as described in section 2.7.1. Next, we set $M_{pk} = (n, e)$ and $M_{sk} = (n, e, d)$. Also, $H : \{0,1\}^* \rightarrow HRange$ and $G : \{0,1\}^* \rightarrow GRange$ are hash functions, modeled as random oracles.
- Extract: Compute $x = H(ID)^d \bmod n$ and return the signing key $d_{ID} = (n, e, x)$.
- Sign: Randomly select $t \leftarrow \mathbb{Z}_n^*$ and compute $T = t^e \bmod n$, $c = G(T||m)$ and $s \leftarrow xt^c \bmod n$. The signature is $\sigma = (T, s)$.
- Verify: If $s^e = H(ID)T^{G(T||m)} \bmod n$ then return 1, else return 0.

This scheme is EUF-CMA secure under the RSA assumption [Kiltz and Neven, 2009].

Forking Lemma

Forking Lemma was introduced by David Pointcheval and Jacques Stern [Pointcheval and Stern, 2000], which is specified in terms of an adversary that attacks a digital signature scheme instantiated in the random oracle model. The forking lemma states that if an adversary (typically a probabilistic Turing machine), on inputs drawn from some distribution, produces an output that has some property with non-negligible probability, then with non-negligible probability, if the adversary is re-run on new inputs but with the same random tape, its second output will also have the property. They show that if an adversary can forge a signature with non-negligible probability, then there is a non-negligible probability that the same adversary with the same random tape can create a second forgery in an attack with a different random oracle. The forking lemma has been used to prove the security of a variety of digital signature schemes and other random-oracle based cryptographic constructions.

The forking Lemma of [Pointcheval and Stern, 2000] is designed for the generic digital signature schemes that are based on any three-pass honest-verifier zero-knowledge identification protocol. Let (σ_1, h, σ_2) be a round of the identification protocol, we get a digital signature scheme by replacing the query of the verifier by the hash value of the message m to be signed together with the commitment σ_1 which is bound not to change, namely, $h = H(m, \sigma_1)$, where H is the hash function. Thus, a signature of a message m is a triple (σ_1, h, σ_2), where σ_1 is the commitment sent by the prover, $h = H(m, \sigma_1)$ is the random challenge chosen by the verifier and σ_2 is the answer of the prover, which satisfies the verification test for the signature scheme.

Theorem 2.2. *Let A be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and R the number of queries that A can ask to the random oracle and the number of queries that A can ask to the signer. Assume that, within a time bound T, A produces, with probability $\epsilon \geq 10(R+1)(R+Q)/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from A replacing interac- tion with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$ in expected time $T' \leq 120686QT/\epsilon$.*

Here the hash function H outputs k-bit long elements, where k is the security parameter of the signature scheme.

In [Pointcheval and Stern, 2000] an application of the forking lemma to the Schnorr Digital Signature Scheme is presented, where the authors only have to prove that the triples (σ_1, h, σ_2) produced by the signer and the random oracle can be simulated without the knowledge of the signer's secret.

The forking lemma was later generalized in [Bellare and Neven, 2006], which is defined on input the public parameters x as follows.

1. Pick a random tape r for A.
2. Pick $h_1, ..., h_q$ uniformly from H.
3. Run A on input $(x, h_1, ..., h_q; r)$ to produce (J, σ_2).
4. If $J = 0$, then return $(0, 0, 0)$.
5. Pick $h'_J, ..., h'_q$ uniformly from H.
6. Run A on input $(x, h_1, ..., h_{J-1}, h'_J, ..., h'_q; r)$ to produce (J', σ'_2).
7. If $J' = J$ and $h_J \neq h'_J$ then return $(1, \sigma_2, \sigma'_2)$, otherwise, return $(0, 0, 0)$.

For example, let A be an algorithm for breaking a digital signature scheme in the random oracle model. Then x would be the public parameters (including the public key) A is attacking, and h_i would be the output of the random oracle on its i^{th} distinct input. The forking lemma is of use when it would be possible, given two different random signatures of the same message, to solve some underlying hard problem. An adversary that forges once, however, gives rise to one that forges twice on the same message with non-negligible probability through the forking lemma. When A attempts to forge on a message m, we consider the output of A to be (J, σ_2) where σ_2 is the forgery, and J is such that m was the J^{th} unique query to the random oracle (it may be assumed that A will query m at some point, if A is to be successful with non-negligible probability). (If A outputs an incorrect forgery, we consider the output to be $(0, \sigma_2)$) By the forking lemma, the probability of obtaining two good forgeries σ_2 and σ'_2 on the same message but with different random oracle outputs (that is, with $h_J \neq h'_J$ is non-negligible when the probability of forging a signature is also non-negligible. This allows us to prove that if the underlying hard problem is indeed hard, then no adversary can forge signatures.

2.9.2 Identity Based Encryption

Identity-based encryption was introduced by Shamir in 1984 [Shamir, 1984], which aims to encrypt a message by just using the identity of the intended recipient. Several partial and inefficient solutions were proposed after Shamir's initial challenge but it was only in 2000 that Sakai et al. [Sakai et al., 2000], Boneh and Franklin [Boneh and Franklin, 2003], and Cocks [Cocks, 2001] came up with very practical solutions. Although [Sakai et al., 2000]'s work is the first practical IBE scheme that is almost

identical to the scheme presented in [Boneh and Franklin, 2003], the work of Boneh-Franklin included also a security reduction, appropriate assumptions, definitions, the choice of right curves, how to encode and map elements into points, etc. Besides Cocks' scheme is the first IBE that does not use pairings but rather it works in standard RSA groups and its security relies on the standard quadratic residuosity assumption. The scheme of [Cocks, 2001] encrypts the message bit by bit and thus it is considered very bandwidth consuming, however it can be used in practice to encrypt short session keys.

An IBE scheme consists of four algorithms: Setup, Extract, Encrypt and Decrypt.

- Setup: Given a security parameter k, Setup generates the parameters of the scheme, master public key M_{pk} and the master secret key M_{sk}, where M_{sk} is only known to PKG. In addition, the description of a finite message space M and the description of a finite ciphertext space C are part of the scheme parameters.
- Extract: Given an arbitrary identifier string $ID = \{0,1\}^*$ and the system parameters, the algorithm returns the private key d_{ID} associated to the given identity.
- Encrypt: Given the system parameters, a message $m \in M$ and an identity ID, the algorithm returns a ciphertext $c \in C$
- Decrypt: Given a ciphertext $c \in C$, and a private key d_{ID} of identity ID, the algorithm returns either $\perp$ or the message m.

To be consistent, an IBE scheme must satisfy the following condition for all messages in M. $\forall\ m \in M$ Decrypt $(c, d_{ID}) = m$ where $c =$ Encrypt(ID, m).

In IBE, an arbitrary string is given as a public key, which is input to the extract algorithm to generate the corresponding secret key. When joining the system, identity Id_k of an entity k which is unique and easily verifiable by the PKG, could be the email address of k with temporal or spatial properties (e.g.,a@b.com@date@site). For instance, by using a hash function H_1 and a specific IBE scheme called Boneh-Franklin IBE [Boneh and Franklin, 2003], PKG extracts the secret key only one time from the master secret key x of the system and Id_k, namely $d_k = xH_1(Id_k)$ which is sent back to k in a secure, out-of-band side channel. If the identity of k has the form user@time, the PKG can proactively refresh the identity of the entity by updating the time portion of the identity and generate a new private key using the updated identity in case of exposure of the private key. Thus, ephemeral identities like user@time cause key updating process to be very easy. Due to compromise of the secret key, Bob requests a new secret key with a partial update in the time portion of the identity bob@time. The same update method could be applied in proactive refreshing of identity and secret key, since the time portion determines valid period of the secret key.

Let us describe the advantages of IBE when used in mobile ad hoc networks. First, there is no need for certificates, which results in better performance than other CA based systems when the bandwidth is limited as in the case of mobile ad hoc networks. Also, the computing complexity is much more low in IBE compared to RSA-based systems due to the use of elliptic curve cryptography. For instance a public key in RSA is a number several thousand bits long without a concept of identity, requiring a certificate to tie the public key to an identity, whereas the public key of Bob in IBE is like bob@b.com. Also, by adding different parameters to the identity, namely bob@b.com@date@site, Bob can use the same identity in different systems with different life time of keys, which provides advantages in case of compromise of his secret keys. This representation provides high semantics due to the date and the location information, it suggests a routing path and an easier resource discovery of special nodes in the ad hoc network. Moreover, if an entity is malicious or compromised, the PKG leaves the peer out of the system either by making its identity or key invalid. A sender-only-user does not need to request any private key, only users who wants to receive information from other nodes need to pass by the offline PKG regularly to request keying. To avoid that the PKG becomes a single point of failure, a group of n PKGs could be used to share the system's master secret key x so that any node can derive its secret key by combining the shares of its secret key from any t PKGs. Hence, if the number of compromised PKGs does not exceed the threshold value, the privacy of the nodes is protected. Thus, multiple PKGs enabled by threshold cryptography or hierarchical PKGs could provide secret keys to avoid single point of failure of one PKG and decrease the effect of compromised PKGs. Also to support large ad hoc networks, hierarchical PKG structures could be employed, which is also useful for nodes changing their locations frequently between different systems. The offline entity PKG can have different policies depending on the behaviors of the nodes, namely cooperativeness in relaying, reputable or badly behaving nodes. This may include extracting keys valid for different time periods, excluding peers from the system by identity blacklisting or key expiring. A peer can join, leave, change location or status anywhere and anytime in the network because it has a time and location invariant identity uniquely identifying itself within the system.Asynchronous communication is enabled since any node can receive information from other nodes before obtaining his private key or if it is in idle state to save energy. Any form of communication is possible such as relayed, multi-hop communication.

Classification of IBE systems

In [Boyen, 2007], a classification of the known identity-based encryption schemes is presented, which we summarize as below.

- Quadratic Residuosity IBE (without pairings): Cock's IBE that is secure in the

Random Oracle Model, and further extensions of it presented in [Boneh et al., 2007, Ateniese and Gasti, 2009] both secure in ROM. Except for this class, the rest of the IBE schemes are based on pairings.

- Full Domain Hash IBE: This is the class of the Boneh-Franklin IBE [Boneh and Franklin, 2003], and to which the earlier Sakai-Ohgishi-Kasahara identity-based key exchange [Sakai et al., 2000] also belongs. Both secure in the random oracle model.
- Exponent Inversion IBE: This approach to IBE can be traced to an idea of Mitsunary, Sakai, and Kasahara in the context of traitor tracing. A benefit of this type of construction is that there is no need to hash directly on the curve. This category includes the Sakai-Kasahara scheme originally described in [Sakai and Kasahara, 2003] and later proven secure in [Chen and Cheng, 2005] in the random oracle model. The category also includes the second of two IBE schemes proposed by Boneh and Boyen [Boneh and Boyen, 2004], which has a selective-identity proof of security in the standard model. All these schemes rely on the $BDHI$ complexity assumption called Bilinear Diffie-Hellman Inversion ($BDHI$).
- Commutative Blinding IBE: The last category of IBE systems descends from Boneh and Boyen's scheme, the first scheme given in [Boneh and Boyen, 2004]. It is actually this scheme that is the basis for the first fuzzy IBE scheme described in [Sahai and Waters, 2005]. The systems in this category are based on the same BDH assumption as the Boneh-Franklin IBE scheme.

2.9.3 Bilinear Groups and Maps

Pairing-based IBE systems makes use of a bilinear map or pairing, which is implemented using a weil or a tate pairing on elliptic curves. For a more detailed explanation on pairings the reader is referred to [Galbarith et al., 2006]. We briefly review the necessary facts about pairings and the groups over which they are defined. For a prime p, we denote the finite field of order p by $\mathbb{Z}_p$. Wet let $\mathbb{Z}_p^*$ denote the multiplicative group of order $p-1$ consisting of the elements in $\mathbb{Z}_p \setminus \{0\}$. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two (possibly distinct, but isomorphic) cyclic groups of prime order p. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be respective generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. A pairing is a bilinear function $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{F}$ that maps pairs of elements in $\mathbb{G}_1, \mathbb{G}_2$ to elements of a group $\mathbb{F}$, where $\mathbb{F}$ is another multiplicative group of order p. $\hat{e}$ and the group operations in $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{F}$ can be performed efficiently. The identity elements are denoted by $1_{\mathbb{G}_1}$, $1_{\mathbb{G}_2}$ and $1_{\mathbb{F}}$ respectively. Furthermore, we assume that:

- Bilinear: The map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{F}$ is bilinear if $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$ $\forall u \in \mathbb{G}_1$, $\forall v \in \mathbb{G}_2$ and $\forall a, b \in \mathbb{Z}$.

- Non-degenerate: For $u \in \mathbb{G}_1$, $\hat{e}(u, v) = 1_{\mathbb{F}}$ $\forall v \in \mathbb{G}_2$ iff $u = 1_{\mathbb{G}_1}$. Non-degeneracy means the mapping cannot be the trivial map which sends every pair of elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ to the identity element of $\mathbb{F}$. Because all are groups of prime order, it follows that if g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$, then $e(g_1, g_2)$ is a generator of $\mathbb{F}$.

- Computable: $\hat{e}(u, v)$ is efficiently computed $\forall u \in \mathbb{G}_1$, $\forall v \in \mathbb{G}_2$

We say that $\mathbb{G}_1, \mathbb{G}_2$ forms a bilinear group pair, and that $\hat{e}$ is a bilinear map from $(\mathbb{G}_1, \mathbb{G}_2)$ into $\mathbb{F}$. The above definition is general in the sense that no special constraint is placed on $\mathbb{G}_1$ and $\mathbb{G}_2$ other than having prime order p. In a number of applications, however, it is important to require that $\mathbb{G}_1$ and $\mathbb{G}_2$ be the same group (i.e. $\mathbb{G}_1 = \mathbb{G}_2$); this leads to the notion of symmetric bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$. The designers of encryption schemes have tended to prefer symmetric pairings, favoring simplicity over generality [Boneh and Boyen, 2004]. In order to be consistent with the notation of the current IBE and fuzzy IBE schemes, we opt for the simple symmetric formulation. Despite the fact that CDH is hard to solve in $\mathbb{G}$, the decisional DH can be easy in $\mathbb{G}$. Indeed, there is an important class of groups in which the DDH is easy and the DH and DL problems are believed to be hard. These are the "Diffie-Hellman gap groups" that are used in pairing-based cryptography [Koblitz and Menezes, 2010]. To prove that we define a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$, where $\mathbb{G}$ is an additive group of points of an elliptic curve and $\mathbb{F}$ is an multiplicative group of a finite field.

Given (g, g^x, g^y, g^z), where $z = xy \bmod q$, distinguishing between the distributions (g, g^x, g^y, g^z) and (g, g^x, g^y, g^r) is easy since $\hat{e}(g, g)^z = \hat{e}(g, g^z) = \hat{e}(g^x, g^y)$, which can easily be computed whereas $\hat{e}(g, g)^r = \hat{e}(g, g^r) \neq \hat{e}(g^x, g^y)$.

Hence CDH is a stronger problem than DDH due to some particular groups for which detecting DDH tuples is easy, but solving CDH problems is hard. If the Diffie-Hellman problem is hard on a group $\mathbb{G}$ with an easily computable pairing, then $\mathbb{G}$ is a DH gap group. In fact, groups with pairings are the only known examples of gap groups. For all other groups we solve DDH simply by finding discrete logs; there is no known way to solve DDH that is faster than that [Koblitz and Menezes, 2010].

2.9.4 Assumptions based on Bilinear Pairings

For the following assumptions, we have the following notation. $\mathbb{G}^* = \mathbb{G} \setminus \{1_{\mathbb{G}}\}$, where $1_{\mathbb{G}}$ is the identity element of the bilinear group $\mathbb{G}$ and g denotes the generator of $\mathbb{G}$. If S is a set, then $|S|$ is its cardinality and $x \xleftarrow{R} S$ denotes the operation of assigning to x an element of S chosen uniformly at random.

Assumption 2.8. *(Bilinear Diffie-Hellman (BDH)). Let $x, y, z \xleftarrow{R} \mathbb{Z}_q^*$, g be a generator*

of $\mathbb{G}$ *and* $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ *be a bilinear map. Given* (g, g^x, g^y, g^z)*, computing* $\hat{e}(g,g)^{xyz}$ *is hard.*

It can be easily seen that CDH assumption in $\mathbb{G}$ or $\mathbb{F}$ implies the BDH assumption in $< \mathbb{G}, \mathbb{F}, \hat{e} >$. However, the reverse implication is still an open problem. The BDH problem can be reduced to the CDH problem in $\mathbb{G}$ by giving the oracle CDH for the group $(g, \mathbb{G})$. With the input (g, g^x, g^y) to the CDH oracle, the oracle returns g^{xy}. Having this value, the BDH problem can be solved easily since $\hat{e}(g^{xy}, g^z) = \hat{e}(g,g)^{xyz}$.

Furthermore, the BDH problem can be reduced to the CDH problem in $\mathbb{F}$ by defining $k = \hat{e}(g,g) \in \mathbb{F}$, which implies $k^x = \hat{e}(g, g^x), k^y = \hat{e}(g, g^y)$ and $k^z = \hat{e}(g, g^z)$. The input k^x, k^y given to the CDH oracle for the group $(k, \mathbb{F})$ results in the value of k^{xy}, which is again given as input to the oracle with k^z to receive the value k^{xyz}. Having obtained this value, the solution to the BDH problem is found since $k^{xyz} = \hat{e}(g, g^{xyz})$.

Assumption 2.9. *(Bilinear Decisional Diffie-Hellman (BDDH)). Let* $x, y, z \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_q^*$*,* g *be a generator of* $\mathbb{G}$ *and* $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ *be a bilinear pairing. Given* (g, g^x, g^y, g^z) *distinguishing between the distributions* $(g, g^x, g^y, g^z, \hat{e}(g,g)^{xyz})$ *and* $(g, g^x, g^y, g^z, \hat{e}(g,g)^r)$ *is hard.*

Assumption 2.10. *(Diffie-Hellman Inversion (DHI)). Let* $x \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_q^*$ *and* g *be a generator of* $\mathbb{G}$*. Given* $g^x \in \mathbb{G}$ *computing* $g^{\frac{1}{x}} \in \mathbb{G}$ *is hard.*

It can be shown that the DHI problem can be easy with an oracle access to a CDH oracle that can compute for any pair of $g^x, g^y \in \mathbb{G}$ the value $g^{xy} \in \mathbb{G}$. Hence, given an instance of the DHI problem, one can compute $g^{\frac{1}{x}} \in \mathbb{G}$ in polynomial time as follows. Computing $(g^x)^{\varphi(q)-1}$ is the solution of the problem where $t^{\varphi(q)} \equiv 1 \bmod q, \forall t \in \mathbb{Z}_q^*$ due to the Euler theorem, which implies $t^{\varphi(q)-1} \equiv \frac{1}{t} \bmod q$. If $t = x$ then, $(g^x)^{\varphi(q)-1} = g^{\frac{1}{x}}$. The value $(g^x)^{\varphi(q)-1}$ is computed using the CDH oracle and repeated squaring method.

Assumption 2.11. *(Collision Attack-1 (k-CAA1)). Let* $k \in \mathbb{Z}$*,* $x \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_q^*$*,* g *be a generator of* $\mathbb{G}$*. Given* $(g, g^x, h_0, (h_1, g^{\frac{1}{(h_1+x)}}), ..., (h_k, g^{\frac{1}{(h_k+x)}}))$ *where* $h_i \in Z_q^*$ *that are chosen uniformly random and distinct for* $0 \leq i \leq k$*, computing* $g^{\frac{1}{(h_0+x)}}$ *is hard.*

Assumption 2.12. *(Collision Attack-2 (k-CAA2)). Let* $k \in \mathbb{Z}$*,* $x \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_q^*$*,* g *be a generator of* $\mathbb{G}$*. Given* $(g, h_0, (h_1, g^{\frac{1}{(h_1+x)}}), ..., (h_k, g^{\frac{1}{(h_k+x)}}))$ *where* $h_i \in \mathbb{Z}_q^*$ *that are chosen uniformly random and distinct for* $0 \leq i \leq k$*, computing* $g^{\frac{1}{(h_0+x)}}$ *is hard.*

In [Chen and Cheng, 2005], it is proven that $(k-1) - DHI \Leftrightarrow k - CAA2$.

Assumption 2.13. *(Strong CAA (k-sCAA1)). Let* $k \in \mathbb{Z}$, $x \xleftarrow{R} \mathbb{Z}_q^*$, g *be a generator of* $\mathbb{G}$. *Given* $(g, g^x, h_0, (h_1, g^{\frac{1}{(h_1+x)}}), ..., (h_k, g^{\frac{1}{(h_k+x)}}))$ *where* $h_i \in \mathbb{Z}_q^*$ *that are chosen uniformly random and distinct for* $0 \leq i \leq k$, *computing* $[h, g^{\frac{1}{(h+x)}}]$ *for some* $h \in \mathbb{Z}_q^*$ *but* $h \notin \{h_1, ..., h_k\}$ *is hard.*

It is vital that the value of h in the computation for $(h, g^{\frac{1}{(h_0+x)}})$ must be output, otherwise the problem is not hard since one only needs to find an $r \in \mathbb{Z}_q^*$ in the form of $r = \frac{1}{(h+x)} \bmod q$ where h and r are not shown in the problem above [Chen and Cheng, 2005].

Assumption 2.14. *(Strong DH (k-sDH)). Let* $k \in \mathbb{Z}$, $x \xleftarrow{R} \mathbb{Z}_q^*$, g *be a generator of* $\mathbb{G}$. *Given* $(g, g^x, g^{x^2}, ..., g^{x^k})$ *computing* $[h, g^{\frac{1}{(h+x)}}]$ *for some* $h \in \mathbb{Z}_q^*$ *is hard.*

In [Chen and Cheng, 2005], it is proven that $(k-1) - sCAA1 \Leftrightarrow k - sDH$.

Assumption 2.15. *(Exponent Problem (k+1)-EP). Let* $k \in \mathbb{Z}$, $x \xleftarrow{R} \mathbb{Z}_q^*$, g *be a generator of* $\mathbb{G}$. *Given* $(g, g^x, g^{x^2}, ..., g^{x^k})$ *computing* $g^{x^{k+1}}$ *is hard.*

Again it is proven that $k - DHI \Leftrightarrow k + 1 - EP$.

Assumption 2.16. *(Bilinear DH Inversion (k-BDHI)). Let* $k \in \mathbb{Z}$, $x \xleftarrow{R} \mathbb{Z}_q^*$, g *be a generator of* $\mathbb{G}$ *and* $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$ *a bilinear pairing. Given* $(g, g^x, g^{x^2}, ..., g^{x^k})$ *computing* $\hat{e}(g, g)^{\frac{1}{x}}$ *is hard.*

In [Chen and Cheng, 2005], it is proven that $BDH \Leftrightarrow (1 - BDHI)$.

Assumption 2.17. *(Decisional Bilinear DH Inversion (k-DBDHI)). Let* $k \in \mathbb{Z}$, $x \xleftarrow{R} \mathbb{Z}_q^*$, g *be a generator of* $\mathbb{G}$ *and* $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$ *a bilinear pairing. Distinguishing between the distributions* $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g, g)^{\frac{1}{r}})$ *and* $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g, g)^{\frac{1}{x}})$ *is hard.*

Additionally, there exists Gap-{.} assumptions, where {.} can be any of the computational problems that are listed in this chapter. Basically, they assume that a computational problem is still hard despite the fact that the corresponding decisional problem is solvable.

Assumption 2.18. *(Gap Bilinear Diffie-Hellman (Gap BDH)). Let* $k \in \mathbb{Z}$, $a, b, c \xleftarrow{R} \mathbb{Z}_q^*$, g *be a generator of* $\mathbb{G}$ *and* $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$ *a bilinear pairing. Given* g^a, g^b *and* g^c, *computing* $\hat{e}(g, g)^{abc}$ *with the help of the DBDH oracle is hard.*

Let us recall the public key encryption scheme called Bilinear ElGamal. Given the bilinear group $(g, \mathbb{G}, \mathbb{F})$ and a pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$, choose $Q \in \mathbb{G}$, $s \in \mathbb{Z}_q^*$ uniformly at random and compute $T = g^s$, where s is the secret key and the rest of the values is the public key. To encrypt a message $m \in \mathbb{F}$, we choose $r \in \mathbb{Z}_q^*$ at random and compute $(U, V) = (g^r, m\hat{e}(Q, T)^r)$. To decrypt this ciphertext, $m = V/\hat{e}(U, Q)^s$.

It is easy to see that the above Bilinear ElGamal scheme is OW-PCA secure assuming that the Gap-BDH problem is intractable. Here, the plaintext checking (PC) oracle checks on input of a certain message m', the public parameters g, Q, T and a ciphertext (U, V), whether $(g, U, Q, T, V/m')$ is a Bilinear Diffie-Hellman tuple. Hence, the running time and advantage of the OW-PCA attacker is exactly the same as those of Gap-BDH attacker.

2.9.5 Security Notions of IBE

Similar to the security notions described for public key encryption schemes, Indistinguishability is also the right formalization of the security goal for IBE schemes since it is equivalent to semantic security and it is simple, easy to use and appealing. When combined with the means of the adversary, we obtain two security notions, where the weaker notion of security is called as IND-ID-CPA.

IND-ID-CPA security

Here, the adversary is not allowed to issue any decryption query. IND-ID-CPA game between the adversary and challenger is defined as follows.

- Setup: The challenger runs the Setup algorithm with the security parameter k and returns the adversary the system parameters and the master public key M_{pk}.
- Phase 1: The adversary issues private key extraction queries adaptively and the challenger responds with the private keys corresponding to the public key ID_i. Any query made in this phase cannot be presented as the challenge identity in the next stage.
- Challenge: The adversary outputs equal length plaintexts $m_0, m_1 \in M$ and an identity ID_{ch}, provided that it was not queried in Phase 1. The challenger picks a random bit $b \in \{0, 1\}$ and sends the adversary the encryption of m_b under ID_{ch} and M_{pk} as the challenge.
- Phase 2: The adversary issues extraction queries as in Phase 1, with the restriction that ID_{ch} is not queried.

- Guess: The adversary outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

Such an adversary is called an IND-ID-CPA adversary A, and its advantage against the scheme Π with security parameter k is defined as below. The random bits used by the adversary and challenger defines the probability.

$$Adv_{\Pi,A}(k) = |Pr[b = b'] - \tfrac{1}{2}| < negl(k)$$

Since, IND-ID-CPA provides a weak security level, the standard notion of security for IBE is IND-ID-CCA, which is a natural extension of IND-CCA notion of PKE. The only difference is that the adversary is allowed to issue decryption queries, with the restriction that the challenge identity-ciphertext pair is not queried.

Further Attack models

Although the security notions of PKE and IBE are in parallel, IBE schemes have additional attack models that provide the adversary with more power than in PKE schemes. This difference is caused by the adversary's attack on the arbitrary public key, namely the identity. The adversary is allowed to run the adaptive identity attack or the selective identity attack, where the former one is stronger than the latter one since the adversary obtains any private key of the corresponding identity he wishes other than the challenge identity. This model is called as adaptive chosen identity attack or full identity attack. However, in selective chosen identity attack, the challenge identity has to be selected in advance by the adversary before the public parameters are generated. Although selective identity attack is a weak model, it is the standard attack model for fuzzy IBE schemes due to the structure of the identity of the users.

IND-sID-CPA

Basically, an IBE scheme is IND-sID-CPA secure, if no polynomially bounded adversary A has a non-negligible advantage against the Challenger in the following IND-sID-CPA game.

- Select: The adversary A selects a target identity $ID^* \in \{0,1\}^*$.
- Setup: The challenger runs the Setup algorithm with the security parameter k and returns the adversary the system parameters and the master public key M_{pk}.
- Phase 1: The adversary issues private key extraction queries and the challenger responds with the private keys corresponding to the public key $ID_i \neq ID^*$.

- Challenge: The adversary outputs equal length plaintexts $m_0, m_1 \in M$. The challenger picks a random bit $b \in \{0, 1\}$ and sends the adversary the encryption of m_b under ID^* and M_{pk} as the challenge.
- Phase 2: The adversary issues adaptively extraction queries as in Phase 1, with the same restriction on ID^*.
- Guess: The adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

Such an adversary is called an IND-sID-CPA adversary A, who successfully breaks the scheme if he guesses the random bit correctly with a probability significantly better than just random guessing.

Current pairing based fuzzy IBE schemes are proven secure according to this model. To obtain a higher security level, namely, IND-sID-CCA, it is suggested to use a generic construction that converts an IND-ID-CPA secure scheme to an IND-ID-CCA one.

2.10 Fuzzy IBE

In Eurocrypt'05, Sahai and Waters [Sahai and Waters, 2005] proposed a new Identity Based Encryption (IBE) system called fuzzy IBE that uses biometric attributes as the identity instead of an arbitrary string like an email address. Besides, fuzzy IBE could be applied in the context of Attribute-Based Encryption [Pirretti et al., 2006, Sahai and Waters, 2005], where the sender encrypts data using a set of attributes such as {university, faculty, department} and the ciphertext could only be decrypted if the receiver has the secret key associated to all of these attributes or sufficient number of them. The properties of this new system is summarized in section 2.2.7. Although introduced as a new concept in [Sahai and Waters, 2005], current fuzzy IBE schemes are based on previous constructions on Hierarchical IBE (HIBE), multi-receiver identity based broadcast encryption and Boneh-Franklin IBE. These constructions are combined with Shamir's secret sharing scheme that provides the error-tolerance property. We review this scheme as below.

2.10.1 Shamir's secret sharing

A (t, n) *threshold* secret sharing scheme is a method for n parties to carry shares s_i of a message s such that any t of the them to reconstruct the message, but so that no $t-1$ of them can easy do so. The threshold scheme is *perfect* if knowledge of $t-1$ or fewer shares provides no information regarding s.

Shamir's (t, n)*-threshold scheme* provides an elegant construction of a perfect (t, n)-threshold scheme using a classical algorithm called Lagrange interpolation. First we introduce Lagrange interpolation as a theorem.

Theorem 2.3. *(Lagrange interpolation). Given t distinct points (x_i, y_i) of the form $(x_i, f(x_i))$, where $f(x)$ is a polynomial of degree less that t, then $f(x)$ is determined by*

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, i \neq j} \frac{x - x_j}{x_i - x_j}. \tag{2.1}$$

Shamir's scheme is defined for a secret $s \in \mathbb{Z}_p$ with p prime, by setting $a_0 = s$, and choosing $a_1, \ldots, a_{t-1}$ at random in $\mathbb{Z}_p$. The trusted party computes $f(i)$, where

$$f(x) = \sum_{k=0}^{t-1} a_k x^k,$$

for all $1 \leq i \leq n$. The shares $(i, f(i))$ are distributed to the n distinct parties. Since the secret is the constant term $s = a_0 = f(0)$, the secret is recovered from any t shares $(i, f(i))$, for $I \subset \{1, \ldots, n\}$ by

$$s = \sum_{i \in I} c_i f(i), \text{ where each } c_i = \prod_{j \in I, j \neq i} \frac{i}{j - i}.$$

Properties: Shamir's secret sharing scheme is (1) *perfect* — no information is leaked by the shares, (2) *ideal* — every share is of the same size p as the secret, and (3) involves no unproven hypotheses.

Current fuzzy IBE/IBS schemes combine pairing based encryption/signature schemes and Shamir's secret sharing in order to achieve error-tolerant encryption for biometric identities. Briefly, the first construction of fuzzy IBE for small universe of attributes is simplified version of threshold bilinear ElGamal encryption and the large universe construction is almost identitical to the Boneh-Boyen IBE [Boneh and Boyen, 2004] scheme. In [Pirretti et al., 2006], the large universe construction of Sahai and Waters is used by replacing the computationally expensive T function with a hash function to design an efficient Attribute based encryption scheme in the ROM. The third paper on fuzzy IBE is almost identitical to the broadcast IBE scheme of [Baek et al., 2005] and finally, the only scheme that considers the anonymity of the fuzzy IBE is based on Boneh-Franklin IBE. In the following sections, we summarize the base IBE schemes of current fuzzy IBE systems.

2.10.2 Based on Boneh-Boyen IBE

In [Boneh and Boyen, 2004], the authors construct a selective identity secure Hierarchical IBE (HIBE) without random oracles, where selective identity secure IBE is a slightly weaker security model than the standard security model for IBE. In a Hierarchical IBE, identities are vectors. A vector of dimension n represents an identity at depth n.

- Setup: Given a security parameter k_0 that determines the size of the group, the parameters of the scheme are generated as follows. Choose a bilinear group $(\mathbb{G}, \mathbb{F}, \hat{e})$ where $\mathbb{G}, \mathbb{F}$ are of prime order p and g is a generator of $\mathbb{G}$. For now, we assume identities (ID) of depth of n are vectors of elements in $\mathbb{Z}_p^n$. We write $ID = (w_1, ..., w_n) \in Z_p^n$. The j-th component corresponds to the identity at level j. We also assume messages to be encrypted are elements in $\mathbb{G}$. The HIBE system works as follows:

  To generate system parameters for an HIBE of maximum depth n, select a random generator $g \in \mathbb{G}$, a random $\alpha \in \mathbb{Z}_p$, and set $g_1 = g^\alpha$. Next, pick random elements $t_1, ..., t_n \in \mathbb{G}$ and a random element $g_2 \in \mathbb{G}$. The public parameters $params$ and the master secret key are given by $params = (g, g_1, g_2, t_1, ..., t_n)$ and $M_{sk} = g_2^\alpha$.

  For $j = \{1, ..., n\}$ we define $F_j : Z_p \to \mathbb{G}$ to be the function: $F_j(x) = g_1^x t_j$.

- Extract: To generate the private key d_{ID} for an identity $ID = (w_1, ..., w_j) \in Z_p^j$ of depth j, pick random $r_1, ..., r_j \in Z_p^j$ and output

$$d_{ID} = (d_0, ..., d_j) = (g_2^\alpha \cdot \prod_{k=1}^{j} F_k(w_k)^{r_k}, g^{r_1}, ..., g^{r_j})$$

- Encrypt: Given the plaintext $m \in \mathbb{G}$, $params$ and identity $ID = (w_1, ..., w_j) \in Z_p^j$, pick a random $s \in \mathbb{Z}_p$, and output

$$c = (A, B, C_1, ...C_j) = \hat{e}(g_1, g_2)^s \cdot m, g^s, F_1(w_1)^s, ..., F_j(w_j)^s$$

- Decrypt: Given the ciphertext $(A, B, C_1, ...C_j)$ encrypted using the public key ID and $d_{ID} = (d_0, ..., d_j)$, the receiver decrypts as

$$m = A \cdot \frac{\prod_{k=1}^{j} \hat{e}(C_k, d_k)}{\hat{e}(B, d_0)}$$

Indeed, for a valid ciphertext, we have

$$\frac{\prod_{k=1}^{j} \hat{e}(C_k, d_k)}{\hat{e}(B, d_0)} = \frac{\prod_{k=1}^{j} \hat{e}(F_k(w_k), g)^{sr_k}}{\hat{e}(g, g_2)^{s\alpha} \prod_{k=1}^{j} \hat{e}(g, F_k(w_k))^{sr_k}} = \frac{1}{\hat{e}(g_1, g_2)^s}$$

Boneh-Boyen IBE is IND-sID-CPA secure based on the decisional BDH problem.

The first paper on fuzzy IBE describes two schemes, where the first scheme is designed for small universe of attributes (i.e. biometric features) based on a threshold version of bilinear ElGamal, and the second scheme is designed for large universe of attributes based on the Boneh-Boyen scheme with slight modifications. In particular, we replace the identity vector, by the feature vector of the user, and choose a random polynomial $q(.)$ such that $q(0) = \alpha$ in order to replace the d_0 value in the Extract algorithm with $g_2^{q(w_i)} \cdot \prod_{k=1}^{j} F_k(w_k)^{r_k}$.

2.10.3 Based on Boneh-Franklin IBE

The fuzzy IBE scheme of [van Liesdonk, 2007] is based on the multi-receiver version of Boneh-Franklin IBE [Boneh and Franklin, 2003] and shamir's secret sharing. Let us briefly review Boneh-Franklin IBE, which is also called as BasicIdent.

- Setup: Given a security parameter k, the parameters of the scheme are generated as follows.

  1. Choose a bilinear group $(\mathbb{G}, \mathbb{F}, \hat{e})$ where $\mathbb{G}, \mathbb{F}$ are of prime order q and g is a generator of $\mathbb{G}$.
  2. Choose $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = g^s \in \mathbb{G}^*$ as the master public key.
  3. Select two cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{F} \rightarrow \{0,1\}^n$ for some n.

  The message space is $M = \{0,1\}^n$ and the ciphertext space is $C = \mathbb{G}^* \times \{0,1\}^n$.

  The system parameters are $(q, \mathbb{G}, \mathbb{F}, \hat{e}, n, g, P_{pub}, H_1, H_2)$.

  The master public key is $P_{pub} = g^s \in \mathbb{G}^*$ and the master secret key is $M_{sk} = s$.

- Extract: Given an identifier string $ID \in \{0,1\}^*$ and the master secret key s, the algorithm returns the private key $d_{ID} = Q_{ID}^s \in \mathbb{G}^*$ associated to the given identity, where $Q_{ID} = H_1(ID) \in \mathbb{G}^*$.

- Encrypt: Given the plaintext $m \in M$, P_{pub} and identity ID compute $Q_{ID} = H_1(ID) \in \mathbb{G}^*$ choose $r \in \mathbb{Z}_q^*$ uniformly at random and compute the ciphertext as $c = (g^r, m \oplus H_2(\hat{e}(Q_{ID}, P_{pub})^r)$

- Decrypt: Given the ciphertext $c = (U, V) \in C$ encrypted using the public key ID and d_{ID}, the receiver decrypts as $m = V \oplus H_2(\hat{e}(U, d_{ID}))$ since

$$\hat{e}(d_{ID}, U)) = \hat{e}(Q_{ID}^s, g^r) = \hat{e}(Q_{ID}, g)^{rs} = \hat{e}(Q_{ID}, g^s)^r$$

For the security reduction in ROM, a public key encryption scheme called BasicPub is defined and an IND-ID-CPA attack on BasicIdent is converted to a IND-CPA attack on BasicPub to show that private key extraction queries are useless for the adversary. Finally the reduction from the BDH assumption into an IND-CPA attack on BasicPub is shown.

Since BasicIdent is malleable, it is not secure in the sense of IND-ID-CCA. In [Boneh and Franklin, 2003], BasicIdent is converted into IND-ID-CCA scheme by applying the Fujisaki-Okamoto transformation, which results in the FullIdent scheme in the random oracle model. Hence, BasicIdent is converted to the IND-ID-CCA secure scheme FullIdent.

2.10.4 Based on Baek et al.'s IBE

An obvious way to construct an IBE scheme for the multi user setting is simply encrypting a message n times using BasicIdent, where n is the number of receivers. Written in additive notation, the message m is encrypted as $\{(r_1P, m \oplus H_2(\hat{e}(H_1(ID_1), P_{pub})^{r_1}), ..., (r_nP, m \oplus H_2(\hat{e}(H_1(ID_n), P_{pub})^{r_n})\}$ where $r_i \in \mathbb{Z}_q^*$ are uniformly chosen at random. However, the performance of that construction is very bad since, one needs n pairing computations to compute $\hat{e}(H_1(ID_i), P_{pub})$, n scalar multiplications with elements in $\mathbb{G}$ to compute r_iP, n exponentiations in group $\mathbb{F}$ to compute $(\hat{e}(H_1(ID_1), P_{pub})^{r_n}$ and finally the evaluations of the hash function H_1 are done on the points of an elliptic curve. The ciphertext length is $n(l_1 + l_2)$ where l_1 is the length of an element of the group $\mathbb{G}$ and l_2 is the length of the message. It is exactly this construction that is used in [van Liesdonk, 2007] to design an anonymous fuzzy IBE scheme, since Boneh-Franklin IBE guarantees recipient anonymity.

In [Baek et al., 2005], to improve the performance of the multi-receiver Boneh-Franklin IBE, the randomness re-use technique is proposed to decrease the number of multiplications in the group $\mathbb{G}$, namely instead of having a different r_iP for each user, a common rP can be used, which results in $n - 1$ less multiplications. However, this does not effect the exponentiations and pairing computations, where each paring computation is equal to roughly 10 exponentiation computations.

In [Baek et al., 2005], the selective multi identity attack (sMID) is chosen as the main attack model to prove an IBE scheme in multi-user setting. This attack model is a weak model where the attacker commits ahead of time the list of identities of the receivers that it intends to attack, whereas in full identity attack (MID) model, the attacker adaptively chooses the identities that will be challenged.

In [Baek et al., 2005], the following scheme is proven secure in the sense of indistinguishability of encryptions under selective multi-ID, chosen plaintext attack (IND-sMID-CPA) based on the $BDDH$ problem in ROM. To be consistent with the notation

of the scheme, we continue with the additive notation.

- Setup: Given a security parameter k, the parameters of the scheme are generated as follows.

  1. Choose a bilinear group $(\mathbb{G}, \mathbb{F}, \hat{e})$ where $\mathbb{G}$ and $\mathbb{F}$ are of prime order q and P is a generator of $\mathbb{G}$.
  2. Choose s and Q uniformly at random $s \in \mathbb{Z}_q^*$, $Q \in \mathbb{G}^*$ and compute $T = sP$.
  3. Select a hash function $H : \{0,1\}^* \to \mathbb{G}^*$.

  The message space is $M = \{0,1\}^l$ and the ciphertext space is $C = (\mathbb{G}^*)^{n+1} \times \mathbb{F}$. The system parameters are $(q, \mathbb{G}, \mathbb{F}, \hat{e}, P, T, Q, H)$ and the master secret key is $M_{sk} = s$.

- Extract: Given an identifier string $ID_i \in \{0,1\}^*$ and the master secret key s, the algorithm returns the private key $d_i = sH(ID_i) \in G^*$ associated to the given identity.

- Encrypt: Given the plaintext $m \in M$, the system parameters and a list of identities $ID^* = (ID_1, ..., ID_n)$ choose $r \in \mathbb{Z}_q^*$ uniformly at random and compute the ciphertext as $c = (U, V_1, ..., V_n, W, L) = (rP, rH(ID_1) + rQ, ..., rH(ID_n) + rQ, \hat{e}(Q,T)^r m, L)$,
  where L is a label that contains information about how V_i is associated with each receiver. There is no need for the sender to perform a pairing computation if $\hat{e}(Q,T)$ is precomuted and provided as a PKG's common parameter.

- Decrypt: Given the ciphertext $c = (U, V_1, ..., V_n, W, L) \in C$ encrypted using ID^* and d_i, the receiver decrypts by finding the corresponding V_i as $m = \frac{\hat{e}(U,d_i)}{\hat{e}(T,V_i)} W$ since

$$\begin{aligned} \frac{\hat{e}(U,d_i)}{\hat{e}(T,V_i)} W &= \frac{\hat{e}(rP, sH(ID_i))}{\hat{e}(sP, rH(ID_i)+rQ)} W = \frac{\hat{e}(rP, sH(ID_i))}{\hat{e}(rP, sH(ID_i)+sQ)} W \\ &= \frac{\hat{e}(rP, sH(ID_i))}{\hat{e}(rP, sH(ID_i))\hat{e}(rP, sQ)} \hat{e}(Q,T)^r m \end{aligned}$$

The idea of the proof is again to construct an adversary B that tries to solve the $BDDH$ problem using an adversary A that attacks the above scheme.

As noted before selective multi-identity attack model is a weak model compared to the fully adaptive model, where the attacker adaptively chooses which identity to attack and outputs the list of challenge multiple identities in the challenger phase after it sees public parameters instead of ahead of time. The cost of this stronger attack model is the inefficient reduction in the security proof due to the difficulty in generating

a challenge ciphertext while handling the random oracle and key extraction queries, whereas in selective attack model every phase of the game could be programmed at the beginning according to the $BDDH$ parameters. Consequently, the probability that the list of identities A chooses before the challenge phase to be challenged on matches the guess of B is at least $(\frac{1}{q_H})^n$ where n denotes the number of receivers.

By integrating the shamir's secret sharing into this scheme, we obtain the fuzzy IBE schemes described in [Baek et al., 2007].

2.10.5 Based on Sakai-Kasahara IBE

In [Sakai and Kasahara, 2003], the authors describe efficient identity based cryptosystems including an efficient IBE scheme, which could only be proven secure in 2005 by the authors of [Chen and Cheng, 2005]. The main difference of Sakai-Kasahara Key Construction to the above presented schemes is that it does not require a MapToPoint hash function. A MapToPoint hash function converts a user's identity to a point on the underlying elliptic curve in IBE schemes. The above presented IBE schemes and current efficient fuzzy IBE schemes [Pirretti et al., 2006, Baek et al., 2007] employ this special function, which is usually implemented as a probabilistic algorithm and is more expensive than a point scalar multiplication in terms of computation time [Chen and Cheng, 2005, Chen et al., 2006]. This operation is also time consuming and cannot be treated as a conventional hash operation which is commonly ignored in performance evaluation. It is known that the cost of a MapToPoint hash operation is bigger than one inversion in $\mathbb{Z}_p$. Besides, as it is noted in [Barreto et al., 2005, Smart and Vercauteren, 2007], it is difficult to find groups as the range of the MapToPoint hash function and to define an efficient isomorphism at the same time.

Briefly, we review the setup and key generation of Sakai-Kasahara IBE scheme [Sakai and Kasahara, 2003] as below. Since our new constructions for biometric IBE/IBS is based on this efficient key construction, detailed analysis is presented in the corresponding chapters.

- Generate two cyclic groups $\mathbb{G}$ and $\mathbb{F}$ of prime order p, and a bilinear pairing map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$. Pick a random generator $g \in \mathbb{G}$.
- Pick a random $s \in Z_p^*$ to compute g^s and pick a hash function $H_1 : \{0,1\}^* \rightarrow Z_p^*$. Here, $M_{sk} = s$.
- Given an identifier string $ID \in \{0,1\}^*$, the secret key of this identity is $d_{ID} = g^{\frac{1}{s+H_1(ID)}}$.

Remark 2.1. *The secret key is a short signature d_A on the message ID_A signed under the private signing key s, which is existentially unforgeable under chosen-message attack*

in the random oracle model, provided that the k-sCAA1 assumption is sound in $\mathbb{G}$ *[Chen and Cheng, 2005].*

Also, based on this key construction, the authors of [Barreto et al., 2005] designed an IBS scheme, which is shown to be EUF-CMA (Existential Unforgeability under Chosen Message Attack) secure based on the $k-DHI$ problem. Moreover, the scheme of [Barreto et al., 2005] is currently the most efficient pairing-based IBS scheme in the literature.

- Setup: Given the security parameter 1^k, generate two cyclic groups $\mathbb{G}$ and $\mathbb{F}$ of prime order p, and a bilinear pairing map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$. Pick a random generator $g \in \mathbb{G}$ and two hash functions $H_3 : \{0,1\}^* \to \mathbb{Z}_p^*$, $H_4 : \{0,1\}^* \times \mathbb{F} \to \mathbb{Z}_p^*$.
- Extract: The signing key is $d = g^{1/(x+H_3(ID))}$, where $M_{sk} = x$.
- Sign: In order to sign the message m,

  1. Pick at random $r \in \mathbb{Z}_p^*$ and compute $h = H_4(m, \hat{e}(g,g)^r) \in \mathbb{Z}_p^*$.
  2. Compute $S = d^{r+h}$.

  Hence, the signature on m is $\sigma = (h, S)$.
- Verify: To verify a signature $\sigma = (h, S)$ on m, compute

$$\begin{aligned} V &= \hat{e}(S, g^{H_3(ID)} \cdot g^x) \cdot \hat{e}(g,g)^{-h} \\ &= \hat{e}(D^{r+h}, g^{H_3(ID)} \cdot g^x) \cdot \hat{e}(g,g)^{-h} \\ &= \hat{e}(g^{(r+h)/(x+H_3(ID))}, g^{H_3(ID)+x}) \cdot \hat{e}(g,g)^{-h} \\ &= \hat{e}(g,g)^{r+h} \cdot \hat{e}(g,g)^{-h} = \hat{e}(g,g)^r \end{aligned}$$

  and check whether $H_4(m, V) = h$

Chapter 3

Distributed Biometric Remote Authentication Systems

In this chapter, we describe the components of biometric-based remote authentication and focus on a special type of system introduced by Bringer et al., where the server-side functionalities are performed in a distributed fashion using a detached biometric database $\mathcal{DB}$ and non-colluding system components.

We start with the review of the security notions (i.e. identity privacy and transaction anonymity) and analyze existing schemes designed according to this security model. We show that the schemes differ from each other based on the homomorphic encryption scheme chosen, incorporation of a secure sketch scheme, the biometric storage mechanism and whether an additional security factor is required as in the case of multi-factor biometric authentication. Next, we propose a new and efficient biometric storage mechanism, where the biometric features of the users are stored at the $\mathcal{DB}$ as a random pool of features instead of storing each complete reference template. This way, common features belonging to different users are not stored multiple times, which results in a reduced storage cost. Besides, the $\mathcal{DB}$ and the authentication server $\mathcal{AS}$ do not know which set of features belong to which user, since the indices of the database locations of a user's features are stored in the smartcard of the user, thus the index list of each user is kept secret from the $\mathcal{DB}$ and $\mathcal{AS}$. In current authentication systems that provide a security reduction, biometrics is assumed as a binary string such as a an 2048 bits iris code, whereas the general representation of biometrics is a set of features that can be either ordered such as face, voice, handwritten signatures or unordered such as fingerprint minutia. For this general representation, we describe a new biometric remote authentication scheme based on this new storage method. To guarantee the security notions, we present the security reductions, which prove that using an adversary breaking the identity privacy (and transaction anonymity) notion, one can

construct another adversary that breaks the semantic security of ElGamal encryption scheme (and user privacy of PIR protocol).

Secondly, we analyse the security model of distributed biometric remote authentication (DBRA). By considering the schemes designed according to this model, we prove that identity privacy can never be achieved for the existing schemes, if biometrics is assumed as public data and a publicly stored sketch is employed for improved accuracy. Besides, a statistical attack is shown that is effective even if the sketch is stored as encrypted. To prevent statistical attacks, we propose a weaker notion of identity privacy, where the adversary has limited power. In view of our attacks, we describe a new biometric remote authentication system by combining distributed biometric authentication and cancelable biometrics, where the new system is also applicable for biometrics represented as a set of features. Next, we define "identity privacy for cancelable biometrics" as a new notion and show that existing schemes vulnerable to our attacks are secure in the cancelable biometric setting if the new notion is assumed.

This chapter is based on our work presented in [Sarier, 2009a,b, 2010b, 2011a].

3.1 Introduction

Biometric-based authentication systems can be classified as remote or local authentication, where the former system authenticates a user over a network by performing the matching of his transmitted fresh biometrics to his stored biometric data at the remote server. Here, the client side is only responsible from capturing the fresh biometrics of the user through a biometric sensor and transmitting the extracted features to the remote server. Hence, the client side does not require sophisticated devices such as match-on-card (MOC) systems that store a biometric template and perform the matching on card, whereas the remote server should store the biometric templates of the users centrally and performs the matching. The advantage of remote authentication is that the user does not need to store any biometric data on an IC card that can be lost easily and matching at the server allows for an complex matching algorithm that may not be implemented on card, in particular, as a MOC system. Also, MOCs are more expensive than the traditional template-on-card systems as they require smart cards with embedded microprocessor and operating systems to run the match application. Other MOC systems that release the biometric template to another device (either directly or over a network) to perform the matching function may compromise the security [Heyer, 2008].

In ACISP'07, a special type of biometric remote authentication system and a new security model is introduced by Bringer et al., where security against insider attacks are considered. In this model, the server-side functionalities are performed in a distributed

fashion using a detached biometric database and non-colluding system components. Basically, this system is composed of three entities, the authentication server $\mathcal{AS}$, the sensor $\mathcal{S}$ capturing the biometrics and the detached biometric database $\mathcal{DB}$. $\mathcal{AS}$ only stores the identity information of the users and provides the communication between $\mathcal{S}$ and $\mathcal{DB}$. Besides, $\mathcal{AS}$ does not have access to the reference biometrics that is stored as encrypted using homomorphic encryption, thus all the computations performed by $\mathcal{AS}$,$\mathcal{S}$ and $\mathcal{DB}$ stay in the encrypted domain. This leads to a new security notion called identity privacy that guarantees the privacy of the link between the identity (name) and the biometrics of the user although biometrics is assumed as public data. The intuition of this notion is that a malicious $\mathcal{AS}$ that generates two templates for a user, cannot identify from the protocol runs, which of the two biometric templates is registered to the $\mathcal{DB}$ with probability significantly better than that of random guessing. Moreover, $\mathcal{AS}$ performs the matching after a Private Information Retrieval (PIR) protocol that prevents a curious $\mathcal{DB}$ from tracking the user that authenticates to the system. Thus, transaction anonymity against a (malicious) database is satisfied which is the second notion for biometric remote authentication.

3.1.1 Motivation and Contributions

As it is noted in the previous chapter, the first biometric-based authentication system that assumes biometrics as public data and that presents a security is analysis from a cryptographic point of view is described in 2007, although automated biometric authentication systems are employed for more than two decades. Despite the fact that systems of Bringer et al. provide a security reduction, recently, different attacks have emerged that break the security notions with a simple brute-force attack. Thus, a thorough analysis of the schemes and the security notions is required. Besides, one should also consider the efficiency of these systems, especially for the large scale deployment of them.

We start with the review of the security notions (i.e. identity privacy and transaction anonymity) and analyze existing schemes designed according to this security model. Basically, these systems are based on the primitives of homomorphic encryption, Private Information Retrieval (PIR) and secure sketch. In this framework, we propose a new and efficient biometric storage mechanism, where the biometric features of the users are stored at the $\mathcal{DB}$ as a random pool of features instead of storing each complete reference template. This way, common features belonging to different users are not stored multiple times, which results in a reduced storage cost. Besides, the $\mathcal{DB}$ and the authentication server $\mathcal{AS}$ do not know which set of features belong to which user, since the indices of the database locations of a user's features are stored in the smartcard of the user, thus the index list of each user is kept secret from the $\mathcal{DB}$ and $\mathcal{AS}$. In current authentication systems that provide a security reduction, biometrics

is assumed as a binary string such as a an 2048 bits iris code, whereas the general representation of biometrics is a set of features that can be either ordered such as face, voice, handwritten signatures or unordered such as fingerprint minutia. For this general representation, we describe a new two-factor biometric remote authentication scheme based on this new storage method. Our system can easily and securely integrate a sketch for improved accuracy since it is designed as a two-factor authentication system, where the second factor is the smartcard of the user that stores some secret data. In the second part of this chapter, we will show why this additional factor is required in order to achieve the security notions.

Following Bringer et al.'s security model, we present the security reductions, which prove that using an adversary breaking the identity privacy (and transaction anonymity) notion, one can construct another adversary that breaks the semantic security of ElGamal encryption scheme (and user privacy of PIR protocol).

Next, we consider DBRA schemes that require a fuzzy sketch scheme for improved accuracy. We analyze the security based on the model of Bringer et al., where we prove that if biometrics is assumed as public data and the fuzzy sketch required for error-correction is stored publicly, the notion of identity privacy against a malicious authentication server $\mathcal{AS}$ can never be satisfied. Basically, this notion guarantees the secrecy of identity-biometrics relation through a security game between the (malicious) $\mathcal{AS}$ and a simulator (i.e. challenger) $\mathcal{C}$. If $\mathcal{AS}$ can correctly distinguish the registered reference template $\sim$that is one of the two templates output by $\mathcal{AS}\sim$ by listening to the protocol runs, $\mathcal{AS}$ wins this game, thus breaks the scheme in the sense of identity privacy.

In identity privacy game, the malicious $\mathcal{AS}$ has to output two biometric templates describing the user U. Since the definition of this notion does not restrict $\mathcal{AS}$ on how he chooses the two biometric templates, $\mathcal{AS}$ can output a pair of templates (b_1, b_2) for U, where the distance between the two templates is either $\mathsf{dis}(b_1, b_2) < t$ or $\mathsf{dis}(b_1, b_2) > t$. Here, t is the error correction threshold of the secure sketch scheme that is used to correct the errors given a similar biometrics and a public helper data PAR. For the two cases, we prove separately that the adversary can easily compute the exact biometric template that is registered by the challenger $\mathcal{C}$ of the game using the helper data PAR of the secure sketch that is publicly available. Thus, the schemes of [Bringer et al., 2007c, Tang et al., 2008] and any biometric remote authentication scheme that assumes biometrics and the required secure sketch as public data are vulnerable to this attack and cannot satisfy identity privacy. Although the scheme of [Bringer and Chabanne, 2008] stores the helper data PAR as encrypted, we propose a statistical attack to break identity privacy, where the adversary uses the (known) distribution of U's biometrics and outputs the two templates (b_1, b_2) for U in a special way. To our knowledge, no concrete attack has been presented against the sketch-based schemes of [Bringer and

Chabanne, 2008, Bringer et al., 2007c, Tang et al., 2008], although [Simoens et al., 2011] presents attacks against the schemes in [Bringer et al., 2007b, Barbosa et al., 2008].

Thus, we observe that the security model of Bringer et al. does not consider the attacks that reveal the cleartext of the stored reference biometrics with the help of the public sketch. Besides, if the sketch is stored secretly, then identity privacy game should be modified so that there is a restriction on the templates generated by the adversary $\mathcal{AS}$ to prevent $\mathcal{AS}$ breaking the notion with statistical attacks. Thus, we describe a new notion called Weak-Identity privacy that does not allow the adversary to generate the possible templates for a particular user, instead the templates are given to him by the challenger. Under this new notion, the scheme of [Bringer and Chabanne, 2008] is resistant against our statistical attacks.

Secondly, we discuss alternative solutions to guarantee the security of DBRA schemes requiring public sketches. The trivial solution for the schemes [Bringer et al., 2007c, Tang et al., 2008] is to store the sketch PAR secretly, namely, in the tamper-proof smartcard of the user. This will result in a two-factor authentication scheme, thus, the system is not anymore a pure biometric-based authentication scheme. Besides, if these systems are implemented for biometrics that are represented as a set of features, this solution still does not cover brute-force attacks for biometrics with a small feature space. We note that current provably secure schemes are only defined for biometrics represented as a fixed length binary string such as an 2048 bits long Iris code except for the schemes of [Sarier, 2010b, Barbosa et al., 2008] that assume biometrics as a set of features, i.e. k-tuple of integers.

As a first solution, we describe a new DBRA protocol where we combine cancelable biometrics and distributed remote authentication. Briefly, cancelable biometrics perform a distortion of the biometric image or features before matching. The variability in the distortion parameters provides the cancelable nature of the scheme. Distortion (i.e masking) is performed either using a one-way transformation or a high entropy randomness that is stored in the user's smart card to be used later for authentication in the transformed space. Our protocol is applicable for biometrics represented as a set of features and resistant against brute-force attacks if the feature space is small. Next, we define a stronger notion as "Identity privacy for cancelable biometrics", where breaking this notion implies breaking the underlying encryption scheme in the sense of indistinguishability. The schemes of [Bringer et al., 2007c, Tang et al., 2008] that are vulnerable to our attack are secure in cancelable biometric setting based on this new notion.

Finally, we employ the detached biometric storage in DBRA, which is not considered in current cancelable biometric systems and in their security analysis. Thus, a trusted biometric database can serve different service providers due to its distributed structure.

Besides, a major difference of our model to existing schemes of Bringer et al. [Bringer et al., 2007b, Bringer and Chabanne, 2008, Bringer et al., 2007c, Tang et al., 2008] is the use of bilinear pairings, which allows the $\mathcal{AS}$ to compute the final authentication decision without any decryption operation. Thus, $\mathcal{AS}$ does not need to store a secret key, whose leakage endangers the system's security drastically.

3.1.2 Related Work

Existing distributed biometric remote authentication (DBRA) schemes differ from each other based on the homomorphic encryption scheme chosen, incorporation of a secure sketch scheme, the biometric storage mechanism and whether an additional security factor is required as in the case of multi-factor biometric authentication. DBRA schemes that are designed according to the security model of Bringer et al. [Bringer and Chabanne, 2008, Bringer et al., 2007c, Tang et al., 2008] combine homomorphic encryption, secure sketches and Private Information Retrieval (PIR) to achieve the security notions of identity privacy and transaction anonymity. The first biometric system in this model [Bringer et al., 2007b] employs Goldwasser-Micali encryption and a special PIR in order to compare two binary biometric strings in encrypted domain using hamming distance. Next, the systems of [Bringer et al., 2007c, Tang et al., 2008] require a secure sketch scheme to error-correct the biometric string such as an 2048 bits Iris code and use ElGamal encryption for equality testing [Gamal, 1984] together with an efficient PIR scheme. Similarly, the work of [Bringer and Chabanne, 2008] combines a secure sketch, Goldwasser-Micali and Paillier encryption in Lipmaa's PIR protocol to prevent the attacks against the scheme in [Bringer et al., 2007b]. Besides, in [Sarier, 2010b], elliptic curve ElGamal and a PIR scheme is employed together with a special secure sketch scheme applicable to an ordered biometric feature set. Another work that assumes biometrics as a set of features [Barbosa et al., 2008] provides a secure biometric identification scheme using a Support Vector Machine and Paillier encryption by adapting the security notions for biometric features (usually an k-tuple of numbers). A survey of these systems is given in [Sarier, 2009b]. Recently, [Simoens et al., 2011] presents a survey of attacks against the schemes of [Bringer et al., 2007b, Barbosa et al., 2008] and some other biometric schemes. No attacks are known for the schemes presented in [Bringer and Chabanne, 2008, Bringer et al., 2007c, Tang et al., 2008], which require the use of secure sketches. Except for the works of [Barbosa et al., 2008, Sarier, 2010b, 2009a], the biometrics is assumed as a binary string such as a an 2048 bits iris code, whereas the general representation of biometrics is a set of features that can be either ordered such as face, voice, handwritten signatures or unordered such as fingerprint minutia.

3.2 Architecture of the System

The system structure for biometric-based remote authentication schemes designed according to the security model of Bringer et al. consists of five components. Here, the user U and the sensor S denote the client side and the remaining components denote the server-side of the system.

- Human user U, which uses his biometrics to authenticate himself to an authentication server. The user may possess a smart card for storing additional data such as error correcting information or user specific data other than biometrics.
- Sensor client $\mathcal{S}$, which captures the raw biometric data and extracts a biometric template, and communicates with the authentication server by performing cryptographic operations such as public key encryption. We also assume a liveness link between the sensor and the server-side components, to provide confidence that the biometric data received on the server-side is from a present living person.
- Authentication server $\mathcal{AS}$, which deals with human user's authentication request by communicating with the user and organizing the entire server-side procedure. The data stored at the $\mathcal{AS}$ consists of a list $\mathcal{L}$= $\{ID_1, ..., ID_N\}$ of user identities $ID_l \in \{0,1\}^*$. The index of the user in this list will be $j \in \{1, ..., N\}$. In a successful authentication the $\mathcal{AS}$ will obviously learn the user's identity, which means that it should learn nothing about the biometric data being submitted.
- Database $\mathcal{DB}$, which stores biometric information for users, and works as a biometric template matcher by providing the matching service to the authentication server. Since the $\mathcal{DB}$ is aware of privileged biometric data, it should learn nothing about the user's identity, or even be able to correlate or trace authentication runs from a given (unknown) user.
- Verification Unit $\mathcal{VU}$ that helps the authentication process by taking the output produced by the $\mathcal{DB}$ and performing the intermediate operations to help for the final decision. Again that it should not be able to learn anything about the user's real identity. This unit takes place in the schemes of [Barbosa et al., 2008], in [Bringer et al., 2007b] under the name of matcher $\mathcal{M}$, and in [Bringer and Chabanne, 2008] under the name of Hardware Security Module (HSM). In our new proposals, we will also employ this unit.

3.2.1 Authentication Workflow

Like most existing biometric-based cryptosystems, we also assume that a biometric-based authentication scheme consists of two phases: an enrollment phase and a verifi-

cation phase.

1. In the enrollment phase, user U registers his reference biometrics at the database $\mathcal{DB}$ and his personalized username ID at the authentication server $\mathcal{AS}$. The user may have multiple registrations at the same $\mathcal{AS}$ under different usernames.

2. In the verification phase, user U issues an authentication request to the authentication server $\mathcal{AS}$ through the sensor client $\mathcal{S}$. $\mathcal{AS}$ decides based on U's biometrics with help from the database $\mathcal{DB}$ and the verification unit $\mathcal{VU}$.

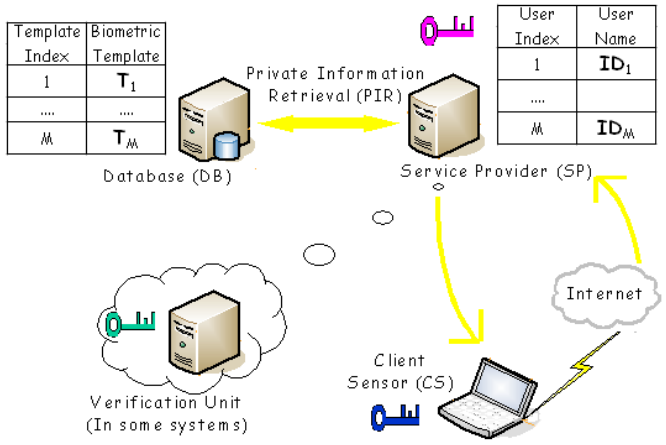

Figure 3.1: Overview of DBRA

3.3 Overview of the required cryptographic techniques

DBRA schemes that are designed according to the security model of Bringer et al. combine homomorphic encryption, secure sketches and Private Information Retrieval (PIR) to achieve the security notions of identity privacy and transaction anonymity. The first biometric system in this model [Bringer et al., 2007b] employs Goldwasser-Micali encryption [Goldwasser and Micali, 1982] and the PIR system of [Kushilevitz and Ostrovsky, 1997]. Next, the systems of [Bringer et al., 2007c, Tang et al., 2008] require a secure sketch scheme to error-correct the biometric string such as an 2048 bits Iris code and use ElGamal encryption for equality testing [Gamal, 1984] together with an efficient PIR scheme such as [Lipmaa, 2005, Gentry and Ramzan, 2005]. Similarly, the work of [Bringer and Chabanne, 2008] combines a secure sketch, Goldwasser-Micali

and Paillier encryption [Paillier, 1999] in Lipmaa's PIR protocol [Lipmaa, 2005] to prevent the attacks against the protocol in [Bringer et al., 2007b]. Another work that assumes biometrics as a set of features [Barbosa et al., 2008] provides a secure biometric identification scheme using a Support Vector Machine and Paillier encryption. In order to analyse the differences between these biometric systems, we briefly define the necessary components of these systems.

3.3.1 Homomorphic encryption

To make an authentication decision in the encryption domain based on a certain metric or to construct a number-theory based PIR protocol, we need a secure cryptosystem that is homomorphic over an abelian group. For a given cryptosystem with $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, the message space M and the ciphertext space C that are groups, $\mathsf{Decrypt}(\mathsf{Encrypt}(a) \star \mathsf{Encrypt}(b)) = a * b$, where $a, b \in M$, and $*, \star$ represent the group operations of M, C respectively. The homomorphic encryption scheme that is employed in [Bringer et al., 2007c, Tang et al., 2008] and that we employ for our new system is ElGamal encryption scheme with the homomorphic property

$$\mathsf{Encrypt}(a) \times \mathsf{Encrypt}(b) = \mathsf{Encrypt}(a \times b)$$

We refer the reader to the section 2.7.5 of the background chapter for the details.

3.3.2 Secure Sketches

Let $\mathcal{H}$ be a metric space with distance function dis. A secure sketch scheme allows recovery of a hidden value $w \in \mathcal{H}$ from any value $w' \in \mathcal{H}$ close to this hidden value with the help of some public value PAR, which does not leak too much information about w. The reader is referred to the section 2.2.3 of the background chapter for the details. The first scheme of [Bringer et al., 2007c] and the scheme of [Tang et al., 2008] implement a secure sketch protocol to test for equality in the encryption domain using the homomorphic property of the encryption system. An example sketch scheme is given in [Tang et al., 2008] as follows. Let $\mathcal{C}$ be a $[n, k, 2t+1]$ error- correction code over a field $\mathbb{F}$. With input $x \in \mathbb{F}^n$, PAR is computed by the function SS as $\mathsf{PAR} = x - c$, where c is a random codeword. With input (x', PAR), Rec computes x'' as $c' = x' - \mathsf{PAR}$, decode c' to obtain c'', and set $x'' = c'' + \mathsf{PAR}$.

The fuzzy sketch for iris biometrics based on the code-offset construction is used in the biometric authentication schemes of [Bringer and Chabanne, 2008]. Let C be an $(n, k, 2t+1)$ binary linear error correcting code in Hamming space. Let $\mathsf{PAR} = c \oplus b$, where c is a random codeword in C. From the corrupted codeword $c' = \mathsf{PAR} \oplus b' = c \oplus (b \oplus b')$, one can recover c if the hamming distance $\mathsf{dis}_{\mathcal{H}}$ between b and b' is $\mathsf{dis}_{\mathcal{H}}(b, b') < t$.

For biometrics that can be represented as an ordered set of features such as face, iris, voice, handwritten signatures [Li et al., 2006], we implement the white noise sketch of [Li et al., 2006] that corrects the white noise on each quantized component (i.e. quantized feature) w_i, $1 \leq i \leq k$ of the biometric vector is as follows:

- The SS function takes the quantized biometrics $Q_\lambda(b) = (w_1, ..., w_k) \in M_\lambda$ as input and computes for each w_i, $c_i = C_\lambda(w_i)$ and outputs the public parameter $\mathsf{PAR}= (\Delta_1, ..., \Delta_k) = (w_1 - c_1, ..., w_k - c_k)$. Here, $C_\lambda(\cdot)$ is the function that finds the unique codeword $c \in C_\lambda$ that is nearest to w_i in the codebook C_λ [Li et al., 2006].
- The Rec function takes a quantized fresh biometric $Q_\lambda(b') = (w'_1, ..., w'_k)$ and PAR as input and computes $c_i = C_\lambda(w'_i - \Delta_i)$ for $1 \leq i \leq k$ and outputs $Q_\lambda(b) = (c_1 + \Delta_1, ..., c_k + \Delta_k)$.

3.3.3 Private Information Retrieval

In order to provide Transaction Privacy, the systems in [Bringer et al., 2007b,c, Bringer and Chabanne, 2008, Tang et al., 2008, Sarier, 2009a, 2010b] employ a number-theory based PIR system, which allows the $\mathcal{AS}$ to retrieve the i-th bit (more generally, the i-th item) from the $\mathcal{DB}$ consisting of m bits while keeping the value i private, which is defined as user privacy. The PIR of [Gentry and Ramzan, 2005] has an additional benefit of retrieving more then one bit, and in particular many consecutive bits. In this context, a Private Block Retrieval (PBR) protocol enables a user to retrieve a block from a block-database. For biometric authentication, [Bringer et al., 2007c] implements a client/server architecture that consists of $\mathcal{DB}$ containing a list of N blocks $(R_1, ..., R_N)$ and the $\mathcal{AS}$ that runs an efficient PBR protocol (such as of [Gentry and Ramzan, 2005]) to retrieve R_l for any $1 \leq l \leq N$. Another advantage of the PBR protocol of [Gentry and Ramzan, 2005] is the retrieval of many non-consecutive blocks efficiently.

Briefly, the PBR protocol of [Gentry and Ramzan, 2005] works as follows.

1. The server partitions the m-bit database $\mathcal{DB}$ into t blocks $\mathcal{DB}= C_1||C_2....||C_t$ of size at most l bits.
2. $S = \{p_1, ..., p_t\}$ is a set of small distinct prime numbers.
3. Each block C_i is associated to a prime power $\pi_i = p_i^{c_i}$, where c_i is the smallest integer so that $p_i^{c_i} \geq 2^l$.
4. All parameters above are public

5. Server precomputes an integer e that satisfies $e = C_i \bmod \pi_i$ using Chinese Remainder Theorem.

6. To query for block C_i, the user generates an appropriate cyclic group $\mathbb{G} = \langle g \rangle$ with order $|G| = q\pi_i$ for some suitable integer q and sends $(\mathbb{G}, g)$ to server, keeping q private. $\mathbb{G}$ contains a subgroup $\mathbb{H}$ of smooth order π_i, and that $h = g^q$ is a generator of $\mathbb{H}$.

7. Server responds with $g_e = g^e \in \mathbb{G}$

8. To retrieve C_i it suffices to retrieve $e \bmod \pi_i$ by setting $h_e = g_e^q \in \mathbb{H}$ and performing a (tractable) discrete logarithm computation $\log_h h_e$, which occurs entirely in the subgroup $\mathbb{H}$ of order $p_i^{c_i}$ and can be quite efficient if p_i is small.

Computational complexity for the Querier side is no more than $4\sqrt{nl}$ group operations and for the Server side $\Theta(n)$ group operations. Communication complexity is $3l_{\mathbb{G}}$ bits, where $l_{\mathbb{G}}$ denotes the size of an element in the group $\mathbb{G}$. The scheme can be converted into a scheme that recovers d l-bit blocks with total communication complexity $(2+d)l_{\mathbb{G}}$ [Gentry and Ramzan, 2005].

Also, in [Ishai et al., 2004], the solution to the problem of retrieving k items that are not necessarily consecutive is presented using hashing. This way, the complexity is much smaller than the naive solution, namely $s \cdot PIR$, where $s = \sigma \log(k\mu)$ for $\mu \in \mathbb{Z}_p^*$. Furthermore, better performance is derived via explicit batch codes instead of hashing, since small values of k do not work with hashing. The reader is referred to [Ishai et al., 2004] for a more detailed discussion of application of batch codes for amortizing the time complexity of PIR. Recently, [Melchor and Gaborit, 2008] introduced an efficient noise-based PIR scheme, which is 100 times faster than all of the number-theory based PIR systems. The communication cost of [Melchor and Gaborit, 2008] is not optimal as of [Gentry and Ramzan, 2005], however, communication cost is not the main performance measurement of PIR as shown in the following table due to the enormous computational cost at the $\mathcal{DB}$-end for number-theory based PIR schemes [Melchor and Gaborit, 2008].

Scheme	Query		Download time	Bandwidth usage
	size	time		
Lipmaa's PIR	162 Kb	0,16s	33h	0.003%
Gentry and Ramzan's PIR	3Kb	$\approx$ 0s	17h	0.016%
Noise-based PIR	19Mb	19s	10min	7.2%

Table 3.1: Comparison of different PIR systems [Melchor and Gaborit, 2008]

3.4 Security Model

The security model of the biometric remote authentication systems introduced by Bringer et al. have the following properties.

3.4.1 Trust Relationships

Different from the local authentication environment, sensor client and authentication server are assumed to be independent components in this model. In [Tang et al., 2008], this is considered to be an appropriate assumption in the remote authentication environment, where human users access the authentication server through sensor clients, which are not owned by the authentication server but have a business agreement with the authentication server.

- Liveliness Assumption: This is an indispensable assumption on the sensor client $\mathcal{S}$ for any biometric system as it guarantees with high probability that the biometrics is coming from a live human user.
- Security link Assumption: To provide the confidentiality and integrity of sensitive information, the communication channel between U, $\mathcal{S}$, $\mathcal{AS}$, $\mathcal{DB}$ and $\mathcal{VU}$ should be encrypted using standard protocols.
- Collusion Assumption: Due to the distributed system structure, we assume that U, $\mathcal{DB}$,$\mathcal{VU}$ and $\mathcal{AS}$ are malicious but they do not collude. Additionally, the sensor client $\mathcal{S}$ is always honest.

3.4.2 Security Notions

Identity Privacy

Informally, this notion guarantees the privacy of the sensitive relationship between the user identity and its biometrics against a malicious authentication server even in case of multiple registrations of the same user with different personalized usernames. Briefly, it means that the authentication server or the database (or an attacker that has compromised one of them) cannot recover the biometric template of the user [Bringer et al., 2007b, Tang et al., 2008]. Here, l denotes the security parameter of the protocol and the symbol $\emptyset$ means that there is no explicit output (besides the state information) for the adversary.

Given an adversary $\mathcal{A}$ running against the biometric authentication scheme and a challenger $\mathcal{C}$ that simulates the registration phase of the scheme, we consider the following game between $\mathcal{A}$ and $\mathcal{C}$.

Experiment $Exp_{\mathcal{A}}(l)$
$(i, ID_i, b_i^0, b_i^1, (ID_j, b_j)_{\{j \neq i\}}) \leftarrow \mathcal{A}(1^l)$
$b_i = b_i^\beta \xleftarrow{R} \{b_i^0, b_i^1\}$
$\emptyset \leftarrow Enrollment((ID_j, b_j)_j)$
$\beta' \leftarrow \mathcal{A}(Challenger; Verification)$
if $\beta' = \beta$ return 1 else return 0

A biometric authentication scheme satisfies the notion of Identity Privacy if

$$Adv_{\mathcal{A}}(l) = Pr[Exp_{\mathcal{A}} = 1|\beta = 1] - Pr[Exp_{\mathcal{A}} = 1|\beta = 0] \tag{3.1}$$

is negligible for all PPT A. Here, the adversary $\mathcal{A}$ generates the authentication data for the users U_j together with two biometric (binary) templates b_i^0, b_i^1 for an additional user U_i. The challenger $\mathcal{C}$ picks at random biometrics $b_i = b_i^\beta$ of U_i and simulates the enrollment phase by registering the encryption of the biometrics of each user at the $\mathcal{DB}$. After running the verification protocol polynomially many times, $\mathcal{A}$ outputs a guess for the biometrics of U_i that $\mathcal{C}$ has chosen. The intuition of this notion is that a malicious authentication server, who knows that the registered biometric template is one of the two templates that he has generated, cannot identify the random choice β of the challenger from listening to the protocol runs with probability significantly better than that of random guessing. Since the sensor is honest, independent and thus not under the control of the adversary, A cannot guess the template by checking the sensor.

Transaction Anonymity

Informally, transaction anonymity means that a malicious database cannot learn anything about the personal identity of the user for any authentication request made to the authentication server [Bringer et al., 2007b, Tang et al., 2008]. Formally, given an adversary $\mathcal{A}$ running against the biometric authentication scheme and a challenger C that simulates the registration phase of the scheme, we consider the following game.

Experiment $Exp_{\mathcal{A}}(l)$
For $1 \leq j \leq N, (ID_j, b_j) \leftarrow \mathcal{A}(1^l)$
$\emptyset \leftarrow Registration(ID_j, b_j)_j$
$(i_0, i_1) \leftarrow \mathcal{A}(Challenger; Verification)$
$i_\beta \overset{R}{\leftarrow} \{i_0, i_1\}$
$\emptyset \leftarrow Verification(i_\beta)$
$\beta' \leftarrow \mathcal{A}(Challenger; Verification)$
if $\beta' = \beta$ return 1 else return 0

A biometric authentication scheme satisfies the notion of Transaction Privacy if equation (3.1) is satisfied for all PPT A.

Here, the adversary $\mathcal{A}$ generates the authentication data for N users and $\mathcal{C}$ simulates the registration phase. After running the verification protocol polynomially many times, $\mathcal{A}$ outputs two users with indices i_0, i_1, where $\mathcal{C}$ picks a random user i_β to initiate the verification for that user. After running the verification protocol polynomially many times, $\mathcal{A}$ outputs a guess for the user that $\mathcal{C}$ has chosen.

3.5 The first protocol

The first remote biometric verification scheme for distributed environments is described in [Bringer et al., 2007b], where the biometric template of user U_i is assumed as a fixed (M bit) binary string b_i=$(b_{i,1}, ..., b_{i,M})$ that is stored as a plaintext in $\mathcal{DB}$ during the registration phase. For authentication, each bit of the fresh biometrics b'_i is encrypted using the public key of the matcher $\mathcal{M}$, where the encryption is performed using the Goldwasser-Micali scheme that is homomorphic (i.e. $\varepsilon(b'_{i,k}, pk) \cdot \varepsilon(b_{i,k}, pk) \bmod n = \varepsilon(b'_{i,k} \oplus b_{i,k}, pk)$). Next, to provide transaction anonymity, $\mathcal{AS}$ runs a PIR protocol to obtain U_i's encrypted biometric template $\varepsilon(b_i)$ computed by the $\mathcal{DB}$, where the communication cost of the PIR is linear in the size N of the users in the $\mathcal{DB}$. The enrollment and verification phases are summarized as below:

Registration Phase: The user U_i registers his personalized identity at the $\mathcal{AS}$ and his biometrics as a fixed binary string $b_i = (b_{i,1}, ..., b_{i,M})$ at the $\mathcal{DB}$.

Verification Phase: The authentication protocol is summarized according to the figure 2 of [Bringer et al., 2007b].

1. The sensor client $\mathcal{S}$ computes a fresh encryption of U_i's biometrics $\varepsilon(b'_i, pk)$=$(\varepsilon(b_{i,1}, pk), ..., \varepsilon(b_{i,M}, pk))$ using the public key pk of the matcher $\mathcal{M}$ and forwards it to $\mathcal{AS}$.

2. $\mathcal{AS}$ runs a PIR protocol by sending $\varepsilon(t_1, pk), ..., \varepsilon(t_N, pk)$ to $\mathcal{DB}$, where $t_j = 1$

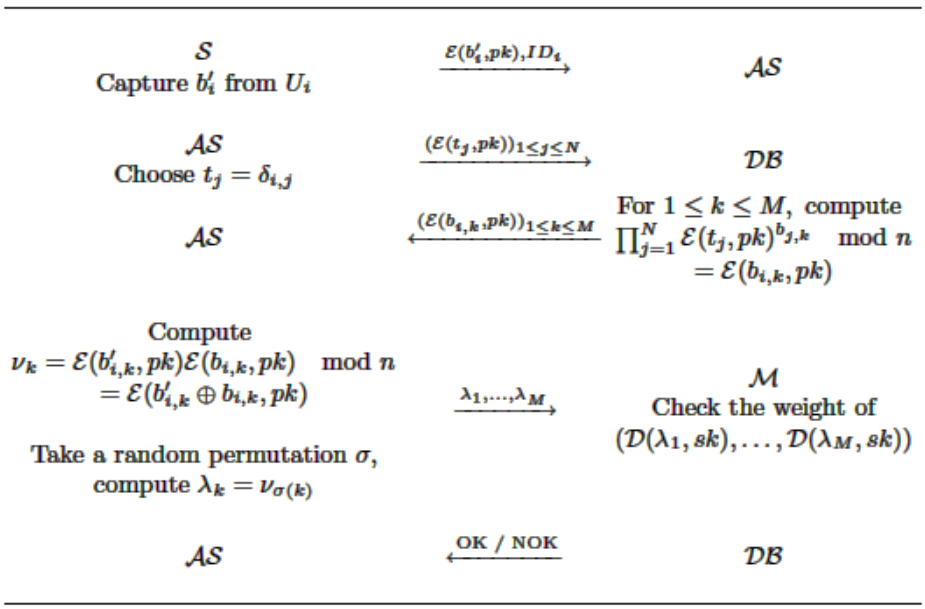

Figure 3.2: Overview of the protocol in [Bringer et al., 2007b]

for $i = j$, else $t_j = 0$. Here, $1 \le j \le N$.

3. $\mathcal{DB}$ computes $\varepsilon(b_{i,k}, pk) = \prod_{j=1}^{N} \varepsilon(t_j, pk)^{b_{j,k}} \mod n$, for $1 \le k \le M$.

4. $\mathcal{AS}$ computes $\nu_k = \varepsilon(b'_{i,k})\varepsilon(b_{i,k}) \mod n = \varepsilon(b'_{i,k} \oplus b_{i,k})$ for $1 \le k \le M$ and sends a permutation of ν_k's (denoted by λ_k) to the matcher $\mathcal{M}$.

5. The detached matcher $\mathcal{M}$ with the secret key of the Goldwasser-Micali scheme decrypts the λ_k's to compute the hamming weight and the decision based on a predefined threshold.

6. Finally, $\mathcal{DB}$ forwards the OK/NOK decision to the $\mathcal{AS}$ [Bringer et al., 2007b].

3.5.1 Attacks against Bringer et al.'s scheme

The scheme of [Bringer et al., 2007b] is provably secure in the framework of Bringer et al. [Bringer et al., 2007b]. However, an attack with complexity exponential in N against this scheme is described in [Barbosa et al., 2008] that reveals the user's biometric data to $\mathcal{AS}$. Thus, the authors of [Barbosa et al., 2008] suggest re-randomization of the ciphertexts by the $\mathcal{DB}$ before sending them to the $\mathcal{AS}$.

At the question & answer session of BIOSIG 2009, David Naccache proposed another attack against the scheme of [Bringer et al., 2007b], which is linear in M. The attack can be described as follows:

1. Assume that the threshold of the system is 0, namely zero error tolerance is applied. Since $\mathcal{AS}$ obtains an encrypted fresh biometrics $\varepsilon(b'_{i,k})$ the user U_i, a malicious $\mathcal{AS}$ replaces the first encrypted bit with the encryption of the plaintext (i.e. 1) that he chooses and checks the verification result.

2. If the matcher $\mathcal{M}$ returns reject, than the first bit of the template of U_i is 0, else if U_i is authenticated, $\mathcal{AS}$ obtains the first bit as 1.

3. $\mathcal{AS}$ fixes the obtained bit and continues with the next bit. In M queries, $\mathcal{AS}$ will extract the biometrics of U_i, where M is the size of the biometrics.

For error tolerant systems, the number of attempts of the $\mathcal{AS}$ will be at most $M - t$, where $t > 0$ denotes the threshold of the system.

3.5.2 Analysis of the Attack and its Extension

In this section, we present a new attack from the perspective of $\mathcal{DB}$ that will break the transaction anonymity notion. As one can note from the step 3 of Bringer et al.'s scheme [Bringer et al., 2007b], the $\mathcal{DB}$ computes the encryption of U_i's stored biometrics and sends it to $\mathcal{AS}$. Since at the last step, the $\mathcal{DB}$ forwards the authentication result based on the computations of the $\mathcal{M}$, the $\mathcal{DB}$ can also replace the first encrypted bit generated at step 3 of the verification protocol of [Bringer et al., 2007b] and depending on the authentication result, $\mathcal{DB}$ can extract the template of the U_i. This way, $\mathcal{DB}$ can track which user is authenticating, namely PIR protocol is useless and transaction anonymity is not guaranteed.

The solution of [Barbosa et al., 2008], namely rerandomization of the ciphertexts by the $\mathcal{DB}$ does not prevent the above stated attacks due to the nature of Goldwasser Micali encryption system. However, we can fix the scheme of [Bringer et al., 2007b] with a simple modification in the verification phase as below.

3.5.3 Modified Scheme

In this section we modify the scheme of [Bringer et al., 2007b] to avoid the above presented attacks. The intuition of our proposal is that, the sensor $\mathcal{S}$ does not send the fresh encrypted biometrics to the $\mathcal{AS}$, but to the $\mathcal{DB}$ directly. Thus, $\mathcal{AS}$ cannot modify the bits as in the attacks of David Naccache.

1. The sensor client $\mathcal{S}$ computes a fresh encryption of U_i's biometrics $\varepsilon(b'_i, pk)$=$(\varepsilon(b_{i,1}, pk), ..., \varepsilon(b_{i,M}, pk))$ using the public key pk of the $\mathcal{M}$, signs the hash of $\varepsilon(b'_i, pk)$ and forwards them together to $\mathcal{DB}$.

2. $\mathcal{AS}$ runs a PIR protocol by sending $\varepsilon(t_1), ..., \varepsilon(t_N)$ to $\mathcal{DB}$, where $t_j = 1$ for $i = j$, else $t_j = 0$.

3. $\mathcal{DB}$ checks the signature of $\mathcal{S}$ and computes $\varepsilon(b_{i,k}, pk) = \prod_{j=1}^{N} \varepsilon(t_j, pk)^{b_{j,k}} \bmod n$, for $1 \leq k \leq M$. Next, $\mathcal{DB}$ computes $\nu_k = \varepsilon(b'_{i,k})\varepsilon(b_{i,k}) \bmod n$=$\varepsilon(b'_{i,k} \oplus b_{i,k})$ for $1 \leq k \leq M$.

4. $\mathcal{DB}$ rerandomizes and permutes the ciphertexts before sending them to the $\mathcal{M}$.

5. The detached unit $\mathcal{M}$ with the secret key of the Goldwasser-Micali scheme decrypts the ciphertexts to compute the hamming weight and the decision is forwarded to the $\mathcal{AS}$.

6. Finally, $\mathcal{AS}$ either rejects or authenticates U_i.

Lemma 3.1. *The modified scheme guarantees Identity and Transaction Privacy based on the semantic security of the Goldwasser-Micali encryption scheme.*

Sketch of the Proof: Since $\mathcal{AS}$ only knows the index of the user U_i and the authentication result, $\mathcal{AS}$ cannot extract the biometrics of U_i using the properties of the Goldwasser Micali scheme. Similarly, $\mathcal{DB}$ does not have access to the secret keys of the $\mathcal{M}$ and the authentication result of U_i due to the security link property, so $\mathcal{DB}$ cannot identify which index i is trying to authenticate. This way, the security reduction presented in [Bringer et al., 2007b] becomes valid. Although the communication cost of the modified scheme is not optimal, the computation cost is only $O(N)$ modular multiplications for the $\mathcal{DB}$ instead of $O(N)$ modular exponentiations as in [Bringer and Chabanne, 2008], which combines Goldwasser Micali encryption with Paillier encryption to implement the (communication efficient) PIR of Lipmaa [Lipmaa, 2005] and store biometrics as encrypted sketches using a slightly different architecture than [Bringer et al., 2007b] to provide security against malicious $\mathcal{AS}$ and $\mathcal{DB}$.

3.6 Schemes based on ElGamal Encryption

In [Bringer et al., 2007c], the authors present another DBRA scheme based on the ElGamal encryption and a PIR scheme. The main difference of this biometric authentication system is the integration of a secure sketch scheme for error correcting a biometric (binary) string such as an 2048 bits Iris code and the use of ElGamal encryption. This way, there is no need for a similarity metric (i.e. hamming distance) for

the final decision, instead the system is used for equality testing. Here, each biometric string is stored at the $\mathcal{DB}$ as encrypted with the public key pk of the $\mathcal{AS}$ as opposed to the scheme of [Bringer et al., 2007b], where each biometric string is stored as a plaintext. Thus, $\mathcal{AS}$ generates an ElGamal key pair (pk, sk) during the setup phase of the protocol, where $pk = y = g^x$.

In the registration phase, the user U registers at the $\mathcal{DB}$ by sending $R = (R^1, R^2) = \mathsf{Enc}(g^b, pk) = (g^r, y^r g^b)$, namely the ElGamal encryption of its biometrics b to $\mathcal{DB}$ and the parameter PAR is publicly available for reconstruction of the same biometrics b using the secure sketch scheme. The user also registers his unique pseudorandom identifier ID at the $\mathcal{AS}$.

For authentication, the following steps are performed.

- The sensor client $\mathcal{S}$ sends the user identity ID to the $\mathcal{AS}$ and the encrypted fresh biometrics $X = (X^1, X^2) = \mathsf{Enc}(g^{b'}, pk)$ to the $\mathcal{DB}$ using the PAR for error-correction and ElGamal encryption.
- For each entry $1 \leq j \leq N$, the $\mathcal{DB}$ selects a random $r_j, r'_j \in \mathbb{Z}_q$ and computes

  $$C_j = ((g^{r'_j}(X^1(R_j^1)^{-1})^{r_j}, (y^{r'_j}(X^2(R_j^2)^{-1})^{r_j})) = (g^{r'_j}(g^r(R_j^1)^{-1})^{r_j}, y^{r'_j}(y^r g^{b'}(R_j^2)^{-1})^{r_j})$$

  where R_j, $1 \leq j \leq N$ is the ElGamal encryption of each user biometrics stored in the system ($\mathcal{DB}$).
- Finally, $\mathcal{AS}$ runs an efficient PIR protocol to obtain the value C corresponding to the user U and decrypts it using his secret key sk. If $\mathsf{Dec}(C)$=1, $\mathcal{AS}$ authenticates U, else rejects.

In addition, [Tang et al., 2008] presents a slightly modified version of this scheme by simplifying the randomization step of the $\mathcal{DB}$. Again, the same components, namely a PIR, secure sketch and ElGamal encryption scheme is considered. Apart from the computational cost of the PIR, the number of exponentiations computed by the $\mathcal{DB}$ is reduced from $O(4N)$ as in [Bringer et al., 2007c] to $O(2N)$ due to the use of a single random number instead of two different random numbers for the randomization of the ciphertexts. Also, as one can notice from the first step of the authentication phase of [Bringer et al., 2007c], the sensor client $\mathcal{S}$ communicates with the $\mathcal{DB}$ to send the fresh encryption of the biometrics, which could be impractical. In practice, there might be only very few organizations that can be trusted by human users to store their biometric information though they may want to use their biometrics for the authentication purpose at many authentication servers. Therefore, in [Tang et al., 2008], the authors suggest a scenario like that of Single Sign-On systems, where biometric information

for all authentication servers are centralizedly stored and managed. In this security model the centralized database will not be a bottleneck in the sense of security [Tang et al., 2008]. Specifically, each biometric string is stored at the $\mathcal{DB}$ as encrypted with the public key $pk_{\mathcal{AS}}$ of the $\mathcal{AS}$ as in [Bringer et al., 2007c], however, when the user authenticates to the system, the sensor client first encrypts the fresh biometrics using the public key $pk_{\mathcal{AS}}$ of the $\mathcal{AS}$ and next re-encrypts the result with the public key $pk_{\mathcal{DB}}$ of the $\mathcal{DB}$. Hence, the sensor does not need to communicate with the $\mathcal{DB}$ during the verification phase as in [Bringer et al., 2007c], instead $\mathcal{S}$ only communicates with the $\mathcal{AS}$. $\mathcal{AS}$ and $\mathcal{DB}$ generate each an ElGamal key pair during the setup phase of the protocol. The registration phase is the same as in [Bringer et al., 2007c].

For authentication, the following steps are performed in[Tang et al., 2008].

- The sensor client $\mathcal{S}$ sends the user identity ID and the encrypted fresh biometrics $Z = (Z^1, Z^2) = \mathsf{Enc}(X, pk_{\mathcal{DB}})$, where $X = (X^1, X^2) = \mathsf{Enc}(g^{b'}, pk_{\mathcal{AS}})$ to the $\mathcal{AS}$ using the PAR and ElGamal encryption. $\mathcal{S}$ also sends his signature σ on Z to $\mathcal{AS}$. Note that X has two components since the encryption scheme is ElGamal.
- $\mathcal{AS}$ first retrieves the index for ID and then forwards (Z, σ) to the database $\mathcal{DB}$.
- $\mathcal{DB}$ first verifies the signature σ. If the verification succeeds, $\mathcal{DB}$ decrypts Z to recover X. Next, for each entry $1 \leq j \leq N$, the $\mathcal{DB}$ selects a random $r_j \in \mathbb{Z}_q$ and computes $C_j = (((X^1(R_j^1)^{-1})^{r_j}, (X^2(R_j^2)^{-1})^{r_j})) = ((g^r(R_j^1)^{-1})^{r_j}, (y^r g^{b'}(R_j^2)^{-1})^{r_j}))$, where R_j is the ElGamal encryption of each user biometrics stored in the system.
- Finally, $\mathcal{AS}$ runs an efficient PIR protocol to obtain the value C corresponding to U and decrypts it using his secret key sk. If $\mathsf{Dec}(C)$=1, $\mathcal{AS}$ authenticates U, else rejects.

The security reductions of [Bringer et al., 2007c] and [Tang et al., 2008] are almost the same and presented in the full version of the papers.

3.7 Schemes based on Paillier Encryption Scheme

In [Barbosa et al., 2008], the authors describe a new distributed remote identification scheme for biometrics represented as a set of features (i.e. feature vector) after feature extraction by integrating a Support Vector Machine (SVM) to work as a multi-class authentication classifier. In pattern recognition and in image processing, feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be redundant then the input data will be transformed into a reduced representation set of features (also named

features vector). Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

This system is actually the first system that works for any type of biometrics represented as a set of features (i.e. feature vector). Particularly, the $|\mathbb{U}|$-class SVM implemented in [Barbosa et al., 2008] is described as follows: For each user $U_i \in \mathbb{U}$ with biometrics b_i, a mono classifier is trained using the remaining users ($\mathbb{U}/U_i$) as the rejected class after extracting the biometric feature vector b_i of U_i. Next, a user profile $w^*_{\mathbb{U}}$ for each user U_i is constructed. Each user profile $w^*_{\mathbb{U}}$ consists of support vectors $SV_{i,j}$ and their weights $\alpha_{i,j}$, where $i = 1...S, j = 1...|\mathbb{U}|$. This will finish the registration phase of the system. For identification, each component of the feature vector b_i is encrypted by $\mathcal{S}$ using Paillier encryption scheme and sent to the $\mathcal{AS}$. $\mathcal{AS}$ forwards the encrypted biometric data to $\mathcal{DB}$, which computes the SVM classification values $class$ in the encryption domain by using the homomorphic properties of Paillier encryption system. Specifically, $\mathcal{DB}$ takes the profile data $w^*_{|\mathbb{U}|}$ and computes for each class $1 \leq j \leq |\mathbb{U}|$ the distance of b_i to the $w^*_{|\mathbb{U}|}$ in the encryption domain. Next, $\mathcal{DB}$ re-randomizes the resulting ciphertexts and sends the final vector $class$ of size $|\mathbb{U}|$ to $\mathcal{AS}$, which permutes and re-randomizes this vector to $sclass$. Next, $\mathcal{VU}$ decrypts each component of $sclass$ and finds the index d of the maximum positive scaler contained in the decrypted vector. If there exists not such a positive index, $\mathcal{VU}$ sends $\perp$ to $\mathcal{AS}$, else it sends d. Finally, $\mathcal{AS}$ recovers the identity of U_i using d and the inverse of the permutation used in $sclass$. The communication cost of this scheme is $O(N)$ ($N = |\mathbb{U}|$) and the computation cost is $O(N)$ exponentiations mod q^2. Thus, the system is not an efficient solution in terms of both communication and computational cost.

Secondly, [Bringer and Chabanne, 2008] uses Goldwasser-micali encryption and the PIR scheme of [Lipmaa, 2005] that is based on Paillier encryption scheme for storing biometrics as encrypted sketches, which we summarize as below.

In the enrollment phase, the user U registers at the $\mathcal{DB}$ by sending $R = (R^1, R^2) = \mathsf{Enc}(\mathsf{PAR}, pk)$ and $H(c)$, namely the encryption of its biometric sketch $\mathsf{PAR} = c \oplus b$ using Goldwasser-Micali encryption scheme and the hash of the codeword c, i.e. $H(c)$ to $\mathcal{DB}$, where the parameter PAR is not publicly available as in [Bringer et al., 2007c, Tang et al., 2008]. The user also registers his pseudorandom identifier ID and $H(c)$ at the $\mathcal{AS}$. For authentication, the following steps are performed.

- $\mathcal{S}$ sends the user identity ID to the $\mathcal{AS}$ and the encryption of the fresh biometrics $X = (X^1, X^2) = \mathsf{Enc}(b', pk)$ using Goldwasser-Micali encryption.
- $\mathcal{S}$ integrates the encrypted biometrics of the user into the PIR request that is sent to the $\mathcal{DB}$, which returns the encryption of $c \oplus b' \oplus b$ and the encryption of

$H(c)$ to the $\mathcal{AS}$.

- Finally, $\mathcal{AS}$ decrypts the values with the help of the hardware security model that stores the secret keys of the system and obtains $c' = c \oplus b' \oplus b$ and $H(c)$. If $\mathsf{dis}(b, b') < t$, then $\mathcal{AS}$ is able to decode c' and obtains a codeword c''. Next, he checks $H(c) = H(c'')$ to accept/reject the authentication request of U.

3.8 A first attempt for an efficient Biometric Remote Authentication

In this section, we present our first protocol that combines ElGamal encryption, the PIR of [Gentry and Ramzan, 2005] and an efficient biometric data storage mechanism different from the systems described previously. We note that the systems of [Bringer et al., 2007b, Bringer and Chabanne, 2008, Bringer et al., 2007c, Tang et al., 2008] are only applicable to biometrics that can be represented as binary strings in the Hamming space in order to perform the authentication through a binary string matching. For example, iris is one type of such biometrics, but for other biometrics, how to construct a secure authentication scheme in this security model is left as an open problem in [Tang et al., 2008]. The scheme of [Barbosa et al., 2008] is designed for biometrics represented as short sequences of integer numbers (i.e. features), but its efficiency is incomparible to Bringer et al.'s schemes since both the communication and computational cost of [Barbosa et al., 2008] is of $O(N)$. An overview of the existing schemes are presented in table 3.3. Thus, our protocol is designed for the more general representation of biometrics, i.e. which can be an ordered/unordered set of features, however, we will introduce a more efficient storage mechanism compared to the previous schemes. The architecture will be the same of [Bringer et al., 2007b, Barbosa et al., 2008, Bringer and Chabanne, 2008], where the systems employ an independent verification unit $\mathcal{VU}$. In order to improve the accuracy, we employ the secure sketch scheme of [Li et al., 2006] scheme that corrects the white noise in biometrics, where biometrics is assumed as a k-tuple of ordered features.

3.8.1 An efficient storage mechanism for biometrics

In [Li et al., 2006] it is observed that many biometric templates can be represented in a general form: The original biometrics can be considered as a list of k points that can be ordered. Under noise, each point can be perturbed by a distance less than δ, and on top of that, at most t points can be replaced. Here, the first noise is defined as the white noise, and the second noise as replacement noise. We note that this similarity

measure can be applied to handwritten online signatures, iris patterns, voice features, and face biometrics [Li et al., 2006].

However, the schemes designed according to the model of Bringer et al. all require a secure sketch for correcting a binary string such as an Iris code, thus the systems are only applicable to biometrics that can be represented as a binary string. Although the first scheme in this model [Bringer et al., 2007b] does not employ a secure sketch, the system also assumes biometrics as a binary string. Consequently, [Tang et al., 2008] leaves the design of a secure remote biometric authentication scheme for other biometric modalities as an open problem.

In our design, we assume biometrics as a set of ordered biometric features [Li et al., 2006, Sutcu et al., 2007, Chang and Li, 2006] where we map each feature to an element of a finite field (for instance using a hash function as in [Sahai and Waters, 2005, Baek et al., 2007]). Thus any type of biometrics that can be represented as a set of ordered features is suitable for our system. Next, we design the biometric database as a random pool of features, which is a new storage mechanism introduced in [Sarier, 2009a] and further developed in [Sarier, 2010b]. Hence, instead of storing at each database index the complete biometric template of a user (either in encrypted form or in clear), we store a single feature without the information to which user it belongs. We store each feature after the mapping as in [Sahai and Waters, 2005, Baek et al., 2007], in form of g^{w_i}, namely as exponentiations of the generator of the ElGamal group. Thus an attacker who compromised the server-end or colluding authentication server and database cannot obtain the identity-biometrics relation. During the registration phase, the user registers each feature at a random database location and stores the index of the database location at his tamper-proof smart card as encrypted. If some of the features of his biometrics are already stored at the database, than the user does not need to re-register these common features. Also, each feature w_i is stored at the $\mathcal{DB}$ as g^{w_i} and no further encryption is performed, which prevents ciphertext expansion. This way, the storage cost of the database reduces as the common features are not stored more than once and ciphertext expansion is avoided. To quantify this storage cost, [Sarier, 2010b] presents a simple analysis by referring to the experiment in [Mansukhani et al., 2007], which measures minutiae pair matches for fingerprint verification on a small fingerprint database of 100 users with 8 prints of the same finger as shown in table 3.2. In this experiment, the total number of pairs of matched minutiae (i.e. fingerprint feature) is counted for $\binom{50}{2} = 1225$ comparisons of fingerprints belonging to 50 different users. Since a fingerprint is represented by 30-50 minutiae [Mansukhani et al., 2007], one can easily compute that when our system is applied even on such a small database, we can reduce the storage cost approximately by 10% (i.e, $3991/(50 \cdot 1225 - 3991)$). In case of large identification systems, the storage cost will decrease much more [Sarier, 2010b].

Besides, since no biometric template is stored as an entry, there is no need to apply a

Table 3.2: The number of common features [Mansukhani et al., 2007]

	No.of Users	No. of Fingerprint Pairs Compared	Total Matched Point Pairs
Same User	50	1400	37705
Diff. User	50	1225	3991

homomorphic encryption scheme to store the biometric template as encrypted, where the ciphertext size is twice the plaintext size as in [Tang et al., 2008, Bringer et al., 2007c] and the storage cost of each user in [Bringer and Chabanne, 2008] is given as 128kbytes as each bit of the sketch of the 2048-bits Iris code is stored as encrypted using Goldwasser-Micali scheme. Finally, the choice of the system parameters of [Bringer and Chabanne, 2008, Bringer et al., 2007b] results in a constraint on the size of $\mathcal{DB}$. However, the database storage cost of our system is $(k - c) \cdot P$ for each user due to the c common features that are not stored twice, where P is the size of an element of the elliptic curve ElGamal group $\mathbb{G}$ and k is the number of biometric features of each user. For instance, $P = 171$ bits for a 160-bit elliptic curve on which the ElGamal encryption is implemented.

To improve the accuracy further, our system can also incorporate a secure sketch for ordered feature sets, which is described in section 3.8.5.

3.8.2 A concrete scheme based on ElGamal

In this section, we present our new protocol that implements a new and more efficient storage mechanism for the biometric database and results in a reduced storage cost and applicable for various biometric modalities that can be represented as an ordered set of features opposed to the schemes of Bringer et al. An overview of the authentication flow of the new protocol is given in figure 3.3.

Registration Phase

- The sensor client $\mathcal{S}$ generates a key pair $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ for a signature scheme and publishes the public key $pk_{\mathcal{S}}$.
- The authentication server $\mathcal{AS}$ generates an ElGamal key pair $(pk_{\mathcal{AS}}, sk_{\mathcal{AS}})$, where $pk_{\mathcal{AS}} = (\mathbb{G}, g, y_{\mathcal{AS}})$, $y_{\mathcal{AS}} = g^{x_{\mathcal{AS}}}$ and $sk_{\mathcal{AS}} = x_{\mathcal{AS}}$ and publishes $pk_{\mathcal{AS}}$.
- The verification unit $\mathcal{VU}$ generates an ElGamal key pair $(pk_{\mathcal{VU}}, sk_{\mathcal{VU}})$, where $pk_{\mathcal{VU}} = (\mathbb{G}, g, y_{\mathcal{VU}})$, $y_{\mathcal{VU}} = g^{x_{\mathcal{VU}}}$ and $sk_{\mathcal{VU}} = x_{\mathcal{VU}}$ and publishes $pk_{\mathcal{VU}}$.

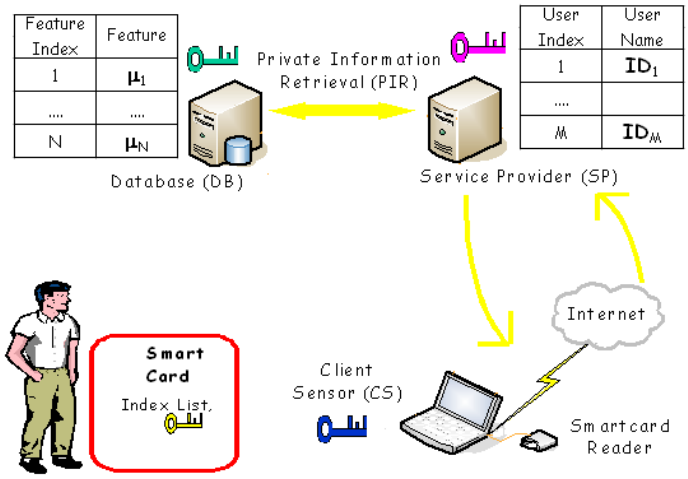

Figure 3.3: Overview of the protocol in [Sarier, 2010b]

- The user U generates his personalized username ID and registers it at the $\mathcal{AS}$, and registers his biometric features in form of g^{w_i} for $1 \leq i \leq k$ at the $\mathcal{DB}$ by choosing a random location s_i for each feature. If some of the features w_i of the user is already stored, than the database returns the indices of these features so that no feature is stored twice. The indices of these locations $\{s_1, ...s_k\}$ are stored at the tamper-proof smartcard of U as encrypted with the public key of $\mathcal{VU}$, namely as $Index_i = \mathsf{Enc}(s_i, pk_{\mathcal{VU}})$ for $1 \leq i \leq k$.

Verification Phase

- The sensor client $\mathcal{S}$ sends the user identity ID to $\mathcal{AS}$ to start an authentication request. Next, $\mathcal{S}$ computes the encrypted fresh biometrics B' of user U, i.e. for each feature $w_i' \in B'$, $Z_i = (Z_i^1, Z_i^2) = \mathsf{Enc}(X_i, pk_{\mathcal{VU}})$, where $X_i = (X_i^1, X_i^2) = \mathsf{Enc}(g^{w_i'}, pk_{\mathcal{AS}})$ using the ElGamal encryption. Also, $\mathcal{S}$ retrieves from the smartcard of the user the encrypted indices $Index_i$ and sends Z_i's and $Index_i$'s to the $\mathcal{VU}$ together with his signature on $(Z_1||...Z_k||Index_1||...||Index_k||T)$. Here T denotes the time stamp.

- $\mathcal{VU}$ first verifies the signature of the sensor S. If the verification succeeds, $\mathcal{VU}$ decrypts Z_i to recover X_i. $\mathcal{VU}$ also decrypts $Index_i = \mathsf{Enc}(s_i, pk_{\mathcal{VU}})$ to obtain the indices of the user for the PIR protocol.

- Next, for each entry $1 \leq j \leq N$, the $\mathcal{DB}$ computes $R_j = \mathsf{Enc}(g^{w_j}, pk_{\mathcal{AS}})$, where

R_j is the ElGamal encryption of each biometric feature stored in the $\mathcal{DB}$.

- $\mathcal{VU}$ runs the PIR protocol of [Gentry and Ramzan, 2005] to obtain the values R_i corresponding to the user U's biometrics. Next, for each $1 \leq i \leq k$, the $\mathcal{VU}$ selects a random $r_i \in \mathbb{Z}_q$ and computes $C_i = (((X_i^1(R_i^1)^{-1})^{r_i}, (X_i^2(R_i^2)^{-1})^{r_i}))$, where X_i is the ElGamal encryption of the fresh biometric feature of the authenticating user U. The C_i's are permuted before sending them to the $\mathcal{AS}$.

- Finally, $\mathcal{AS}$ obtains the permuted C_i's from the $\mathcal{VU}$ and decrypts them using his secret key. If most of the C_i's satisfy $\mathsf{Dec}(C_i)$=1 out of k features of user U, namely the total number of matching features of user U is above the system's threshold, $\mathcal{AS}$ authenticates U, else rejects.

Remark 3.1. *By considering the authentication workflow of [Tang et al., 2008], we can also allow the sensor to communicate only with the $\mathcal{AS}$ instead of communicating both with the detached $\mathcal{VU}$ and $\mathcal{AS}$. The signature of the sensor and the time stamp prevents a malicious $\mathcal{AS}$ from replay attacks and brute-force attacks in order to obtain some information about the reference template of the user.*

3.8.3 Security Analysis

Theorem 3.1. *The proposed scheme achieves identity privacy against a malicious $\mathcal{AS}$, based on the semantic security of the ElGamal encryption scheme.*

Proof. If the proposed scheme does not achieve identity privacy against malicious $\mathcal{AS}$ or an attacker who has compromised it, we construct an algorithm A', which receives a public key pk from the ElGamal challenger and runs A as a subroutine to break the semantic security of the ElGamal scheme. Here, A denotes $\mathcal{AS}$ or the attacker who compromised it. For the security reduction, we will use the same technique as in [Bringer and Chabanne, 2008].

The reduction A' sets $pk_{\mathcal{VU}} = pk$ which is obtained from its challenger. Next, the adversary A generates two possible templates B_e^0, B_e^1, where $B_e^0 = \{w_i\}_{1 \leq i \leq k}$ and $B_e^1 = \{w_i'\}_{1 \leq i \leq k}$ describe the same user U_e. Thus, $|B_e^0 \cap B_e^1| \geq t$, where t denotes the system's error tolerance threshold. Also, A generates the reference templates of the remaining users in the system and forwards them to A'.

A' sends the reference templates generated by A and the two possible templates B_e^0, B_e^1 for user U_e to his challenger. Since the templates are first encrypted with the public key of the $\mathcal{AS}$ and then with the public key of the $\mathcal{VU}$ during verification, A' computes $m_i^0 = y_{\mathcal{AS}}^{a_i} g^{w_i}$ and $m_i^1 = y_{\mathcal{AS}}^{a_i} g^{w_i'}$ for $a_i \xleftarrow{R} \mathbb{Z}_q$ corresponding to the challenge user U_e and sends m_i^0, m_i^1 to its challenger.

Suppose A' receives the challenge $c_i^* = \mathsf{Enc}(m_i^\beta, pk)$, where β is the coin toss of the challenger. The indices of the database locations of each user in the system are only known to the challenger of A'.

A' answers the Verification queries from A as follows:

- For any user U_l, $1 \leq l \leq M$,

  If $l = e$, A' randomly selects $r_i \overset{R}{\leftarrow} \mathbb{Z}_q$ and generates for $1 \leq i \leq k$, (Z_i^1, Z_i^2), where $Z_i^1 = \mathsf{Enc}(g^{a_i + r_i}, pk)$ and $Z_i^2 = c_i^* \otimes \mathsf{Enc}(y_{\mathcal{AS}}^{r_i}, pk)$. Here, for any two ElGamal ciphertexts (c_1, c_2) and (c_3, c_4), the operator $\otimes$ is defined as follows: $(c_1, c_2) \otimes (c_3, c_4) = (c_1 c_3, c_2 c_4)$.

  If $l \neq e$, A' generates for $1 \leq i \leq k$, (Z_i^1, Z_i^2) by directly following the protocol specification.

  We note that some of the features of a user's biometric template that are selected randomly can be corrupted by A' to simulate the white noise in the biometrics. The total number of the corrupted features is less than the system threshold.

- A' sends (Z_i^1, Z_i^2) to $\mathcal{VU}$ together with the encrypted indices of user U_l obtained from his challenger. The signature of the sensor is also simulated by A'.

- To simulate the result of the computations performed by $\mathcal{VU}$, A' picks at random some values $t_i \overset{R}{\leftarrow} \mathbb{Z}_q$ and returns A $(\mathsf{Enc}(C_i, pk_{\mathcal{AS}}))$ for $1 \leq i \leq k$, where $C_i = g^{t_i}$, $t_i \overset{R}{\leftarrow} \mathbb{Z}_q$ for the features that are corrupted by A' at the first step of the game to simulate the noise. For the majority of the features we have $C_i = 1$.

After listening to polynomially many authentication runs, A outputs β'. A' terminates by outputting $\beta = \beta'$. Since A' uses the same coin toss as the ElGamal challenger, then A' wins the game against the semantic security of ElGamal scheme with the same advantage of A. □

It is clear that the verification unit $\mathcal{VU}$ and the database $\mathcal{DB}$ have zero advantage in distinguishing between (ID_e, B_e^0) and (ID_e, B_e^1), because they have no access to any information about the users' identities [Bringer et al., 2007b].

Theorem 3.2. *The proposed scheme achieves transaction anonymity against malicious $\mathcal{DB}$, based on the security (user privacy) of the PIR protocol.*

Proof. If the proposed scheme does not achieve transaction anonymity against malicious $\mathcal{DB}$, then we can construct an algorithm A', which receives the public parameters from the PIR challenger and runs A as a subroutine to break the user privacy of the PIR scheme. Here, A denotes $\mathcal{DB}$ or the attacker who compromised it that tries to break the transaction anonymity of the biometric scheme.

A generates the reference templates of the users in the system and returns these data to A'. A' forwards them to his PIR challenger and receives the public parameters of the PIR protocol from his PIR challenger and faithfully answers the Verification queries from A. Since the database only communicates with the $\mathcal{VU}$, the reduction A' has to simulate the $\mathcal{VU}$ that runs the PIR protocol. If we instantiate the PIR protocol with the protocol of [Gentry and Ramzan, 2005], the challenger returns A' the public parameters generated before querying for the blocks.

- On receiving (e_0, e_1) from A, A' forwards this to his PIR challenger that flips a coin β and gives the (public) PIR parameters for the retrieval of the feature set of the randomly picked user to the A' so that A' simulates the $\mathcal{VU}$ that runs the PIR protocol to retrieve the data from $\mathcal{DB}$. For the protocol of [Gentry and Ramzan, 2005], the challenger generates the cyclic group $\mathbb{G}$ and its generator g and sends them to A'. The challenger keeps the factorization of the order of $\mathbb{G}$ as secret.

- A' faithfully answers the oracle queries of A by forwarding these requests to his challenger and simulating $\mathcal{VU}$ with the parameters obtained from his challenger. A finally outputs β' and A' forwards this to his challenger.

The simulation is faithful, and the advantage of A in attacking the transaction privacy of the biometric system is equal to the advantage of A' that runs against the user privacy of the PIR protocol. □

3.8.4 Efficiency Analysis

In table 3.3, we summarize various remote biometric-based authentication schemes that satisfy the security model of Bringer et al. [Bringer et al., 2007b, Bringer and Chabanne, 2008, Tang et al., 2008]. When we take typical values for the parameters as listed in table 3.3, we obtain the following relations. For biometric modalities with M=512 bytes template sizes [Itakura and Tsujii, 2005] and for 160-bit elliptic curves, we have $M \approx kP$, if $20 \leq k \leq 30$ as implemented in [Li et al., 2006, Mansukhani et al., 2007]. Also, for current PIR systems with communication cost PIR, we have $PIR << O(N)$.

3.8.5 Improving the accuracy

In order to improve the accuracy of our biometric scheme, we use of a secure sketch that error-corrects the white noise in biometric that is represented as a set of ordered biometric features. Face is an example of such biometrics [Li et al., 2006]. Here, white

Table 3.3: Comparison of various biometric authentication systems

Scheme	Computation Cost	Storage at $\mathcal{DB}$-index	Storage per user	Comm. cost
[Bringer et al., 2007b]	M exp + $(MN)/2$ mult	M bits	M bits	$O(N)$
[Barbosa et al., 2008]	$O(N)$ exp	$\|n\|k$ bits	$\|n\|k$ bits	$O(N)$
[Bringer et al., 2007c]*	$O(N)$ exp	$2M$ bits	$2M$ bits	PIR
[Bringer and Chabanne, 2008]	$O(N)$ exp	$\|n\|M$ bits	$\|n\|M$ bits	PIR
[Tang et al., 2008]	$O(N)$ exp	$2M$ bits	$2M$ bits	PIR
[Sarier, 2010b]	$O(N)$ exp	P bits	$(k-c)P$ bits	PIR

*The first biometric scheme
Abbreviations: N= number of entries in $\mathcal{DB}$; k=dimension of the feature vector; M= size of the biometric template; P=size of a single stored feature; c= number of common features of a user; $|n|$=size of an RSA modulus

noise means that each point (i.e. feature) can be perturbed by a distance less than δ. In our biometric remote authentication protocol [Sarier, 2010b], the replacement noise is tolerated by the threshold t of the set difference metric.

Secure Sketch for ordered biometrics

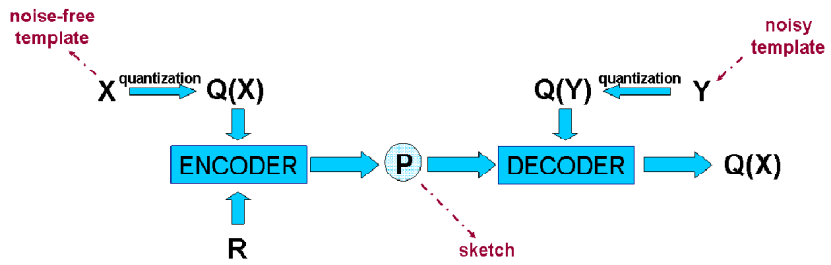

Figure 3.4: Sketch Generation and Reconstruction [Sutcu et al., 2007]

The secure sketch of [Li et al., 2006, Sutcu et al., 2007, Chang and Li, 2006] assumes that many biometric templates can be represented as a sequence of points in some bounded continuous domain. (e.g., real numbers resulted from some signal processing techniques). To handle points in some continuous domain, the authors of [Li et al., 2006, Sutcu et al., 2007] quantize (discretize) the points such that they become points in a discrete domain so that a known sketch scheme in discrete domain is applicable to construct the sketch. When a fresh measurement of the same biometrics is given, it is

quantized using the same quantizer and the corresponding reconstruction algorithm in the discrete domain is used to recover the quantized version of the original data points, thus only the quantized original is reconstructed. There are two types of noise that can occur. The first noise, white noise, perturbs each points by a small distance δ, and the second noise, replacement noise, replaces some points by different points, whose number is blow a threshold t. We summarize the properties of the sketch described in [Li et al., 2006] as follows. Each biometrics can be written as a sequence $b = (\nu_1, ..., \nu_k)$, where a feature ν is an element of the universe U such that $\nu \in \mathbb{R}$ and $0 \leq \nu < 1$. R denotes a similarity relation $R \subset U \times U$ on U, where U is a set that may be uncountable. For each pair of biometrics (b, b'), one can write $(b, b') \in R$, if there exists a set $S \subset b \cap b'$ with $|S| \geq t$ for some threshold t, and for every $\nu \in S$, $|\nu - \nu'| < \delta$ for some threshold δ. Let M be a set of finite points, and let $Q : U \to M$ be a function that maps points in U to points in M. We will refer to such a function Q as a quantizer. The quantizer Q_λ is a member of a family of quantizers Q parameterized by the step size λ, which is defined as $Q_\lambda : U \to M$. In other words, a quantization is applied to transform the points in the continuous domain to a discrete domain and the step size $\lambda \in \mathbb{R}$ as a measure of the precision of the quantized biometrics. We assume that $0 < \lambda \leq \delta$.

Let us give an example. For a feature $\nu \in \mathbb{R}$ we employ a scalar quantizer Q_λ with step size $\lambda = 0.001$ to map the feature to an integer in [0,1000], such that $Q_\lambda(\nu) = w$. The quantization of b is defined as $Q_\lambda(b) = \langle Q_\lambda(\nu_1), ..., Q_\lambda(\nu_k) \rangle$ and the corresponding quantized domain is $M_\lambda = [0, \lceil \frac{1}{\lambda} \rceil]$. Thus, $\lambda Q_\lambda(\nu_i) \leq \nu_i \leq \lambda(Q_\lambda(\nu_i)+1)$ for each $\nu \in U$.

Similar to the case in the continuous domain, we have $|w - w'| < \delta_\lambda$ in the quantized domain, where $\delta_\lambda = \lceil \frac{\delta}{\lambda} \rceil$. Thus, under white noise, a point w in the quantized domain can be shifted by a distance of at most δ_λ.

Furthermore, for each quantized domain M_λ we consider a codebook C_λ, where every codeword $c \in C_\lambda$ has the form $c = z(2\delta_\lambda + 1)$ for some non-negative integer z. We use $C_\lambda(\cdot)$ to denote the function such that given a quantized feature w, it returns a value $c = C_\lambda(w)$ such that $|w - c| \leq \delta_\lambda$. Here, $C_\lambda(\cdot)$ is the function that finds the unique codeword $c \in C_\lambda$ that is nearest to w in the codebook C_λ [Li et al., 2006].

- The SS function takes the quantized biometrics $Q_\lambda(b) = (w_1, ..., w_k) \in M_\lambda$ as input and computes for each w_i, $c_i = C_\lambda(w_i)$ and outputs the $\mathsf{PAR} = (\Delta_1, ..., \Delta_k) = (w_1 - c_1, ..., w_k - c_k)$, In other words, for every w_i, the encoder SS outputs the distance of w_i from its nearest codeword in the codebook C_λ.

- The Rec function takes a quantized fresh biometric $Q_\lambda(b') = (w'_1, ..., w'_k)$ and PAR as input and computes $c_i = C_\lambda(w'_i - \Delta_i)$ for $1 \leq i \leq k$ and outputs $Q_\lambda(b) = (c_1 + \Delta_1, ..., c_k + \Delta_k)$. In other words, the decoder Rec shifts every w'_i by Δ_i, maps it to the nearest codeword in C_λ, and shifts it back by the same distance. The encoders and the decoders work only on the quantized domain.

The details of the construction of the codebooks, the computations and selection of the parameters for a concrete application of face biometrics are presented in [Li et al., 2006, Sutcu et al., 2007].

3.8.6 The concrete protocol with improved accuracy

Registration Phase

- $\mathcal{S}$, $\mathcal{AS}$, $\mathcal{VU}$ generate their keys $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ and publishes them as in section 3.8.2.
- The user U generates his personalized username ID and registers it at the $\mathcal{AS}$, computes his quantized biometrics $Q_\lambda(b) = (w_1, ..., w_k)$ and registers his biometric features g^{w_i} for $1 \leq i \leq k$ at the $\mathcal{DB}$ by choosing a random location s_i for each feature. The indices of these locations $\{s_1, ...s_k\}$ are stored at the tamper-proof smartcard of U as encrypted with the public key of $\mathcal{VU}$, namely as $Index_i = \mathsf{Enc}(s_i, pk_{\mathcal{VU}})$ for $1 \leq i \leq k$. Finally, the encoder SS of the sketch computes for each w_i, $c_i = C_\lambda(w_i)$ and outputs the public parameter $\mathsf{PAR}= (\Delta_1, ..., \Delta_k) = (w_1 - c_1, ..., w_k - c_k)$ that is stored in the smartcard of the user. We will explain in the next section why the helper data PAR should be stored in the smartcard of the user.

Verification Phase

- The sensor client $\mathcal{S}$ sends the user identity ID and the encrypted fresh biometrics $Z_i = (Z_i^1, Z_i^2) = \mathsf{Enc}(X_i, pk_{\mathcal{VU}})$, where $X_i = (X_i^1, X_i^2) = \mathsf{Enc}(g^{w_i}, pk_{\mathcal{AS}})$ to the $\mathcal{AS}$ using the PAR and ElGamal encryption. Here, w_i's are computed using the (fresh) quantized biometrics $Q_\lambda(b) = (w'_1, ..., w'_k)$ and PAR as described in section 3.8.5. The rest of the operations are as in section 3.8.2.
- Finally, $\mathcal{AS}$ obtains C_i's from the $\mathcal{VU}$ and decrypts them using his secret key. If $\mathsf{Dec}(C_i)$=1 and the total number of matching features is above the threshold t of the set difference metric, $\mathcal{AS}$ authenticates U, else rejects. Hence, the threshold t allows the protocol to tolerate the replacement noise in the biometrics using the set difference metric.

3.9 New Attacks

As it is described in section 3.4.2, identity privacy notion means that a malicious authentication server $\mathcal{AS}$ cannot distinguish the stored biometrics of a chosen user

Attack $Atk1_{\mathcal{A}}$
$(i, ID_i, b_i^0, b_i^1, (ID_j, b_j)_{\{j \neq i\}}) \leftarrow \mathcal{A}(1^l)$.
$b_i = b_i^\beta \xleftarrow{R} \{b_i^0, b_i^1\}$
$\emptyset \leftarrow Enrollment((ID_j, b_j)_j)$
Use public data of ID_i: $\mathsf{PAR}_i = c \oplus b_i^\beta$
Compute $b_i^1 \oplus \mathsf{PAR}_i = c'$
If $\mathsf{Decode}(c') = c'$
Return $\beta = 1$
Else if $\mathsf{Decode}(c') = b_i^0 \oplus \mathsf{PAR}_i$
Return $\beta = 0$

Attack $Atk2_{\mathcal{A}}$
$(i, ID_i, b_i^0, b_i^1, (ID_j, b_j)_{\{j \neq i\}}) \leftarrow \mathcal{A}(1^l)$
$b_i = b_i^\beta \xleftarrow{R} \{b_i^0, b_i^1\}$
$\emptyset \leftarrow Enrollment((ID_j, b_j)_j)$
Use public data of ID_i: $\mathsf{PAR}_i = c \oplus b_i^\beta$
Compute $b_i^1 \oplus \mathsf{PAR}_i = c'$
If $\mathsf{Decode}(c') = \perp$
Return $\beta = 0$
Else If $\mathsf{Decode}(c') = b_i^1 \oplus \mathsf{PAR}_i$
Return $\beta = 1$

even though $\mathcal{AS}$ knows that the stored biometrics is one of two possible templates generated by $\mathcal{AS}$. This notion is actually the analogue notion of indistinguishability of ciphertexts defined for encryption systems, where the adversary A generates two equal length messages and has to distinguish from the ciphertext that is the encryption of one of these two messages generated by A.

Considering the security model for identity privacy as described in section 3.4.2, we first assume that the adversary produces two biometric templates (b_i^0, b_i^1) for the target user U_i with ID_i such that $\mathsf{dis}(b_i^0, b_i^1) < t$, where t is the error correction threshold of the secure sketch scheme. We call this first attack as $Atk1_{\mathcal{A}}$, which successfully distinguishes the template that was registered for the challenge user ID_i using the public helper data PAR_i, which is the output of the secure sketch in order to be used to error correct the biometrics.
For the attack $Atk1_{\mathcal{A}}$, the adversary can easily distinguish which template was chosen by the challenger to be registered for U_i by looking at the output of the decoding function of the secure sketch. If he correctly guessed the template b_i^1, then the computation of $b_i^1 \oplus \mathsf{PAR}_i$ will result in a correct codeword, which does not need to be error corrected. Otherwise, he returns $\beta = 0$.

The second case we consider is that the adversary produces two biometric templates (b_i^0, b_i^1) for the target user ID_i with $\mathsf{dis}(b_i^0, b_i^1) > t$, which we call as $Atk2_{\mathcal{A}}$. We note that this pair of templates still describe the same user U_i, since the variation of the biometrics can be larger then the error-correction capacity of the secure sketch. Our attack successfully distinguishes the template that was registered for the challenge user ID_i using the public helper data PAR_i. The difference to the previous attack is that, if b_i^1 is not the template that was registered by the challenger $\mathcal{C}$, then, since the distance between the two templates (b_i^0, b_i^1) is above the error-correction capacity, the decoding procedure will not work. Thus, the registered template is b_i^0, and $\mathcal{A}$ returns $\beta = 0$.

The reason that the public data PAR of the secure sketch scheme helps the adversary

in the identity privacy game is due to the fact that for secure sketch construction the standard notions of security do not fit. The statement "PAR leaks no information about the biometric template b" is normally formalized by requiring that b and PAR be almost statistically independent. Even the analogue requirement for computationally bounded adversaries, semantic security, is impossible here: if Eve knows that b is one of two similar strings (b_1, b_2), then she can compute b from PAR and b_1. The difficulty, then, is that the standard definitions of security require secrecy even when Eve knows a lot about b, which is in contrast to the security of sketches, where Eve is sufficiently uncertain about b, since biometrics is assumed as secret data. In [Dodis and Smith, 2005], it is shown that secure sketches can only guarantee entropic security, which assumes that the adversary is sufficiently uncertain about the user's biometrics, which implies that secure sketches can never guarantee the notion of indistinguishability for computationally bounded adversaries. Thus, the schemes of [Bringer et al., 2007c, Tang et al., 2008] and any biometric remote authentication scheme that assumes biometrics and the required secure sketch as public data are vulnerable to this attack and cannot satisfy identity privacy.

As opposed to the schemes of [Bringer et al., 2007c, Tang et al., 2008], the scheme of [Bringer and Chabanne, 2008] stores the sketch as encrypted in the $\mathcal{DB}$. Thus, a malicious $\mathcal{AS}$ has only access to different corrupted codewords $c'_{ik} = \mathsf{PAR}_i \oplus b'_{ik}$, where b'_{ik} is the fresh biometrics of the user U_i at the k^{th} authentication run. However, this data can also help the malicious $\mathcal{AS}$ when playing the identity privacy game, since there is no restriction on the two templates the adversary generates for the challenge user U_i. Assume that the adversary knows that biometrics of U_i behaves according to some distribution, and has determined the mean of this distribution after taking enough samples; a well-motivated adversary can take more measurements, and thus determine the mean more accurately. Let the adversary set one of the two templates he generates in the game as equal to the mean value of this distribution, i.e. $b_i^0 = \mu$ and the second template he has to output equal to the value that is the maximum (allowable) distance to the mean, i.e. $b_i^1 = \mu + \delta$, where 2δ denotes the variability of the biometrics of U_i with identity ID_i, namely the range of U_i's biometrics. Enough number of samples $\{b_{ir}^S\}_{1<r<N}$ of U_i's biometric data b_i allows the adversary to compute this range information. Since the malicious $\mathcal{AS}$ performs the decoding of the corrupted codeword c'_i for user U_i and obtains the correct codeword c_i that was used in $\mathsf{PAR}_i = c_i \oplus b_i^\beta$, $\mathcal{AS}$ has access to c'_{ik}'s for $1 < k < M$ obtained at the k^{th} authentication run of U_i and the unique codeword c_i after decoding each corrupted codeword c'_{ik}. The attack is denoted by $Atk3^*_{\mathcal{A}}$.

The intuition of this attack is that by setting one of the templates to the mean of the distribution of U_i's biometrics, and the other template to the maximum value of its range, listening to enough protocol runs (polynomially many) of U_i allows the adversary to distinguish which template was registered using a statistical attack on the

Attack $Atk3^{*}_{\mathcal{A}}$

$(i, ID_i, b_i^0, b_i^1, (ID_j, b_j)_{\{j \neq i\}}) \leftarrow \mathcal{A}(1^l)$ s.t. $b_i^0 = \mu$ and $b_i^1 = \mu + \delta$
$b_i = b_i^\beta \xleftarrow{R} \{b_i^0, b_i^1\}$
$\emptyset \leftarrow Enrollment((ID_j, b_j)_j)$
At the k^{th} authentication run of ID_i, where $1 < k < M$:
Obtain the data of ID_i, $\mathsf{PAR}_i \oplus b'_{ik} = c_i \oplus b_i^\beta \oplus b'_{ik} = c'_{ik}$
If $\mathsf{Decode}(c'_{ik}) = c_i$, store $e_{ik} = c'_{ik} \oplus c_i$.
Compute $a = Mean(HW(e_{ik}))$
For $1 \leq r \leq N$, compute $b = Mean(HW(b_{ir}^S \oplus b_i^0))$ and $c = Mean(HW(b_{ir}^S \oplus b_i^1))$
If $a \approx b$ return $\beta = 0$, else if $a \approx c$ return $\beta = 1$

Attack $Atk3^{**}_{\mathcal{A}}$

$(i, ID_i, b_i^0, b_i^1, (ID_j, b_j)_{\{j \neq i\}}) \leftarrow \mathcal{A}(1^l)$ s.t. $b_i^0 = \mu$ and $b_i^1 = \mu + \delta$
$b_i^\beta \xleftarrow{R} \{b_i^0, b_i^1\}$
$b_i = b_i^\beta$
$\emptyset \leftarrow Enrollment((ID_j, b_j)_j)$
Compute $b_i^2 \approx \mu + \delta/2$
At the k^{th} authentication run of ID_i, where $1 < k < M$:
Obtain the data of ID_i, $\mathsf{PAR}_i \oplus b'_{ik} = c_i \oplus b_i^\beta \oplus b'_{ik} = c'_{ik}$
If $\mathsf{Decode}(c'_{ik}) = c_i$, store $e_{ik} = c'_{ik} \oplus c_i$.
Compute $a = Mean(HW(e_{ik}))$, $b = (HW(b_i^2 \oplus b_i^0))$
If $a < b$ return $\beta = 0$, else return $\beta = 1$

errors. Since the hamming weight HW of the error $e_{ik} = b_i^\beta \oplus b'_{ik}$ when $b_i^\beta = b_i^0$ will be significantly less than the hamming weight of the error when $b_i^\beta = b_i^1$, we can make various statistical analysis by comparing the errors obtained from the authentication runs of U_i to the simulated errors based on the distribution of the U_i's biometrics and determine the value of β.

An alternative way to analyze the error and determine the value of β could be described by the following algorithm $Atk3^{**}_{\mathcal{A}}$. Similar to the attack $Atk3^{*}_{\mathcal{A}}$, in this attack we expect that the majority of the fresh templates presented to the sensor to be concentrated around the mean template b_i^0 of user U_i. Thus, computing an intermediate value b_i^2 can help us to determine the value of β. The exact value of b_i^2 could be set based on the distribution of the biometrics and other experiments.

Thus, the condition on the two templates generated by $\mathcal{A}$ must be specified in a concrete way to avoid such statistical attacks. However, with this current definition of identity privacy, this is not possible since the generation of the two templates is controlled by the

adversary. Hence, one should modify the identity privacy notion to avoid statistical attacks. One possible solution is adapting a weaker security notion of public key encryption to our setting. This weaker notion is called as Weak-Indistinguishability where the adversary cannot select challenge plaintexts (m_0, m_1), instead the challenger computes (m_0, m_1) and returns them to the adversary [Yang et al., 2010]. The same idea could be applied to identity privacy notion, where the two possible templates for U_i are computed by the challenger using the biometric template space BtSp associated to the user U_i. Then, one of the two templates presented by the challenger to the adversary is registered to the database. If the two templates $\{b_i^0, b_i^1\}$ are chosen close to each other, then we may refer to the notion of *Indistinguishability of Errors*, which prevents an insider adversary to obtain some information about the reference template of U_i based on the errors he collects. Weak-Identity Privacy is defined as follows:

Experiment $Exp_{\mathcal{A}}^{W-IP}(l)$
$(i, ID_i, (ID_j, b_j)_{\{j \neq i\}}) \leftarrow \mathcal{A}(1^l)$
$\{b_i^0, b_i^1\} \leftarrow \mathsf{BtSp}(U_i)$
$b_i^\beta \xleftarrow{R} \{b_i^0, b_i^1\}$
$b_i = b_i^\beta$
$\emptyset \leftarrow Enrollment((ID_j, b_j)_j)$
$\beta' \leftarrow \mathcal{A}(Challenger; Verification)$
if $\beta' = \beta$ return 1 else return 0

A biometric authentication scheme satisfies Weak-Identity Privacy if equation (3.1) is satisfied for all PPT A. Under this weaker notion, [Bringer and Chabanne, 2008] is secure against statistical attacks. The security analysis based on this weaker notion is identical to the analysis presented in [Bringer and Chabanne, 2008].

3.10 Preventing the Attacks

As we show in the previous section, for each different scheme, we have a different attack based on the properties/architecture of the system. For statistical attacks against schemes with encrypted sketches, we suggest to evaluate the security of the scheme based on our new notion called Weak-Identity privacy. Other sketch-based schemes used for equality testing can be made resistant against our attacks through the following solutions. The first solution is to store the sketch PAR secretly for the schemes of [Bringer et al., 2007c, Tang et al., 2008], for instance in the tamper-proof smartcard of the user. This will result in a multi-factor authentication scheme, thus, the system is not anymore a pure biometric based authentication scheme. Still, this solution does not cover a brute-force attack if these systems are employed for biometrics that can be represented as a set of features with a small feature space. Since encryption of each

feature is performed individually, an insider adversary can try different feature sets to obtain some information on the stored template of the user from the authentication result. For a large feature space, he can mount an attack similar to the statistical attack of the previous section. Specifically, if the biometrics is represented as an ordered set of features as in face biometrics, the adversary can generate the two templates in such a way that the first template includes some particularly chosen features, whereas the second template does not. By observing the matching/non-matching of these particular features, the malicious server can distinguish which template is registered by the challenger. It is cancelable biometrics that can prevent this attack, if the stored template is somehow distorted, where the distortion parameters are unknown to the insider adversary. Specifically, if we define identity privacy in a different setting, then biometric remote authentication schemes assuming biometrics as public data can achieve Identity privacy if they are combined with cancelable biometrics. The cancelable biometrics system we use requires a high entropy randomness that is stored in the user's smart card to be used later for authentication in the transformed space. This way, biometric data stored at the server is protected through this transformation and biometrics can be updated by changing the transformation function or the randomness. This system also prevents the user's traceability across different biometric databases, even if the (distorted) biometric templates are stored in clear. Example systems employing a high entropy randomness stored in a smart card for cancelable biometrics are given in [Hirata and Takahashi, 2009, Cambier et al., 2002, Sakashita et al., 2009].

Our proposed design is a multi-factor solution that requires each user to possess a smartcard to store some high entropy randomness that will be hashed with the biometrics before the encryption (and storage in the $\mathcal{DB}$). So the same randomness is used during verification by hashing it with the fresh biometrics and after that, the encryption of the result is transmitted to the server side for matching. If a secure sketch is applied, then first biometrics are corrected with the help of PAR, then the randomness is hashed with the corrected biometrics and encryption is performed afterwards. Also, our proposal allows for the integration of a secure sketch without endangering the security of the scheme, since the value PAR is only stored in the tamper-proof smart card of the user. This way, the secrecy of the relationship between the identity and the stored (distorted) biometrics of the user is maintained based on the privacy of the randomness used in the distortion of the biometrics, which is stored in the tamper-proof smartcard of the user. This solution guarantees the two security notions even if we employ a secure sketch and biometrics with small feature space. Finally, we use a cryptographic hash function for the computation of the distorted biometrics, thus, statistical attacks are not possible as even one bit of change of the input of the hash function leads to a complete different hash value.

3.10.1 A New Protocol for Cancelable Biometric Setting

In this section, we describe an example scheme that achieves weak-identity privacy for biometrics represented as an ordered set of features and (standard) identity privacy for biometrics represented as a binary string. The new scheme is defined in cancelable biometrics setting, where we assume biometrics as public data but the randomness used in the distortion of the biometric features is kept as secret. We assume biometrics as an ordered set of features such as face, iris, voice, handwritten signatures [Li et al., 2006], however, the system also works for biometrics defined as a binary string such as an 2048-bit Iris code. The matching of the fresh biometrics and the stored template is performed as in [Sarier, 2010b] with the help of bilinear pairings, where the authentication server $\mathcal{AS}$ does not need a secret key for its operations. This is an important difference to the existing schemes [Bringer et al., 2007c, Tang et al., 2008, Bringer and Chabanne, 2008], which store the biometrics as encrypted with the public key of the $\mathcal{AS}$. Thus, if the secret key of the $\mathcal{AS}$ is leaked, then each user in the system has to re-register in the best case scenario, i.e. before the compromise of the $\mathcal{DB}$, whereas the compromise of the $\mathcal{AS}$ does not affect the security of our system as $\mathcal{AS}$ does not need its secret key for its computations due to the use of bilinear pairings, hence, does not store any secret key. Finally, we assume the general representation of biometrics, where a biometric template B_e consists of k features, i.e. $B_e = \{w_i\}_{1 \leq i \leq k}$. A possible attack for this type of biometrics occurs when the feature space is small. A malicious $\mathcal{AS}$ may compare the encryption of different features to the authentication data and using pairings, he decides whether he correctly guessed the feature. Since we concatenate a different random string to each feature, based on the secrecy of these distortion values applied to each feature, the adversary cannot launch this brute-force attack. In our scheme, we use the same architecture of [Tang et al., 2008] as summarized in section 3.6, which does not require a detached verification unit $\mathcal{VU}$ and the sensor does not communicate with the biometric database as in many real-life applications.

Enrollment Phase

- $\mathcal{S}$ generates his key pair $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ and publishes them. In addition, $\mathcal{AS}$ is given an elliptic curve ElGamal public key $pk_{\mathcal{AS}} = g^y$ without the associated secret key, for instance, a trusted third party can generate this public key. Finally, a cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$ and a bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ is required.

- The user U generates his personalized username ID and registers it at the $\mathcal{AS}$, computes his distorted biometrics by picking at random $r_i \in \mathbb{Z}_q$ for $1 \leq i \leq k$ to compute $H(w_i, r_i)$ and registers his distorted biometric features as $R_i =$

$(R_i^1, R_i^2) = (g^{r_i}, g^{yr_i}H(w_i, r_i))$ for $1 \leq i \leq k$ at the $\mathcal{DB}$. The distortion numbers $\{r_1, ..., r_k\}$ are stored at the tamper-proof smartcard of U.

Remark 3.2. *To further increase the accuracy, a secure sketch for ordered biometrics can be used, whose public parameter PAR is only stored in the tamperproof smartcard of the user together with the distortion numbers, thus PAR is not publicly available as in the schemes of [Bringer et al., 2007c, Tang et al., 2008, Bringer and Chabanne, 2008]. This is required to guarantee the identity privacy notion if a secure sketch is employed.*

Verification Phase

- $\mathcal{S}$ sends the user U's identity ID and the encrypted fresh biometrics for $1 \leq i \leq k$, $X_i = (X_i^1, X_i^2) = \mathsf{Enc}(H(w_i', r_i), pk_{\mathcal{AS}}) = (g^{x_i}, g^{yx_i}H(w_i', r_i))$ to the $\mathcal{AS}$ using ElGamal encryption and the distortion values r_i's stored in the smartcard. $\mathcal{S}$ sends his signature σ on $X = \{X_i : 1 \leq i \leq k\}$ to $\mathcal{AS}$.

- $\mathcal{AS}$ verifies the signature of $\mathcal{S}$ and communicates with the $\mathcal{DB}$.

- $\mathcal{DB}$ computes for each entry $1 \leq j \leq N$ the rerandomization of R_{ji}, where R_{ji} is the encryption of the i^{th} feature of the j^{th} user's distorted biometrics. For instance, the rerandomization for the user U's biometric template is computed as $C_i = (C_i^1, C_i^2) = (g^{\beta_i}R_i^1, g^{y\beta_i}R_i^2) = (g^{\beta_i + r_i}, g^{y\beta_i + yr_i}H(w_i, r_i))$ for $1 \leq i \leq k$.

- $\mathcal{AS}$ first retrieves the index for ID and runs an efficient PIR protocol to obtain the user U's rerandomized biometrics denoted as C_i for each feature of U. Next, $\mathcal{AS}$ selects a random $s_i \in \mathbb{Z}_q$ and computes for each biometric feature of U, $Z_i = (X_i \oslash C_i)^{s_i}$, where, for any integer x and two ElGamal ciphertexts (c_1, c_2) and (c_3, c_4), the operator $\oslash$ is defined as follows: $((c_1, c_2) \oslash (c_3, c_4))^x = ((\frac{c_1}{c_3})^x, (\frac{c_2}{c_4})^x)$. Thus, for the matching features, we obtain

  $$Z_i = (Z_i^1, Z_i^2) = ((g^{x_i} \cdot (g^{\beta_i + r_i})^{-1})^{s_i}, (g^{yx_i} \cdot (g^{y\beta_i + yr_i})^{-1})^{s_i}).$$

  Finally, $\mathcal{AS}$ finds the total number matched features using bilinear pairings. Here, $\mathcal{AS}$ obtains $\hat{e}(pk_{\mathcal{AS}}, Z_i^1) = \hat{e}(g, Z_i^2)$ for the matching features by computing in total $2k$ bilinear pairings. If the number of Z_i's satisfying this equation is above the threshold, $\mathcal{AS}$ authenticates U, else rejects.

Theorem 3.3. *The proposed scheme achieves identity privacy against the $\mathcal{AS}$, based on the Gap DH problem and the tamper-proofness of the user smartcard.*

Proof. If the proposed scheme does not achieve identity privacy against malicious $\mathcal{AS}$ or an attacker who has compromised it, we construct an algorithm A', which receives an Gap DH challenge from its challenger and runs A as a subroutine to solve the Gap

DH problem. Here, the attacker A with advantage ϵ denotes $\mathcal{AS}$ or the attacker who compromised it and the hash function H is controlled by A'.

A' obtains from his challenger a Gap DH challenge $(g, g^x, g^y))$ and sets $pk_{\mathcal{AS}} = g^y$. Next, A' obtains the two possible biometric templates B_e^0, B_e^1 generated by A for the user with ID_e. A' chooses random elements $\alpha_i \in \mathbb{Z}_q$ and $\xi_i \in \mathbb{G}$ and computes $c_i^* = (c_i^1, c_i^2) = (g^{x\alpha_i}, \xi_i)$ for $1 \leq i \leq k$. Finally, A' registers c_i^*'s to the $\mathcal{DB}$ as the distorted biometric feature set of the user with ID_e. Intuitively, ξ_i denotes $g^{xy\alpha_i}$ multiplied by the hash of the random distortion parameter (i.e. $r_i = x\alpha_i$) and the biometric feature w_i, where the distortion parameter is not known by A and w_i's are chosen randomly from the two biometric templates output by A. Thus, the challenge ciphertext is random and independent from the biometric templates due to the random distortion parameters.

A' faithfully answers the verification queries by A as follows.

- Simulation of the sensor: For any $1 \leq l \leq M$, A' simulates the message (i.e. the encrypted fresh biometrics) sent from the sensor as follows:

  If $l = e$, A' randomly selects $d_i \xleftarrow{R} \mathbb{Z}_q$ and generates for $1 \leq i \leq k$, (Z_i^1, Z_i^2) and (ID_e, σ_e), where $Z_i^1 = g^{d_i} c_i^1 = g^{d_i + x\alpha_i}$ and $Z_i^2 = g^{yd_i} c_i^2 = g^{yd_i} \xi_i$, and σ_e is the signature of the sensor on these data. Here, we basically compute the rerandomization of the registered biometrics c_i^*.

  If $l \neq e$, A' generates for $1 \leq i \leq k$, (Z_i^1, Z_i^2) and (ID_l, σ_l) by directly following the protocol specification.

- Simulation of the $\mathcal{DB}$: A' simulates the $\mathcal{DB}$ by following the protocol specification.

- Simulation of the H-queries: Upon receiving a query (w_j, s_j),

  1. If there exists (w_j, s_j, h_j) in HList, return h_j.
  2. Else if $w_j \in B_e^0$ or $w_j \in B_e^1$, A' checks whether $\hat{e}((g^y)^{\frac{s_j - d_i}{\alpha_i}}, g) = \hat{e}(g^x, g^y)$ for each $1 \leq i \leq k$. If one of the equations is satisfied, A' returns $(g^y)^{\frac{s_j - d_i}{\alpha_i}}$ to his challenger and finishes the game. Here, d_i is the randomness used during the simulation of the sensor in the randomization of the challenge ciphertext associated to the feature $w_i = w_j$. Additionally, $\hat{e}((g^y)^{s_j/\alpha_i}, g) = \hat{e}(g^x, g^y)$ could also be checked, again if the equation is satisfied, A' returns $(g^y)^{s_j/\alpha_i}$ to his challenger and finishes the game.
  3. Otherwise, A' picks a random $h_j \in \mathbb{G}$, adds the tuple (w_j, s_j, h_j) in HList, return h_j to A.

- If A outputs β', then A' terminates by picking a random entry s_j from the HList and returns $(g^y)^{\frac{s_j - d_i}{\alpha_i}}$ to his challenger.

The intuition of this security game is that an attacker A has to query the random oracle H in order to be able to distinguish the template stored in the database. Since A only obtains the randomization of the challenge template c_i^* for $1 \leq i \leq k$, from his queries to the random oracle, A' tries to solve the Gap DH problem. Let $\mathbb{H}$ be the event that algorithm A issues the queries $H(w_j, s_j)$ for $w_j \in B_e$ at some point during the simulation. By the definition of A, $|Pr[\beta' = \beta] - \frac{1}{2}| > \epsilon$. Pr[$\mathbb{H}$] in the simulation above is equal to Pr[$\mathbb{H}$] in the real attack. Also, in the real attack we have Pr[$\mathbb{H}$]$\geq 2\epsilon$ due to the following facts.

If the HList does not contain the values for w_j, s_j's, then we have $Pr[\beta' = \beta | \neg\mathbb{H}] = \frac{1}{2}$.

Combining all the results and defining the event E as $E = Pr[\beta = \beta']$, we obtain the following as in [Boneh and Franklin, 2003]

$$
\begin{aligned}
E &= Pr[\beta = \beta' | \mathbb{H}] Pr[\mathbb{H}] + Pr[\beta = \beta' | \neg\mathbb{H}] Pr[\neg\mathbb{H}] \\
&\iff Pr[\beta = \beta'] \geq \tfrac{1}{2}(1 - Pr[\mathbb{H}]) \\
&\iff Pr[\beta = \beta'] \leq \tfrac{1}{2}(1 + Pr[\mathbb{H}]).
\end{aligned}
$$

Thus, $\epsilon \leq |Pr[\beta = \beta' | \mathbb{H}] - \frac{1}{2}| \leq \frac{1}{2} Pr[\mathbb{H}]$, namely $Pr[\mathbb{H}] \geq 2\epsilon$

Following from the above claims, we have that A' produces the correct answer with probability at least $2\epsilon/q_2$, where q_2 denotes the total number of queries to the H-oracle.

□

We note that our scheme achieves weak-identity privacy for biometrics represented as an ordered set of features and (standard) identity privacy for biometrics represented as a binary string. This difference results from the fact that the adversary may arrange the two possible biometric sets of the challenge user in a special way so that he can distinguish the registered reference template from the matching results of the particular features he arranged. The matching result analysis does not work for biometrics as a binary string since there is a single authentication result: accept or reject.

Theorem 3.4. *The proposed scheme achieves transaction anonymity against a malicious $\mathcal{DB}$, based on the security (user privacy) of the PIR protocol.*

The proof is identical to the proof presented in section 3.8.3. The only difference is that the entity $\mathcal{VU}$ is replaced by $\mathcal{AS}$, as in this protocol we do not employ a detached verification unit.

3.10.2 Identity Privacy for Cancelable Biometrics

Our first solution presented in the previous section guarantees identity privacy due to the one-wayness property of the cancelable biometrics and the secrecy of the helper data PAR. Thus, in order to distinguish one of the biometric templates, the adversary playing the identity privacy game as described in [Bringer et al., 2007b] has to break the one-wayness of the cancelable biometrics, where one-wayness is a weaker security notion than indistinguishability. To overcome this limitation, we define the following notion, where breaking this new notion implies breaking the underlying encryption scheme in the sense of indistinguishability, which is a stronger security notion.

Given an adversary $\mathcal{A}$ running against the biometric authentication scheme and a challenger $\mathcal{C}$ that simulates the registration phase of the scheme, we consider the following game between $\mathcal{A}$ and $\mathcal{C}$.

Experiment $Exp_A(l)$
$((ID_j, b_j, r_j, PAR_j)_{\{j \neq e\}}) \leftarrow \mathcal{A}(1^l)$
$(e \neq j, ID_e, b_e, r_e^0, r_e^1, PAR_e) \leftarrow \mathcal{A}(1^l)$
$r_e^\beta \overset{R}{\leftarrow} \{r_e^0, r_e^1\}$
$r_e \leftarrow r_e^\beta$
$\emptyset \leftarrow Enrollment^*(Distortion(b_j, r_j)_j)$
$\beta' \leftarrow \mathcal{A}(Challenger; Verification)$
if $\beta' = \beta$ return 1 else return 0

A biometric authentication scheme satisfies the notion of "Identity Privacy for Cancelable Biometrics" if equation (3.1) is satisfied for all PPT A. Here, the adversary $\mathcal{A}$ generates the authentication data for $N-1$ users together with the reference biometrics b_j, the secure sketch PAR, and two different distortion parameters for an additional user U_e. $\mathcal{C}$ picks at random a distorion parameter $r_e = r_e^\beta$. Next, the chosen distortion parameter is applied to the reference biometric template and the enrollment phase is completed. The difference of our notion to the Bringer et al.'s identity privacy notion [Bringer et al., 2007b, Bringer and Chabanne, 2008, Tang et al., 2008] is that the $\mathcal{C}$ does not need to choose randomly one of the two similar biometrics generated by the adversary $\mathcal{A}$, since with the public value PAR, the error-corrected template can be easily computed and a unique reference template b_e is obtained. Thus, $\mathcal{C}$ only needs to apply the random distortion r_j^β to this reference template b_j and then register the encryption of this distorted biometrics in the $Enrollment^*$ phase. This application could be performed as in the protocol described in section 3.10.1, by simply picking at random $r_e^1, r_e^2 \in \mathbb{Z}_q$ as input to the hash function. After running the verification protocol, $\mathcal{A}$ outputs a guess for the distortion parameter that $\mathcal{C}$ has chosen. One can easily show that the schemes of [Bringer et al., 2007c, Tang et al., 2008] achieve identity privacy for cancelable biometrics against a malicious $\mathcal{AS}$, based on the semantic security of the

ElGamal encryption although the sketch PAR is public. The proof is identical to the proofs presented in [Bringer et al., 2007c, Tang et al., 2008] for biometrics represented as a fixed length binary string. If biometrics is represented as a set of features, a set of randomly picked distortion parameters is applied instead of a single parameter.

3.11 Comparison

In this section, we present an overview of the protocols designed according to the model of Bringer et al. We compare the schemes based on the security notions they achieve and whether the schemes are still secure even if the secret key of the verification unit in [Bringer et al., 2007b, Barbosa et al., 2008] or the secret key of the authentication server in [Tang et al., 2008, Bringer et al., 2007c] is leaked, where this key is required for the matching stage and the final decision. In our scheme the authentication server does not know his secret key and uses bilinear pairings for the matching in the encrypted domain, thus, our scheme is resistant against this attack. $^+$ denotes the first biometric scheme.

Table 3.4: Properties of various DBRA schemes

Scheme	Identity Privacy	Transaction Anonymity	Security against Key Compromise	Current Attacks
System 1 [Bringer et al., 2007b]	No	No	No	D. Naccache's attack at BIOSIG'09
System 2 [Barbosa et al., 2008]	Yes	Yes	No	Attack of [Simoens et al., 2011]
System$^+$3 [Bringer et al., 2007c]	No	Yes	No	$Atk1_{\mathcal{A}}, Atk2_{\mathcal{A}}$
System 4 Bringer et al. [2008]	No	Yes	No	$Atk3^*_{\mathcal{A}}, Atk3^{**}_{\mathcal{A}}$
System 5 [Tang et al., 2008]	No	Yes	No	$Atk1_{\mathcal{A}}, Atk2_{\mathcal{A}}$
New System [Sarier, 2011a]	Yes	Yes	Yes	-

3.12 Conclusion

In this chapter, we analyze the security model for distributed biometric remote authentication (DBRA). We review different schemes that guarantee the security notions with a security reduction and present a new protocol with a different biometric template storage. Our first improvement is in terms of efficiency, i.e. reducing the database storage cost significantly. Secondly, the compromise of the biometric database does not allow an adversary to obtain the biometric templates of each user, as in our system, the database stores only a random pool of biometric features. The database does not know which feature belongs to which user, since the database locations of a user's feature set $Index$ is stored only in the smartcard of that user.

Next, we focus on the notion of identity privacy and present three new attacks that reveal the reference biometric template of the user to the malicious server. The first type of attack applies to any system that assumes biometrics and the sketch as public data since a secure sketch can only guarantee a weak level of security. However, if the sketch is stored secretly, i.e. in a tamper-proof smartcard, then the systems are secure for biometrics represented as a fixed length binary string. The second type of attack is a statistical attack, which works even if the sketch is stored as encrypted at the database. Consequently, the security of pure biometric remote authentication schemes is questionable if they are evaluated in the framework of a realistic and strong security model. Thus, we suggest that DBRA systems should be implemented as a two-factor authentication system, which employs a tamper-proof smartcard for storing additional data as the second factor. Besides, the current systems are not suitable for other biometric traits that are represented as an ordered/unordered feature set, whereas our new protocol for cancelable biometric setting is both secure against the three types of attacks and resistant for attacks as a result of different representations of biometrics. Finally, if identity privacy is redefined in cancelable biometric setting, the schemes vulnerable to the first type of attack are secure for public sketches.

Chapter 4

Practical Multi-factor Biometric Remote Authentication

In this chapter, we evaluate the security properties of Multi-Factor Biometric Authentication (MFBA), where biometrics is assumed as a set of features that can be either ordered or unordered depending on the biometric modality. We propose efficient schemes for MFBA that are suitable for a different template extraction method used in bipartite biotokens. In particular, a bipartite biotoken describes the biometric template of a user as two sets of data. The first set consists of the stable parts of the features and the second set is the non-stable parts. By separating the stable and non-stable parts of each feature, a cryptographic protocol is applied to encrypt the stable parts and the matching score is computed in the encryption domain at the remote server, whereas another (optional) matching can be performed at the client-side by checking whether each non-stable part is within its predefined range. We formally describe the security model for MFBA, where the server-side computations are performed in the encrypted domain but without requiring a decryption key for the authentication decision of the server. Thus, leakage of the secret key of any system component does not affect the security of the scheme as opposed to the current biometric systems involving cryptographic techniques. Finally, we reduce the security of our design to the unforgeability of the Schnorr Signature Scheme according to our new security model that captures simultaneous attacks against a MFBA. In this context, we define the notion of user privacy, where the goal of the adversary is to impersonate a client to the server. The adversary has access to different oracles that model the adversaries capabilities such as eavesdropping on the communication channel -even in the case of a compromised session key that is used to build a secure communication link before the start of the protocol execution-, and compromise of *either* the sensor (namely biometrics of the user) *or* the smart card of the user that stores the secret parameters used in the stable/non-stable part separation.

We note that, our design is the first biometric system based on a CCA secure encryption system. However, when implemented on a pairing friendly elliptic curve, the server can make the authentication decision without any decryption operation through the use of bilinear pairings.

We also show that there is a tradeoff between the security the scheme achieves (OW-CCA instead of IND-CCA) and the requirement for making the authentication decision without using any secret key. For unordered biometrics such as fingerprint minutia, we employ RSA encryption combined with a zero knowledge proof of knowledge of plaintext for RSA. The results described in this chapter were published in [Sarier, 2010a].

4.1 Introduction

Biometrics provides a stronger authentication mechanism compared to password-based systems, as passwords can be easily lost, forgotten or compromised using various attacks. In addition to identification/authentication purposes, biometrics can also be used for key release, key binding or key generation, where this key could be the input to a symmetric cryptosystem such as AES. For different applications of biometrics, the assumptions on the biometric data varies. Specifically, current biometric identification/authentication systems that are provably secure in a cryptographic sense, assume that biometrics is public data whereas the link between the identity of the user and his biometrics should be kept secret. On the other hand, biometric cryptosystems that lock/generate a secret key using biometric features assume that biometric template of a user is secret data. Despite different views about the secrecy of biometrics, the common principle is that biometrics is sensitive data and the privacy of biometrics that is either stored on a central database or on a tamper-proof smart card should be protected using cryptographic techniques.

To protect the privacy of biometrics in the above listed settings, we combine biometrics and cryptography. There exists different types of biometric cryptosystems such as fuzzy extractors, fuzzy vault and recently introduced bipartite biotokens, which could be used for biometric key generation, key binding and key release, respectively. Also, cancelable biometrics perform a distortion of the biometric image or features before matching. The variability in the distortion parameters provides the cancelable nature of the scheme. Besides, fuzzy vault is a key binding system that hides an encoded secret among some chaff points, where the secret key is encoded as the coefficients of a polynomial that is evaluated at the biometric feature locations such as fingerprint minutia locations. Recently introduced bipartite biotoken is an extension of revocable biotoken and fuzzy vault, where the stable parts of biometrics is stored as encrypted and the evaluation points of the embedded polynomial are not stored as opposed to

the fuzzy vault.

However, the implementation of these systems come along with various attacks that question the security of them. For fuzzy vault, the broad categorization of these attacks consists of known plaintext and ciphertext-only attacks, where the former assumes that an attacker can gain access to the secret key hidden in the fuzzy vault, which leads to the biometric template by verifying the secret polynomial on the points in the vault. The second group of attacks, namely ciphertext-only attacks do not require any insider knowledge and brute force attack or different instances of the vault encoded with the biometrics of the same user is enough to obtain the biometric template of that user. For fuzzy extractors, a similar attack based on the reusability of the same (or a noisy variant) of the biometrics for multiple extractions of independent public strings is described due to improper fuzzy sketch constructions or wrongly chosen error correcting codes. From these public strings, an attacker can exactly regenerate the corresponding secret keys that are output by the fuzzy extractor.

Apart from the application specific attacks, remote biometric authentication systems are vulnerable to four classes of attacks: Attacks to the sensor via spoofing or compromising the sensor, attacks to the database (tampering with the templates, substitution attacks), attacks to the matcher and intercepting/eavesdropping to the communication channel. The first and second classes of attacks can be avoided by additional security factors (password, smart cards) and by storing the (cancelable) templates as encrypted and signed. Also, if a decryption is performed during the matching stage a Trojan horse type attack can lead to the disclosure of the raw biometric. Thus, the comparison should be made in the encrypted domain without any decryption operation. Finally, the communication between the entities should be encrypted with session keys to prevent the last classes of attacks. Clearly, all of these countermeasures assumes the secrecy of the system's private keys and session keys. Since, for biometric authentication systems, although the encoding of the raw biometrics can be encrypted, if the encryption is ever broken due to a leaked secret key or due to the leakage of the session key of a previous authentication, the reference template is not anymore a secret, thus, biometrics cannot provide a security factor on that application as it cannot be revoked.

New approaches for remote biometric based verification guarantee provable security against these attacks, which consists of a hybrid protocol that distributes the server side functionality in order to detach the biometric data storage from the service provider SP. This is required as the biometric templates of each user is stored in the database as encrypted using the public key of the service provider SP that only stores the identities of each user (Name, personalized usernames). The main advantage of this approach is that the user does not need to store any secret data and/or his biometric template in a smart card as in match on card systems. However, the collusion of the server end components results in the violation of the user privacy. Other attacks against this type

of systems are presented recently in [Sarier, 2011a, Simoens et al., 2011].

Besides, multi-factor biometric authentication (MFBA) protocols that store the user's biometric template in a smart card combine a local authentication on the card by a remote authentication at the server. For instance, the local matching of the fresh biometrics to the stored template on card generates a biometric secret key that is used to sign a challenge sent by the remote server for the remote authentication [Itakura and Tsujii, 2005, Bringer et al., 2008]. Other multi-factor solutions combine this biometric key with encryption or zero knowledge proofs for a remote verification [Bhargav-Spantzel et al., 2006]. Cancelable biometrics is another approach that stores the masked biometrics at the server, where the masking is performed using a one-way transformation or a high entropy randomness that is stored in the user's smart card to be used later for authentication in the transformed space. This way, biometric data stored at the server is protected through this transformation and biometrics can be updated by changing the transformation function or the randomness [Ratha et al., 2001, Sakashita et al., 2009].

4.1.1 Related Work

Juels and Wattenberg [Juels and Wattenberg, 1999] introduced the fuzzy commitment scheme, which assumes biometrics as a binary string (for instance a 2048 bit Iris code) and replaces biometric matching algorithms by error-correction techniques. Further systems that follow the same approach are presented in [Boyen, 2004, Dodis et al., 2004] which include secure sketches and fuzzy extractors that are used for biometric authentication. Also, Juels and Sudan have developed the *fuzzy vault* [Juels and Sudan, 2006], which assumes biometrics as an unordered set of features and is designed for the set difference metric in order to hide a secret key (i.e. an AES key) using biometrics. Implementations of fuzzy vault for fingerprints are given in [Clancy et al., 2003, Nandakumar et al., 2007a].

Attacks against biometric cryptosystems are presented in [Boyen, 2004, Mihăilescu et al., 2009] and an overview of the attacks that are specific for traditional biometric authentication systems are given in [Li and Jain, 2009]. Also, Scheirer and Boult present three different attacks (including the correlation attacks) against the fuzzy vault and biometric encryption [Scheirer and Boult, 2007] and in view of that, they proposed revocable biotokens [Boult et al., 2007] and its implementation in [Scheirer and Boult, 2009], where their system could be considered as an example of cancelable biometrics, a concept that was introduced in [Ratha et al., 2001].

Besides, Bringer et al. [Bringer and Chabanne, 2008, Tang et al., 2008] defined the security notions for biometric remote authentication and described a new architecture based on homomorphic encryption, Private Information Retrieval (PIR), detached bio-

metric storage and secure sketches for error correction of the biometric string. Also, the authors of [Barbosa et al., 2008, Sarier, 2009a, 2010b] describe a secure remote authentication scheme in the same framework assuming biometrics as a feature set. A survey of these systems could be found in [Sarier, 2009b]. The common property of these systems is the high computational costs due to the PIR systems and strong assumptions on the server-end components resulting in small-scale biometric systems with highest security. Although a simple client server biometric authentication system is proposed in [Upmanyu et al., 2009], the decision can be computed after a decryption operation similar to the schemes of [Tang et al., 2008, Bringer and Chabanne, 2008, Barbosa et al., 2008], thus the leakage of the system's secret keys endangers the security of the system. Besides, MFBA systems are proposed in [Bhargav-Spantzel et al., 2006, Apampa et al., 2008, Sarier, 2009a, 2010b, Bringer et al., 2008, Itakura and Tsujii, 2005], where the last two schemes performs the matching on card for a local authentication followed by a remote authentication. Also, cancelable biometrics is combined with a smart card for storing only the helper information [Hirata and Takahashi, 2009, Sakashita et al., 2009] resulting in a MFBA protocol.

4.1.2 Motivation and Contributions

When we analyse different biometric authentication systems, we see that the most dangerous event is the leakage of the session keys encrypting the communication channel between the client and the server. Similarly, the storage of the system's secret keys causes another bottleneck as they are required for decryption of the stored templates at the matching stage or for decrypting the final decision when homomorphic encryption is used. The natural question that arises is whether it is possible to have user's privacy even if these keys are compromised. In other words, is there a way to store the biometrics as encrypted and perform the matching in encrypted domain without any decryption operation. Partially, current systems achieve this using homomorphic encryption schemes, however, for the final decision, the system's secret key is still needed. Thus, we need a different encryption method that also determines the final decision without using any secret key. Besides, an attacker that compromised the session key between the server and client could eavesdrop to the communication channel and later perform a replay attack by sending the same ciphertext (i.e. encrypted biometrics) without even knowing the true biometrics of the user. How do we prevent replay attacks? A solution could be attaching a proof of knowledge of the plaintext to the ciphertext, which proves that the user knows the biometrics without revealing it to the server. However, this zero knowledge proof (ZKP) must include a time stamp and additional data such as user specific information to become non-malleable, namely we require a non-malleable ZKP to avoid a replay attack that sends the ciphertext and the corresponding ZKP obtained from a previous session or to prevent a more powerful

attacker that transforms the (malleable) ZKP to a valid ZKP for the current session. Besides, the combination of a (weakly secure) encryption scheme and a non-malleable ZKP prevents Chosen Ciphertext Attacks as shown in [Sahai, 1999, Katz, 2002, Jao, 2009]. Another point one should consider is the compromise of the sensor. In this case, can we have still privacy?

In this chapter, we try to answer these questions and design a new biometric verification protocol that does not require additional detached components at the server end and strong assumptions on the system. Instead, we propose a simple client server architecture for a MFBA by combining bipartite biotokens and cryptographic techniques, where the complete biometric template of the user is not stored in any system component. We formally design the security model for MFBA based on the privacy/security issues summarized as in figure 4.1, where we allow an adversary trying to impersonate a user against a honest-but-curious server to access different oracles. Basically, these oracles model the adversaries capabilities such as eavesdropping on the communication channel -even in the case of a compromised session key that is used to build a secure communication link before the start of the protocol execution- and compromise of *either* the sensor (namely biometrics of the user) *or* the smart card of the user through side channel analysis. We present a security reduction according to this strong model.

Firstly, we follow the biometric template extraction method used in bipartite biotokens [Boult et al., 2007], where the biometrics is transformed using a scaling and a translation in order to separate the stable and non stable part of each biometric feature. The encrypted stable parts and the signature of the user on this data are stored at the service provider and the non-stable parts can be stored in clear together with the separation (i.e. transformation) parameters in the tamper-proof smart card of the user if a second check is performed. This operation results in a cancelable biometric template as changing the parameters and/or the public key for encrypting the stable parts will lead to a different template. In addition, security against the honest-but-curious server is guaranteed by storing the stable parts as encrypted with the user's public key, where the corresponding secret key is not needed after the registration to the server and thus not stored anywhere. Also, this storage mechanism at the server avoids substitution/masquerade attacks due to the secret transformation parameters and encrypted storage, and prevents tampering with the templates due to the signature of the user on the encrypted template.

Our system is based on the signed ElGamal encryption scheme [Tsiounis and Yung, 1998], which is IND-CCA secure and plaintext aware in Random oracle model (ROM). Due to the combination of a (weakly secure) encryption scheme and non-malleable proof of knowledge of the randomness used, the adversary proves his knowledge of the (stable) biometric features, thus a decryption oracle would be useless to the adversary. This way, security against Chosen Ciphertext Attacks (CCA) is provided without loosing

the homomorphic property of the IND-CPA secure part of the scheme that encrypts the message as opposed to the generalized signed ElGamal encryption [Schnorr and Jakobsson, 2000] or padded ElGamal [Fujisaki and Okamoto, 1999], which are also CCA secure in ROM. We note that, our design is the first biometric system based on a CCA secure homomorphic encryption system, i.e. non-malleable ElGamal encryption that combines ElGamal encryption with a non-malleable zero knowledge proof [Tsiounis and Yung, 1998]. However, when implemented on a pairing friendly elliptic curve, the server can make the authentication decision without any decryption operation through the use of bilinear pairings. In chapter 2, we presented the first biometric authentication scheme [Sarier, 2010b] that is based on this method, namely, encrypting the biometric templates using elliptic curve ElGamal and for verification, we use bilinear pairings to test whether two biometric templates are equal in the encrypted domain. This can be considered as an instance of a new concept called "PKE with equality testing" introduced in a later publication of [Yang et al., 2010], where the authors describe a slightly different scheme that is OW-CCA secure with a reduction to the CDH problem in ROM and use bilinear pairings to test for equality. In this chapter, we show that elliptic curve signed ElGamal achieves OW-CCA security in ROM if bilinear pairings are used to test for equality of biometric data in the encrypted domain. We also show that there is a tradeoff between the security the scheme achieves (OW-CCA instead of IND-CCA) and the requirement for making the authentication decision without using any secret key. Clearly, if the final decision is made by decrypting the resulting computation as in current biometric authentication systems, our construction achieves IND-CCA security. For unordered biometrics such as fingerprint minutia, we employ RSA encryption combined with a zero knowledge proof of knowledge of plaintext for RSA.

The main difference of our system to the previously defined systems is that we do not need to use any decryption key at any stage of the protocol and the authentication is performed in the encrypted domain. Currently, the systems perform authentication in the encrypted domain using the homomorphic properties of the encryption scheme but later require a decryption for the final decision as in [Hirata and Takahashi, 2009, Barbosa et al., 2008, Tang et al., 2008, Bringer and Chabanne, 2008]. However, in our design, the leakage of the secret key of any entity does not affect the security of the system. Besides, we do not have to employ a secure sketch or error correcting procedure to obtain the exact template that was stored in the biometric database of the service provider. Since no template is stored in the server end in our system, there is no need for a detached database and to employ the computationally expensive Private Information Retrieval (PIR) system to retrieve any template from the database privately. Instead, we propose a simple client server architecture for biometric verification that could be implemented also for large scale systems such as border control applications.

Moreover, the stable parts of a user are stored as encrypted using the public key of

that user, whereas currently defined provably secure systems store the templates of each user at the detached database as encrypted using the public key of the service provider. Hence, if the service provider and the database collude, identity (Name) and biometrics relation cannot be kept secret and compromise of the systems secret key requires all the users to re-register to the system with a new public key in the best-case scenario (i.e before the compromise of the biometric database). In our system, the server (or an adversary that compromised it) has to invert the encrypted stable parts and compromise the smart card of the user at the same time in order to obtain this relation.

Another difference to the previous systems is that our system can be implemented a hybrid system that combines server-side matching and client side-matching, where the matching score of the both sides cannot be obtained by a passive attacker due to the use of a range proof that does not reveal the matching score but proves that the score lies in a range based on a threshold. This way, attacks depending on the matching score (for instance hill climbing attacks, Trojan horse attacks) are avoided. Finally, revocation of the biometric templates can be easily performed by changing the transformation parameters and/or picking a different public key for the user to encrypt the stable parts.

4.2 Preliminaries and Definitions

In this section, we review the definitions of the primitives used in this chapter briefly. The reader is referred to the background chapter for the details of ElGamal encryption, Schnorr Signature, forking lemma, zero knowledge proof, plaintext awareness and the presented definitions.

Definition 4.1. *Bilinear Pairing:*

Let $\mathbb{G}$ and $\mathbb{F}$ be multiplicative groups of prime order q and let g be generator of $\mathbb{G}$. A bilinear pairing is denoted by $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ if the following two conditions hold. $1_{\mathbb{G}}, 1_{\mathbb{F}}$ denote the identity elements of $\mathbb{G}$ and $\mathbb{F}$, respectively.

1. *$\forall\ (u, v) \in \mathbb{G} \times \mathbb{G}$ and $\forall\ (a, b) \in \mathbb{Z}$ we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$*
2. *If $\hat{e}(u, v) = 1_{\mathbb{F}}\ \forall v \in \mathbb{G}$, then $u = 1_{\mathbb{G}}$, namely the pairing is non-degenerate.*

ElGamal encryption system can be implemented on elliptic curves, which allow for the construction of a special map called bilinear pairing. Using bilinear pairings, one can check the equality of two plaintexts without decrypting their corresponding ciphertexts.

4.2.1 Forking Lemma

Forking Lemma was introduced by David Pointcheval and Jacques Stern [Pointcheval and Stern, 2000], which is specified in terms of an adversary that attacks a digital signature scheme instantiated in the random oracle model. The forking lemma states that if an adversary (typically a probabilistic Turing machine), on inputs drawn from some distribution, produces an output that has some property with non-negligible probability, then with non-negligible probability, if the adversary is re-run on new inputs but with the same random tape, its second output will also have the property. The authors show that if an adversary can forge a signature with non-negligible probability, then there is a non-negligible probability that the same adversary with the same random tape can create a second forgery in an attack with a different random oracle. The forking lemma has been used to prove the security of a variety of digital signature schemes and other random-oracle based cryptographic constructions. The reader is referred to the background chapter for the details of this method.

4.2.2 Zero Knowledge Proof

A proof of knowledge is an interactive proof in which the prover P succeeds "convincing" a verifier V that it knows something. Specifically, a ZKP allows a user to have a private data, and prove its possession without releasing it. As shown by [Katz, 2002, Sahai, 1999], it is possible to construct zero knowledge proofs having the property that a proof of one statement cannot be adapted or mutated into a proof of another statement (a property known as non-malleability). Using non-malleable zero knowledge proofs, it is then possible to construct cryptosystems that achieve CCA security [Jao, 2009].

4.2.3 Plaintext Awareness

Plaintext-Aware encryption is means that an adversary is aware of the decryption of the messages which she encrypts in the sense that she cannot produce a ciphertext without knowing the corresponding plaintext. Plaintext awareness is defined both in ROM and standard model. In [Teranishi and Ogata, 2006], it is shown that combining a one-way secure encryption and plaintext awareness implies CCA security.

For our new MFBA scheme, we employ the non-malleable ElGamal encryption scheme of [Tsiounis and Yung, 1998], which combines ElGamal encryption with non-malleable ZKP based on the Schnorr Signature scheme. This CCA-secure encryption scheme is also called as signed ElGamal and shown to be plaintext aware in [Schnorr and Jakobsson, 2000]. This is achieved by a generic extractor that extracts the secret key r from a signed ciphertext $(u, v, c, z) = (g^r, y^r m, c, z)$ produced by the adversary. Given

(r, u, v) the plaintext m can be extracted in one generic step. Thus, signed ElGamal encryption is plaintext aware as defined in [Bellare and Rogaway, 1994].

4.3 A New Design for MFBA

In this section, we describe how to combine the ingredients defined in the previous section to obtain a provably secure and efficient MFBA protocol. We start with a simple client-server application and analyse its security against various attacks, that are summarized in section 4.4.1. For better understandability of the security concepts, we present different examples and their weaknesses step by step.

4.3.1 Choosing the cryptographic method

For simplicity, assume that biometrics b of user U is represented as an ordered set of features, where each feature b_i is stored as encrypted with U's public key $y = g^x$ using elliptic curve ElGamal as $\mathsf{Encrypt}(b_i)=(u_i, v_i) = (g^{r_i}, y^{r_i} b_i)$.

When U wants to authenticate, he sends a fresh encryption of his biometrics as $\mathsf{Encrypt}(b'_i)=(u'_i, v'_i) = (g^{r'_i}, y^{r'_i} b'_i)$. The server SP has to compute for each feature, $(a_i, b_i) = (\frac{u_i}{u'_i}, \frac{v_i}{v'_i})$ and check $\hat{e}(a_i, y) = \hat{e}(b_i, g)$. If $b'_i = b_i$, then, they cancel out in the division $(a_i, b_i) = (\frac{g^{r_i}}{g^{r'_i}}, \frac{y^{r_i}}{y^{r'_i}}) = (g^{r_i - r'_i}, y^{r_i - r'_i})$, thus, $\hat{e}(a_i, y) = \hat{e}(g^{r_i - r'_i}, g^x) = \hat{e}(g, g)^{x(r_i - r'_i)} = \hat{e}(g^{x(r_i - r'_i)}, g) = \hat{e}(b_i, g)$ due to the (1) property of the pairing. Lastly, SP decides based on the number of matching features to accept or reject U.

However, ElGamal encryption system is malleable. For example, given an encryption (u_i, v_i) of some (possibly unknown) message m, one can easily construct a valid encryption $(u_i, 2v_i)$ of the message $2m$. Or, assume that the attacker eavesdrops on the communication channel and obtains the encryption (u_i, v_i) of each feature of the user. At a later time, the attacker can use the same ciphertext (u_i, v_i) (or the re-encryption of it via $(g^k u_i, y^k v_i)$) and authenticates to the system without even knowing the biometrics. Thus, ElGamal encryption is combined with a zero knowledge proof (ZKP) of plaintext (i.e. biometrics), which proves to the server that the user knows the biometrics. In [Tsiounis and Yung, 1998], Schnorr proofs of knowledge (which is based on the Schnorr signature) is combined with ElGamal encryption to obtain a non-malleable encryption scheme that we present as an example as below.

Assume that each biometric feature b_i of a user is stored as $\mathsf{Encrypt}(b_i)=(u_i, v_i)$. When the user wants to authenticate, he sends a fresh encryption of his biometrics as $\mathsf{Encrypt}(b'_i)=(u'_i, v'_i) = (g^{r'_i}, y^{r'_i} b'_i)$ and the Schnorr ZKP (z_i, c_i), where $z_i = g^{k_i}, c_i = r'_i \cdot H(g, u'_i, v'_i, z_i, U) + k_i$. Here, U denotes a user specific data (i.e. name, identity

data) and H is a cryptographic hash function. The server first checks the ZKP as $g^{c_i} = (u'_i)^{H(g,u'_i,v'_i,z_i,U)} \cdot z_i$. If this equality holds, then server computes the bilinear pairings as before.

One should note that this non-malleable scheme is still vulnerable to replay attacks as an attacker that compromised the session key and eavesdrop on the communication channel, can obtain the encrypted biometrics and the corresponding ZKP of a user and later impersonates this user. This can only be prevented by adding a time stamp t to the ZKP in addition to the user specific data as $H(g, u'_i, v'_i, z_i, U, t)$, thus the attacker cannot use the same ciphertext for a replay attack at a later time t'.

Besides, the systems of [Bringer and Chabanne, 2008, Tang et al., 2008] require the use of secure sketches and store biometrics in the detached database as encrypted using the public key of the service provider SP. If the secret key of SP is leaked, than every user has to re-register to the system before the compromise of the database. Therefore, the new design should store each biometrics as encrypted with the user's public key.

4.3.2 Biometric Template Generation

Basically, biometrics of a user is represented as a set of features, where each feature can be mapped to a finite field element [Juels and Sudan, 2006, Nandakumar et al., 2007a] or error corrected using a secure sketch [Sutcu et al., 2006]. In our design, we represent the biometrics of each user as a set of features, where k denotes the size of this set and depending on the biometric modality chosen, the features could be ordered. Besides, we do not have any assumption on the secrecy of the biometrics, whereas the biometric template that is stored should be private but easily revocable.

In addition to the classical representation of biometrics, one can also implement the biometric template extraction method of [Boult et al., 2007], where each feature is transformed using a scaling and a translation in order to separate the stable ν and non stable r part of each feature. This way, the stable part ν can be encrypted using standard public key encryption schemes, as one bit of change in the plaintext (i.e. nonstable biometrics) will result in a completely different ciphertext when encrypted with standard cryptosystems. The encrypted stable parts are stored at the service provider and the non stable parts can be stored as a plaintext in the smart card of the user which is tamper proof. The main difference of this approach is that there is no need to employ a secure sketch or error correcting procedure to obtain the exact template that was stored in the biometric database of the service provider. Moreover, this method of template generation applies to any type of biometrics that can be processed as stable and non stable parts.[Boult et al., 2007, Scheirer and Boult, 2009] implements this approach for fingerprints.

In our system, we store the non-stable (residual) part, the transformation (i.e. scaling

and translation) parameters and the window parameter E in the user's tamper proof smart card. This is different from the systems of [Boult et al., 2007, Scheirer and Boult, 2009], which allow to send the biotoken over the network for authentication or key release purposes. Assuming the biometric produces a value v'' that is transformed via scaling s and translation t to $v' = (v''-t)s$, the resulting v' is split into the overall stable component ν, and the the residual component r. The amount of stable/unstable data is a function of the biometric modality. In [Boult et al., 2007], the separation parameter E that depends on the expected variations on v'' separates the stable (integer) part $\nu = int(v'/E)$ and non stable part $r = rmod(v', E)$ using a simple mod operation:

$$r = rmod(v', E) = v'\%2E \text{ if } (v'\%2E) < E \text{ and}$$
$$r = rmod(v', E) = 2E - (v'\%2E), \text{ otherwise}$$

Other methods (if they exist) that separate stable/non stable parts of a feature are also applicable in our setting.

4.4 Security Model

We propose a MFBA scheme that consists of three components, which communicate via an encrypted channel.

- *Sensor Client SC*: This is the entity that obtains the fresh biometrics of the user during verification. The liveliness assumption should be satisfied as it guarantees with high probability that the biometrics is coming from a live human user. The sensor client is always honest as in any biometric system and it is trusted by everyone.

- *Service Provider SP*: This entity stores the identity information (name, personalized usernames...) for each user and the encrypted stable parts of each user's biometric template. Since no complete reference biometric template of a user either as a plaintext or in encrypted form is stored at the SP, there is no need for a detached biometric database.

- *User U with a smart card*: Each user possesses a tamper proof smart card that stores the non stable parts of his biometrics (optional) and the parameters of the biometric template extraction method. We emphasize that no complete template is stored as in Match On Card (MOC) systems.

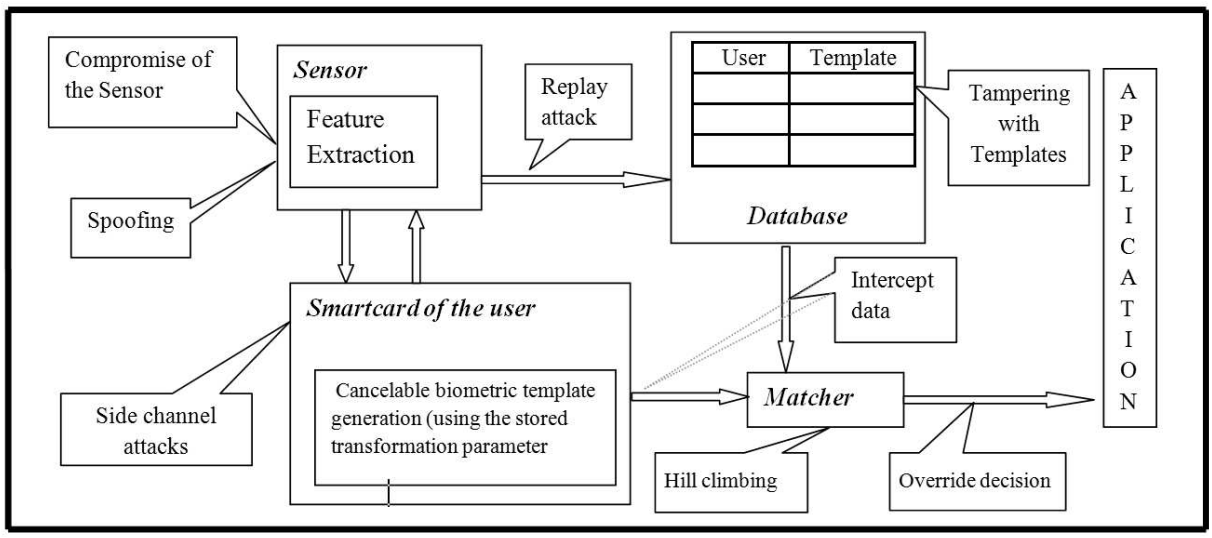

Figure 4.1: Security and Privacy Issues of Multi-factor Biometric Systems

4.4.1 Adversarial Capabilities and Goals

In order to define the adversaries capabilities and goals, one has to determine the security and privacy issues for MFBA systems, which is summarized in section 4.4.1. In our security model, the goal of the adversary is to impersonate a user. We model the adversary's power by allowing him to interact with protocol instances through several oracles as below.

- Reveal: This query models the leakage of information about the authentication requests, where an eavesdropper listening to the communication channel can obtain the encrypted stable parts if the session key is compromised. Namely, it models the leakage of information about the session key agreed on by the sensor client and the server as in the case of a misuse of it afterward. Moreover, the authentication data of a user (i.e. encrypted stable parts) can also be leaked from the server due to an insider attack.

- Corrupt: This query models corruption capabilities of the adversary. She can indeed steal/break either one of the authentication factors of the user. In particular, the oracle can output the biometrics of the user. It models the attack against the sensor client, i.e. the compromise of the biometrics of the user. Alternatively, the oracle can output *either* the transformation parameters (t, s, E) *or* some part of the non-stable parts (r_js) that are stored in the tamperproof smart card of the user. It models the side channel attack against the smart card of the

user. Clearly, the adversary is restricted to query the corrupt oracle at most for one authentication factor. Besides, no corruption can be performed during an authentication session, but before a new session starts. For even higher security, a user password can be added as a third factor. Then, we allow the adversary to corrupt two authentication factors, one of which must be the password.

4.4.2 User Privacy

We define the security notion for MFBA as User Privacy. To formally model this notion, we describe a security game between a challenger that simulates the server and an adversary that tries to impersonate a user. The adversary can ask several queries, but to the server only: We only consider adversaries whose goal is to impersonate a client to the server. Briefly, user privacy means that the adversary cannot impersonate a user to the server and thus cannot access user-specific applications. The formal definition of user privacy is as follows:

Given an adversary A running against the MFBA scheme and a simulator S that simulates the registration phase of the scheme, we consider the following game between A and S. At the end of the game, A makes an authentication request for the user U. If successfully authenticated, A wins, otherwise, A looses.

Experiment $Exp_A(l)$
$(pk, sk) \leftarrow \mathsf{Keygen}(1^l)$
$c \leftarrow \mathsf{Encrypt}(\nu, pk)$
$\emptyset \leftarrow Registration(ID, pk, c, \mathsf{ZKP})$
c', $\mathsf{ZKP}' \leftarrow A^{\mathcal{O}_1, \mathcal{O}_2}(Verification, pk)$
If $c' \approx c$ and ZKP' is verified, return 1, else return 0

A biometric authentication scheme satisfies the notion of User Privacy if
$Succ_A(l) = Pr[Exp_A(l) = 1] < negl(l)$ for all PPT adversaries A.

Here, the simulator S simulates the enrollment phase by registering the encrypted authentication data c and the corresponding zero knowledge proof ZKP. S registers the encryption of the stable parts $c = \mathsf{Encrypt}(\nu, pk) = \langle \mathsf{Encrypt}(\nu_1, pk), ..., \mathsf{Encrypt}(\nu_k, pk) \rangle$ and the corresponding proofs $\mathsf{ZKP} = \langle \mathsf{ZKP}(\nu_1), ..., \mathsf{ZKP}(\nu_k)) \rangle$ of knowledge of plaintext ν_js. Having access to Reveal and $\mathsf{Corrupt}$ oracles denoted by $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively, the adversary A tries to impersonate the user U. If A succeeds, namely, impersonates the user U to the simulator, A wins the game.

4.4.3 The concrete scheme

In this section, we present our MFBA scheme, which consists of three components: The user U with a smart card, a sensor client SC and a service provider SP. An overview of the registration and verification phases are presented in figure 4.2 and 4.3. For the first construction that is based on the elliptic curve ElGamal encryption scheme and a non-malleable ZKP based on the Schnorr signature, we assume that biometrics is represented as an ordered set of features such as face [Sutcu et al., 2006].

- *Setup Phase:* The parameters of the elliptic curve ElGamal encryption scheme are initialized with pairing friendly elliptic curve group $\mathbb{G}$ and bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ and a map $G : \{0,1\}^* \rightarrow \mathbb{G}$. Each user U posesses an ElGamal key pair (pk_U, sk_U) that is used to encrypt the stable parts of the biometrics. We note that after the registration phase, the user does not need to store the secret key in his card. Also, SP and SC generate their key pairs to build a secure communication channel between the entities.

- *Enrollment Phase*: U registers to the system as follows:

  1. SC extracts U's raw biometrics b and the raw data is transformed via a translation and scaling as described in section 4.3.2. Next, each transformed biometrics v'_j for $1 \leq j \leq k$, is separated to fraction (residual parts) r_js and integer (stable) part ν_js using a reflected modulus $rmod$ that does not increase the distance between points [Boult et al., 2007, Scheirer and Boult, 2009]. Each stable part ν_j is mapped using G and the resulting value $\mu_j = G(\nu_j)$ is encrypted using the public key pk_U of the user U to obtain $w_j = \mathsf{Encrypt}(\mu_j, pk_U) = (w_j^1, w_j^2)$, whereas the residual r_js are stored in the smart card in clear for each $1 \leq j \leq k$. To enforce the secrecy of the non-encrypted r_js, we use an approach similar to the Match On Card (MOC) system [Bringer et al., 2008], where fresh biometrics are acquired by the sensor client but the matching of the residual parts are made inside the card. This way, the confidentiality of the r_js relies on inherent protections of smart cards against physical threats, where r_js do not go out of the card.
  2. U registers his ID and the encrypted stable parts together with the ZKPs and his signature on w_js at the SP. U stores the parameters (i.e. transformation parameters, reflected modulus $rmod$, windowing parameter E) and (optionally) the residual parts (i.e. r_js) in his smart card. U does not store the secret key of his public key pk_U as it will not be used after signing the encrypted parts.

- *Verification Phase*: U authenticates to SP as follows:

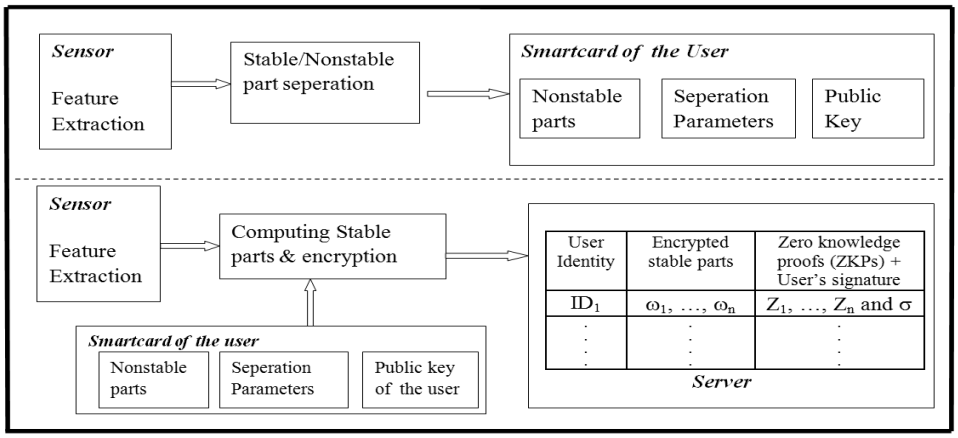

Figure 4.2: Registration Phase

1. The sensor client SC extracts U's fresh b' and communicates with the smart card of U to send b'.
2. The user's smart card seperates the stable and residual parts of each feature using the parameters stored in his card and encrypts the stable parts of each feature using the ElGamal public key pk_U of U. If the residual parts are stored in U's smartcard, an additional check can be performed on card by matching the fresh residuals to the stored ones based on the predefined thresholds for each residual part. The encrypted stable parts $w'_j = (w'^1_j, w'^2_j)$ are sent to SC together with the associated ZKPs.
3. SC and SP agree on a session key. SC sends the encryption of the data obtained from U using this session key.
4. SP decrypts the data using the session key, verifies the ZKPs and compares the fresh encrypted stable parts w'_j's of U to the previously stored data w_j's by using the homomorphic property of ElGamal encryption scheme. For $1 \leq j \leq k$, SP selects $s_j \xleftarrow{R} \mathbb{Z}_p^*$ to compute

$$R_j = \left(R_j^1, R_j^2\right) = \left(\left(\frac{w_j^1}{w_j'^1}\right)^{s_j}, \left(\frac{w_j^2}{w_j'^2}\right)^{s_j}\right)$$

5. SP checks for $1 \leq j \leq k$ whether $\hat{e}(g^x, R_j^1) = \hat{e}(g, R_j^2)$ by computing $2k$ pairings. Here, $pk_U = g^x$ is the public key of U.
6. Finally, SP counts the number of the equations satisfying the above condition and computes the matching score ms, which can be compared to the

matching score mns of the non-stable parts stored on card. The comparison is made by using an efficient range proof such as [Peng and Bao, 2010], which does not reveal the mns even to the server but proves that $mns \approx ms$. If the user can prove to the server that the mns lies within the range determined by the predefined threshold of the system, SP decides to authenticate U.

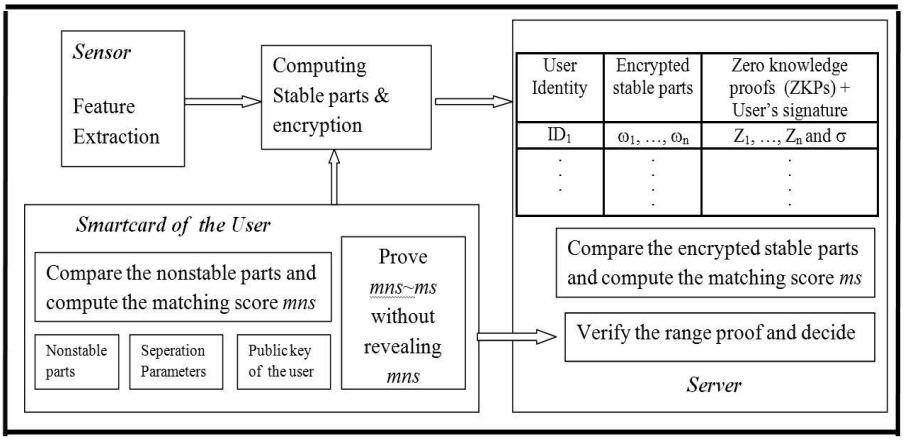

Figure 4.3: Verification Phase

Theorem 4.1. *Assume that an attacker running against our protocol breaks the user privacy by making at most two queries to the* Corrupt*, q_H queries to the random oracle H and q_R queries to* Reveal *oracle, then the simulator can break the existential unforgeability of the Schnorr Signature scheme.*

Proof. User privacy is achieved by playing a game between the simulator that simulates the environment (i.e. the server) to the adversary A. The simulator generates a non-malleable ElGamal ciphertext (i.e. an ElGamal encryption and the corresponding zero knowledge proof based on Schnorr signature scheme) that represents the encryption of the stable parts of the user U^* with public key pk_{U^*}. Using the adversary running against our MFBA, the simulator is able to obtain a forgery of the Schnorr signature.

- *Simulation of the Enrollment:* The challenger sets the (non-malleable) ElGamal public key of U^* as $pk_{U^*} = pk$ and generates the registration data for U^* as $c = \langle \mathsf{Encrypt}(\mu_1), ..., \mathsf{Encrypt}(\mu_k) \rangle$ encrypted with pk and the corresponding Schnorr $\mathsf{ZKP} = \langle \mathsf{ZKP}(\mathsf{Encrypt}(\mu_1)), ..., \mathsf{ZKP}(\mathsf{Encrypt}(\mu_k)) \rangle$. The simulator S registers this data for the user U^* and returns the adversary A the public key $pk_{U^*} = pk$. Here, $\mu_i \in \mathbb{G}$ denotes the stable parts of U^*'s biometrics, $\mathsf{Encrypt}(\mu_i) = (u_i, v_i) =$

$(g^{r_i}, y^{r_i}\mu_i)$ and $z_i = g^{k_i}, c_i = r_i \cdot \mathrm{H}(g, u_i, v_i, z_i, U^*, t_R) + k_i$ for $1 \leq i \leq k$ are computed for the registration time t_R.

- *Simulation of the H oracle*: At each new input $(g, u_i, v_i, z_i, U^*, t)$, the simulator picks a random value h_i, inserts the tuple $(g, u_i, v_i, z_i, U^*, t, h_i)$ to the HList and returns h_i to the attacker A.

- *Simulation of the Reveal Queries*: To simulate the past authentication requests of user U^*, the attacker queries for the authentication data that was communicated to the server at time t'. S generates a rerandomization of the challenge ciphertext as $\mathsf{Encrypt}(\mu_i) = (u_i', v_i') = (g^{r_i} g^{r_i'}, y^{r_i} y^{r_i'} \mu_i) = (g^{r_i + r_i'}, y^{r_i + r_i'} \mu_i)$. Next S picks at random $e_i', s_i' \in \mathbb{Z}_q^*$ and computes $(u_i')^{-e_i'} g^{s_i'}$ as z_i'. Finally, the tuple $(g, u_i', v_i', z_i', U^*, t', e_i')$ is inserted to the HList and S returns (u_i', v_i', z_i', s_i') as the answer of A's query to the reveal oracle.

  It is easy to check that the answer of the simulator is correct since the Schnorr ZKP $(z_i', c_i') = ((u_i')^{-e_i'} g^{s_i'}, s_i')$ is verified via

$$g^{c_i'} = g^{(r_i + r_i')e_i' + k_i'} = g^{(r_i + r_i')e_i'} g^{(r_i + r_i')(-e_i')} g^{s_i'} = g^{s_i'}$$

- *Simulation of the Corrupt Queries*: The attacker can *either* query for the true biometrics of U^* *or* for the transformation parameters stored at U^*'s smart card. If a password is also used as a third factor, A can query for U^*'s password. Thus, at most two queries are allowed to this oracle, one of which is for the user's password.

After polynomial number of queries to the oracles, the attacker impersonates U^* by returning the authentication data for U^* for the current time t^* as $\mathsf{Encrypt}(\mu_i) = (u_i^*, v_i^*) = (g^{r_i^*}, y^{r_i^*} \mu_i)$ and the Schnorr ZKP $z_i^* = g^{k_i^*}, c_i^* = r_i^* \cdot \mathrm{H}(g, u_i^*, v_i^*, z_i^*, U^*, t^*) + k_i^*$, which is actually a Schnorr signature on the message $(g, u_i^*, v_i^*, z_i^*, U^*, t^*)$.

Thus, the simulator S obtains a forgery on the message $(g, u_i^*, v_i^*, z_i^*, U^*, t^*)$ and breaks the existential unforgeability of the Schnorr Signature Scheme. The intuition of the game is that the adversary having access to either the biometrics or the transformation parameters (translation, scaling,windowing parameter E) cannot compute the stable parts μ_is, thus the only way for impersonating the user U^* is the forgery on the Schnorr signature.

Hence, we can state that our protocol guarantees user privacy based on the existential unforgeability of the Schnorr signature scheme. Although the security reduction for Schnorr signatures are given in the random oracle model, recently, [Neven et al., 2009] enhances confidence in the instantiation of Schnorr signatures, in particular its elliptic-curve variant, with hash functions like SHA-1 and MD5 by analyzing its security in another popular idealization, the generic group model. □

Corollary 4.1. *Assume that an attacker running against our protocol breaks the user privacy by making at most two queries to the* Corrupt*, q_H queries to the random oracle H and q_R queries to* Reveal *oracle, then the simulator can solve the discrete logarithm problem.*

If we go one step further, we can solve the discrete logarithm problem using the security proof of [Pointcheval and Stern, 2000] for Schnorr Signature scheme that is based on the forking lemma technique. The intuition is that if the attacker A can forge a signature, then the simulator can construct a modified adversarial algorithm which (1) constructs a random oracle H and runs the attacker A until she produces a forged signature (z_i, c_i) on the message $m = (g, u_i, v_i, z_i, U^*, t^*)$, (2) fabricates a second random oracle H' which is identical to H except for its output on m, i.e. $H(m) \neq H'(m)$ and re-runs the adversary on the same inputs (i.e. m) to obtain another signature (z_i, c_i'). Finally, the simulator obtains r_i of the forged signature, namely, solves the discrete logarithm problem on g^{r_i} by computing $z_i = g^{c_i} g^{-r_i H(m)} = g^{c_i'} g^{-r_i H'(m)}$, implying $g^{r_i(H(m)-H'(m))} = g^{(c_i'-c_i)}$. Hence, $g^{r_i} = g^{(c_i'-c_i)/(H'(m)-H(m))}$ and $r_i = (c_i' - c_i)/(H'(m) - H(m))$.

The computation of the discrete logarithm is achieved using this oracle replay attack [Pointcheval and Stern, 2000], which means that by the polynomial replay of the attack with different random oracles, the simulator is able to obtain two signatures on the identical message $m = (g, u_i, v_i, z_i, U^*, t^*)$ and the value $z_i = g^{k_i}$ but for different hash values $H(m) \neq H'(m)$.

Hence, we can state that our protocol guarantees user privacy based on the difficulty of solving discrete logarithms.

Clearly, from the value r_i of the forged signature, the simulator computes the plaintext (i.e. the stable parts of U^*'s biometrics), thus breaks the one-wayness of the ElGamal encryption. Identical to the arguments of the proof for non-malleable ElGamal [Tsiounis and Yung, 1998], since signed ElGamal encryption scheme is plaintext aware [Schnorr and Jakobsson, 2000] with respect to the challenge time t^* and user U^*, the party that included t^* and U^* in the encryption (i.e. the party who produced the Schnorr signature) can compute the value r_i corresponding to the signature, and from this compute the stable part $\mu_i = v_i / y^{r_i}$, where $y = pk$ is U^*'s public key. If the attacker returned the data by changing any part of the previously obtained authentication data after querying the Reveal oracle, she needs to obtain a signature on the message $(g, u_i^*, v_i^*, z_i^*, U^*, t^*) \neq (g, u_i', v_i', z_i', U^*, t')$, which she has not seen before. Thus, she has to know or efficiently compute r_i. Thus, for any modified ciphertext, the attacker knows the randomness r_i he used in generating the authentication data, thus knows the plaintext (i.e. stable parts of U^*'s biometrics). Hence, the attacker who can generate a valid ZKP for the encrypted stable parts of U^* for the authentication time t^*, does actually know the underlying plaintext.

Lemma 4.1. *For Signed ElGamal encryption implemented on a pairing friendly elliptic*

curve, achieving IND-ATK is impossible. However, the same scheme is an IND-CCA secure PKE on a non-bilinear group.

Proof. When playing the indistinguishability game, the adversary knows the challenge plaintexts m_0 and m_1, he does not even need to resort its plaintext choosing capability. After challenge phase, what the adversary knows is the public key, challenge plaintexts m_0 and m_1 and the challenge ciphertext. Using a bilinear pairing computation, he can test whether the challenge ciphertext is an encryption of m_0 or m_1. However, in a non-bilinear group, namely a group where the Decisional Diffie-Hellman problem (DDH) is intractable, signed ElGamal is proven IND-CCA secure in the random oracle model. □

Lemma 4.2. *Signed ElGamal encryption implemented on a pairing friendly elliptic curve can only guarantee OW-CCA security in the random oracle model based on the one-wayness of ElGamal encryption and strong EUF-CMA secure Schnorr signature.*

Proof. Before presenting the proof, we note that EUF-CMA secure Schnorr signature should be transformed so that strong EUF-CMA security is achieved. A signature system is said to be strongly unforgeable if the signature is existentially unforgeable and, given signatures on some message m, the adversary cannot produce a new signature on m. Strongly unforgeable signatures are used for constructing chosen ciphertext secure systems and group signatures. This is a stronger notion than (standard) EUF-CMA, which basically requires the adversary to come up with any valid message-signature pair that does not equal to the output of the signing oracle. There exists various generic constructions that convert any EUF-CMA secure signature to a strong one, both in ROM and standard model. The reader is referred for the details to [Liu et al., 2010].

We make the proof by contradiction. Assume that elliptic curve signed ElGamal is not secure against CCA attacks, which implies that the decryption oracle can help the adversary to invert the challenge ciphertext. Thus, an attacker can construct a ciphertext by adapting or mutating the challenge ciphertext so that the resulting signed ElGamal ciphertext is a valid query to the decryption oracle. And from the answer of the decryption oracle, the adversary is able to invert the challenge ciphertext. Again, this requires to (strongly) forge a signature $\sigma = (z, c)$ on the (modified) ElGamal ciphertext $m = (u, v)$. However, due to the strong EUF-CMA secure signature on the message m, the adversary cannot even produce a new signature on m for which he has a valid signature. Thus, we have a contradiction and the lemma follows. In [Schnorr and Jakobsson, 2000], it is emphasized that a signed ElGamal ciphertext consists of an ElGamal ciphertext (u, v) and a Schnorr signature (c, z) of the message $m = (u, v)$ for the public signature key u. The signature does not contain any information about m

as (c, z) depends on m exclusively via some hash value that is statistically independent of m.

□

As it is shown in [Tsiounis and Yung, 1998, Schnorr and Jakobsson, 2000], signed ElGamal encryption is IND-CCA secure in the random oracle model (ROM) since Schnorr signature is also proven secure in ROM. An alternative construction could be the combination of ElGamal encryption with a strong EUF-CMA secure signature scheme or with an interactive non-malleable ZKP of knowledge of plaintext as described in [Katz, 2002], which results in an IND-CCA secure scheme in the standard model. We note that ElGamal encryption on elliptic curve groups equipped with bilinear pairings can only guarantee OW-PCA security based on the GDH problem in the standard model. When this scheme is combined with a non-malleable ZKP, the construction can achieve at most OW-CCA security in the standard model instead of IND-CCA security. Hence, there exists a tradeoff between the security the scheme achieves and the requirement for making the authentication decision without using any secret key.

The above described biometric scheme is partly based on an earlier publication of [Sarier, 2010b], which constructs a remote biometric authentication scheme using OW-PCA secure ElGamal encryption (due to the GDH problem) implemented on a pairing friendly elliptic curve and for verification, the equality test of two encrypted biometric templates is performed through bilinear pairings in the encrypted domain. This construction can be considered as an instance of a later publication that introduced the concept of "PKE with equality test" [Yang et al., 2010], which is very similar to our application on biometrics if generalized to message encryption. In their system, the authors design an encryption scheme on elliptic curves that is OW-CCA secure in the random oracle model based on the CDH problem. Again, the equality test is performed via bilinear pairings. The disadvantage of that scheme is that, when ElGamal encryption implemented on a pairing friendly elliptic curve is combined with a (interactive) non-malleable ZPK of [Katz, 2002], OW-CCA security can be achieved in the standard model, whereas the author's scheme is secure in the random oracle model. However, for efficiency reasons, our system based on signed ElGamal or the recently introduced scheme of [Yang et al., 2010] that are both secure in ROM can be applied for biometric setting.

4.4.4 Biometrics as an Unordered Set

Although some biometric modalities can be represented as an ordered set of features such as face biometric, for fingerprints this is not a trivial task [Boult et al., 2007]. Fuzzy vault based systems try to find a solution for biometrics that consists of an unordered

set of features, however, as shown recently, there exists many attacks against these systems that reveal both the secret used for authentication and the biometric template that hides this secret. Since ordering or grouping of features are not possible for some biometric modalities, we cannot use a probabilistic encryption scheme such as ElGamal encryption system since the comparison cannot be made in the encrypted domain. Also, the matching of the non stable part on card will not be consistent with the matching of the encrypted stable parts at the remote server. (There could be accidental matches on card that results in different matching scores). However, if we use a deterministic scheme like RSA, the remote server can send the indices of the fresh encrypted parts that exactly match the stored encrypted stable parts and thus, the match on card system performs the matching according to the instruction of the remote server, which will result in similar matching scores at the both entities. We note that the stored stable features at the SP and the non-stable parts stored at the smart card share the same order at the enrollment phase, i.e. if a specific feature is stored as the second feature in the server, than the unstable part of this feature is also stored at the 2. place on card. Besides, computing the indices of the matching stable parts is also possible when elliptic curve ElGamal is used, however, the remote server has to compute in worst case $O(k^2)$ bilinear pairings and modular divisions, where the computation of one bilinear pairing is approximately 9 modular exponentiations. (k is the size of the feature set). Thus, our previous system is impractical compared to a deterministic encryption scheme for unordered biometric features. Finally, replay attacks should be considered when a deterministic scheme is used as encryption of the same message results in the same ciphertext, whereas the encryption of the same message results in a different ciphertext due to the random coins used in the probabilistic encryption scheme. Thus, the communication channel should be encrypted using a session key and ZKPs designed for RSA [Rivest, 2001] should be attached to the ciphertext with a time stamp as before. In [Katz, 2002], the author also presents an interactive non-malleable ZKP of knowledge of plaintext for RSA encryption.

- *Setup Phase*: The RSA keys of each user is initialized, where $pk_U{=}(n, e)$ is the public key of the user that is only used in the encryption of the stable part. Also, SP and SC possesses two key pairs for an encryption and signature scheme to generate the session keys.

- *Enrollment Phase*:

    1. SC extracts U's raw biometrics b, which is processed as in the previous section to obtain $w_j = (\mathsf{Encrypt}(\nu_j), pk_U)$ and the corresponding ZKPs for RSA such as the system described in [Rivest, 2001].
    2. U registers his ID and the encrypted stable parts together with the ZKP proof at the SP and U stores the residual parts (i.e. r_js) following the same

order of their corresponding stable parts and the parameters (i.e. transformation parameters, reflected modulus $rmod$) in his smart card. We note that, we cannot use the locations of the matched residual parts on card to determine the corresponding stable parts at the server since there could be accidental matches on card. The secret key of U that encrypts the stable parts is not stored as it will not be used during matching.

- *Verification Phase*:

  1. SC extracts U's fresh b' and communicates with the smart card of U to send the raw data b'.
  2. The user's smart card performs as before to compute the encrypted stable parts $w'_j = (\mathsf{Encrypt}(\nu_j), pk_U)$ and the corresponding ZKPs for RSA to send to SC, which signs them before sending to SP.
  3. SP verifies the signature and compares the fresh encrypted stable parts w'_js of U to the stored w_js.
  4. If the number of matched stable features is above the threshold, SP sends the signed order information of the matched stable parts to the SC. For instance, if the first stable part in the fresh query matched the third stable part stored in the gallery, than SP sends $[1 \rightarrow 3]$ to the SC. In order to leak no information about the actual matching score, SP sends to the client random order information for the non-matching parts.
  5. SC checks the signature and forwards the order information to the smart card, where the residual parts are matched on card to the fresh residual parts following this order. For instance, if SP has sent $[1 \rightarrow 3]$, then the smart card compares also the first non-stable part in the query to the third non-stable part stored at the smart card.
  6. Finally, the matching score ms computed by SP and the matching score mns of the non-stable parts stored on card are compared privately as before.

4.5 Discussion

In table 4.1, we analyze the success of the attacker against our system in case of 4 classes of attacks. In our security model, we only assume that the attacker cannot compromise both the sensor and the smart card of the user, otherwise, the attacker with the true biometrics of the user and the transformation parameters stored at the card can impersonate a user trivially. We note that this assumption may be relaxed if the user has its own biometric smart card reader, then we can obtain higher security against sensor compromise.

Table 4.1: Attacks against multi-factor biometric systems

	Compromise of User biometrics	Impersonation
(1) Server Compromise	×	×
(2) Side channel attack	×	×
(3) Session key compromise	×	×
(4) Sensor Compromise	Unavoidable	× if no (2)

Also, the server stores the encrypted stable parts together with the ZKPs, thus, tampering with the stable parts is not possible since they are signed with the secret key of each user (which is not required after this operation, thus not need to be stored in the smartcard of the user) and the ZKPs are non-malleable i.e. cannot be modified to work with the new stable parts. Another advantage of the new system is that revoking of the templates is possible since the user can choose a different public key in the encryption of the stable parts and use different transformation parameters in the separation of the stable/nonstable parts. This also prevents linkability of the stored templates of the same user at different servers. We emphasize that the smart card of the user does not output a matching score, but a range proof on this score, which does not leak any information about the score and proves the server that the score lies within a range that the server accepts. Thus, no information useful for a hill climbing attack can be obtained. We note that, the second matching on card is optional, it may give a higher confidence to the server and thus, it can be considered as a second layer of authentication. Alternatively, SP can also store the non-stable parts and can perform the matching of the two parts himself. This way, there is no need for a range proof and a matching score, instead the server outputs an accept/reject decision. Thus, hill climbing attacks are not applicable and the server cannot compute the true biometrics of the user due to the encrypted stable parts and the secrecy of the the transformation parameters that are stored in the tamperproof smart card of the user.

As a final note, the features of some biometric modalities (i.e. fingerprints) can take small integer values or the feature space of some biometric traits could be a small universe of features. Thus, an adversary can try to find out the stable parts of some user by computing the stable parts using the captured biometric features of the user and randomly picked transformation parameters. One can prevent this particular attack for biometrics with small feature (and thus small universe of stable parts) by allowing each user to implement a different map for encoding each stable part to a group element before encrypting them. Specifically, in the registration phase of section 4.4.3, each stable part ν_j is mapped using $G : \{0,1\}^* \to \mathbb{G}$ before applying ElGamal encryption. This randomized map prevents the adversary to perform a brute force attack on the

stable parts since the randomization parameters are stored at each user's smart card. Alternatively, we can implement the same idea presented in section 3.10 of the previous chapter to prevent attacks resulting from small feature space. Besides, encoding of arbitrary bit sequences into sequences of group elements is easy for particular groups such as $\mathbb{Z}_q^*$ that correspond to an interval of integers. In [Tang et al., 2008], the encoding problem is solved by mapping the biometrics to an element of $\mathbb{Z}_q^*$ and the resulting value b is encrypted as $\mathsf{Encrypt}(g^b, pk)$.

4.6 Conclusion

In this chapter, we present the security model for MFBA and describe two schemes for ordered/unordered set of biometric features that combine a different extraction method, zero knowledge proofs and homomorphic encryption schemes. The security notion for MFBA is defined as user privacy, which is achieved for our protocols even in the case of simultaneous attacks against the system. Finally, our schemes are provably secure in our security model but less complex than existing biometric schemes that provide a security reduction.

Chapter 5

Efficient Biometric Identity Based Signature

In this chapter, we present a new biometric Identity Based Signature (IBS) scheme that is applicable for any type of biometrics (i.e. represented as an ordered or unordered set of features) and is more efficient compared to the current fuzzy IBS of [Yang et al., 2008] and threshold Attribute Based Signature (t-ABS) scheme of [Shahandashti and Safavi-Naini, 2009], when implemented for biometric identities. Moreover, the new scheme could function as a fuzzy IBS or t-ABS scheme if the biometric features are replaced by attributes defining the identity of the signer.

We prove the security of our new scheme in the framework of the exisiting adversarial models for fuzzy IBS and t-ABS and additionally in the framework of a stronger model, which basically simulates the leakage of partial secret key components of the challenge identity. This property is not considered in the current security model of fuzzy IBS (and t-ABS), which return to the adversary only the private key components belonging to any identity other than (i.e. not similar to) the challenge identity. However, in our stronger security model, we allow the adversary to query for some of the private key components belonging to the challenge identity.

Our new scheme is based on the currently most efficient pairing based IBS scheme. In order to show the efficiency of our new scheme, we first describe an intermediate scheme called "modified t-ABS", which is obtained by replacing the computationally expensive T function in t-ABS of [Shahandashti and Safavi-Naini, 2009] with a MapToPoint hash function similar to the conversion presented in [Pirretti et al., 2006] for fuzzy IBE systems.

However, our new scheme is even more efficient since we do not require a MapToPoint hash function as in the modified t-ABS instead we use only an ordinary hash function. It is known that the cost of a MapToPoint hash operation is bigger than one inversion

in $\mathbb{Z}_q$. Our scheme is compared to other error tolerant signature schemes and shown to be much more efficient in terms of its each phase.

Based on the recently defined privacy notions, we show that our scheme achieves weak signer-attribute privacy and our intermediate proposal modified t-ABS achieves full signer attribute privacy if the additional protocols and architecture described in the t-ABS scheme of [Shahandashti and Safavi-Naini, 2009] is employed. The contributions of this chapter is based on the publication [Sarier, 2010c].

5.1 Introduction

Introduced in [Sahai and Waters, 2005], fuzzy IBE uses biometric attributes as the identity instead of an arbitrary string like an email address. This new system combines the advantages of IBE with those of biometric identities, where IBE avoids the need for an online Public Key Infrastructure (PKI), which is the most inefficient and costly part of Public Key Encryption (PKE). Fuzzy IBE could be used in an ad-hoc setting where the users are unprepared, namely without having any public key or even predefined e-mail addresses. Instead, the signer could present his biometrics to the verifier, who can check the signature for validity using the biometric identity of the signer. Besides, the use of biometric identities in the framework of IBE simplifies the process of key generation at the Private Key Generator (PKG). Since biometric information is unique, unforgettable and non-transferable, the user only needs to provide his biometrics at the PKG under the supervision of a well-trained operator to avoid biometric forgery and to obtain his private key instead of presenting special documents and credentials to convince the PKG about his identity. It should be noted that biometrics is assumed as public information, hence the compromise of the biometrics does not affect the security of the system. This point of view is also accepted in the biometrics community, where the raw biometric data is assumed as public data whereas the revocable biometric template that is stored in a central database or on a smartcard for biometric authentication is considered as private data. The reader is referred to the next chapter for the other advantages of fuzzy IBE.

The signature analogue of fuzzy IBE is introduced in [Yang et al., 2008], where a provably secure fuzzy Identity Based Signature (IBS) scheme is described. Since the error tolerance property is satisfied, fuzzy IBS of [Yang et al., 2008] is applicable for biometric identities and it shares the same advantages of fuzzy IBE. The private key components of a fuzzy system are generated by combining the values of a unique polynomial on each feature of the biometrics with the master secret key ms of PKG. However, due to the noisy nature of biometrics, a fuzzy system allows for error tolerance in the decryption stage for fuzzy IBE (or in the verification stage for fuzzy IBS). Particularly, a signature constructed using the biometrics ID could be verified by the

receiver using a set of publicly computable values corresponding to the identity ID', provided that ID and ID' are within a certain distance of each other. Moreover, fuzzy IBS could be considered in the context of Attribute Based Signature (ABS), which allows the signer to generate a signature using the attributes she possess.

Another approach for incorporating biometrics into IBS is presented in [Burnett et al., 2007], where the error tolerance is provided by a different identity structure compared to fuzzy IBS, namely by integrating a fuzzy extractor into the IBS scheme. This way, both the signer and verifier operate with the same public key, which is required for standard cryptographic schemes. The limitation of this approach is that it requires a special type of biometrics which can be represented as a binary string, which can be error corrected and hashed to be used as a unique identity string. Thus, it is not suitable for biometrics that can be represented as an unordered set of features such as fingerprint minutia.

5.1.1 Related Work

The first fuzzy IBE scheme is described by Sahai and Waters in [Sahai and Waters, 2005] and the security is reduced to the MBDH problem in the standard model, where the size of the public parameters is linear in the number of the attributes of the system or the number of attributes (or features) of a user. More efficient fuzzy IBE and biometric IBE schemes are achieved with short public parameter size by employing the random oracle model (ROM) [Pirretti et al., 2006], [Baek et al., 2007], [Furukawa et al., 2008], [Sarier, 2008]. The signature analogue of fuzzy IBE, i.e. fuzzy IBS is first defined in [Yang et al., 2008]. Similarly, a threshold Attribute Based Signature (t-ABS) scheme and its extension to threshold attribute based anonymous credential systems is presented in [Shahandashti and Safavi-Naini, 2009], where the authors also define the security notions of weak/full signer attribute privacy for t-ABS.

Burnett et al [Burnett et al., 2007] described the first biometric IBS scheme called BIO-IBS for a biometrics that can be represented as a binary string such as Iris, where they used the biometric information as the identity and construct the public key (namely the identity) of the signer using a fuzzy extractor, which is then used in the modified SOK-IBS scheme [Bellare et al., 2004].

Besides, the fuzzy IBS scheme of [Yang et al., 2008] is provably secure in the standard model, where the scheme is based on the Sahai-Waters construction [Sahai and Waters, 2005] and the two level hierarchical signature of Boyen and Waters [Waters, 2005]. However, the scheme is very inefficient due to the $d(n+4)$ exponentiations and the $d+2$ bilinear pairing computations during the verification process, where d is the error tolerance parameter of the scheme and n is the size of the feature (i.e. attribute) set of each user. Recently, a threshold ABS (t-ABS) scheme [Shahandashti and Safavi-Naini,

2009] with the same key generation phase as of fuzzy IBS and with threshold attribute based verification is designed, which suffers from the same disadvantages described for this fuzzy IBS. Due to the threshold verification, t-ABS can also be implemented as a biometric IBS scheme as opposed to other ABS schemes [Maji et al., 2008], [Khader, 2007], [Shanqing and Yingpei, 2008], which are proven secure in the ROM or generic group model. Thus, there is a need to devise an efficient and provably secure signature scheme with error-tolerance property in order to integrate biometric data.

5.1.2 Our Contributions

Our new scheme is based on the Sakai Kasahara Key Construction [Sakai and Kasahara, 2003] and the security is reduced to the k-DHI computational problem in the ROM with a more complex security reduction compared to [Chen and Cheng, 2005], [Chen et al., 2006], [Barreto et al., 2005]. In order to show the efficiency of our new scheme, we first describe an intermediate scheme called "modified t-ABS", which is obtained by replacing the computationally expensive T function in t-ABS of [Shahandashti and Safavi-Naini, 2009] with a MapToPoint hash function similar to the conversion presented in [Pirretti et al., 2006] for fuzzy IBE systems. Specifically, in previous fuzzy IBS (and fuzzy IBE) constructions, there is a fixed value, n, on the number of attributes that is defined at the setup in order to label a ciphertext. The setup function publishes values $t_1, ..., t_n$. The function $T(i)$ is computed in both the key generation and verify algorithms as: $T(i) = g^{x^i} \prod_{j=1}^{n+1} t_j^{\Delta_{i,N}}$ where N is the set $\{1, ..., n+1\}$. By applying the conversion presented in [Pirretti et al., 2006], the $n+1$ exponentiations needed to solve T at each verification have been replaced with a single MapToPoint hash function used as a random oracle. Besides, each ciphertext does not have to contain exactly n attributes to describe the identity of the user as in other fuzzy IBS and t-ABS schemes.

However, our new scheme is even more efficient since we do not require a MapToPoint hash function as in the modified t-ABS instead we use only an ordinary hash function. A MapToPoint hash function converts a user's identity to a point on the underlying elliptic curve in IBE schemes. Current efficient fuzzy IBE schemes [Pirretti et al., 2006, Baek et al., 2007] employ this special function, which is usually implemented as a probabilistic algorithm and is more expensive than a point scalar multiplication in terms of computation time [Chen and Cheng, 2005, Chen et al., 2006]. This operation is also time consuming and cannot be treated as a conventional hash operation which is commonly ignored in performance evaluation. Besides, as it is noted in [Barreto et al., 2005, Smart and Vercauteren, 2007], it is difficult to find groups as the range of the MapToPoint hash function and to define an efficient isomorphism at the same time.

Our scheme is compared to other error tolerant signature schemes and shown to be much more efficient in terms of its each phase. Specifically, the verification phase of

the new scheme requires d exponentiations in group $\mathbb{G}$ and d pairing computations instead of $d(n+4)$ exponentiations and $d+2$ pairings as in the schemes of [Yang et al., 2008], [Shahandashti and Safavi-Naini, 2009] and achieves much shorter public parameter size, private key and signature sizes compared to these schemes. Also, we have a structurally simpler key generation algorithm compared to [Yang et al., 2008], [Shahandashti and Safavi-Naini, 2009], where the number of exponentiations in the group $\mathbb{G}$ is reduced from $n(n+4)$ as in [Yang et al., 2008], [Shahandashti and Safavi-Naini, 2009] to n and the cost of signing is half of the existing schemes. Based on the recently defined privacy notions, we show that our scheme achieves weak signer-attribute privacy and our intermediate proposal modified t-ABS achieves full signer attribute privacy if the additional protocols and architecture described in the t-ABS scheme of [Shahandashti and Safavi-Naini, 2009] is employed.

Finally, we present our arguments against the architecture of this scheme, which uses an additional party called "signature holder" that has access to the (biometric) attributes of the signer and the verification attribute set that is known by the verifier to be used during the verification of the signature. By knowing the two similar (biometric) attributes, the signature holder determines the common biometric attributes, i.e. performs an error correction based on set difference, and then communicates with the verifier to send a converted signature using the common attributes. This new model is designed in order to achieve weak and/or full attribute privacy, however, the model is against the definition of fuzzy IBS, which does not allow a third party to compute an error-corrected converted signature. Instead, the verifier should be able to verify the fuzzy IBS without any help from a third party even if the signature is generated by using an attribute set similar to the one the verifier. Thus, the t-ABS scheme of [Shahandashti and Safavi-Naini, 2009] cannot be considered as a fuzzy IBS scheme with this new model. This t-ABS scheme is actually identitical to the fuzzy IBS scheme presented in 2008 by [Yang et al., 2008] without this new model.

5.2 Definitions and Building Blocks

In order to introduce the new biometric IBS scheme, at first, we review the definitions and required computational primitives. The reader is referred to the background chapter for the details of Shamir's secret sharing, forking lemma and IBS based on Sakai-Kasahara key construction. Given a set S, $x \xleftarrow{R} S$ defines the assignment of a uniformly distributed random element from the set S to the variable x. Biometric identities will be element subsets of some universe, $\mathbb{U}$, of size $|\mathbb{U}|$, where each element is associated with a unique integer in $\mathbb{Z}_p^*$ as in [Baek et al., 2007], [Sahai and Waters, 2005]. Finally, we define the Lagrange coefficient $\Delta_{\mu_i,S}$ for $\mu_i \in \mathbb{Z}_p$ and a set S of

elements in $\mathbb{Z}_p$ as

$$\Delta_{\mu_i,S}(x) = \prod_{\mu_j \in S, \mu_j \neq \mu_i} \frac{x - \mu_j}{\mu_i - \mu_j}$$

The security of our scheme is reduced to the well-exploited complexity assumption (k-DHI), which is stated as follows.

Assumption 5.1. *(DH Inversion (k-DHI)). For $k \in \mathbb{N}$, and $x \xleftarrow{R} \mathbb{Z}_p^*$, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{F}$, given $(g_1, g_2, g_2^x, g_2^{x^2}, ..., g_2^{x^k})$, computing $g_1^{(1/x)}$ is hard.*

5.2.1 Forking Lemma

Forking Lemma was introduced by David Pointcheval and Jacques Stern [Pointcheval and Stern, 2000], which is specified in terms of an adversary that attacks a digital signature scheme instantiated in the random oracle model. The forking lemma states that if an adversary (typically a probabilistic Turing machine), on inputs drawn from some distribution, produces an output that has some property with non-negligible probability, then with non-negligible probability, if the adversary is re-run on new inputs but with the same random tape, its second output will also have the property. The authors show that if an adversary can forge a signature with non-negligible probability, then there is a non-negligible probability that the same adversary with the same random tape can create a second forgery in an attack with a different random oracle. The forking lemma has been used to prove the security of a variety of digital signature schemes and other random-oracle based cryptographic constructions. The reader is referred to the background chapter for the details of this method.

5.2.2 Fuzzy Identity Based Signature

In [Yang et al., 2008], the generic fuzzy IBS scheme is defined as follows. The same definition applies for t-ABS [Shahandashti and Safavi-Naini, 2009], where the identity consists of a set of attributes such as $\{university, faculty, department, group\}$.

- Setup: Given a security parameter l, the PKG generates the master secret key ms and the public parameters of the system.
- Extract: Given a user's identity $ID = \{\mu_1, ..., \mu_n\}$ and the master secret key ms, the PKG returns the corresponding private key D^{ID}. Here, n denotes the size of the set ID.

- Sign: A probabilistic algorithm that takes as input the private key D^{ID} associated to the identity ID, public parameters and a message $m \in M$ and outputs the signature σ.

- Verify: A deterministic algorithm that given an identity ID' such that $|ID \cap ID'| \geq d$, the signature σ together with the corresponding message m and the public parameters, returns a bit b. Here $b = 1$ means that σ is valid and d denotes the error tolerance parameter of the scheme.

Correctness: A fuzzy IBS scheme has to satisfy the correctness property, i.e., a signature generated by a signer with identity ID must pass the verification test for any ID' if $|ID \cap ID'| \geq d$.

5.2.3 Security Model

A fuzzy IBS scheme is selectively unforgable under adaptive chosen message and given identity attacks (SUF-FIBS-CMA) if no probabilistic polynomial time (PPT) adversary A has a non-negligable advantage in the following game.

- **Phase 1**: The adversary A declares the challenge identity $ID^* = \{\mu_1^*, ..., \mu_n^*\}$.

- **Phase 2**: The challenger runs the Setup algorithm and returns the system parameters to A.

- **Phase 3**: A issues private key queries for any identity ID' such that $|ID' \cap ID^*| < d$. The adversary issues signature queries for any identity.

- **Phase 4**: A outputs a forgery (ID^*, m^*, σ^*), where A does not make a signature query on (m^*, σ^*) for ID^*.

The success of A is defined as $\text{Succ}_A^{\text{SUF-FIBS-CMA}}(l) = \Pr[\mathsf{Verify}(ID^*, m^*, \sigma^*) = 1]$.

Collusion Resistance: It is important to note that the above definition of unforgeability guarantees collusion resistance since users with common biometric features cannot collude to generate a signature that is not generable by one of the colluders.

Remark 5.1. *The second security reduction of our scheme allows the adversary A to have as much power as possible by providing A with some of the private key components of the challenge identity ID^* except for the component $\mu^* \in ID^*$. Thus, our security model is stronger than the (SUF-FIBS-CMA) model of [Yang et al., 2008], [Shahandashti and Safavi-Naini, 2009] and the details of this model is presented in section 5.3.3.*

5.2.4 Signer-Attribute Privacy

In [Shahandashti and Safavi-Naini, 2009], privacy of the signer is guaranteed with an additional algorithm that is run by an additional entity called "signature holder" which knows the set of signer attributes that are known to the verifier, namely ID' in our setting. A signature holder can always check a signature against possible verification attribute sets to deduce information about the signer's attributes since it stores both the ID that is used to sign the message and the verification attribute set ID'. To preserve privacy of signers, the t-ABS scheme is equipped with an additional algorithm for converting the signature to another signature that is verifiable against the verifier and only reveals the d chosen attributes of the signer. This way, the converted signature reveals only the d attributes of ID that are common with ID' chosen by the signer at the time of conversion. This property is defined as *weak signer-attribute privacy* and it is achieved by the following algorithms for our setting.

- **Convert**: Given the public parameters of the fuzzy IBS, a message signature pair (m, σ), and an identity ID' (i.e. the verification attribute set), the signature holder generates a converted signature $\tilde{\sigma}$ on the message.
- **CvtVerify**: An algorithm run by the verifier that given an identity ID' (i.e. the verification attribute set), a message converted-signature pair $(m, \tilde{\sigma})$ and the public parameters, returns a bit b. Here $b = 1$ means that $\tilde{\sigma}$ is a valid converted signature by a signer who has at least d of the attributes in ID', namely $|ID \cap ID'| \geq d$.

Weak signer-attribute privacy ensures that only the d attributes of the signer that are chosen by the signature holder are revealed to the verifier given a converted signature.

In addition, the authors of [Shahandashti and Safavi-Naini, 2009] define the full signer-attribute privacy, which guarantees that the verifier learns nothing more than the fact that $|ID \cap ID'| \geq d$ by combining the converted signature with an interactive verification protocol, which is a zero knowledge proof of knowledge of a valid converted signature with respect to the public inputs. Attribute privacy is obtained by using an interactive verification protocol **iVerify**, that allows the signature holder to prove possession of a valid converted signature without revealing the chosen d attributes in common between the signer and the verifier.

5.3 A New Efficient Biometric IBS Scheme

The first idea for an efficient biometric IBS Scheme is to modify the t-ABS scheme of [Shahandashti and Safavi-Naini, 2009] by replacing T with a hash function used as a

random oracle, which will reduce computational overhead in the key generation and verification algorithms dramatically. The same approach was used in [Pirretti et al., 2006] to obtain an efficient Attribute Based Encryption (ABE) scheme in the random oracle model.

Since a new random polynomial is chosen for each private key, the modified t-ABS is secure against collusion attacks. The $n+1$ exponentiations needed to solve the function T in [Shahandashti and Safavi-Naini, 2009], [Yang et al., 2008] have been replaced with a single MapToPoint hash and signatures can contain a variable number of attributes, rather than be required to contain n as in [Shahandashti and Safavi-Naini, 2009], [Yang et al., 2008]. Verification can be optimized to reduce the number of bilinear map operations by bringing the Lagrange coefficients in [Pirretti et al., 2006]. This optimization reduces the number of bilinear map operations from $3d$ to $d+2$ at the expense of increasing the number of exponentiations from d to $3d$, thus the overall speed of verification is improved. The modified t-ABS scheme [Sarier, 2010c] consists of the following phases.

5.3.1 Modified t-ABS

- Setup: Given a security parameter l, the parameters of the scheme are generated as follows.

  1. Generate two cyclic groups $\mathbb{G}$ and $\mathbb{F}$ of prime order $p > 2^l$ and a bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$. Pick a random generator $g \in \mathbb{G}$.
  2. Pick randomly $y \in \mathbb{Z}_p^*$ and $h, g_2 \in \mathbb{G}$ and compute $g_1 = g^y$.

  The public parameters are (g, g_1, g_2, h) and the master secret key is y.

- Extract: Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ be a collision resistant hash function and let $T : \mathbb{Z}_p \rightarrow \mathbb{G}$ be a MapToPoint hash function modeled as a random oracle. Let Γ be the set defined as $\Gamma = \bigcup_{\mu \in ID} H(\mu)$. A new random degree $d-1$ polynomial $q(\cdot)$ over $\mathbb{Z}_p$ is selected such that $q(0) = y$ and $\forall i \in \Gamma$, a random r_i is chosen and $D_i^{ID} = (g^{q(\mu_i)} T(\mu_i)^{r_i}, g^{r_i})$ for each $\mu_i \in \Gamma$

- Sign: Given a message $m \in M$ and D^{ID}, the following steps are performed.

  1. Pick a random $s_i \in \mathbb{Z}_p$ for $1 \leq i \leq n$
  2. Compute $\sigma_{1i} = g^{q(\mu_i)} T(\mu_i)^{r_i} (g_1^m \cdot h)^{s_i}$, $\sigma_{2i} = g^{r_i}$, $\sigma_{3i} = g^{s_i}$ for each $1 \leq i \leq n$.

  The signature on the message m for identity Γ is $\sigma = (\sigma_{1i}, \sigma_{2i}, \sigma_{3i})$ for $1 \leq i \leq n$.

- Verify: Given σ, m and Γ', choose an arbitrary set $S \subseteq \Gamma \cap \Gamma'$ such that $|S| = d$ and check

$$\hat{e}(g_2, g_1) = \prod_{\mu_i \in S} \left(\frac{\hat{e}(\sigma_{1i}, g)}{\hat{e}(T(\mu_i), \sigma_{2i})\hat{e}(g_1^m \cdot h, \sigma_{3i})}\right)^{\Delta_{\mu_i, S}(0)}$$

The modified t-ABS scheme satisfies both weak signer-attribute and full signer-attribute privacy if the additional protocols for signature conversion and interactive verification are applied. The reader is referred to [Shahandashti and Safavi-Naini, 2009] for the details of this application.

The main disadvantage of the modified t-ABS is the use of a MapToPoint hash function, which converts a user's identity to a point on the underlying elliptic curve in IBE schemes. Thus, our new biometric IBS scheme uses the Sakai Kasahara Key Construction [Sakai and Kasahara, 2003] for the generation of the private keys. This way, the problems stated above for the modified t-ABS are prevented and better performance is obtained due to the use of an ordinary hash function instead of MapToPoint hash function, which is called n times for the key generation and verification algorithms, respectively. Besides, the total number of exponentiations and bilinear pairings required for the remaining phases are also reduced. Finally, the size of the public parameters and the signature is also much shorter compared to the fuzzy IBS scheme of [Yang et al., 2008], [Shahandashti and Safavi-Naini, 2009]. The details of the new scheme [Sarier, 2010c] is presented as follows.

5.3.2 Our Efficient Biometric IBS Scheme

- Setup: Given a security parameter l, the parameters of the scheme are generated as below.

  1. Generate three cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{F}$ of prime order $p > 2^l$ and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{F}$. Pick a random generator $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ such that $\psi(g_2) = g_1$.
  2. Pick random $x, y \in \mathbb{Z}_p^*$, compute $P_{pub} = g_2^x \in \mathbb{G}_2$ and $\kappa = \hat{e}(g_1, g_2)^y$.
  3. Pick two cryptographic hash functions $H_1 : \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*$ and $H_2 : \{0,1\}^{k_1} \times \mathbb{F} \rightarrow \mathbb{Z}_p^*$.

  The message space is $M = \{0,1\}^{k_1}$. The master public key is $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{F}, \psi, \hat{e}, g_1, g_2, P_{pub}, \kappa, H_1, H_2)$ and the master secret key is $ms = x, y$.

- Extract: First, the set of biometric attributes $ID = \{\mu_1, ..., \mu_n\}$ of the signer are obtained from the raw biometric information as in [Baek et al., 2007], [Sahai and Waters, 2005]. Next, the PKG picks a random polynomial $q(\cdot)$ of degree $d-1$ over $\mathbb{Z}_p$ such that $q(0) = y$ and returns $D_i^{ID} = g_1^{q(\mu_i)/t_i}$ for each $\mu_i \in ID$. Here $t_i = x + H_1(\mu_i)$.

- Sign: Given a message $m \in M$ and D^{ID}, the following steps are performed.

  1. Pick a random $z \in \mathbb{Z}_p^*$ and compute $h = H_2(m, \kappa^z) = H_2(m, r)$
  2. $\sigma_i = (D_i^{ID})^{z+h}$ for each $\mu_i \in ID$.

  The signature on the message m for identity ID is $\sigma = (\Sigma, h)$, where $\Sigma = \{\sigma_i : \mu_i \in ID\}$.

- Verify: Given σ, m and ID', choose an arbitrary set $S \subseteq ID \cap ID'$ such that $|S| = d$ and check $h = H_2(m, r')$ by computing

$$\begin{aligned} r' &= \big[\prod_{\mu_i \in S} \hat{e}(\sigma_i, P_{pub} \cdot g_2^{H_1(\mu_i)})^{\Delta_{\mu_i,S}(0)}) \big] \kappa^{-h} \\ &= \big[\prod_{\mu_i \in S} \hat{e}((D_i^{ID})^{z+h}, g_2^{t_i})^{\Delta_{\mu_i,S}(0)}) \big] \kappa^{-h} \\ &= \big[\prod_{\mu_i \in S} \hat{e}(g_1^{q(\mu_i)(z+h)}, g_2)^{\Delta_{\mu_i,S}(0)}) \big] \kappa^{-h} \\ &= \hat{e}(g_1^{y(z+h)}, g_2) \kappa^{-h} \\ &= \kappa^z \end{aligned}$$

Here, the polynomial $q(\cdot)$ of degree $d-1$ is interpolated using d points by polynomial interpolation in the exponents using Shamir's secret sharing method [Shamir, 1979].

Theorem 5.1. *Suppose the hash functions H_1, H_2 are random oracles and there exists an adaptively chosen message and given identity attacker A that produces a forgery within a time t and with probability $\epsilon \geq 10(q_s+1)(q_s+q_2)/2^l$ by making q_1, q_2 random oracle queries, and q_s signature queries. Then there exists an algorithm B that solves the k-DHI problem for $k = q_1$ in an expected time $t' \leq 120686 q_2(t + O(q_s\tau_p))/(\epsilon(1 - k/2^l)) + O(k^2\tau_{mult})$ where τ_{mult} and τ_p respectively denote the cost of a scalar multiplication in $\mathbb{G}_2$ and the required time for a pairing evaluation.*

Proof. Assume that a polynomial time attacker A produces a forgery, then using A, we show that one can construct an attacker (i.e. a simulator) B solving the k-DHI problem. Suppose that B is given the k-DHI problem $(g_1, g_2, g_2^x, g_2^{x^2}, ..., g_2^{x^k})$, B will compute $g_1^{1/x}$ using A as follows.

- **Phase 1**: A declares the challenge identity $ID^* = \{\mu_1, ...\mu_n\}$.
- **Phase 2**: B picks a random feature $\mu^* \in ID^*$ and simulates the public parameters for A. The following three steps are identical to [Chen et al., 2006].

  1. B selects $h_0, ..., h_{k-1} \in \mathbb{Z}_p^*$ and sets $f(z) = \prod_{j=1}^{k-1}(z + h_j)$, which could be written as $f(z) = \sum_{j=0}^{k-1} c_j z^j$. The constant term c_0 is non-zero because $h_j \neq 0$ and c_j are computable from h_j. Here, h_0 denotes the hash value of the challenge attribute $\mu^* \in ID^*$, where μ^* is picked at random by B.

  2. B computes $p_2 = \prod_{j=0}^{k-1}(g_2^{x^j})^{c_j} = g_2^{f(x)} \in \mathbb{G}_2$ and $p_1 = \psi(p_2) = g_1^{f(x)} \in \mathbb{G}_1$. Next, $p_2^x = g_2^{xf(x)} = \prod_{j=0}^{k-1}(g_2^{x^{j+1}})^{c_j}$ and $p_1^x = \psi(p_2^x)$. The public key is fixed as $P_{pub} \in \mathbb{G}_2 = p_2^{x-h_0}$. If $p_2 = 1$, then $x = -h_j$ for some j, then k-DHI problem could be solved directly [Chen et al., 2006].

  3. B computes $f_j(z) = \frac{f(z)}{z+h_j} = \sum_{v=0}^{k-2} d_{j,v} z^v$ for $1 \leq j < k$ and $p_1^{1/(x+h_j)} = g_1^{f_j(x)} = \prod_{v=0}^{k-2} \psi((g_2^{x^v}))^{d_{j,v}}$.
  4. Finally, B computes $p_1^{x/(x+h_j)} = g_1^{xf_j(x)} = \prod_{v=0}^{k-2} \psi((g_2^{x^{v+1}}))^{d_{j,v}}$. This way, the signature queries can be simulated for any identity chosen by A.

  B picks a random $y \in \mathbb{Z}_p^*$ to compute $\kappa = \hat{e}(p_1, p_2)^y$ and returns A the public parameters $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{F}, \psi, \hat{e}, p_1, p_2, P_{pub}, \kappa, H_1, H_2)$, where H_1, H_2 are random oracles controlled by B as follows.

- **Phase 3:** The reduction B continues this phase with the simulation of the oracles that A has access to.

  H_1-queries: For a query on μ_i,

  1. If $\mu_i \in ID^*$ and $\mu_i = \mu^*$, return h_0 and add $\langle \mu^*, h_0, \perp \rangle$ to H_1List.
  2. Else return $h_i + h_0$, add the tuple $\langle \mu_i, h_i + h_0, p_1^{1/(x+h_i)} \rangle$ to H_1List.

  Key extraction queries: Upon receiving a query for ID such that $|ID \cap ID^*| < d$, we first define three sets θ, θ', S: The first set is $\theta = ID \cap ID^*$, next denote with θ'

any set such that $\theta \subseteq \theta' \subseteq ID$ and $|\theta'| = d-1$, and finally, $S = \theta' \cap \{0\}$. In order to understand why we define these sets and the intuition of this simulation, we refer the reader to section 6.2.3 of the next chapter, which summarizes in detail the small universe construction of the first fuzzy IBE system in the literature. If the reader is familiar with the topic and this simulation of key extraction phase, he may skip this summary presented in section 6.2.3 of chapter 6.

Next, we define the decryption key components, D_i^{ID}, for $\mu_i \in \theta'$ as:

If $\mu_i \in \theta$: $D_{\mu_i}^{ID} = p_1^{s_i}$ where s_i is chosen randomly in $\mathbb{Z}_p$.

If $\mu_i \in \theta' - \theta$: $D_{\mu_i}^{ID} = p_1^{\lambda_i/(x+h_i)}$ where λ_i is chosen randomly in $\mathbb{Z}_p$.

The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(0) = y$.

For $\mu_i \in \theta$ we have $q(\mu_i) = xs_i$ and for $\mu_i \in \theta' - \theta$ we have $q(\mu_i) = \lambda_i$.

The simulator B can calculate the other D_i^{ID} values where $\mu_i \notin \theta'$ since the simulator knows the values of $p_1^{1/(x+h_i)}$, $p_1^{x/(x+h_i)}$ and y. The simulator makes the assignments for the final case $\mu_i \notin \theta'$ as:

$$D_{\mu_i}^{ID} = (\prod_{\mu_j \in \theta} p_1^{xs_j \Delta_{\mu_j,S}(\mu_i)/(x+h_i)})(\prod_{\mu_j \in \theta'-\theta} p_1^{\lambda_j \Delta_{\mu_j,S}(\mu_i)/(x+h_i)}) p_1^{y\Delta_{0,S}(\mu_i)/(x+h_i)}$$

Using interpolation the simulator is able to calculate $D_{\mu_i} = p_1^{q(\mu_i)/(x+h_i)}$ for $\mu_i \notin \theta'$, where $q(x)$ was implicitly defined by the random assignment of the other $d-1$ variables $D_{\mu_i} \in \theta'$ and the variable y. Therefore, the simulator is able to construct a private key for the identity ID. Furthermore, the distribution of the private key for ID is identical to that of the original scheme.

Signature queries: For a query on a message-identity pair (m, ID),

1. If $|ID \cap ID^*| \geq d$, for the first query on the challenge identity (or a similar identity ID), B picks at random a $d-1$ degree polynomial $q(\cdot)$ such that $q(0) = y$. At each new query with $|ID \cap ID^*| \geq d$, B picks randomly $a, h \in \mathbb{Z}_p^*$, computes $r = \hat{e}(p_1^{ax} \cdot p_1^{-h}, p_2)^y = \hat{e}(p_1^{ax-h}, p_2)^y$ and backpatches to define the value $H_2(m, r)$ as h. Next, he computes $\sigma_i = p_1^{axq(\mu_i)/(x+h_i)}$ for each $\mu_i \neq \mu^*$. For the feature $\mu_i = \mu^*$, he computes $\sigma_{\mu^*} = p_1^{aq(\mu^*)}$. Lastly, B returns $\sigma = (\Sigma, h)$ to A, where $\Sigma = \{\sigma_i : \mu_i \in ID\}$.
2. Else, B picks randomly $z, h \in \mathbb{Z}_p^*$, computes $r = \hat{e}(p_1^z, p_2)^y$ and backpatches to define $H_2(m, r)$ as h. Finally, B obtains the corresponding private key components by simulating the key extraction oracle and returns $(D_{\mu_i}^{ID})^{z+h}$ for each $\mu_i \in ID$.

B aborts in the unlikely event that $H_2(m,r)$ is already defined.

Remark 5.2. *The simulation of the signature queries on any ID such that $|ID \cap ID^*| > d$ is correct since given (σ, m), A chooses an arbitrary set $S \subseteq ID$ such that $|S| = d$ and checks $h = H_2(m,r)$ as below. Lets assume that $\mu^* \in S$,*

$$
\begin{aligned}
r &= \big[\prod_{\mu_i \in S} \hat{e}(\sigma_i, P_{pub} \cdot p_2^{H_1(\mu_i)})^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu^* \neq \mu_i \in S} \hat{e}(p_1^{axq(\mu_i)/(x+h_i)}, p_2^{x-h_0} \cdot p_2^{H_1(\mu_i)}) \cdot \hat{e}(\sigma_{\mu^*}, p_2^{x-h_0} \cdot p_2^{H_1(\mu^*)}))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu^* \neq \mu_i \in S} \hat{e}(p_1^{axq(\mu_i)/(x+h_i)}, p_2^{x-h_0} \cdot p_2^{h_i+h_0}) \cdot \hat{e}(\sigma_{\mu^*}, p_2^{x-h_0} \cdot p_2^{h_0}))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu^* \neq \mu_i \in S} \hat{e}(p_1^{axq(\mu_i)/(x+h_i)}, p_2^{x+h_i}) \cdot \hat{e}(p_1^{aq(\mu^*)}, p_2^{x}))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu^* \neq \mu_i \in S} \hat{e}(p_1^{axq(\mu_i)}, p_2) \cdot \hat{e}(p_1^{aq(\mu^*)}, p_2^{x}))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu_i \in S} \hat{e}(p_1^{axq(\mu_i)}, p_2))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \hat{e}(p_1^{axy}, p_2)\hat{e}(p_1, p_2)^{-hy} \\
&= \hat{e}(p_1^{ax-h}, p_2)^{y}
\end{aligned}
$$

Thus, the simulation of the signature queries on any ID is correct since given (σ, m), A chooses an arbitrary set $S \subseteq ID$ such that $|S| = d$ and checks $h = H_2(m,r)$ by computing the same way as above.

$$
\begin{aligned}
r &= \big[\prod_{\mu_i \in S} \hat{e}(\sigma_i, P_{pub} \cdot p_2^{H_1(\mu_i)})^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu_i \in S} \hat{e}(D_{\mu_i}^{z+h}, p_2^{x-h_0} \cdot p_2^{h_i+h_0}))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \big[(\prod_{\mu_i \in S} \hat{e}(p_1^{q(\mu_i)(z+h)}, p_2))^{\Delta_{\mu_i,S}(0)}\big]\kappa^{-h} \\
&= \hat{e}(p_1^{y(z+h)}, p_2)\hat{e}(p_1, p_2)^{-hy} \\
&= \hat{e}(p_1, p_2)^{yz} = \kappa^{z}
\end{aligned}
$$

- **Phase 4**: After the queries to the random oracles, the adversary has to forge a signature (m, r, σ) on the exact challenge identity $ID^* = (\mu_1, .., \mu^*, ..\mu_n)$.

Next, the forking lemma [Pointcheval and Stern, 2000], [Bellare and Neven, 2006] is applied on (m, r, h, Σ). If the triples (r, h, Σ) can be simulated without knowing the private key components of ID^*, then there exists a Turing machine B' that replays a sufficient number of times on the input (P_{pub}, ID^*) to obtain two valid signatures (m^*, r, h', Σ') and (m^*, r, h'', Σ'') such that $h' \neq h''$ for the same message m^* and commitment r. If both forgeries satisfy the verification equation for all the sets $S \subseteq ID^*$ such that $|S| = d$ and $\mu^* \in S$, namely,

$$\begin{aligned} r &= \big[\prod_{\mu_i \in S} (\hat{e}(\sigma'_i, P_{pub} \cdot p_2^{H_1(\mu_i)})^{\Delta_{\mu_i, S}(0)}) \big] \kappa^{-h'} \\ &= \big[\prod_{\mu_i \in S} (\hat{e}(\sigma''_i, P_{pub} \cdot p_2^{H_1(\mu_i)})^{\Delta_{\mu_i, S}(0)}) \big] \kappa^{-h''} \end{aligned}$$

By verifying all the possible combinations for the set S, B is assured that each partial signature σ'_i and σ''_i is valid. B finds the solution to the k-DHI problem from the forgeries associated to the feature $\mu^* \in ID^*$, namely $\sigma'_{\mu^*}, \sigma''_{\mu^*}$.

Then, the computations are performed as in [Barreto et al., 2005],

$$\begin{aligned} &\hat{e}(\sigma'_{\mu^*}, P_{pub} \cdot p_2^{H_1(\mu^*)}) \hat{e}(p_1, p_2)^{-h'} = \hat{e}(\sigma''_{\mu^*}, P_{pub} \cdot p_2^{H_1(\mu^*)}) \hat{e}(p_1, p_2)^{-h''} \\ &\Rightarrow \hat{e}(\sigma'_{\mu^*}, p_2^x) \hat{e}(p_1, p_2)^{-h'} = \hat{e}(\sigma''_{\mu^*}, p_2^x) \hat{e}(p_1, p_2)^{-h''} \\ &\Rightarrow \hat{e}(\sigma'_{\mu^*} / \sigma''_{\mu^*}, p_2^x)^{(h'-h'')^{-1}} = \hat{e}(p_1, p_2) \end{aligned}$$

Similar to the proof in [Barreto et al., 2005], we set $T = p_1^{q(\mu^*)/x} = (\sigma'_{\mu^*} / \sigma''_{\mu^*})^{(h'-h'')^{-1}}$.

Finally, we obtain the solution to the k-DHI problem, namely $g_1^{1/x}$, by computing $(T^{1/q(\mu^*)} / \prod_{j=1}^{k-1} \psi(g_2^{x^{j-1}})^{c_j})^{1/c_0}$, since

$$T^{1/q(\mu^*)} = p_1^{1/x} = \psi(p_2)^{1/x} = \prod_{j=0}^{k-1} (\psi(g_2^{x^{j-1}}))^{c_j} = \psi(g_2)^{c_0/x} \cdot \prod_{j=1}^{k-1} \psi(g_2^{x^{j-1}})^{c_j} \tag{5.1}$$

The computation for the advantage and the running time is identical to the computation described in [Pointcheval and Stern, 2000], [Barreto et al., 2005]. The only difference in the computation of the running time is the removal of the factor q_1 since the security model of fuzzy IBS (and t-ABS) is based on given identity attack model instead of adaptive chosen identity model as in standard IBS schemes. □

5.3.3 A Stronger Security Model

We prove the security of our new scheme in the framework of a stronger security model [Sarier, 2010c] compared to exisiting adversarial models for fuzzy IBS and t-ABS, which basically simulates the leakage of partial secret key components of the challenge identity. A biometric IBS scheme is selectively unforgeable under adaptive chosen message and given identity attacks (SUF-FIBS-CMA*) if no probabilistic polynomial time (PPT) adversary A has a non-negligible advantage in the game between a challenger and the adversary as follows.

- **Phase 1**: The adversary A declares the challenge identity $ID^* = \{\mu_1^*, ..., \mu_n^*\}$.
- **Phase 2**: The challenger runs the Setup algorithm and returns the system parameters to A.
- **Phase 3**: The challenger returns the private key components of any identity such that $|ID' \cap ID^*| < d$ except for a particular feature $\mu^* \in ID^*$ (if $\mu^* \in ID'$). For the challenge identity, A obtains any $d-1$ private key components he selects other than $\mu^* \in ID^*$. The adversary issues signature queries for any identity and the challenger returns the partial signatures for the all components of the queried identity.
- **Phase 4**: A outputs a forgery (ID^*, m^*, σ^*), where A does not make a signature query on (m^*, σ^*) for ID^*.

The success of A is defined as $\text{Succ}_A^{\text{SUF-FIBS-CMA}^*}(l) = \Pr[\mathsf{Verify}(ID^*, m^*, \sigma^*) = 1]$.

This security model is stronger than the model of fuzzy IBS since the adversary has access to private key components of any ID including the case of $|ID \cap ID^*| \geq d$, as opposed to the security model of [Yang et al., 2008], [Shahandashti and Safavi-Naini, 2009]. We can further relax the requirement for the forgery on the exact challenge identity ID^* by allowing the adversary to output a forgery on the similar identity ID^+ such that $|ID^+ \cap ID^*| \geq d$ and $\mu^* \in ID^+$. We should note that in current fuzzy IBS model, the forgery should be on the exact challenge identity.

The proof of our new scheme in the framework of this stronger model slightly differs from the above proof.

Proof. Assume that a polynomial time attacker A produces a forgery, then using A, we show that one can construct an attacker B solving the k-DHI problem.

Suppose that B is given the k-DHI problem $(g_1, g_2, g_2^x, g_2^{x^2}, ..., g_2^{x^k})$, B will compute $g_1^{1/x}$ using A as follows.

- **Phase 1**: A declares the challenge identity $ID^* = \{\mu_1, ...\mu_n\}$.

- **Phase 2**: B picks a random feature $\mu^* \in ID^*$ and simulates the public parameters for A as before.

  B picks a random $y \in \mathbb{Z}_p^*$ to compute $\kappa = \hat{e}(p_1, p_2)^y$ and returns A the public parameters $(p_1, p_2, \hat{e}, \psi, \mathbb{G}_1, \mathbb{G}_2, \mathbb{F}, P_{pub}, \kappa, H_1, H_2)$, where H_1, H_2 are random oracles controlled by B as follows.

- **Phase 3**: H_1-queries: Identical to the previous proof.

  Key extraction queries: Upon receiving a query for ID, for every $\mu_i \neq \mu^* \in ID$, run the H_1-oracle simulator and obtain $\langle \mu_i, h_i + h_0, p_1^{1/(x+h_i)} \rangle$ from H_1List. Pick a random $d-1$ degree polynomial $q(\cdot)$ such that $q(0) = y$ and return $D_{\mu_i} = p_1^{q(\mu_i)/(x+h_i)}$ for each $\mu_i \in ID$ except for $\mu^* \in ID$ (if $\mu^* \in ID$). For the challenge identity ID^*, a random $d-1$ degree polynomial $q(\cdot)$ such that $q(0) = y$ is picked and A is given the $d-1$ private key components that A selects, namely $D_{\mu_i} = p_1^{q(\mu_i)/(x+h_i)}$ except for the feature μ^*.

  Signature queries: For a query on a message-identity pair (m, ID),

  1. If $|ID \cap ID^*| \geq d$ and $\mu^* \in ID$, B picks randomly $a, h \in \mathbb{Z}_p^*$, computes $r = \hat{e}(p_1^{ax} \cdot p_1^{-h}, p_2)^y = \hat{e}(p_1^{ax-h}, p_2)^y$ and backpatches to define the value $H_2(m, r)$ as h. Next, B obtains the corresponding private key components by simulating the key extraction oracle on ID and computes $\sigma_i = p_1^{axq(\mu_i)/(x+h_i)}$ for each $\mu_i \neq \mu^*$. For the feature $\mu_i = \mu^*$, he computes $\sigma_{\mu^*} = p_1^{aq(\mu^*)}$. Lastly, B returns $\sigma = (\Sigma, h)$ to A, where $\Sigma = \{\sigma_i : \mu_i \in ID\}$.
  2. Else if $|ID \cap ID^*| < d$ and $\mu^* \in ID$, step 1 is repeated.
  3. Else, B picks randomly $z, h \in \mathbb{Z}_p^*$, computes $r = \hat{e}(p_1^z, p_2)^y$ and backpatches to define $H_2(m, r)$ as h. Finally, B obtains the corresponding private key components by simulating the key extraction oracle and returns $(D_{\mu_i}^{ID})^{z+h}$ for each $\mu_i \in ID$.

  B aborts in the unlikely event that $H_2(m, r)$ is already defined. The simulation of the signature queries on any ID with $\mu^* \in ID$ is correct as before.

- **Phase 4:** After the queries to the random oracles, the adversary has to forge a signature (m, r, σ) on the exact challenge identity $ID^* = (\mu_1, .., \mu^*, ..\mu_n)$.

Next, the forking lemma [Pointcheval and Stern, 2000], [Bellare and Neven, 2006] is applied on (m, r, h, Σ). If the triples (r, h, Σ) can be simulated without knowing the private key components of ID^*, then there exists a Turing machine B' that replays a sufficient number of times on the input (P_{pub}, ID^*) to obtain two valid signatures

(m^*, r, h', Σ') and (m^*, r, h'', Σ'') such that $h' \neq h''$ for the same message m^* and commitment r. If both forgeries satisfy the verification equation for all the sets $S \subseteq ID^*$ such that $|S| = d$ and $\mu^* \in S$, as before.

By verifying all the possible combinations for the set S, B is assured that each partial signature σ_i' and σ_i'' is valid. B finds the solution to the k-DHI problem from the forgeries associated to the feature $\mu^* \in ID^*$, namely $\sigma'_{\mu^*}, \sigma''_{\mu^*}$. Then, the computations are performed as before. Again, the solution to the k-DHI problem, $g_1^{1/x}$ is obtained by outputting $(T^{1/q(\mu^*)} / \prod_{j=1}^{k-1} \psi(g_2^{x^{j-1}})^{c_j})^{1/c_0}$ due to the equation 5.1.

The computation for the advantage and the running time is identical to the computation described in [Pointcheval and Stern, 2000], [Barreto et al., 2005]. The only difference in the computation of the running time is the removal of the factor q_1 since the security model of fuzzy IBS (and t-ABS) is based on given identity attack model instead of adaptive chosen identity model as in standard IBS schemes. □

We can further relax the requirement for the forgery on the exact challenge identity ID^* by allowing the adversary to output a forgery on the similar identity ID^+ such that $|ID^+ \cap ID^*| \geq d$ and $\mu^* \in ID^+$. As long as both forgeries satisfy the verification equation for all the sets $S \subseteq ID^*$ (or $S \subseteq ID^+$) such that $|S| = d$ and $\mu^* \in S$, B is assured that each partial signature σ_i' and σ_i'' is valid. Since B finds the solution to the k-DHI problem from the forgeries associated to the feature $\mu^* \in ID^*$, namely $\sigma'_{\mu^*}, \sigma''_{\mu^*}$, we can relax the requirement for the forgery on the exact challenge identity ID^*. We should note that in current fuzzy IBS model, the forgery should be on the exact challenge identity.

5.3.4 Weak Signer-Attribute Privacy

In the t-ABS scheme of [Shahandashti and Safavi-Naini, 2009], the verifier is able to identify which d common attributes are used in the generation of the converted signature, since $ID' \setminus S$ components of the converted signature are publicly simulatable. If only weak signer-attribute privacy is considered, more efficient **Convert** and **CvtVerify** algorithms could be designed by removing the bilinear pairings and exponentiations computed for the *dummy* components, namely $ID' \setminus S$. For applications that require full signer-attribute privacy, our "modified t-ABS" scheme with the additional protocols presented in the original t-ABS paper could be a more efficient solution than t-ABS.

- **Convert**: On input the public parameters of the fuzzy IBS, the message signature pair (m, σ) computed by the signer, and the identity ID', the signature holder selects $S \subseteq ID \cap ID'$ such that $|S| = d$ and sets $\forall \mu_i \in S$, $\tilde{\sigma_i} = \sigma_i$. Next, $\forall \mu_i \in ID' \setminus S$, the signer sets $\tilde{\sigma_i} = \perp$ and returns the verifier $(m, \tilde{\sigma})$.

- **CvtVerify**: Given an identity ID', a message converted-signature pair $(m, \tilde{\sigma})$ and the public parameters, the verifier can easily identify the d common attributes and verifies the signature as before.

5.3.5 Some arguments against the architecture of t-ABS

We present our arguments against the architecture of the t-ABS scheme of [Shahandashti and Safavi-Naini, 2009], which uses an additional party called "signature holder" that has access to the (biometric) atributes ID of the signer and the verification attribute set ID' that is known by the verifier to be used during the verification of the signature. By knowing the two similar (biometric) attributes ID and ID', the signature holder determines the common biometric attributes, i.e. performs an error correction based on set difference to select the d signature components that will be used in the actual verification, and then communicaties with the verifier to send a converted signature generated by the common attributes, i.e. the d signature components. This new model is designed in order to achieve weak and/or full attribute privacy, however, the model is against the definition of fuzzy IBS, which does not allow a third party to compute an error-corrected converted signature. Instead, the verifier should be able to verify the fuzzy IBS even if the signature is generated by using an attribute set similar to the one the verifier has. Thus, this t-ABS scheme cannot be considered as a fuzzy IBS scheme with the model described in [Shahandashti and Safavi-Naini, 2009]. Without this new model, t-ABS scheme is actually identitical to the fuzzy IBS scheme presented in 2008 [Yang et al., 2008], where the only difference is the replacement of the Water's function $W(\cdot)$ on the message m, i.e. $W(m) = h \prod h_i^{m_i}$, with the value g_1^m, where m_i denotes the i^{th} bit of m and h_i are random elements from $\mathbb{G}$ defined as part of the public parameters of the fuzzy IBS of [Yang et al., 2008].

5.4 Efficiency Discussions and Comparison

In this section, we compare different fuzzy IBS and ABS schemes applicable for biometric identities. For simplicity of the comparison, ψ is taken as the identity map (i.e. $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$) and the computational cost for multiplication in $\mathbb{G}$ is omitted. All the computations are performed according to the optimization introduced in [Pirretti et al., 2006], where the dominant operations are considered as bilinear pairings followed by exponentiations. The abbreviations used in figure 5.1 denote the following: $|B|$ is the bit-length of an element in set (or group) B; n is the number of features in ID; T_e is the computation time for a single exponentiation in $\mathbb{G}$; T'_e is the computation time for a single exponentiation in $\mathbb{F}$; T_H is the computation time for a MapToPoint hash function; T_i is the computation time for a single inverse operation in $\mathbb{Z}_p$; T_p is the

computation time for a single pairing operation; T_i' the computation time for a single inverse operation in $\mathbb{F}$; d is the error tolerance parameter; k_1 the size of the message; k_2 output size of the H_2 hash function.

Figure 5.1: Comparison of error tolerant IBS schemes

	fuzzy IBS	t-ABS	Modified t-ABS	Our Scheme
Size of public parameters	$(n+k_1+4)\|\mathbb{G}\|+\|\mathbb{F}\|$	$(n+5)\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$2\|\mathbb{G}\|+\|\mathbb{F}\|$
Size of D^{ID}	$2n\|\mathbb{G}\|$	$2n\|\mathbb{G}\|$	$2n\|\mathbb{G}\|$	$n\|\mathbb{G}\|$
Size of σ	$3n\|\mathbb{G}\|$	$3n\|\mathbb{G}\|$	$3n\|\mathbb{G}\|$	$n\|\mathbb{G}\|+k_2$
Cost of Key Generation	$n(n+4)T_e$	$n(n+4)T_e$	$n(3T_e+T_H)$	$n(T_i+T_e)$
Cost of Sign	$(k_1+2n)T_e$	$2nT_e$	$2nT_e$	nT_e+T_e'
Cost of Verify	$d((n+4)T_e+T_p)$ $k_1T_e+2T_p$	$d((n+4)T_e+T_p)$ $2T_p+2T_i'$	$d(3T_e+T_p+T_H)$ $2T_p+2T_i'$	$d(T_p+T_e)$ $+T_e'$
Security Model	Standard Model	Standard Model	ROM	ROM

5.5 Conclusion

In this chapter, we review the existing IBS schemes applicable for biometric identities and propose a more efficient biometric IBS scheme by employing the Sakai Kasahara Key Construction in the ROM. In addition, our scheme could function as a practical fuzzy IBS or t-ABS scheme with the claim that the new scheme is faster than all known pairing-based IBS methods for fuzzy identities as it is based on the currently the most efficient pairing-based IBS scheme. Besides, examining the signer-attribute privacy for fuzzy IBS and our scheme without requiring an intermediate (error-correcting) party such as the "signature holder" of t-ABS could be an interesting future work since the user may use his biometrics in other applications such as biometric encryption or authentication systems, where the latter assumes the privacy of the identity-biometrics relationship rather than the secrecy of the biometrics of the user.

Chapter 6

Biometric Identity Based Encryption

In this chapter, we present two efficient biometric Identity Based Encryption (IBE) schemes based on pairings following the security model of fuzzy IBE systems. The first construction works for biometrics or in general for attributes that can be ordered/grouped, whereas the second construction called as BIO-IBE is suitable for any type of biometrics. Our designs are based on the Sakai Kasahara Key Construction and the security reduction is presented for large universe of biometric attributes in the Random Oracle Model (ROM) and for small universe in the standard model. We will show that for the large universe of attributes, BIO-IBE is more efficient compared to other fuzzy IBE schemes and for the small universe, it is more efficient compared to the small universe construction of [Sahai and Waters, 2005]. Similar to our fuzzy IBS system in the previous chapter, we describe a stronger security model and prove the security of BIO-IBE based on this stronger model that basically simulates the leakage of partial secret key components of the challenge identity. This property is not considered in the current security model of fuzzy IBE, which return to the adversary only the private key components belonging to any identity other than (i.e. not similar to) the challenge identity. However, in our stronger security model, we allow the adversary to query for some of the private key components belonging to the challenge identity. Besides, BIO-IBE is the first biometric IBE scheme that allows for the use of multi-modal biometrics for defining the identity of the user. Specifically, we introduce a new method for key generation, where a unique biometric identity string ID obtained from the biometric attributes is used instead of picking a different polynomial for each user as in other fuzzy IBE schemes. At the key generation phase, we combine the master secret key, features of any biometric trait and this unique ID to bind the private key components to the user and thus, avoid collision attacks. This new combination does not only prevent this attack, but also has the advantage of better accuracy/identification compared to the use of uni-modal biometrics as in current fuzzy IBE. From the efficiency point of view, the fuzzy extraction of ID is performed only by the sender, is independent of

the message, and hence can be done once and for all. Finally, our new method can be applied in other IBE systems that are not based on pairing based cryptography, as we will see in the next chapter. The contributions of this chapter are based on the papers [Sarier, 2008] and [Sarier, 2011b].

6.1 Introduction

In Eurocrypt'05, Sahai and Waters proposed a new Identity Based Encryption (IBE) system called fuzzy IBE that uses biometric attributes as the identity instead of an arbitrary string like an email address. Before this application, other combinations of biometrics and cryptography have been discussed in many research papers, which mainly focused on the derivation of a secret key from a biometric trait. Clearly, biometrics is assumed as secret data in these applications. One can argue whether this assumption is realistic. After all, biometric information can be easily captured and can be used to impersonate a user. Fingerprints, for example, are left everywhere and can be easily lifted. But since biometrics can identify a person uniquely, it makes sense to use them as the public key in an identity-based encryption scheme. The problem with this approach is that biometrics usually consist of noisy data, i.e. two measures w and w' of the same biometric are not completely the same. However, the main feature of fuzzy IBE is the construction of the secret key based on the biometric data of the user which can decrypt a ciphertext encrypted with a slightly different measurement of the same biometrics. Specifically, fuzzy IBE allows for error tolerance in the decryption stage, where a ciphertext encrypted with the biometrics w could be decrypted by the receiver using the private key corresponding to the biometrics w', provided that w and w' are within a certain distance of each other according to the 'set overlap' distance metric. This is in contrast to regular IBE schemes, which view the identity of a person as a unique string like an e-mail address, thus they are not suitable for error-prone identities. Thus, fuzzy IBE combines the advantages of IBE with using biometrics as an identity, where IBE avoids the need for an online Public Key Infrastructure (PKI), which is the most inefficient and costly part of public key encryption. The use of biometrics as the identity in the framework of IBE simplifies the process of key generation at the Private Key Generator (PKG). Since biometric information is unique, unforgettable and non-transferable, the user only needs to provide his biometrics at the PKG to obtain his secret key instead of presenting special documents and credentials to convince the PKG about his identity. Also, biometrics is attached to the user, hence the public key of the user is always with him to be used for encryption during an ad hoc meeting. Finally, biometric data could be easily integrated with fuzzy IBE due to its error tolerance property, which is required for the noisy nature of biometrics. Besides, fuzzy IBE could be applied in the context of Attribute-Based Encryption [Pir-

retti et al., 2006, Sahai and Waters, 2005], where the sender encrypts data using a set of attributes such as {university, faculty, department} and the ciphertext could only be decrypted if the receiver has the secret key associated to all of these attributes or sufficient number of them. In current fuzzy IBE schemes, the private key components are generated by combining the values of a unique polynomial evaluated on each attribute with the master secret key. Besides, the biometrics is considered as public information, hence the compromise of the biometrics does not affect the security of the system.

6.1.1 Motivation and Contributions

Using biometrics in Identity-Based Encryption

Using biometric-based identity in an IBE system has a number of important advantages over "standard" IBE that are listed in [Sahai and Waters, 2005] as follows.

- The process of obtaining a secret key from an authority is very natural and straightforward. In standard IBE schemes, a user with a an identity such as an e-mail address 'proves' to the trusted authority that he is indeed entitled to this identity. This will typically involve presenting supplementary documents or credentials. The type of authentication that is necessary is not always clear and robustness of this process is questionable (the supplementary documents themselves could be subject to forgery) [Sahai and Waters, 2005]. Typically, there exists a tradeoff between a system that is expensive in this step and one that is less reliable [Sahai and Waters, 2005]. However, in biometric IBE the user ony presents his biometrics to the trusted authority under the supervision of a well trained operator. If the operator is able to detect imitation attacks, for example playing the recording of a voice, then the security of this phase is only limited by the quality of the biometric technique itself [Sahai and Waters, 2005]. Here, our proposal for multi-modal biometric identities prevents even attempts for such impersonation attack as forging two different biometric traits is more difficult compared to the use of one modality. In [Ross and Jain, 2004] multimodal biometric systems are shown to be more reliable due to the presence of multiple, (fairly) independent pieces of evidence. They also deter spoofing since it would be difficult for an impostor to spoof multiple biometric traits of a genuine user simultaneously. This fact also relax the requirement for a well trained supervisor since a challenge response type of mechanism by requesting the user to present a random subset of biometric traits thereby ensuring that a 'live' user is indeed present at the point of data acquisition [Ross and Jain, 2004].
- We emphasize that the biometric measurement for an individual need not be kept secret since it is used as a public key. This assumption is also accepted by

the biometrics community who consider the biometrics as public data, whereas the biometric template that is stored in a database for authentication purposes should be kept private.

- Also, a biometric identity is an inherent trait and will always with a person. Using biometrics in IBE will mean that the person will always have their public key handy. In several situations a user will want to present an encryption key to someone when they are physically present when a user is traveling and another party encrypts an ad-hoc meeting between them [Sahai and Waters, 2005, Baek et al., 2007].

- Finally, using a biometric as an identity has the advantage that identities are unique if the underlying biometric is of a good quality. Some types of standard identities, such as the name "Bob Smith" will clearly not be unique or change owners over time [Sahai and Waters, 2005]. Here, our proposal for multi-modal biometric identities also strengthens this assumption since multi-modal biometrics identifies a person even more than a uni-modal system which is the current default in fuzzy IBE schemes.

In this chapter, we present two efficient biometric Identity Based Encryption (IBE) schemes based on pairings following the security model of fuzzy IBE systems. The first construction works for biometrics or, in general, for attributes that can be ordered/grouped. Let us give this example: Assume the identites are represented as a set of ID ={university, faculty, department, section, division, group, student}. If we preserve this order for the representation of each user than a user with identity ID' = {university, faculty, department, section, 0, 0, student} is able to decrypt a ciphertext encrypted for identity ID, due to the error-tolerance of the system. Similarly, let ID denote the face biometrics of the user, i.e. {mouth, eye, nose, forehead, jaw, eyebrow, cheek}. If we preserve this order for the representation of each user than a user with identity ID' = {mouth, eye$'$, nose, forehead, 0, eyebrow, 0} is able to decrypt a ciphertext encrypted for identity ID. Here, 0 indicates that the feature (or attribute) of a particular region could not be extracted and eye$'$ denotes that the extracted eye feature is slightly different than the actual eye feature of the user as a result of noise.

The second construction called as BIO-IBE is suitable for any type of biometrics including the biometrics represented as an unordered set of features such as fingerprint minutia. The security reduction is presented for large universe of biometric attributes in the Random Oracle Model (ROM) and for small universe in the standard model. For the former case, the security is based on the well-exploited k-BDHI computational problem and for the latter case, it is based on the decisional k-BDHI problem. We will show that for the large universe of attributes, BIO-IBE is more efficient compared to the fuzzy IBE schemes secure in ROM [Pirretti et al., 2006, Baek et al., 2007, van

Liesdonk, 2007] and for the small universe, it is more efficient compared to the small universe construction of [Sahai and Waters, 2005]. Moreover, we describe a stronger security model and prove the security of BIO-IBE based on this stronger model that basically simulates the leakage of partial secret key components of the challenge identity. This property is not considered in the current security model of fuzzy IBE, which return to the adversary only the private key components belonging to any identity other than (i.e. not similar to) the challenge identity. However, in our stronger security model, we allow the adversary to query for some of the private key components belonging to the challenge identity. As different from the fuzzy IBE scheme of [Sahai and Waters, 2005], the ciphertexts can contain a variable number of attributes but the error tolerance parameter is a fixed threshold value as in current systems.

For our new construction, we have three main motivations. To begin with, our new scheme BIO-IBE [Sarier, 2008, 2011b] is the first biometric IBE scheme that allows for the use of multi-modal biometrics for defining the identity of the user. In particular, we mean to indicate that (1) the biometric sources are different; that is, multiple biometric traits are involved such as face + fingerprint (2) Only a single sensor is employed to obtain the raw data from a single biometric trait; this data is then used by multiple matchers, such as a matcher working on minutia-based fingerprint and a matcher working on non-minutia based fingerprint. Multi-modal biometrics overcomes the limitations of uni-modal systems, namely, unacceptable performance and inability to operate on a large user population. First of all, identification using multiple biometrics utilizes information from multiple sensors to increase fault tolerance capability, to reduce uncertainty and noise, and to overcome incompleteness of individual sensors. A multimodal approach can increase the reliability of the decisions made by a biometric system. By using multiple biometric characteristics, the system will be applicable on a larger target population. Finally, a multimodal biometric system is generally more robust to fraudulent technologies, because it is more difficult to forge multiple biometric characteristics than to forge a single biometric characteristic [Ross and Jain, 2004].

How do we implement multi-modal identities in biometric IBE schemes? The answer lies in the structure of the key generation algorithm of BIO-IBE, where a unique biometric identity string ID obtained from the biometric attributes is used instead of picking a different polynomial for each user as in other fuzzy IBE schemes. For the case (1) (i.e. for different biometric sources), the private key components of each user is computed by using a biometric trait such as fingerprint, face, palmprint, etc. combined with the unique biometric identity string fuzzy-extracted from a different biometric trait such as the Iris scan of the user. This combination is used to bind the private key components to that user and thus avoid the collision attacks, which means that different users sharing common biometric attributes with the receiver of the ciphertext cannot decrypt this ciphertext by combining their secret key components associated to these common attributes. In current fuzzy IBE schemes, collision

attacks are prevented by picking a unique polynomial for each user that is evaluated at each biometric feature of the user and that is combined with the master secret key to generate the private key components. In BIO-IBE, the final computed private key components of each user can be thought as a biometric fusion at the feature level. This new combination does not only prevent collision attacks, but also has the advantage of better accuracy/identification compared to the use of uni-modal biometrics as in current fuzzy IBE. Some combinations appear more natural than others: fingerprint + iris, face + iris or face+fingerprint, palmprint + fingerprint etc., which are the main combinations for multimodal biometric identification/sketch/vault based on the fusion of different biometric traits [Rattani et al., 2007, Y. Sutcu and Memon, 2007, Rattani and Tistarelli, 2009, Nandakumar and Jain, 2008]. Hence, BIO-IBE combines any type of biometrics represented as an ordered (such as face) or unordered (such as fingerprint minutia) set of biometric features with another biometric trait that can be represented as a binary string (such as Iris), which is input to a fuzzy extractor to obtain a unique biometric identity string. If we employ the case (2) of multi-modal biometrics, namely a single biometric trait with different feature processing methods, then we can combine (unordered) features from fingerprint minutia with the binary string obtained from vicinity-based fingerprint features described in [Bringer and Despiegel, 2010]. From the efficiency point of view, the fuzzy extraction is performed only by the sender, is independent of the message, and hence can be done once and for all. We should note that the use of multi-biometric based encryption using a fuzzy extractor is claimed to be introduced in 2011 by [Zhang et al., 2011], although the first use of this approach was presented at BIO-IBE in 2008. Finally, our new method for preventing collision attacks can be applied in other IBE systems that are not based on pairing based cryptography, as we will see in the next chapter.

Our second motivation for the new design is to eliminate the requirement of a special hash function called MapToPoint hash function that maps a user's identity to a point on the underlying elliptic curve in IBE schemes. Currently, efficient fuzzy IBE schemes [Pirretti et al., 2006, Baek et al., 2007, van Liesdonk, 2007] employ this special function, which is usually implemented as a probabilistic algorithm and is more expensive than a point scalar multiplication in terms of computation time [Chen and Cheng, 2005, Chen et al., 2006]. This operation is also time consuming and cannot be treated as a conventional hash operation which is commonly ignored in performance evaluation. Besides, as it is noted in [Barreto et al., 2005, Smart and Vercauteren, 2007], it is difficult to find groups as the range of the MapToPoint hash function and to define an efficient isomorphism at the same time.

Apart from the efficiency gain resulting from the replacement of the MapToPoint hash function with an ordinary one, BIO-IBE has a structurally simpler key generation algorithm and provides better efficiency in terms of the key generation and decryption algorithms compared to the existing fuzzy IBE schemes [Pirretti et al., 2006, Baek

et al., 2007, van Liesdonk, 2007] secure in ROM. Specifically, we reduce the number of exponentiations in the group $\mathbb{G}$ from $3n$ as in [Pirretti et al., 2006] (and from $2n$ as in [Baek et al., 2007]) to $n + 2$. Also, the decryption algorithm requires d bilinear pairing computations and d exponentiations, whereas the existing schemes require $d+1$ bilinear pairing computations and $2d$ exponentiations. Here, n denotes the size of the biometric feature set of the user and d denotes the error tolerance parameter.

BIO-IBE [Sarier, 2008] is proven secure based on the standard security model for fuzzy IBE and a stronger security model that we introduce in the ROM. The main difference of our stronger security model is that the adversary is allowed to make private key extraction queries on the challenge identity w^*, where A can obtain $d - 1$ private key components of w^* that A chooses. This new model basically simulates the leakage of partial secret key components of the challenge identity. This property is not considered in the current security model of fuzzy IBE, which return to the adversary only the private key components belonging to any identity other than (i.e. not similar to) the challenge identity. Thus, the adversary A has more power compared to the model defined in [Sahai and Waters, 2005, Baek et al., 2007]. Next, we improve the reduction cost of BIO-IBE by reducing its security to the decisional k-BDHI problem instead of computational k-BDHI problem. We see a tradeoff between the tightness of the reduction cost and the hardness of the underlying problem. Besides, for the small universe construction, BIO-IBE is proven secure in the standard model based on the decisional k-BDHI problem and the size of its public parameters is equal to the size of a standard IBE scheme, whereas the small universe construction of [Sahai and Waters, 2005] has public parameters linear in the size of the (small) Universe of attributes. Here, small universe construction means that the universe of features is defined beforehand, thus we cannot use attributes that were not considered during the setup.

Next, we further improve BIO-IBE by eliminating a Denial of Service (DoS) attack that results from the use of the fuzzy extraction process. In this context, we describe a modified version of BIO-IBE and show that it is immune against this attack due to the signature applied on the public value PAR of the user. To prevent DoS attacks, the modified BIO-IBE [Sarier, 2011b] integrates an efficient IBS scheme into BIO-IBE in order to sign the public value PAR of the receiver during the key generation phase of BIO-IBE. Besides, the encryption phase is also modified by requiring the sender to verify the signature on the PAR before the fuzzy extraction and the encryption of the message. The IBS scheme that is used to sign the PAR is currently the most efficient pairing based IBS scheme [Barreto et al., 2005] with the shortest signature length among all IBS schemes as shown by [Galindo and Garcia, 2009]. Since this IBS scheme is based on the Sakai Kasahara Key Construction, it is well-suited to modified BIO-IBE. Alternatively, the recently introduced IBS scheme of [Galindo and Garcia, 2009] can be used for a more efficient verification, which is based on sequentially delegating Schnorr signatures and the verification cost is only 1.5 exponentiations in group $\mathbb{G}$. This way,

the sender can detect whether PAR of the receiver that is stored publicly is modified by an active adversary, thus the generation of a ciphertext based on a wrong identity is avoided. Despite the additional verification of the signature on PAR, the modified BIO-IBE still achieves better efficiency compared to the existing fuzzy IBE schemes in terms of the key generation and decryption algorithms. Similar to the fuzzy extraction computation, the verification of the signature on PAR is performed once and for all. Finally, key escrow problem inherent in all IBE systems effects also fuzzy/biometric IBE systems, namely PKG can decrypt any message as it generates all the secret keys of the users in IBE. However, applying certificateless encryption techniques to BIO-IBE avoids this problem easily.

6.1.2 Related Work

The first fuzzy IBE scheme [Sahai and Waters, 2005] is described by Sahai and Waters in 2005 and its security is reduced to the MBDH problem in the standard model, where the size of the public parameters is linear in the size of the attribute (i.e. feature) space $\mathbb{U}$ for the small universe construction. The authors present another scheme for the large universe of attributes that is based on the first scheme presented in [Boneh and Boyen, 2004]. This scheme reduces the size of the public parameters to the number of attributes n of a user at the cost of an expensive function computed for each ciphertext. Piretti et al [Pirretti et al., 2006] achieved a more efficient fuzzy IBE scheme with short public parameter size by employing the Random Oracle Model (ROM). Baek et al [Baek et al., 2007] described two new fuzzy IBE schemes with efficient key generation algorithms and proved their security in ROM based on the DBDH assumption. Next, the author of [van Liesdonk, 2007] described another fuzzy IBE system that is based on the Boneh-Franklin IBE scheme [Boneh and Franklin, 2003] in ROM in order to achieve anonymity notion. The main disadvantage of the schemes in [Pirretti et al., 2006, Baek et al., 2007, van Liesdonk, 2007] is the use of the MapToPoint hash function, which is inefficient compared to the ordinary hash functions.

Besides, Burnett et al [Burnett et al., 2007] described a biometric Identity Based Signature (IBS) scheme called BIO-IBS, where they used the biometric information represented as a fixed length binary string as the identity and construct the public key of the user using a fuzzy extractor [Dodis et al., 2004], which is then used in the modified SOK-IBS scheme [Bellare et al., 2004]. Finally, the signature analogue of fuzzy IBE is described in [Yang et al., 2008, Shahandashti and Safavi-Naini, 2009] and an efficient biometric IBS scheme is presented in [Sarier, 2010c]. The common property of all these schemes is the use of bilinear pairings.

6.1.3 Organization

In section 6.2, we will state the definitions of the primitives that are used in our scheme. Next, we present our first construction OrdFIBE for ordered features and the security reduction. In section 6.2.6, we describe a new method for preventing collision attacks and section 6.3 describes our first BIO-IBE scheme. Next, we present a stronger security model for biometric IBE and present the security reduction of BIO-IBE in section 6.4. To obtain a tight reduction cost, the security reduction of BIO-IBE is modified in section 6.5. Following this result, we are able to prove BIO-IBE for small universe of attributes in the standard model in section 6.6. Furthermore, a new denial of service attack is analyzed and a simple solution that does not affect the efficiency of the scheme is presented in section 6.7. Finally, we compare our results with existing fuzzy IBE schemes implemented for biometric identities and conclude our results.

6.2 Definitions and Building Blocks

In order to introduce the new biometric IBE scheme, at first, we briefly review the definitions and required computational primitives. Given a set S, $x \xleftarrow{R} S$ defines the assignment of a uniformly distributed random element from the set S to the variable x. $|S|$ denotes the size of the set S and μ_i denotes an attribute (or feature) of the biometric feature set w in the universe $\mathbb{U}$ of biometric attributes. If y is a string then $|y|$ denotes the bit-length of y. Also, ID denotes any identity string such as Name, e-mail address, whereas ID denotes the identity string extracted from biometric information of the user. $\mathcal{ID}$ denotes the identity space, M denotes the message space and C denotes the ciphertext space, whereas $\mathcal{C}$ denotes the error-correcting code with $\mathcal{C}_e$ encoding and $\mathcal{C}_d$ decoding functions. $\mathbb{Z}_p^*$ denotes $\mathbb{Z}_p \setminus \{0\}$. Finally, we remind the reader that the details of the definitions of the following primitives are presented in the background chapter.

Definition 6.1. (Bilinear Pairing). *Let $\mathbb{G}$ and $\mathbb{F}$ be multiplicative groups of prime order p and let g be a generator of $\mathbb{G}$. A bilinear pairing is denoted by $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$ if the following two conditions hold.*

1. *$\forall\ a, b \in \mathbb{Z}_p$, we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$*
2. *$\hat{e}(g, g) \neq 1_{\mathbb{F}}$, namely the pairing is non-degenerate.*

We define the Lagrange coefficient $\Delta_{\mu_i,S}$ for $\mu_i \in \mathbb{Z}_p^*$ and a set S of elements in $\mathbb{Z}_p^*$ as

$$\Delta_{\mu_i,S}(x) = \prod_{\mu_j \in S, \mu_j \neq \mu_i} \frac{x - \mu_j}{\mu_i - \mu_j}$$

For ease of presentation, we work exclusively in the setting where $\hat{e}$ is symmetric; our definitions and results can be generalized to the asymmetric setting where $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{F}$ with $\mathbb{G}_1$ and $\mathbb{G}_2$ being different groups.

The security of our scheme is reduced to the well-exploited complexity assumption (k-BDHI) [Chen and Cheng, 2005], which is stated as follows.

Assumption 6.1. *(Bilinear DH Inversion (k-BDHI)). Let $k \in \mathbb{Z}$, $x \overset{R}{\leftarrow} \mathbb{Z}_q^*$, g be a generator of $\mathbb{G}$ and $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ a bilinear pairing. Given $(g, g^x, g^{x^2}, ..., g^{x^k})$ computing $\hat{e}(g,g)^{\frac{1}{x}}$ is hard.*

In [Chen and Cheng, 2005], it is proven that $BDH \Leftrightarrow (1 - BDHI)$.

Assumption 6.2. *(Decisional Bilinear DH Inversion (k-DBDHI)). Let $k \in \mathbb{Z}$, $x \overset{R}{\leftarrow} \mathbb{Z}_q^*$, g be a generator of $\mathbb{G}$ and $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ a bilinear pairing. Distinguishing between the distributions $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^{\frac{1}{r}})$ and $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^{\frac{1}{x}})$ is hard.*

6.2.1 Fuzzy Identity Based Encryption

In [Sahai and Waters, 2005, Baek et al., 2007], fuzzy IBE is defined as follows.

- Setup: Given a security parameter k_0, the Private Key Generator (PKG) generates the master secret key ms and the public parameters of the system.
- Extract: Given a user's identity w and ms, the PKG returns the corresponding private key.
- Encrypt: A probabilistic algorithm that takes as input an identity w', public parameters and a message $m \in M$ and outputs the ciphertext $c \in C$. Here, M, C and $\mathbb{U}$ denote the message space, the ciphertext space and the universe of attributes, respectively.
- Decrypt: A deterministic algorithm that given the private key and a ciphertext encrypted with w' such that $|w \cap w'| \geq d$, returns either the underlying message m or a reject message. Here d denotes the error tolerance parameter of the scheme.

6.2.2 Security Model

In [Sahai and Waters, 2005, Baek et al., 2007], the Selective-ID model of security for fuzzy IBE is defined using a game between a challenger and an adversary as follows. In other words, a fuzzy IBE scheme guarantees indistinguishability against fuzzy selective identity chosen plaintext attacks (IND-FSID-CPA) if no probabilistic polynomial time (PPT) adversary has a non-negligible advantage in the following game.

- **Phase 1**: The adversary A declares the challenge identity $w^* = (\mu_1^*, ..., \mu_n^*)$.
- **Phase 2**: The challenger runs the Setup algorithm and returns to the adversary the system parameters.
- **Phase 3**: The adversary A issues private key queries for any identity w' such that $|w' \cap w^*| < d$.
- **Phase 4**: The adversary A sends two equal length messages m_0 and m_1. The challenger returns the ciphertext that is encrypted using w^* and the message m_β, where $\beta \xleftarrow{R} \{0, 1\}$.
- **Phase 5**: **Phase 3** is repeated.
- **Phase 6**: A outputs a guess β' for β.

A fuzzy IBE scheme is secure in the sense of IND-FSID-CPA if

$$\mathrm{Adv}_A^{\text{IND-FSID-CPA}}(l) = |Pr[\beta' = \beta] - \frac{1}{2}| < negl(l)$$

for all PPT A. Currently, fuzzy IBE systems are proven secure based on this notion, however, they can be combined with the generic constructions such as REACT [Okamoto and Pointcheval, 2001] to become secure against chosen ciphertext attacks.

6.2.3 The small universe construction of Sahai and Waters

In [Sahai and Waters, 2005], the first scheme for biometric identities is designed for a small universe of attributes $\mathbb{U}$. Here, small universe construction means that the universe of biometric features is defined beforehand, thus we cannot use attributes that were not considered during the setup. This assumption on the size of $\mathbb{U}$ is due to the fact that the public parameters of this scheme are linear in $|\mathbb{U}|$. In addition to the error-tolerance property, this scheme provides security against collusion attacks, which means that different users sharing common biometric attributes with the receiver of the ciphertext cannot decrypt this ciphertext by combining their secret key components associated to these common attributes. Collision attacks are prevented by picking a unique polynomial for each user that is evaluated at each biometric feature of the user and that is combined with the master secret key to generate the private key components. To explain this key generation method and analyse the security reduction, let us review the small universe construction of [Sahai and Waters, 2005]. The second scheme presented in [Sahai and Waters, 2005] is already analyzed in the Background chapter, as it is based on the Boneh-Boyen HIBE scheme [Boneh and Boyen, 2004].

- Setup: Identities will be element subsets of some universe, $\mathbb{U}$, of size $|\mathbb{U}|$. Each element is associated to a unique integer in $\mathbb{Z}_p^*$. (In practice an attribute will be associated with each element so that identities will have some semantics.)

  Let $(g, \mathbb{G}, \mathbb{F})$ be a bilinear group with $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$. First, the universe, $\mathbb{U}$ of elements is defined. For simplicity, the first $|\mathbb{U}|$ elements of Z_p^* to be the universe. Next, choose $t_1, ..., t_{|\mathbb{U}|}$ uniformly at random from $\mathbb{Z}_p$. Finally, choose y uniformly at random $\mathbb{Z}_p$.

  The published public parameters $params$ are: $T_1 = g^{t_1}, ..., T_{|\mathbb{U}|} = g^{t_{|\mathbb{U}|}}, Y = \hat{e}(g, g)^y$. The master secret key is: $ms = t_1, ..., t_{|\mathbb{U}|}, y$.

- Extract(w, ms): To generate a private key for identity $w \in \mathbb{U}$, we do the following. A $d-1$ degree polynomial q is randomly chosen such that $q(0) = y$. The private key consists of components, $(D_i)_{\mu_i \in w}$, where $D_i = g^{q(\mu_i)/t_i}$ for every $\mu_i \in w$.

- Encrypt $(w', m, params)$: Encryption with the public key w' and message $m \in \mathbb{F}$ proceeds as follows. First, a random value $s \in \mathbb{Z}_p$ is chosen. The ciphertext is then published as: $E = (w', \{E_i = T_i^s\}_{\mu_i \in w'}, E' = mY^s)$. Note that the identity, w' is included in the ciphertext.

- Decrypt(D, E): Suppose that a ciphertext, E, is encrypted with a key for identity w' and we have a private key for identity w, where $|w \cap w'| \geq d$. Choose an arbitrary d-element subset, S, of $w \cap w'$. Then, E can be decrypted as:

$$
\begin{aligned}
& E / \textstyle\prod_{\mu_i \in S} (\hat{e}(D_i, E_i))^{\Delta_{\mu_i, S}(0)} \\
& = m\hat{e}(g, g)^{sy} / \textstyle\prod_{\mu_i \in S} (\hat{e}(g^{st_i}, g^{q_{\mu_i}/t_i}))^{\Delta_{\mu_i, S}(0)} \\
& = m\hat{e}(g, g)^{sy} / \textstyle\prod_{\mu_i \in S} (\hat{e}(g, g)^{sq_{\mu_i}})^{\Delta_{\mu_i, S}(0)} = m
\end{aligned}
$$

Theorem 6.1. *[Sahai and Waters, 2005]. If an adversary can break the scheme in the Fuzzy Selective-ID Model, then a simulator can be constructed to play the Decisional MBDH game with a non-negligible advantage.*

Proof. Suppose there exists a polynomial-time adversary, A, that can attack our scheme in the Selective-ID model with advantage ϵ. We build a simulator B that can play the Decisional MBDH game with advantage $\epsilon/2$. The simulation proceeds as follows:

- **Phase 1**: The simulator B runs A and receives the challenge identity, α.

- **Phase 2**: We first let the challenger set the bilinear group $(g, \mathbb{G}, \mathbb{F})$. The challenger flips a fair binary coin, ν outside of B's view. If $\nu = 0$, the challenger sets $(A, B, C, Z) = (g^a, g^b, g^c, \hat{e}(g, g)^{ab/c})$; otherwise it sets $(A, B, C, Z) = (g^a, g^b, g^c, \hat{e}(g, g)^z)$ for random $a, b, c, z \in \mathbb{Z}_p$. We assume the universe, $\mathbb{U}$ is defined. The simulator assigns the public key parameters as follows.

It sets the parameter $Y = \hat{e}(g, A) = \hat{e}(g, g)^a$. For all $\mu_i \in \alpha$ it chooses random $\beta_i \in \mathbb{Z}_p$ and sets $T_i = C^{\beta_i} = g^{c\beta_i}$. For all $\mu_i \in \mathbb{U} - \alpha$ it chooses random $v_i \in \mathbb{Z}_p$ and sets $T_i = g^{v_i}$. It then gives the public parameters to A. Notice that from the view A all parameters are chosen at random as in the construction.

- **Phase 3**: A makes requests for private keys where the identity set overlap between the identities for each requested key and α is less than d. Suppose A requests a private key γ where $|\gamma \cap \alpha| < d$.

  We first define three sets κ, κ', S in the following manner:

  $\kappa = \gamma \cap \alpha$, let κ' be any set such that $\kappa \subseteq \kappa' \subseteq \gamma$ and $|\kappa'| = d-1$, and $S = \kappa' \cap \{ 0\}$.

  Next, we define the decryption key components, D_i, for $\mu_i \in \kappa'$ as:

  If $\mu_i \in \kappa$: $D_i = g^{s_i}$ where s_i is chosen randomly in $\mathbb{Z}_p$.

  If $\mu_i \in \kappa' - \kappa$: $D_i = g^{\lambda_i / v_i}$ where λ_i is chosen randomly in $\mathbb{Z}_p$.

  The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(0) = a$.

  For $\mu_i \in \kappa$ we have $q(\mu_i) = c\beta_i s_i$ and for $\mu_i \in \kappa' - \kappa$ we have $q(\mu_i) = \lambda_i$.

  The simulator can calculate the other D_i values where $\mu_i \notin \kappa'$ since he knows the discrete log of T_i for all $\mu_i \notin \alpha$. The simulator makes the assignments as follows:

  $$\text{If } \mu_i \notin \kappa' : D_i = \Big(\prod_{\mu_j \in \kappa} C^{\beta_j s_j \Delta_{\mu_j, S}(\mu_i)/v_i}\Big)\Big(\prod_{\mu_j \in \kappa' - \kappa} g^{\lambda_j \Delta_{\mu_j, S}(\mu_i)/v_i}\Big) A^{\Delta_{0,S}(\mu_i)/v_i}$$

  Using interpolation the simulator is able to calculate $D_i = g^{q(\mu_i)/t_i}$ for $\mu_i \notin \kappa'$ where $q(x)$ was implicitly defined by the random assignment of the other $d-1$ variables $D_i \in \kappa'$ and the variable A. Therefore, the simulator is able to construct a private key for the identity γ. Furthermore, the distribution of the private key for γ is identical to that of the original scheme.

  As one can notice, the simulator is not able to produce a decryption key when the adversary A queries first for γ, such that $|\gamma \cap \alpha| < d$ and then in some other query, A asks for γ' such that $|\gamma' \cap \alpha| < d$ but $|\gamma \cap \gamma'| \geq d$. This is due to the structure of the secret keys that are computed based on the set κ'. If the adversary asks the secret key for a similar biometrics γ' which will have a slightly different κ' set, then the secret key components of the different features in the set γ' cannot be computed. This fact should not be a considered as a limitation but the security model should include that the adversary cannot ask a similar biometrics γ' as some of its secret key components cannot be computed.

  For BIO-IBE, we will follow the same approach in the generation of the challenge ciphertext, namely by defining the polynomial $q(\cdot)$ by random assignments.

- **Phase 4**: The adversary A, will submit two challenge messages m_0 and m_1 to the simulator. The simulator flips a fair binary coin b, and returns an encryption of m_b. The ciphertext is output as: $E = (\alpha, \{E_i = B^{\beta_i}\}_{\mu_i \in \alpha}, E' = m_b Z)$.

  If $\nu = 0$, then $Z = \hat{e}(g,g)^{ab/c}$. Therefore, if we let $r' = b/c$, then we have $E' = m_b Z = m_b \hat{e}(g,g)^{ab/c} = m_b \hat{e}(g,g)^{ar'} = m_b Y^{r'}$ and $E_i = B^{\beta_i} = g^{b\beta_i} = g^{c\beta_i b/c} = g^{r'c\beta_i} = (T_i)^{r'}$.

  Hence, E is a random encryption of the message m_b under the public key α.

  Otherwise, if $\nu = 1$, then $Z = g^z$. We then have $E' = m_b \hat{e}(g,g)^z$. Since z is random, E' will be a random element of $\mathbb{F}$ from the adversaries view and the message contains no information about m_b.

- **Phase 5**: Identical to **Phase 3**.

- **Phase 6**: A will submit a guess b' of b. If $b = b'$ the simulator will output $\nu' = 0$ to indicate that it was given a MBDH-tuple otherwise it will output $\nu' = 1$ to indicate it was given a random 4-tuple.

□

The reason for the review of this scheme is that this is the first fuzzy IBE scheme secure in the standard model. Following the same construction for the simulation of the key extraction queries, we design an efficient biometric IBE scheme as follows.

6.2.4 A first Attempt for an efficient biometric IBE

As noted before, the main disadvantage of the small-universe construction of Sahai and Waters is the size of the public parameters which is linear in the size of the universe of attributes $\mathbb{U}$. To eliminate this feature, the authors design a second scheme, where the public parameters include the values $t_1, ..., t_n \in \mathbb{G}$ instead of the whole universe. However, a special function T has to be computed in both the key generation and encryption algorithms as: $T(x) = g^{x^i} \prod_{j=1}^{n+1} t_j^{\Delta_{i,N}}$ where N is the set $\{1, ..., n+1\}$. By applying the conversion presented in [Pirretti et al., 2006], the $n+1$ exponentiations needed to solve T at each encryption have been replaced with a single MapToPoint hash function used as a random oracle [Pirretti et al., 2006], [Baek et al., 2007].

In this section, we introduce the use of a different key construction method for biometric IBE, which outperforms all the previous constructions of fuzzy IBE and ABE. We first describe an efficient biometric IBE scheme denoted as OrdFIBE, which is restricted for biometrics that can be represented as an ordered set of features. The system can also be implemented as an ABE scheme, since the attributes {university, department,

section, division, etc} can be grouped/ordered. The reason of this restriction is due to the structure of the security reduction. In the following section, we present another protocol for biometric IBE which also works for unordered set of features/attributes.

- Setup: Given a security parameter k_0, the parameters of the scheme are generated as follows.

  1. Generate two cyclic groups $\mathbb{G}$ and $\mathbb{F}$ of prime order $p > 2^{k_0}$ and a bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$. Pick a random generator $g \in \mathbb{G}$.
  2. Pick at random $x_1, ..., x_n, t \in \mathbb{Z}_p^*$ and set $P_{pub}^1 = g^{x_1}$,...,$P_{pub}^n = g^{x_n}$, $\hat{e}(g, g)^t$.
  3. Pick two cryptographic hash functions $H_1 : \mathbb{Z}_p^* \times \{0,1\}^* \to \mathbb{Z}_p^*$, $H_2 : \mathbb{F} \to \{0,1\}^{k_1}$.

  $M = \{0,1\}^{k_1}$ denotes the message space and $C = \mathbb{U} \times \mathbb{G}^n \times \{0,1\}^{k_1}$ denotes the ciphertext space. The master public key is $(\mathbb{G}, \mathbb{F}, \hat{e}, g, P_{pub}^1, ..., P_{pub}^n, \hat{e}(g,g)^t, H_1, H_2)$ and the master secret key is $ms = x_1, ..., x_n, t$.

- Extract: First, a user's biometric attributes are obtained, where each attribute is mapped to an element of $\mathbb{Z}_p^*$ as in [Sahai and Waters, 2005, Baek et al., 2007, Pirretti et al., 2006]. Given a user's biometric attributes w, the PKG picks a random polynomial $q(\cdot)$ of degree $d-1$ over $\mathbb{Z}_p$ such that $q(0) = t$ and computes the shares $q(\mu_i) = q_i$ for $\mu_i \in w$. It returns $D_{\mu_i} = g^{q_i/(x_i + H_1(\mu_i))} = g^{q_i/(x_i + h_i)}$ for each $\mu_i \in w$.

- Encrypt: On input a (similar) biometrics of the receiver, the biometric attribute set w' is computed. Given a plaintext $m \in M$ and w', the following steps are performed.

  1. Pick at random $r \in \mathbb{Z}_p$.
  2. Compute $L_i = P_{pub}^i \cdot g^{H_1(\mu_i)} = g^{x_i} \cdot g^{h_i} = g^{(x_i + h_i)}$ for $\mu_i \in w'$ and $V = H_2(\hat{e}(g,g)^{tr})$ for each $\mu_i \in w'$.
  3. Compute $C = (w', \{U_i : \mu_i \in w'\}, W) = (w', \{L_i^r : \mu_i \in w'\}, m \oplus V)$.

- Decrypt: Given the ciphertext C and D_{μ_i} for each $\mu_i \in w$, choose an arbitrary

set $S \subseteq w \cap w'$ such that $|S| = d$ and compute $m = W \oplus V$ as

$$\begin{aligned} V &= H_2(\prod_{\mu_i \in S} (\hat{e}(U_i, D_{\mu_i}))^{\Delta_{\mu_i, S}(0)}) \\ &= H_2(\prod_{\mu_i \in S} (\hat{e}(g^{r(x_i+h_i)}, g^{q_i/(x_i+h_i)}))^{\Delta_{\mu_i, S}(0)}) \\ &= H_2(\prod_{\mu_i \in S} (\hat{e}(g,g)^{rq_i})^{\Delta_{\mu_i, S}(0)}) \\ &= H_2(\hat{e}(g,g)^{rt}) \end{aligned}$$

Theorem 6.2. *Suppose the hash functions H_1, H_2 are random oracles and there exists a polynomial time adversary A that can break the scheme by making q_1 H_1-queries, q_2 H_2-queries and q_{ex} private key extraction queries. Then there exists a polynomial time algorithm B that solves the k-BDHI problem with $k = q_1 + q_{ex} + 1$ and advantage*

$$2Adv^{\text{IND-FSID-CPA}}(A) \leq q_2 Adv^{k\text{-BDHI}}(B)$$

Proof. Assume that a polynomial time attacker A breaks our scheme, then using A, we show that one can construct an attacker B solving the k-BDHI problem. Suppose that B is given the k-BDHI problem $(g, g^x, g^{x^2}, ..., g^{x^k})$, B will compute $\hat{e}(g,g)^{1/x}$ using A as follows.

- **Phase 1**: The adversary A declares the challenge identity $w^* = (\mu_1^*, ..., \mu_n^*)$.
- **Phase 2**: B simulates the public parameters for A as follows. The computations of $Q, Q^x, Q^{1/(x+h_j)}, T', T_0$ are the same as in Lemma 3 of [Chen and Cheng, 2005].

  First, B selects $t, a_1, ...a_n \in \mathbb{Z}_p^*$ and $h_0, ...h_{n-1}, ..., h_{k-1} \in \mathbb{Z}_p^*$ at random. Let us denote with $\mathcal{H}$ the set $\{h_n, ..., h_{k-1}\}$. Define $f(z) = \prod_{j=1}^{k-1}(z + h_j)$, which could be written as $f(z) = \sum_{j=0}^{k-1} c_j z^j$. The constant term c_0 is non-zero because $h_j \neq 0$ and c_j are computable from h_j.

  B computes $Q = \prod_{j=0}^{k-1}(g^{x^j})^{c_j} = g^{f(x)}$ and $Q^x = g^{xf(x)} = \prod_{j=0}^{k-1}(g^{x^{j+1}})^{c_j}$.
  If $Q = 1$, then $x = -h_j$ for some j, then k-BDHI problem could be solved directly.

  Next, $f_j(z) = \frac{f(z)}{z+h_j} = \sum_{v=0}^{k-2} d_{j,v} z^j$ for $1 \leq j < k$ and $Q^{1/(x+h_j)} = g^{f_j(x)} = \prod_{v=0}^{k-2}(g^{x^v})^{d_{j,v}}$ is computed.

  In addition, we compute $Q^{x/(x+h_j)} = g^{xf_j(x)} = \prod_{v=0}^{k-2}(g^{x^{v+1}})^{d_{j,v}}$.

  B sets $T' = \prod_{j=1}^{k-1}(g^{x^{j-1}})^{c_j} = g^{(f(x)-c_0)/x}$ and set $T_0 = \hat{e}(T', Q \cdot g^{c_0})$.

B returns A the public parameters $(\mathbb{G}, \mathbb{F}, \hat{e}, Q, P^1_{pub}, ..., P^n_{pub}, \hat{e}(Q,Q)^t, H_1, H_2)$, where $P^1_{pub} = Q^{a_1(x-h_0)}$, $P^2_{pub} = Q^{a_2(x-h_1)}$,..., $P^n_{pub} = Q^{a_n(x-h_{n-1})}$ and H_1, H_2 are random oracles controlled by B as follows.

As one can notice, for the challenge identity, we set $H_1(\mu^*_1) = a_1h_0$, $H_1(\mu^*_2) = a_2h_1$,...,$H_1(\mu^*_n) = a_nh_{n-1}$ and the associated secret key of $\mu^*_i \in w^*$ is $Q^{q(i)/(a_ix)}$. The correction can be verified by

$$\begin{aligned}\hat{e}(Q, Q^{q(i)}) &= \hat{e}(P^i_{pub} \cdot Q^{H_1(\mu^*_i)}, Q^{q(i)/(a_ix)}) \\ &= \hat{e}(Q^{a_i(x-h_{i-1})} \cdot Q^{a_ih_{i-1}}, Q^{q(i)/(a_ix)}) \\ &= \hat{e}(Q^{a_i(x-h_{i-1})+a_ih_{i-1}}, Q^{q(i)/(a_ix)}) \\ &= \hat{e}(Q^{a_ix}, Q^{q(i)/(a_ix)})\end{aligned}$$

Using interpolation, we obtain $\hat{e}(Q, Q^t)$.

- **Phase 3**: B answers the hash and private key queries of A as follows:

  H_1-queries: For a query on identity $w = (\mu_1, ...\mu_n)$, repeat the following for $i = 1, ..., n$. If an element of w is already queried, return the same answer.

  1. If $\mu_i \in w^*$, return a_ih_{i-1} to A and add $\langle \mu_i, a_ih_{i-1}, \perp \rangle$ to H_1List.
  2. Else if $\mu_i \notin w^*$, pick a random element h_j of the set $\mathcal{H}$, return $a_i(h_j + h_{i-1})$ to A and add the tuple $\langle \mu_i, a_i(h_j + h_{i-1}), Q^{1/a_i(x+h_j)}, Q^{x/a_i(x+h_j)} \rangle$ to H_1List.

  We note that the hash queries can also be made individually, i.e. as a query μ_i instead of w since we are able to determine the order of each feature. For instance, an attribute rwth-aachen can easily be recognized as an attribute for university, thus, it should be associated to the public parameter P^1_{pub}. The hash value is $a_1(h_j + h_0)$, where h_j is picked at random. The correction can be verified

$$\begin{aligned}\hat{e}(Q, Q^{q(\text{rwth-aachen})}) &= \hat{e}(P^1_{pub} \cdot Q^{H_1(\text{rwth-aachen})}, Q^{q(\text{rwth-aachen})/a_1(x+h_j)}) \\ &= \hat{e}(Q^{a_1(x-h_0)} \cdot Q^{a_1(h_j+h_0)}, Q^{q(\text{rwth-aachen})/a_1(x+h_j)}) \\ &= \hat{e}(Q^{a_1(x+h_j)}, Q^{q(\text{rwth-aachen})/a_1(x+h_j)})\end{aligned}$$

  H_2-queries: Upon receiving a new query R,

  1. If there exists (R, ξ) in H_2List, return ξ.
  2. Else, choose $\xi \xleftarrow{R} \{0,1\}^{k_1}$ and return ξ to A.

Key extraction queries: B simulates the private key extraction queries of A as follows. Suppose A requests a private key w where $|w \cap w^*| < d$.

We first define three sets κ, κ', S in the following manner: Let $\kappa = w \cap w^*$ and denote with κ' any set such that $\kappa \subseteq \kappa' \subseteq w$ and $|\kappa'| = d - 1$, and $S = \kappa' \cap \{ 0\}$.

Next, we define the decryption key components, D_i for $\mu_i \in \kappa$ and D_j for $\mu_j \in \kappa' - \kappa$ as follows. Here, we denote with subscript i, the elements of w that are in κ and with subscript j, the elements of w that are in $\kappa' - \kappa$ and the elements of w that are not κ'.

If $\mu_i \in \kappa$: $D_i = Q^{s_i}$ where s_i is chosen randomly in $\mathbb{Z}_p$.

If $\mu_j \in \kappa' - \kappa : D_j = Q^{\lambda_j / a_j(x+h_j)}$ where λ_j is chosen randomly in $\mathbb{Z}_p$.

The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(0) = t$.

For $\mu_i \in \kappa$ we have $q(\mu_i) = x a_i s_i$ and for $\mu_j \in \kappa' - \kappa$ we have $q(\mu_j) = \lambda_j$.

The simulator can calculate the other D_j values where $\mu_j \notin \kappa'$ by interpolation.

If $\mu_j \notin \kappa' : D_j$ is computed as

$$(\prod_{\mu_i \in \kappa} Q^{x a_i s_i \Delta_{\mu_i, S}(\mu_j)/a_j(x+h_j)})(\prod_{\mu_k \in \kappa' - \kappa} Q^{\lambda_k \Delta_{\mu_k, S}(\mu_j)/a_j(x+h_j)}) Q^{t \Delta_{0,S}(\mu_j)/a_j(x+h_j)}$$

Since we are able to compute $Q^{x/a_j(x+h_j)}$ and $Q^{1/a_j(x+h_j)}$ for each feature $\mu_j \neq \mu_i^*$, i.e. $\mu_j \in w - \kappa$, using interpolation the simulator is able to calculate $D_j = Q^{q(\mu_j)/a_j(x+h_j)}$ for $\mu_j \notin \kappa'$ where $q(x)$ was implicitly defined by the random assignment of the other $d-1$ variables of the set κ' and the variable Q^t. Therefore, the simulator is able to construct a private key for the identity w. Furthermore, the distribution of the private key for w is identical to that of the original scheme.

- **Phase 4**: Upon receiving the messages (m_0, m_1) with $|m_0| = |m_1|$, B generates the challenge ciphertext as follows.

  1. Pick at random $r^* \in \mathbb{Z}_p$.
  2. Compute $U_{\mu_i^*} = [P_{pub}^i \cdot Q^{H_1(\mu_i^*)}]^{r^*} = [Q^{a_i(x-h_{i-1})} \cdot Q^{a_i h_{i-1}}]^{r^*} = Q^{a_i r}$ for each $\mu_i^* \in w^*$.
  3. B chooses a random $\beta \in \{0,1\}$ and $W^* \xleftarrow{R} \{0,1\}^{k_1}$.
  4. Set the ciphertext to $c^* = (w^*, \{U_{\mu_i^*} : \mu_i^* \in w^*\}, m_\beta \oplus W^*)$.

  The intuition behind these assignments is that we are implicitly choosing the randomness r as r^*/x.

- **Phase 5**: B answers A's queries as in **Phase 3**.
- **Phase 6**: At some point, A responds with the guess β' for the underlying plaintext m_β, which could only be computed from

$$m_\beta = W^* \oplus H_2(\prod_{\mu_i \in S} \hat{e}(U_{\mu_i^*}, D_{\mu_i^*})^{\Delta_{\mu_i^*,S}(0)}) = W^* \oplus H_2(T^{tr^*}).$$

From the queries of A to the H_2-oracle, we obtain T^{tr^*}. Since the simulator knows the value of r^* and t, the value of $T = \hat{e}(Q,Q)^{1/x}$ is computed.

The solution to the k-BDHI problem, $\hat{e}(g, g^{1/x})$, is obtained by outputting $(T/T_0)^{1/c_0^2} = \hat{e}(g, g^{1/x})$ as in [Chen et al., 2006].

$$\begin{aligned} T/T_0 &= \hat{e}(g,g)^{f(x)\cdot f(x)/x} / \hat{e}(g^{(f(x)-c_0)/x}, g^{f(x)+c_0}) \\ &= \hat{e}(g,g)^{f(x)\cdot f(x)/x - f(x)\cdot f(x)/x + c_0^2/x} \\ &= \hat{e}(g,g)^{c_0^2/x} \end{aligned}$$

Since the probability that the session key T^{tr^*} is in the H_2List is $\frac{1}{q_2}$, we have

$$2\mathrm{Adv}^{\text{IND-FSID-CPA}}(A) \leq q_2 \mathrm{Adv}^{\text{k-BDHI}}(B)$$

□

As one can notice, OrdFIBE works only for biometrics that can be ordered/grouped or in the setting of ABE, where the attributes of {university, department, section, division, etc.} can be ordered/grouped for each user. If an attribute is missing, it can take the value 0 and the representation of each user follows the same pattern. Besides, if some of the attributes at the sender's side does not match the ones at the receiver's side, for instance, the values of division and section is different at the both sides, by the error-tolerance of the scheme, the receiver is still able to decrypt the ciphertext. OrdFIBE is more efficient in all of its phases compared to the current fuzzy IBE and ABE schemes as shown in figure 6.3 with the exception that the scheme has n different public keys P_{pub}^i instead of one public key P_{pub} as in other schemes secure in ROM. These additional $n-1$ keys in the public parameters have a much smaller effect than the small universe construction of Sahai and Waters, where the size of the public parameters is linear in the size of the universe of attributes $\mathcal{U}$ such that $n << |\mathcal{U}|$.

As a result, our first attempt cannot be implemented for unordered biometrics such as fingerprint minutia and has a small constant (i.e. n) of overhead in the public parameters of the scheme, which should be improved so that the scheme is more efficient

than existing schemes in ROM in all aspects. For this purpose, we describe BIO-IBE, which is also based on the Sakai Kasahara Key Construction, but it is designed in a slightly different way and has a different security reduction. In particular, BIO-IBE is proven secure both for small and large universe constructions, where the former allows us to reduce the security in the standard model under a specific condition. When we prove BIO-IBE for small universe of attributes, we will simulate the challenge ciphertext similar to the simulation of the private keys of the above scheme, namely by picking random values for the $d-1$ components. Despite this similarity, for BIO-IBE, we require a different method to prevent collision attack. First, we review the necessary tools for our new method.

6.2.5 Error Correcting Codes and Fuzzy Extractors

Let $\mathcal{H} = \{0,1\}^N = \mathbb{F}_2^N$ be the Hamming space of length N, where $\mathbb{F}_2 = \{0,1\}$. The Hamming distance over $\mathcal{H}$ is denoted by $\mathsf{dis}_{\mathcal{H}}()$. An Error Correcting Code (ECC) over $\mathcal{H}$ is a subset $\mathcal{C} \subset \mathcal{H}$, where elements of $\mathcal{C}$ are called as codewords. A binary linear error correcting code $\mathcal{C}$ is a vector subspace of $\mathbb{F}_2^N$. When $\mathcal{C}$ contains 2^k codewords, then $\mathcal{C}$ is denoted as $[N,k,t]$, where t is the correction capacity of $\mathcal{C}$.

The main idea of fuzzy sketches is given a public data $\mathsf{PAR} = c \oplus b$, one tries to correct the corrupted codeword $\mathsf{PAR} \oplus b' = c \oplus (b \oplus b')$. If the Hamming distance $\mathsf{dis}_{\mathcal{H}}(b,b')$ is small, recovering p from $\mathsf{PAR} \oplus b'$ is possible [Bringer et al., 2007a]. An important requirement for such a scheme is that the value PAR should not reveal too much information about the biometric template b, which is obtained as described in section 6.2.6. By applying a strong extractor, one can convert any secure sketch to a fuzzy extractor. The details of fuzzy sketch and fuzzy extractor is given in [Dodis et al., 2004, Juels and Wattenberg, 1999]. Formally, an $(\mathcal{M}, l, t)$ fuzzy extractor is defined as follows. Let $\mathcal{M} = \{0,1\}^v$ be a finite dimensional metric space with a distance function $\mathsf{dis} : \mathcal{M} \times \mathcal{M} \to \mathbb{Z}^+$. Here, dis measures the distance between b and b', where $b, b' \in \mathcal{M}$. An $(\mathcal{M}, l, t)$ fuzzy extractor consists of two functions Gen and Rep.

- Gen: A probabilistic generation procedure that takes as input $b \in \mathcal{M}$ and outputs an biometric identity string $\mathsf{ID} \in \{0,1\}^l$ and a public parameter PAR, that is used by the Rep function to regenerate the same biometric string ID from b' such that $\mathsf{dis}(b,b') \leq t$.
- Rep: A deterministic reproduction procedure that takes as input b' and the publicly available value PAR, and outputs ID if $\mathsf{dis}(b,b') \leq t$.

In [Burnett et al., 2007], the authors describe a concrete fuzzy extractor using a $[n,k,2t+1]$ BCH error correction code, Hamming Distance metric and a one-way hash function $H : \{0,1\}^n \to \{0,1\}^l$. Specifically,

- The Gen function takes the biometrics b as input and returns $\mathsf{ID}= H(b)$ and public parameter $\mathsf{PAR} = b \oplus \mathcal{C}_e(\mathsf{ID})$, where $\mathcal{C}_e$ is a one-to-one encoding function.
- The Rep function takes a biometric b' and PAR as input and computes $\mathsf{ID}' = \mathcal{C}_d(b' \oplus \mathsf{PAR}) = \mathcal{C}_d(b \oplus b' \oplus \mathcal{C}_e(\mathsf{ID}))$. $\mathsf{ID} = \mathsf{ID}'$ if and only if $\mathsf{dis}_{\mathcal{H}}\ (b, b') \leq t$. Here $\mathcal{C}_d$ is the decoding function that corrects the errors upto the threshold t.

6.2.6 Our New Method for Biometric Identities

Any biometric IBE/IBS scheme requires the biometric measurement of the receiver or the signer, respectively. For this purpose, the biometrics of the user is captured using a sensor and the raw biometric data is further processed to extract the feature vector and to obtain the biometric template b of the user. In a biometric encryption scheme, feature extraction is applied on the raw biometric data to obtain the set of features (attributes) and then, each attribute is associated with a unique integer $w_i \in \mathbb{Z}_p^*$ to form the identity $w = (w_1, ..., w_n)$ [Sahai and Waters, 2005, Baek et al., 2007]. Here, n denotes the size of the attributes of each user. Since some of the attributes could be common in some users, a unique polynomial is selected for each user and included in the key generation algorithm to bind the private key to the user. This way, different users cannot collude in order to decrypt a ciphertext that should only be decrypted by the real receiver.

In a biometric IBS scheme such as BIO-IBS [Burnett et al., 2007], the biometric template b assumed as a fixed length binary sting is computed and the hash of b is used as the identity ID. This is in accordance with the framework for biometric template generation, which consists of (1) extracting features; (2) quantization and coding per feature and concatenating the output codes; (3) applying error correction coding (ECC) and hashing [Chen et al., 2007]. During this process, many quantizers produce and use side-information, which could be published to be used later in the reconstruction of the binary template b'.

As different from existing fuzzy IBE systems, BIO-IBE requires the use of the multimodal biometrics, where a biometric feature set extracted from a biometric trait is combined with a unique biometric binary string fuzzy extracted either from the same biometric trait but using a different feature extraction method or from a different biometric trait that can be represented as a binary string. A fuzzy extractor is used to generate a unique biometric string in order to bind the private key components to the user's identity and thus avoid collusion attacks. Specifically, the identity is obtained from the biometric information of the user using a feature extraction algorithm followed by a fuzzy extraction process, where the result of the former procedure (i.e. $w = (w_1, ..., w_n)$) is combined with the output of the latter (i.e. ID) to obtain the biometric attribute set $\mathsf{BID} = \langle H_1(w_1, \mathsf{ID}), ..., H_1(w_n, \mathsf{ID})\rangle$ to be used in the key generation phase.

This way, the privacy of biometric-identity relation and the resistance against collusion attacks is maintained. For multi-modal biometrics using a single biometric trait, the feature set w and the unique identity string ID are obtained from the same biometric trait w and using a single sensor. Instead of choosing a unique polynomial for each user as in current fuzzy IBE schemes, we use the fuzzy extractor to obtain a unique string ID via error correction codes from the biometric template b of the user in such a way that an error tolerance t is allowed. In other words, we will obtain the same string ID even if the fuzzy extractor is applied on a different b' such that $\mathsf{dis}(b, b') < t$. Also, multi-modal approach can be implemented for multi biometric traits that combines fingerprints + Iris or face + iris, where fingerprint minutia can described as a set of unordered features and iris can be represented as a 2048 bit string which is already combined with error correction procedures in different secure sketch/fuzzy extraction applications to have improved accuracy [Bringer et al., 2007a]. Hence, a multi-modal approach for preventing collision attacks has also benefits in security and identification of the users during key generation process.

6.3 BIO-IBE

The second biometric IBE scheme we propose is called as BIO-IBE, which differs from the current fuzzy IBE systems due to the use of a unique biometric string obtained via a fuzzy extractor to prevent the collision attacks. Again, BIO-IBE is based on Sakai-Kasahara's Key Construction for the generation of the private keys, thus it does not require a MapToPoint hash function as opposed to the current fuzzy IBE schemes secure in ROM. Thus, our scheme achieves better performance due to the use of an ordinary hash function instead of MapToPoint hash function since the hash computation is performed n times at each encryption and key generation. We emphasize that, the fuzzy extraction process is only performed by the sender only once and for all and can be efficiently implemented on the finite field $\mathbb{F}_{2^m}$, where $n = 2^m - 1$ is the length of the code. If we use the same fingerprint system described in [Burnett et al., 2007], we have $m \approx 10$ for the $[905, 160, 201]$ BCH error correction code. Besides, this is the first scheme that allows the use of multi-modal biometrics for defining the identity of the user. Thus the final computed private key components of each user can be considered as a biometric fusion at the feature level. We present the details of BIO-IBE as below.

- Setup: Given a security parameter k_0, the parameters of the scheme are generated as follows.

    1. Generate two cyclic groups $\mathbb{G}$ and $\mathbb{F}$ of prime order $p > 2^{k_0}$ and a bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{F}$. Pick a random generator $g \in \mathbb{G}$.

2. Pick at random $x, y \in \mathbb{Z}_p^*$ and compute $P_{pub1} = g^{x/y}$, $P_{pub2} = g^{1/y}$ and $\hat{e}(g, g)$.
3. Pick two cryptographic hash functions $H_1 : \mathbb{Z}_p^* \times \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2 : \mathbb{F} \rightarrow \{0,1\}^{k_1}$. In addition, for the fuzzy extraction process, the PKG picks $H : b \rightarrow \{0,1\}^*$, an encoding function E and a decoding function D and the error correction capacity t. These four components are represented by the fuzzy extraction algorithm FE.

$M = \{0,1\}^{k_1}$ denotes the message space and $C = \mathbb{U} \times \mathbb{G}^n \times \{0,1\}^{k_1}$ denotes the ciphertext space. The master public key is $(g, \mathbb{G}, \mathbb{F}, \hat{e}, P_{pub1}, P_{pub2}, \hat{e}(g, g), H_1, H_2, FE)$ and the master secret key is $ms = x, y$.

- Extract: First, a user's biometric attributes are obtained, where each attribute is mapped to an element of $\mathbb{Z}_p^*$ as before. Besides, the identity string $\mathsf{ID} = H(b)$ is calculated from the same or different biometric trait of the receiver using a fuzzy extractor. Given a user's biometric attributes w and the biometric string ID, the PKG returns $D_{\mu_i}^{\mathsf{ID}} = g^{y/(x+H_1(\mu_i,\mathsf{ID}))} = g^{y/(x+h_i)}$ for each $\mu_i \in w$.

- Encrypt: On input a (similar) biometrics of the receiver, the biometric attribute set w' is computed and a unique string ID' is fuzzy extracted with the help of the associated public parameter PAR as described in section 6.2.6. We should note that the computation of ID' is performed only once and for all.

  Given a plaintext $m \in M$, ID' and w', the following steps are performed.

  1. Pick a random polynomial $q(\cdot)$ of degree $d-1$ over $\mathbb{Z}_p$ such that $q(0) = q$ and compute the shares $q(\mu_i) = q_i$ for $\mu_i \in w'$.
  2. Compute $L_i = P_{pub1} \cdot (P_{pub2})^{H_1(\mu_i,\mathsf{ID}')} = g^{x/y} \cdot (P_{pub2})^{h_i} = g^{x/y} \cdot g^{h_i/y} = g^{(x+h_i)/y}$ for $\mu_i \in w'$ and $V = H_2(\hat{e}(g, g)^q)$.
  3. Return $c = (w', \{U_i : \mu_i \in w'\}, W) = (w', \{L_i^{q_i} : \mu_i \in w'\}, m \oplus V)$.

- Decrypt: Given the ciphertext c and $D_{\mu_i}^{\mathsf{ID}}$ for each $\mu_i \in w$, choose an arbitrary set $S \subseteq w \cap w'$ such that $|S| = d$ and compute $m = W \oplus V$ as

$$\begin{aligned} V &= H_2(\prod_{\mu_i \in S} (\hat{e}(U_i, D_{\mu_i}^{\mathsf{ID}}))^{\Delta_{\mu_i,S}(0)}) \\ &= H_2(\prod_{\mu_i \in S} (\hat{e}(g^{q_i(x+h_i)/y}, g^{y/(x+h_i)}))^{\Delta_{\mu_i,S}(0)}) \\ &= H_2(\prod_{\mu_i \in S} (\hat{e}(g, g)^{q_i})^{\Delta_{\mu_i,S}(0)}) \\ &= H_2(\hat{e}(g, g)^q) \end{aligned}$$

Here, the polynomial $q(\cdot)$ of degree $d-1$ is interpolated using d points by polynomial interpolation in the exponents using Shamir's secret sharing method [Shamir, 1979]. Also, $h_i^{\mathsf{ID}'} = h_i^{\mathsf{ID}}$ for each $\mu_i \in S$ due to the fact that $\mathsf{ID} = \mathsf{ID}'$.

6.4 A Stronger Security Model

Before presenting the security reduction of BIO-IBE, we describe a stronger model of security for fuzzy IBE (IND-sFSID-CPA) using a game between a challenger and an adversary as follows. The main difference of our new security model is that the adversary is allowed to make private key extraction queries on the challenge identity w^*, where A can obtain $d-1$ private key components of w^* that A chooses. This new model basically simulates the leakage of partial secret key components of the challenge identity. This property is not considered in the current security model of fuzzy IBE, which return to the adversary only the private key components belonging to any identity other than (i.e. not similar to) the challenge identity. Thus, the adversary A has more power compared to the model defined in [Sahai and Waters, 2005, Baek et al., 2007].

- **Phase 1**: The adversary declares the challenge identity $w^* = (\mu_1^*, ..., \mu_n^*)$.
- **Phase 2**: The challenger runs the Setup algorithm and returns to the adversary the system parameters.
- **Phase 3**: The adversary issues private key queries for any identity w' such that $|w^* \cap w'| < d$ and the challenger returns the private key components of w'. For the challenge identity w^*, A is given the $d-1$ private key components of w^* that A selects except for the component $\mu^* \in w^*$.
- **Phase 4**: The adversary A sends two equal length messages m_0 and m_1. The challenger picks $\beta \xleftarrow{R} \{0,1\}$ and returns encryption of the message m_β using the challenge identity w^*.
- **Phase 5**: Phase 3 is repeated. A is not allowed to issue private key queries for the remaining $n-d+1$ attributes of w^*.
- **Phase 6**: A outputs a guess β' for β.

Theorem 6.3. *Suppose the hash functions H_1, H_2 are random oracles and there exists a polynomial time adversary A with advantage ϵ that can break the scheme BIO-IBE in the stronger security model by making q_1 H_1-queries, q_2 H_2-queries and q_{ex} private key extraction queries. Then there exists a polynomial time algorithm B that solves the k-BDHI problem with $k = q_1 + q_{ex} + 1$ and advantage*

$$\mathrm{Adv}^{\text{IND-sFSID-CPA}}_{\text{BIO-IBE}}(A) \leq (n-d+1)q_2\mathrm{Adv}^{\text{k-BDHI}}(B)$$

Proof. Assume that a polynomial time attacker A breaks our scheme, then using A, we show that one can construct an attacker B solving the k-BDHI problem. Suppose that B is given the k-BDHI problem $(g, g^x, g^{x^2}, ..., g^{x^k})$, B will compute $\hat{e}(g,g)^{1/x}$ using A as follows.

- **Phase 1**: The adversary A declares the challenge identity $w^* = (\mu_1^*, ..., \mu_n^*)$.
- **Phase 2**: B will implicitly associate a random element of w^*, for instance $\mu^* \in w^*$, to the k-BDHI problem. B simulates the public parameters for A as follows. The computations of this phase are the same as in [Chen and Cheng, 2005] and [Chen et al., 2006].

  First, B selects $y \in \mathbb{Z}_p^*$ and $h_0, ..., h_{k-1} \in \mathbb{Z}_p^*$ at random and computes the parameters $f(z), Q, Q^x, Q^{1/(x+h_j)}, T', T_0$ as in the previous proof.

  B returns A the public parameters $(Q, \mathbb{G}, \mathbb{F}, \hat{e}, P_{pub1}, P_{pub2}, H_1, H_2, FE)$, where $P_{pub1} = Q^{(x-h_0)/y}$, $P_{pub2} = Q^{1/y}$, and H_1, H_2 are random oracles controlled by B as follows. As one can notice, the hash value of $H_1(\mu^*, \mathsf{ID}^*) = h_0$ and the associated secret key of $\mu^* \in w^*$ is $Q^{y/x}$. Here, FE denotes the fuzzy extraction algorithm that is used to extract a unique binary string ID.
- **Phase 3**: B answers the hash and private key queries of A as follows:

  H_1-queries: For a query (μ_i, ID), if there exists $\langle j, l, \mu_i, \mathsf{ID}, h_j + h_0, Q^{y/(x+h_j)} \rangle$ in H_1List, return $h_j + h_0$. Otherwise,

  1. If $\mu_i \in w^*$, $\mathsf{ID} = \mathsf{ID}^*$ and $l \neq d$, return $h_j + h_0$ and add $\langle j, l, \mu_i, \mathsf{ID}^*, h_j + h_0, Q^{y/(x+h_j)} \rangle$ to H_1List. Increment j and l by 1.
  2. If $\mu_i \in w^*$, $\mathsf{ID} = \mathsf{ID}^*$ and $l = d$, then return h_0, add the tuple $\langle j, d, \mu_i = \mu^*, \mathsf{ID}^*, h_0, \perp \rangle$ to H_1List. Increment j and l by 1.
  3. Else, return $h_j + h_0$ and add the tuple $\langle j, l, \mu_i, \mathsf{ID}, h_j + h_0, Q^{y/(x+h_j)} \rangle$ to H_1List. Increment j by 1.

  Here, j and l denotes the values of two counters, where $1 \leq j \leq q_1$ and $1 \leq l \leq n$.

  H_2-queries: Upon receiving a new query R,

  1. If there exists (R, ξ) in H_2List, return ξ.
  2. Else, choose $\xi \overset{R}{\leftarrow} \{0,1\}^{k_1}$ and return ξ to A.

Key extraction queries: B simulates the private key queries of A as follows.

Upon receiving a query (w', ID'), such that $|w' \cap w^*| < d$, for every $\mu_i \in w'$, run the H_1-oracle simulator and obtain $\langle j, l, \mu_i, \mathsf{ID}', h_j + h_0, Q^{y/(x+h_j)} \rangle$ from H_1List. Since $\mathsf{ID}' \neq \mathsf{ID}^*$, return $D_{\mu_i}^{\mathsf{ID}'} = Q^{y/(x+h_j)}$ for each $\mu_i \in w'$. For the challenge identity (w^*, ID^*), return the $d-1$ private key components $D_{\mu_i^*}^{\mathsf{ID}^*} = Q^{y/(x+h_j)}$ that A chooses except for the component associated to the attribute μ^*.

- **Phase 4**: Upon receiving the messages (m_0, m_1) with $|m_0| = |m_1|$, B generates the challenge ciphertext as follows.

  1. Pick $q_i \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_p$ for each $\mu_i \in w^*$ unless $\mu_i = \mu^*$.
  2. Using the values in H_1List, compute $U_{\mu_i} = [P_{pub1} \cdot (P_{pub2})^{H_1(\mu_i, \mathsf{ID}^*)}]^{q_i} = [Q^{(x-h_0)/y} \cdot (P_{pub2})^{h_j+h_0}]^{q_i} = [Q^{(x-h_0)/y} \cdot Q^{(h_j+h_0)/y}]^{q_i} = Q^{q_i(x+h_j)/y}$ for each $\mu_i \in w^*$ except for $\mu_i = \mu^*$.
  3. Pick $q^* \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_p$ and compute $U_{\mu^*} = Q^{q^*}$.
  4. B chooses a random $\beta \in \{0,1\}$ and $W^* \overset{\mathrm{R}}{\leftarrow} \{0,1\}^{k_1}$.
  5. Set the ciphertext to $c^* = (w^*, U_{\mu_i}, m_\beta \oplus W^*)$ where $\mu_i \in w^*$.

  The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(\mu^*) = q^* y/x$. For the remaining $n-d$ (dummy) features we choose the values of $q(i)$ at random.

- **Phase 5**: B answers A's random oracle and private key extraction queries as before. The only condition on the private key extraction queries is that the attacker A cannot query the challenge private key for the remaining $n-d+1$ components of the challenge identity.

- **Phase 6**: At some point, A responds with the guess β' for the underlying plaintext m_β, which could only be computed from

$$m_\beta = W^* \oplus H_2\Big(\prod_{\mu_i \in S} \hat{e}(U_{\mu_i}, D_{\mu_i}^{\mathsf{ID}^*})^{\Delta_{\mu_i, S}(0)}\Big)$$

The only way for A to have any advantage in this game is when H_2List contains

$$R = \prod_{\mu_i \in S} \hat{e}(U_{\mu_i}, D_{\mu_i}^{\mathsf{ID}^*})^{\Delta_{\mu_i, S}(0)}$$

Here, the set $S \subseteq w^*$ with $|S| = d$ denotes the d elements of the challenge identity, where A already knows the secret key components of $d-1$ of the set S. Hence, we

assume that a clever attacker has to pick another element of the challenge identity w^* other than the $d-1$ elements to compose the set S, thus, A can query the H_2-oracle with R to have any advantage.

On the other hand, the only way for B to have any advantage in this game is when H_2List contains the value

$$\begin{aligned} R^* &= \prod_{\mu_i \in S} \hat{e}(U_{\mu_i}, D_{\mu_i}^{\mathsf{ID}^*})^{\Delta_{\mu_i,S}(0)} \\ &= \hat{e}(Q, Q^{y/x})^{q^* \Delta_{\mu^*,S}(0)} \cdot \Lambda \end{aligned}$$

where

$$\Lambda = \prod_{\mu_i \in \bar{S}} \hat{e}(Q, Q)^{q_i \Delta_{\mu_i,S}(0)}$$

Here $S = \{\mu^*\} \cup \bar{S}$ and $\bar{S}$ denotes the $d-1$ elements of the challenge identity for which the A knows the secret key components, namely $\bar{S} \subseteq w^*$ with $|\bar{S}| = d-1$. If we define the advantage of the adversary A (i.e. the probability that A distinguishes the challenge message correctly) as $\mathrm{Adv}(A)$, then advantage of the adversary B (i.e. the probability that B finds the solution to the k-BDHI problem) is
$\mathrm{Adv}(B) \leq \frac{1}{n-d+1}\ \mathrm{Adv}(A)$ since the probability that A chooses μ^* as the remaining element to compose the set S is $\frac{1}{n-d+1}$. B sets $T = (R^*/\Lambda)^{1/(yr^* \Delta_{\mu^*,S}(0))} = \hat{e}(Q, Q^{1/x})$. The solution to the k-BDHI problem, $\hat{e}(g, g^{1/x})$, is obtained by outputting $(T/T_0)^{1/c_0^2} = \hat{e}(g, g^{1/x})$ as shown in the previous proof.

Let $\mathbb{H}$ be the event that algorithm A issues a query for $H_2(R)$. We have $\Pr[\mathbb{H}] \geq 2\epsilon$ due to the following facts.

If the H_2List does not contain the value R, then denoting $SuccA = Pr[\beta' = \beta]$, we have $Pr[\beta' = \beta | \neg\mathbb{H}] = \frac{1}{2}$. By the definition of A, $|Pr[\beta' = \beta] - \frac{1}{2}| > \epsilon$.

Combining all the results and defining the event $SuccA = Pr[\beta = \beta']$, we obtain the following as in [Boneh and Franklin, 2003]

$$\begin{aligned} SuccA = Pr[\beta = \beta' | \mathbb{H}] Pr[\mathbb{H}] + Pr[\beta = \beta' | \neg\mathbb{H}] Pr[\neg\mathbb{H}] \\ \iff Pr[\beta = \beta'] \geq \tfrac{1}{2}(1 - Pr[\mathbb{H}]) \\ \iff Pr[\beta = \beta'] \leq \tfrac{1}{2}(1 + Pr[\mathbb{H}]). \end{aligned}$$

Therefore,

$$\epsilon \leq |Pr[\beta = \beta' | \mathbb{H}] - \frac{1}{2}| \leq \frac{1}{2} Pr[\mathbb{H}] \iff Pr[\mathbb{H}] \geq 2\epsilon$$

The probability that $R = R^*$ is $\frac{1}{n-d+1}$, we have $Pr[\bar{\mathbb{H}}] \geq 2\frac{1}{n-d+1}\epsilon$, where $\bar{\mathbb{H}}$ be the event that algorithm A issues a query for $H_2(R^*)$ at some point during the simulation. Since the probability that the session key R^* is in the H_2List is $\frac{1}{q_2}$, we have

$$2\text{Adv}^{\text{IND-sFSID-CPA}}(A) \leq (n-d+1)q_2\text{Adv}^{\text{k-BDHI}}(B)$$

□

This stronger security model gives the adversary as much power as possible by providing the adversary with $d-1$ private key components of the challenge identity requiring a stronger security model than the Fuzzy Selective-ID model of [Sahai and Waters, 2005, Baek et al., 2007]. We should note that BIO-IBE can be proven in the (weaker) Fuzzy Selective-ID model, then the reduction cost will change slightly due to the factor of $\binom{n}{d}$ since the adversary A will have only $\binom{n}{d}$ different choices for the set S instead of $(n-d+1)$ choices. By replacing $n-d+1$ with $\binom{n}{d}$, we obtain the reduction cost for the Fuzzy Selective-ID model.

B's running time is computed identical to [Baek et al., 2007] as $t_B < t_A + (q_1+q_{ex})O(T_e)$, where T_e denotes the computing time for an exponentiation in $\mathbb{G}$.

6.5 Improving the reduction cost of BIO-IBE

As one can note from the above analysis, the reduction cost of BIO-IBE is not tight. However, we can modify the proof slightly to obtain a tight reduction cost by reducing the security of BIO-IBE to the decisional k-BDHI (k-DBDHI) problem instead of its computational version. This way, H_2 is not assumed as a random oracle, and thus the factor of $\frac{1}{q_2}$ is eliminated from the reduction cost. The tradeoff is the use of a (weaker) decisional problem instead of the computational k-BDHI problem. We can also remove H_2 function and directly multiply the message to the session key $\hat{e}(g,g)^q$ instead of xoring it to the hash of the session key, i.e. $m \oplus H_2(\hat{e}(g,g)^q)$. Then each message m should be an element of the group $\mathbb{F}$ as in [Sahai and Waters, 2005].

Theorem 6.4. *Suppose the hash function H_1 is a random oracle and there exists a polynomial time adversary A with advantage ϵ that can break the scheme BIO-IBE in the stronger security model by making q_1 H_1-queries, and q_{ex} private key extraction queries. Then there exists a polynomial time algorithm B that solves the decisional k-BDHI problem with $k = q_1 + q_{ex} + 1$ and advantage*

$$\frac{\epsilon}{2(n-d+1)}$$

Proof. Assume that a polynomial time attacker A breaks our scheme, then using A, we show that one can construct an attacker B solving the decisional k-BDHI problem. Suppose that B is given an instance of the decisional k-BDHI problem, B will distinguish between $\hat{e}(g,g)^{1/x}$ and $\hat{e}(g,g)^r$ using A as follows.

- **Phase 1**: The adversary A declares the challenge identity $w^* = (\mu_1^*, ..., \mu_n^*)$.
- **Phase 2**: B simulates the public parameters for A as follows. We first let the challenger set the groups $\mathbb{G}$ and $\mathbb{F}$ with an efficient bilinear map, $\hat{e}$ and generator g. The challenger flips a fair binary coin b outside of B's view. If $b = 0$, the challenger returns B $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^{1/x})$ otherwise it returns $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^r)$. The remaining computations of this phase are the same as in the **Phase 2** of the proof of BIO-IBE.

  B returns A the public parameters $(Q, \hat{e}, \mathbb{G}, \mathbb{F}, P_{pub1}, P_{pub2}, H_1, FE)$, where $P_{pub1} = Q^{(x-h_0)/y}$,$P_{pub2} = Q^{1/y}$, and H_1 is a random oracle controlled by B as follows.
- **Phase 3**: B simulates the hash and key extraction queries of A as in BIO-IBE.
- **Phase 4**: Upon receiving the messages (m_0, m_1) with $|m_0| = |m_1|$, B generates the challenge ciphertext as follows.

  1. Pick $q_i \overset{R}{\leftarrow} \mathbb{Z}_p$ for each $\mu_i \in w^*$ unless $\mu_i = \mu^*$.
  2. Compute $U_{\mu_i} = Q^{q_i(x+h_i)/y}$ for each $\mu_i \in w^*$ unless $\mu_i = \mu^*$ as before.
  3. Define $S = \{\mu^*\} \cup \bar{S}$ with $|S| = d$ as before and compute for all elements of S by considering the equation

  $$q_1\Delta_{\mu_1,S}(0) + q_2\Delta_{\mu_2,S}(0) + ... + q^*\Delta_{\mu^*,S}(0) = 1/x \tag{6.1}$$

  which implies

  $$Q^{xq^*\Delta_{\mu^*,S}(0)} = Q \cdot Q^{x(-\sum_{i=1}^{d-1} q_i\Delta_{\mu_i,S}(0))} \tag{6.2}$$

  and finally take the power $1/\Delta_{\mu^*,S}(0)$ of both sides of (6.2) so that $Q^{xq^*} = (Q \cdot Q^{x(-\sum_{i=1}^{d-1} q_i\Delta_{\mu_i,S}(0))})^{1/\Delta_{\mu^*,S}(0)}$. Set $U_{\mu^*} = Q^{xq^*/y}$ for $\mu^* \in w^*$.
  4. Choose a random $\beta \in \{0,1\}$ and set m_β as the challenge message.
  5. Set the ciphertext to $(w^*, \{U_{\mu_i} : \mu_i \in w^*\}, m_\beta\hat{e}(Q,Q)^r)$.

  The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(0) = 1/x$. Here, each $q_i = q(\mu_i)$ and $q^* = q(\mu^*)$ for $\mu_i \in w^*$.

For the set S, where $S = \{\mu^*\} \cup \bar{S}$ and $\bar{S}$ denotes the $d-1$ elements of the challenge identity, for which A knows the secret key components. Thus, $\bar{S} \subseteq w^*$ with $|\bar{S}| = d-1$, the assignments for the features $q(\mu_i) \in \bar{S}$ are valid and computable due to the following.

By considering the equation (6.1) and the case that B flips the coin as $b = 0$ in **Phase 2** of the game, we have $\hat{e}(g,g)^r = \hat{e}(g,g)^{1/x}$. In particular, if the challenger of B flips the coin as $b = 0$, then $\hat{e}(g,g)^r = \hat{e}(g,g)^{1/x}$, from which B computes $T = \hat{e}(Q, Q^r) = \hat{e}(Q, Q^{1/x}) = \hat{e}(g, g^{1/x})^{c_0^2} T_0$, where $T_0 = \hat{e}(T', Q \cdot g^{c_0})$ as defined before. Hence, by knowing Q and Q^x, we can compute

$$Q^{q_1 \Delta_{\mu_1,S}(0) + q_2 \Delta_{\mu_2,S}(0) + ... + q^* \Delta_{\mu^*,S}(0)} = Q^{1/x} \tag{6.3}$$

which implies $Q^{x(q_1 \Delta_{\mu_1,S}(0) + q_2 \Delta_{\mu_2,S}(0) + ... + q^* \Delta_{\mu^*,S}(0))} = Q$. We can write

$$Q^{x(\sum_{\mu_i \in S} q_i \Delta_{\mu_i,S}(0))} = Q^{x(\sum_{\mu_i \in \bar{S}} q_i \Delta_{\mu_i,S}(0) + q^* \Delta_{\mu^*,S}(0))} = Q. \tag{6.4}$$

By denoting $Z = (-\sum_{\mu_i \in \bar{S}} q_i \Delta_{\mu_i,S}(0))$, and multiplying both sides of (6.4) with $(Q^x)^Z$, we obtain $Q^{x(q^* \Delta_{\mu^*,S}(0))} = Q \cdot (Q^x)^Z$, which implies $Q^{xq^*} = [Q \cdot Q^{xZ}]^{1/\Delta_{\mu^*,S}(0)}$ By taking the power of $1/y$ of both sides $Q^{xq^*/y} = [Q \cdot Q^{xZ}]^{1/(y\Delta_{\mu^*,S}(0))}$.

In summary, we can compute the component of the challenge ciphertext associated to the feature μ^* by using the known values of Q, Q^x, q_i, Z but without knowing $q(0) = 1/x$ and $q^* = q(\mu^*)$. Thus, if the adversary A with advantage ϵ picks the set S, where $S = \{\mu^*\} \cup \bar{S}$, then B can distinguish the decisional k-BDHI challenge with advantage ϵ' using A.

Otherwise, if $b = 1$, then $\hat{e}(g,g)^r$ is a random element, thus $T = \hat{e}(Q, Q^r)$ is also a random element computed the same way as above. Since $\hat{e}(g,g)^r$ is random, the session key $\hat{e}(Q,Q)^r)$ will be a random element of $\mathbb{F}$ from the adversaries view and the challenge ciphertext contains no information about m_β.

- **Phase 5**: Identical to BIO-IBE.
- **Phase 6**: At some point, A responds with the guess β' for the underlying plaintext m_β, which could only be computed from

$$m_\beta = W^* / \prod_{\mu_i \in S} \hat{e}(U_{\mu_i}, D_{\mu_i}^{\mathsf{ID}^*})^{\Delta_{\mu_i,S}(0)}.$$

If $\beta = \beta'$, B will output $b' = 0$ to indicate that it was given a decisional k-BDHI tuple otherwise it will output $b' = 1$ to indicate it was given a random tuple.

If the challenger of B flips the coin as $b = 0$ in **Phase 2** of the game, the only way for A to have any advantage in this game is when A computes

$$R = \prod_{\mu_i \in S} \hat{e}(U_{\mu_i}, D_{\mu_i}^{\mathsf{ID}^*})^{\Delta_{\mu_i,S}(0)}$$

Here, the set $S \subseteq w^*$ with $|S| = d$ denotes the d elements of the challenge identity, where A already knows the secret key components of $d-1$ of the set S. Hence, we assume that a clever attacker has to pick another element of the challenge identity w^* other than the $d-1$ elements to compose the set S.

If the challenger of B flips the coin as $b = 0$ in **Phase 2** of the game, the only way for B to have any advantage in this game is when A computes

$$\begin{aligned} R^* &= \prod_{\mu_i \in S} \hat{e}(U_{\mu_i}, D_{\mu_i}^{\mathsf{ID}^*})^{\Delta_{\mu_i,S}(0)} \\ &= \hat{e}(Q, Q^{y/x})^{q^* \Delta_{\mu^*,S}(0)} \cdot \Lambda \end{aligned}$$

Here, $\Lambda = \prod_{\mu_i \in \bar{S}} \hat{e}(Q,Q)^{q_i \Delta_{\mu_i,S}(0)}$ as before.

-If A picks the set $S = \{\mu^*\} \cup \bar{S}$ for the computation of the session key:

- $\Pr[\beta = \beta' | b = 0] = \epsilon + \frac{1}{2}$. Since B guesses $b' = 0$ when $\beta = \beta'$, we have $\Pr[b' = b | b = 0] = \epsilon + \frac{1}{2}$
- $\Pr[\beta \neq \beta' | b = 1] = \frac{1}{2}$. Since B guesses $b' = 1$ when $\beta \neq \beta'$, we have $\Pr[b' = b | b = 1] = \frac{1}{2}$

  Thus, denoting the event that A picks the set S as pick$S$, we have

  $\Pr[b' = b] = \Pr[b' = b | b = 0]\Pr[b = 0] + \Pr[b' = b | b = 1]\Pr[b = 1]$
  $= \frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2}\frac{1}{2} = \frac{1}{2} + \frac{1}{2}\epsilon$

-Else if A does not pick the set $S = \{\mu^*\} \cup \bar{S}$ for the computation of the session key:

- $\Pr[\beta = \beta' | b = 0] = \frac{1}{2}$. Since B guesses $b' = 0$ when $\beta = \beta'$, we have $\Pr[b' = b | b = 0] = \frac{1}{2}$
- $\Pr[\beta \neq \beta' | b = 1] = \frac{1}{2}$. Since B guesses $b' = 1$ when $\beta \neq \beta'$, we have $\Pr[b' = b | b = 1] = \frac{1}{2}$

  Thus, in this case,

$$\begin{aligned}\Pr[b' = b] &= \Pr[b' = b|b = 0]\Pr[b = 0] + \Pr[b' = b|b = 1]\Pr[b = 1] \\ &= \tfrac{1}{2}\tfrac{1}{2} + \tfrac{1}{2}\tfrac{1}{2} = \tfrac{1}{2}\end{aligned}$$

Combining the two cases, the overall advantage of B in the decisional k-BDHI game is

$$\begin{aligned}\Pr[b = b'] - \frac{1}{2} &= \Pr[\text{pick}S]\Pr[b = b'|\text{pick}S] + \Pr[\neg\text{pick}S]\Pr[b = b'|\neg\text{pick}S] - \frac{1}{2} \\ &= \Pr[\text{pick}S]\big[\frac{1}{2} + \frac{1}{2}\epsilon\big] + \Pr[\neg\text{pick}S]\big[\frac{1}{2}\big] - \frac{1}{2} \\ &= \frac{1}{n-d+1}\big[\frac{1}{2} + \frac{1}{2}\epsilon\big] + \frac{n-d}{n-d+1}\big[\frac{1}{2}\big] - \frac{1}{2} \\ &= \frac{1}{n-d+1}\big[\frac{1}{2}\epsilon + \frac{n-d+1}{2}\big] - \frac{1}{2} \\ &= \frac{\epsilon}{2(n-d+1)} + \frac{1}{2} - \frac{1}{2} = \frac{\epsilon}{2(n-d+1)}\end{aligned}$$

As before, we should note that BIO-IBE can be proven in the Fuzzy Selective-ID model, then the reduction cost will change slightly due to the factor of $\binom{n}{d}$ since the adversary A will have only $\binom{n}{d}$ different choices for the set S instead of $(n-d+1)$ choices. By replacing $n-d+1$ with $\binom{n}{d}$, we obtain the reduction cost for the Fuzzy Selective-ID model. B's running time is identical to the previous proof. □

6.6 BIO-IBE in the standard model for small universe

In [Sahai and Waters, 2005], the authors give a construction for the small universe of attributes (i.e. biometric features), where the universe is denoted by $\mathbb{U}$. The reason for this limitation is that, the public parameter size of that scheme is linear in the size of the universe $\mathbb{U}$. If we want to prove the security of BIO-IBE in standard model, then we do not have this limitation, however, in order to simulate the private key extraction queries, we have to compute for all the possible distinct identities $|\mathcal{ID}|$ in the system, namely for $\binom{|\mathbb{U}|}{d} \leq |\mathcal{ID}| \leq \binom{|\mathbb{U}|}{n}$ different identites, the associated private keys beforehand in order to simulate the public parameters of the system and the private key extraction oracle. In other words, we have to compute all the possible private keys beforehand, instead of computing them during each query. This is possible since we use an ordinary hash function, which is efficiently computable. Thus, we have an

upper bound on the total number of queries made to the private key extraction oracle, which is equal to the total number of identites that can be obtained from the small universe of attributes. Thus, our method does not impose a restriction as in the case of bounded CCA secure schemes [Hanaoka and Imai, 2006, Cramer et al., 2007], which have a bound on the decryption queries that should be known before the setup. As a result, we only require a small universe of attributes for BIO-IBE in the standard model, which is more efficient compared to the scheme in [Sahai and Waters, 2005] due to the (small) size of the public parameters.

Theorem 6.5. *Suppose there exists a polynomial time adversary A with advantage ϵ that can break the scheme BIO-IBE in the stronger security model by making q_{ex} private key extraction queries. Then there exists a polynomial time algorithm B that solves the decisional k-BDHI problem with $k = q_{ex} + 1$ and advantage $\frac{\epsilon}{2(n-d+1)}$.*

Proof. Assume that a polynomial time attacker A breaks our scheme, then using A, we show that one can construct an attacker B solving the decisional k-BDHI problem. Suppose that B will distinguish between the distributions $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^{1/x})$ and $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^r)$ using A as follows.

- **Phase 1**: Identical to BIO-IBE.
- **Phase 2**: We first let the challenger set the groups $\mathbb{G}$ and $\mathbb{F}$ with an efficient bilinear map, $\hat{e}$ and generator g. The challenger flips a fair binary coin b outside of B's view. If $b = 0$, the challenger returns B $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^{1/x})$ otherwise it returns $(g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^r)$. We assume the universe of features (i.e. attributes) is defined. After A declares the challenge identity $w^* = \{\mu_1^*, ..., \mu_n^*\}$, B picks a random $\mu^* \in w^*$ and simulates the public parameters for A as follows:

  B computes for all the possible different biometric identities that can be generated from the universe $\mathbb{U}$ of attributes, i.e. for $|\mathcal{ID}|$ different identities, the associated private keys. In particular, B first computes for each different biometric identity $w = \{w_1, ..., w_n\}$ such that $|w^* \cap w| < d$, $H_1(\mu_j, \mathsf{ID}) = h_j$ for each $\mu_j \in w$ using the cryptographic hash function H_1. Since H_1 is not a random oracle anymore, we cannot arrange its outputs as in the previous proofs. Thus, we slightly modify the calculation of Q as follows.

  First, B selects $y \in \mathbb{Z}_p^*$ at random and sets $f(z) = \prod_{j=1}^{k-1}(z+h'_j)$, where $h'_j = h_j - h_0$ are non-zero values. Here $h_0 = H_1(\mu^*, \mathsf{ID}^*)$ is the computed hash value of the feature μ^*. Again, $f(z)$ could be written as $f(z) = \sum_{j=0}^{k-1} c_j z^j$. The constant term c_0 is non-zero because $h'_j \neq 0$ and c_j are computable from h'_j.

  B computes Q and Q^x in the same way as it is shown in the previous proofs.

Next, $f_j(z) = \frac{f(z)}{z+h'_j} = \sum_{v=0}^{k-2} d_{j,v} z^j$ for $1 \leq j < k$ and $Q^{1/(x+h'_j)} = g^{f_j(x)} = \prod_{v=0}^{k-2} (g^{x^v})^{d_{j,v}}$ is computed. Finally, B sets T', T_0 as before.

B returns A the public parameters $(Q, \hat{e}, \mathbb{G}, \mathbb{F}, P_{pub1}, P_{pub2}, H_1, FE)$, where $P_{pub1} = Q^{(x-h_0)/y}$, $P_{pub2} = Q^{1/y}$ As one can notice, the hash value of $H_1(\mu^*, \mathsf{ID}^*) = h_0$ and the associated secret key of $\mu^* \in w^*$ is $Q^{y/x}$.

With this modification, we are able to compute the secret key components of each feature as $Q^{y/(x+h'_j)} = Q^{y/(x+h_j-h_0)}$.

The correctness can be verified by taking a component of the ciphertext $U_j = [P_{pub1} \cdot (P_{pub2})^{h_j}]^{q_j} = [Q^{(x-h_0)/y} \cdot Q^{h_j/y}]^{q_j} = Q^{q_j(x+h_j-h_0)/y}$.

When we perform the decryption of each component using the bilinear pairing $\hat{e}(U_j, D_j) = \hat{e}(Q^{q_j(x+h_j-h_0)/y}, Q^{y/(x+h'_j)}) = \hat{e}(Q^{q_j(x+h_j-h_0)/y}, Q^{y/(x+h_j-h_0)}) = \hat{e}(Q, Q)^{q_j}$.

Here, FE denotes the fuzzy extraction algorithm that is used to extract a unique binary string ID. For simplicity of the proof, each ID associated to the biometrics w is extracted from the same biometric trait but using a different feature extraction method and thus, a unique ID is computed for each different biometrics w to bind the private key components to the user. (Clearly, multi biometric traits can also be employed, then again, the biometric ID associated to the user identity from its own universe of attributes.) Since the computations are performed for each unique identity represented by either a single or multi biometric trait, the total number of private key components that are calculated beforehand could be approximately 2^{60}. Hence, we allow for private key extraction queries for a bounded number, which is a reasonable assumption in case of a small universe of attributes.

After computing each hash value, B computes the private keys as before. The only difference is that, instead of assuming H_1 as a random oracle, we compute the real values of H_1 and the associated private key components. The rest of the computations are performed as in **Phase 2** of BIO-IBE's security reduction presented in section 6.5. The only difference is the replacement of random oracles with real hash functions, i.e. the randomly picked h_i values are replaced with the real values of the H_1 function.

- **Phase 3**: Upon receiving a query (w', ID') such that $|w' \cap w^*| < d$, B simulates the private key extraction queries of A by returning $Q^{y/(x+h'_j)}$ associated to each h_j computed before hand for each $\mu_i \in w'$. For the challenge identity (w^*, ID^*), B returns the $d-1$ private key components $D_{\mu_i^*}^{\mathsf{ID}^*} = Q^{y/(x+h'_i)}$ that A chooses except for the component associated to the attribute μ^*.

- **Phase 4**: Upon receiving the messages (m_0, m_1) with $|m_0| = |m_1|$, B generates the challenge c^* as in the challenge phase of section 6.5.

- **Phase 5**: B answers A's private key extraction queries as in section 6.5. The only condition on the private key extraction queries is that the attacker A cannot query the challenge private key for the remaining $n - d + 1$ components of the challenge identity.
- **Phase 6**: At some point, A responds with the guess β' for the underlying plaintext m_β. If $\beta = \beta'$, B will output $b' = 0$ to indicate that it was given a decisional k-BDHI tuple otherwise it will output $b' = 1$ to indicate it was given a random tuple. The analysis of the reduction cost is the same as in section 6.5.

□

6.7 A New Denial of Service Attack

To prevent collision attacks,BIO-IBE requires the public storage of the value PAR, which is the information needed for error-tolerant reconstruction of the biometric identity string ID and subsequent fuzzy extraction. Since the encryption is performed by combining each biometric feature μ_i with the biometric identity ID of the receiver, the presence of an active adversary who maliciously alters the public string PAR leads the sender to use a wrong public key for the encryption due to a different identity string computed by the fuzzy extractor. By the malicious modification of the public value PAR, an adversary cannot gain any secret information but the receiver of the ciphertext either cannot decrypt it or he obtains a wrong plaintext upon decryption.

The first idea to solve this problem is using a robust fuzzy extractor, which is resilient to modification of the public value PAR [Boyen et al., 2005]. However, the robust fuzzy sketches/fuzzy extractors described in [Boyen et al., 2005] assumes the biometrics as secret data and replaces the value PAR with $\mathsf{PAR}^* = \langle \mathsf{PAR}, H(b, \mathsf{PAR}) \rangle$, where H is a hash function. Since the adversary knows the biometric data b, he can easily modify the value PAR^* by computing a valid hash value, hence, the sender cannot detect the modification of the public value. Another solution could be that the user store the public value PAR in his smart card and present this to the sender during the biometric measurement. However, this defeats the purpose of biometric IBE in the first place, which enables an unprepared user to encrypt in an ad hoc meeting, where the users do not have their smartcards with them.

In [Liu et al., 2007], a similar attack called as Denial-of-Decryption (DoD) Attack in the context of certificateless encryption is defined, whose nature is similar to the well known DoS Attack. In DoD, the attacker can modify the public key of the receiver since the authenticity of the public key is not provided. The authors provide the solution against this attack by requiring the receiver to sign his public key using the private key

associated to a certificateless signature scheme and store the public value together with the signature in a public storage. When the sender wants to encrypt a message, he first verifies the signature on the public value and upon validation, he starts encryption.

In order to prevent a DoS attack on our scheme, we follow a similar approach requiring the receiver of the ciphertext or the PKG to sign the public value PAR using an efficient IBS scheme, and publish both values. A good candidate is the IBS scheme of [Barreto et al., 2005] presented in section 2.10.5 of the background chapter, whose public parameters are almost equal to the parameters of BIO-IBE since both schemes are based on the same Sakai-Kasahara Key Construction method. The only difference in the public parameters of [Barreto et al., 2005] is the use of an arbitrary string such as an e-mail address as the identity and two hash functions, which have a different domain. If the signature is applied by the PKG, then the identity information is taken as the identity of the PKG. Alternatively, the receiver of the ciphertext can sign the PAR using the biometric IBS scheme described in the previous section. Consequently, the signature on the public value PAR makes the modified BIO-IBE immune against a DoS attack.

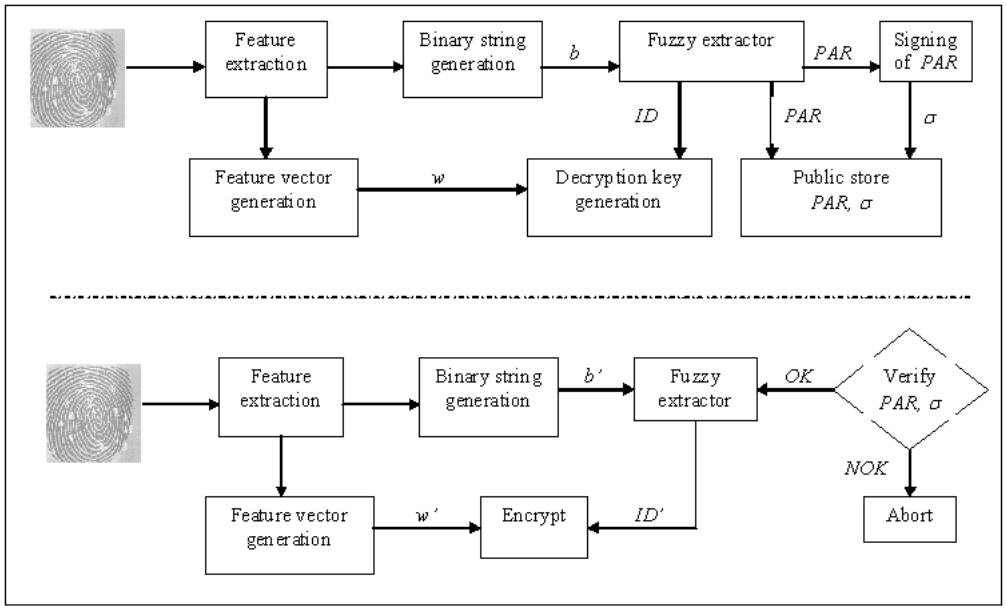

Figure 6.1: Modified BIO-IBE Flow diagram for single-biometric trait

After verifying the signature on the public value PAR, the sender can encrypt a message. The only additional cost for the sender is caused by the verification of the signature,

namely, one exponentiation in $\mathbb{G}$ and in $\mathbb{F}$ and one bilinear pairing if the PKG signes PAR. Despite the additional bilinear pairing computation for the sender, our scheme is still more efficient compared to existing fuzzy IBE schemes due to the removal of n MapToPoint hash computations from each phase. Moreover, the scheme of [Barreto et al., 2005] is currently the most efficient pairing-based IBS scheme in the literature, which is suitable for the modified BIO-IBE. Alternatively, the IBS scheme of [Galindo and Garcia, 2009] can be used for the signing of the PAR, which has an additional verification cost of only 1.5 exponentiation in $\mathbb{G}$ for the encryptor.

As a final note, the key escrow problem inherent with IBE systems also effects fuzzy/biometric IBE systems. Basically, key escrow means that the PKG can decrypt any message as it generates all the secret keys of the users. However, applying certificateless encryption techniques to BIO-IBE avoids this problem, where certificateless encryption is designed as a new system in [Al-Riyami and Paterson, 2003] to solve key escrow problem of IBE and thus avoids the drawbacks of PKE and IBE by combining the functionality of the both systems. In certificateless encryption, a sender requires both the receiver's identity and a public key value produced by the receiver to encrypt a message. For BIO-IBE, key escrow is eliminated by individualizing the value y of the master secret key $ms = x, y$. In particular, if each user U_i in the system selects a unique y_i value and the signature of the receiver is applied on the public values $g^{x/y_i}, g^{1/y_i}$ and PAR instead of only on PAR, the PKG cannot decrypt a message since the new master secret key is $ms = x$ and the decryption keys are of the form $g^{y_i/(x+h_i)}$, where y_i is only known to the user U_i and PKG only generates $g^{1/(x+h_i)}$.

6.7.1 The modified BIO-IBE

Here, we summarize the algorithms of our new scheme, which is obtained by modifying the Key Generation and Encrypt algorithms of BIO-IBE. To avoid the key escrow problem, the technique explained above can also be applied.

- Setup: The parameters of the scheme are generated as in BIO-IBE. Two additional hash functions $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_4 : \{0,1\}^* \times \mathbb{F} \rightarrow \mathbb{Z}_p^*$ are required for the signature scheme as described before.

- Extract: First, a user's biometric attributes w are obtained from the raw biometric information using a reader and the feature extractor and each attribute $\mu_i \in w$ is associated to a unique integer in $\mathbb{Z}_p^*$ as before. Besides, the identity string ID$= H(b)$ is calculated from the biometric template b using a fuzzy extractor, which also outputs the public value PAR that is used in the reconstruction of the ID by the sender (or encryptor). Next, PAR is signed by the PKG or the receiver of the ciphertext.

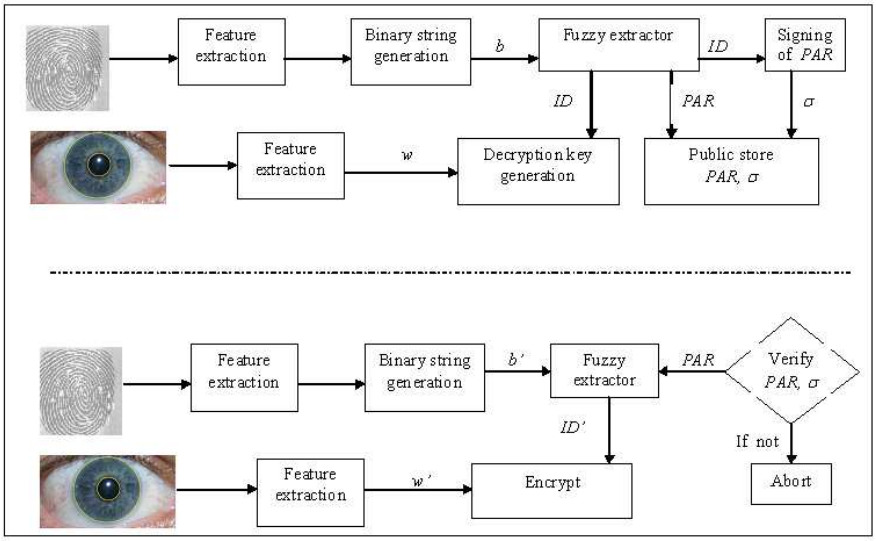

Figure 6.2: Modified BIO-IBE Flow diagram for two biometric traits

Given a user's biometric attributes w and ID, the PKG returns $D_{\mu_i}^{\mathsf{ID}} = g^{y/(x+H_1(\mu_i,\mathsf{ID}))} = g^{y/(x+h_i)}$ for each $\mu_i \in w$. Finally, the PAR and the signature σ are stored in a public file.

- Encrypt: The sender obtains a biometric reading of the receiver together with the signed public parameter PAR, verifies the signature on the PAR, extracts the feature vector w' and computes $\mathsf{ID}' = \mathsf{Rep}(b', \mathsf{PAR})$. Here, if $\mathsf{dis}((b, b') < t$, then $\mathsf{ID} = \mathsf{ID}'$. The encryption of $m \in M$ using ID' and w' is identical to BIO-IBE.
- Decrypt: The same algorithm as in BIO-IBE.

Lemma 6.1. *The modified BIO-IBE is immune against a DoS attack under the existential unforgeability of the IBS scheme of [Barreto et al., 2005].*

6.8 Comparison

We summarize in the following tables the properties of the modified BIO-IBE and compare the computational costs of each algorithm used in the schemes that are provably secure in ROM. The abbreviations used in figure 6.3 are listed in table 6.3. In Table 6.6, we compare the properties of fuzzy IBE schemes in the standard model for small universe of features. The public parameter size of BIO-IBE consists of the size of the master public key, whereas small universe construction of [Sahai and Waters, 2005] has additionally $|\mathbb{U}|$ elements in the public parameters, where $|\mathbb{U}|$ denotes the size of the universe of features. We note that our first construction for ordered biometrics OrdFIBE is even more efficient than BIO-IBE at the cost of $n-1$ additional public parameters. Obviously, all of our constructions are more efficient than existing fuzzy IBE schemes in terms of the key generation and decryption algorithms. Compared to BIO-IBE, the encryption algorithm of the modified BIO-IBE requires additionally one bilinear pairing and two exponentiations due to the signature verification on the PAR, which makes our scheme secure against DoS attacks. However, this verification is performed once and for all similar to the fuzzy extraction operation. Below, figure 6.3 shows the comparison of the costs if the signing of PAR is performed using the IBS scheme of [Barreto et al., 2005] and using the IBS scheme of [Galindo and Garcia, 2009], which avoids the additional pairing computation to sign the public value PAR. Although the fuzzy extraction of the unique identity string and the verification of the signature on it is performed only once, the cost analysis is made by considering the operations for these two one-time computations. The relations between the parameters used in the cost analysis are $T_e < T'_e < T_p$ and $d < n$.

Table 6.1: Properties of Fuzzy IBE Schemes in the Standard Model

Scheme	Decisional Problem	Security Model	Universe Size	Size of public parameters	Multi Modal
SW for Small Universe [Sahai and Waters, 2005]	MBDH	Standard	Small	linear in $\|\mathbb{U}\|$ + master public paramters	×
BIO-IBE	k-BDHI	Standard	Small	master public parameters	✓

Table 6.2: Properties of Various Fuzzy IBE Schemes secure in ROM

Scheme	Assumption	Hash Function	Multi-Modal Application
SW-RO [Pirretti et al., 2006]	DBDH	MaptoPoint	×
EFIBE-I[Baek et al., 2007]	DBDH	MaptoPoint	×
EFIBE-II[Baek et al., 2007]	DBDH	MaptoPoint	×
Fuzzy-BF[van Liesdonk, 2007]	BDH	MaptoPoint	×
BIO-IBE	k-BDHI	Regular	✓

Table 6.3: Abbreviations

$\|S\|$	bit size of an element in the set S	n	number of features of a user
d	error tolerance parameter	FE	time for the fuzzy extraction process
T_e	time for a single exponentiation in $\mathbb{G}$	T'_e	time for a single exponentiation in $\mathbb{F}$
T_H	time for MaptoPoint hash computation	T_p	time for a single pairing operation
T_m	time for a single multiplication in $\mathbb{G}$	T'_m	time for a single multiplication in $\mathbb{F}$
T_i	time for a single inverse operation in $\mathbb{Z}_p$	T'_i	time for a single inverse operation in $\mathbb{F}$
k_1	output size of an ordinary hash function	CRC	time for a checksum computation

Figure 6.3: Comparison to various fuzzy IBE schemes secure in ROM

	Size of the Secret key	Size of the Ciphertext	Cost of Key Generation	Cost of Encrypt	Cost of Decrypt
SW-RO Pirretti et al. [2006]	$2n\|\mathbb{G}\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{F}\|$	$n(T_H+T_m+3T_e)$	$n(T_e+T_H)$ $+2T_e+T_p+T'_m$	$d(2T_e+T_m+T_p)$ $+T_p+T'_i+T'_m$
EFIBE-I Baek et al. [2007]	$2n\|\mathbb{G}\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{F}\|$	$n(T_H+2T_e)$	$n(T_e+T_m+T_H)$ $+2T_e+T_p+T'_m$	$d(2T_e+T_m+T_p)$ $+T_p+T'_i+T'_m$
EFIBE-II Baek et al. [2007]	$2n\|\mathbb{G}\|$	$(n+1)\|\mathbb{G}\|+\|\mathbb{F}\|$	$n(T_H+T_m+2T_e)$	$n(T_e+T_H)$ $+2T_e+T_p+T'_m$	$d(2T_e+T_m+T_p)$ $+T_p+T'_i+T'_m$
Fuzzy-BF van Liesdonk [2007]	$n\|\mathbb{G}\|$	$\|\mathbb{G}\|+nk_1$	$n(T_H+T_e)$	$n(T_p+T_H+T'_e)$ $+nCRC$	$n(T_p)$
OrdFIBE$^+$	$n\|\mathbb{G}\|$	$n\|\mathbb{G}\|+k_1$	$n(T_e+T_i)$	$n(2T_e+T_m)+T_p$	$d(T_e+T_p)$
BIO-IBE Sarier [2008]	$n\|\mathbb{G}\|$	$n\|\mathbb{G}\|+k_1$	$n(T_e+T_i)$ $+FE$	$n(2T_e+T_m)$ $+T_p+FE$	$d(T_e+T_p)$
Modified BIO-IBE Sarier [2011]	$(n+1)\|\mathbb{G}\|$	$n\|\mathbb{G}\|+k_1$	$n(T_e+T_i)$ $+FE+T_e+T'_e$	$n(2T_e+T_m)+T_e$ $2T_p+FE+T'_e$	$d(T_e+T_p)$
Modified BIO-IBE* Sarier [2011]	$(n+1)\|\mathbb{G}\|$	$n\|\mathbb{G}\|+k_1$	$n(T_e+T_i)$ $+FE+T_e$	$n(2T_e+T_m)$ $T_p+FE+1.5T_e$	$d(T_e+T_p)$

OrdFIBE$^+$: Our first construction for ordered biometrics; *: PAR is signed with the scheme of [Galindo and Garcia, 2009]

6.9 Conclusion

In this chapter, we propose efficient biometric IBE schemes that are provably secure in the ROM and standard model depending on the size of the universe of attributes and the representation of the attributes of the user. We start with an efficient fuzzy IBE scheme denoted as OrdFIBE that is restricted to ordered biometrics or attributes that can be grouped/ordered. Thus, OrdFIBE can be generalized to attribute-based encryption. Next, we describe BIO-IBE that is applicable to any type of biometric modality if combined with a fuzzy extractor to avoid collision attacks. For the two universe sizes, BIO-IBE is currently the most efficient biometric IBE scheme with a tight reduction cost among the other pairing based fuzzy IBE schemes applied for biometric identities. Besides, BIO-IBE is the first biometric IBE scheme that is applicable for multi-modal biometrics as opposed to the claim in [Zhang et al., 2011], which combines fuzzy extractors with multi-biometric encryption. Key escrow problem inherent in all IBE (and thus fuzzy IBE) systems can also be solved with a simple modification on BIO-IBE.

Chapter 7

Anonymous Biometric IBE without Pairings

In this chapter, we present a novel framework for the generic construction of biometric Identity Based Encryption (IBE) schemes, which do not require bilinear pairings and result in more efficient schemes than existing fuzzy IBE systems implemented for biometric identities. Also, we analyze the security properties that are specific to biometric IBE namely anonymity, and introduce a new notion for biometric IBE called as identity privacy. Considering these notions, we present generic constructions for biometric IBE and ID-KEM based on weakly secure anonymous IBE schemes, error correcting codes and generic conversion schemes in order to obtain highly secure anonymous biometric IBE schemes. As different from the current fuzzy/biometric IBE systems, our scheme relies on the standard quadratic residuosity (QR) assumption instead of (stronger) bilinear assumptions. In fact, it is the first biometric IBE scheme without depending on pairings. Finally, we describe concrete applications of our framework and compare them to the existing fuzzy IBE systems in terms of time complexity and bandwidth. We design our new constructions for any type of biometrics that can be represented as an ordered set of features (i.e. a sequence of n feature points) such as face, online handwritten signatures, iris, voice etc. [Li et al., 2006]. However, if anonymity property is not required, then our system is applicable for any type of biometrics since we can allow the attachment of the biometrics to the ciphertext as in fuzzy IBE schemes. Collision attacks are prevented using our new method described in the previous chapter. This chapter is based on the paper [Sarier, 2010d], which received the best student paper award.

7.1 Introduction

As described in the previous chapter, encryption using biometric inputs as identities is provided with fuzzy IBE, since the error-tolerance property of a fuzzy IBE scheme is precisely what allows for the use of biometric identities, which inherently will have some noise each time they are sampled. A fuzzy IBE scheme allows for a private key of the receiver's identity set w, to decrypt a ciphertext encrypted with a similar identity set, w', if and only if the set w and set w' are close to each other as measured by the 'set overlap' (i.e. set intersection) distance metric. This is actually the same metric used in fuzzy vault, where Bob, using an unordered set w, can unlock the vault (and access the hidden secret data) only if w overlaps with w' to a large extent. In current fuzzy IBE schemes, the private key components are generated by combining the values of a unique polynomial evaluated on each attribute with the master secret key. This way, different users, each having some portion of the secret keys associated to the attributes of a given ciphertext c cannot collude to decrypt c, which is defined as collusion resistance. The basic fuzzy IBE schemes guarantee a weak level of security for identity based setting i.e. Indistinguishability against Chosen Plaintext Attack (IND-ID-CPA), but they could be combined with well-known generic conversion systems to obtain a high level of security i.e. Indistinguishability against Chosen Ciphertext Attack (IND-ID-CCA). Besides, the biometrics is considered as public information, hence the compromise of the biometrics does not affect the security of the system. Thus, in existing systems, biometrics w' of the receiver is sent together with the corresponding ciphertext so that the receiver with biometrics w can determine the common features between w and w' in order to apply the correct secret key components. Clearly, this could effect the privacy of the user's actions if we consider the notion of anonymity for IBE systems.

7.1.1 Motivation and Contributions

Currently, the secrecy of biometric data is viewed with skepticism since it is very easy to obtain biological information such as fingerprint, iris or face data through fingerprint marking or using a camcorder. However, biometrics is a sensitive information, as in the case of biometric remote authentication, it should not be easy to obtain the biometric data by compromising the central server, where the biometrics of each user is often associated with his personal information. In particular, a user could use its biometrics on a number of applications such as identification, authentication, signing, etc. Thus, the secrecy of identity-biometrics relation should be maintained, which is defined as identity privacy [Bringer et al., 2007b, Tang et al., 2008]. Current fuzzy IBE and biometric IBE systems do not consider anonymity and privacy of user biometrics at the same time, hence, it is vital to describe an efficient and anonymous error-tolerant encryption system for biometric identities in order to avoid traceability of the user's

actions. Although the fuzzy IBE scheme of [van Liesdonk, 2007] provides anonymity, the scheme combines each biometric attribute with the identity (i.e. Name, e-mail address) of the user to avoid the collusion attacks. This approach is not only against identity privacy but also against the main principle of fuzzy IBE or biometric IBE, where the identity of the user should only consist of his biometric data.

In this chapter, we present a novel framework for the generic construction of biometric IBE schemes, which do not require bilinear pairings and result in more efficient schemes than existing fuzzy IBE systems implemented for biometric identities. Also, we analyze the security properties that are specific to biometric IBE namely anonymity, and introduce a new notion for biometric IBE called as identity privacy. Considering these notions, we present generic constructions for biometric IBE and ID-KEM based on weakly secure anonymous IBE schemes, error correcting codes and generic conversion schemes in order to obtain highly secure anonymous biometric IBE schemes. As different from the current fuzzy/biometric IBE systems, our scheme relies on the standard quadratic residuosity (QR) assumption instead of (stronger) bilinear assumptions. In fact, it is the first biometric IBE scheme without depending on pairings. Finally, we describe concrete applications of our framework and compare them to the existing fuzzy IBE systems in terms of time complexity and bandwidth. We may add that the importance of relying on the standard quadratic residuosity assumption should not be underestimated. In fact, this is what motivated the recent work in IBE without pairings. We design our new constructions for any type of biometrics that can be represented as an ordered set of features (i.e. a sequence of n feature points) such as face, online handwritten signatures, iris, voice etc. [Li et al., 2006]. However, if anonymity property is not required, then our system is applicable for any type of biometrics since we can allow the attachment of the biometrics to the ciphertext as in fuzzy IBE schemes.

We start with analyzing the security properties of biometric IBE schemes and review our new method for preventing collusion attacks that we introduced in the previous chapter. Next, we present generic constructions for biometric IBE and ID-KEMs that convert any weakly secure (i.e. IND-ID-CPA) anonymous IBE scheme encrypting a message bit by bit to a highly secure (i.e. IND-ID-CCA) biometric IBE scheme. To build the new generic constructions, we combine fuzzy sketches, error correcting codes and/or modify well known generic conversion schemes to function in the error-tolerant setting. Also, we will describe concrete applications of our generic constructions using anonymous IBE schemes [Boneh et al., 2007, Ateniese and Gasti, 2009] that encrypt each message bit by bit and do not depend on bilinear pairings. This construction allows for the design of biometric IBE schemes relying on weaker assumptions such as interactive QR or standard QR assumption. To avoid collusion attacks and to guarantee the security notions that we present, the anonymous IBE schemes are implemented for biometric identites using our new method described in the previous chapter that

combines each feature with a unique biometric string obtained via a fuzzy extractor as described in the previous chapter. Thus, we achieve more efficient and anonymous biometric IBE schemes compared to current fuzzy IBE systems when implemented for an ordered set of biometric features as in face, iris or voice biometrics (or any type of biometrics if anonymity is not required).

We first implement the anonymous IBE scheme of Boneh et al. [2007] for biometric identities consisting of an ordered set of features to obtain an IND-ID-CPA anonymous biometric IBE scheme in the standard model. Next, we input this anonymous biometric IBE into one of our generic constructions to obtain IND-ID-CCA secure anonymous biometric IBE scheme. To improve the efficiency further, we present another application based on the scheme of [Ateniese and Gasti, 2009] that is secure in ROM and compare it with the current fuzzy/biometric IBE systems in terms of computational cost of encryption/decryption.

As described in the previous chapter, multi-modal biometric identities are used in our new biometric IBE scheme to prevent collision attacks and to achieve higher security and accuracy as in multi-modal biometric identification.

7.1.2 Related Work

The first fuzzy IBE scheme is described by Sahai and Waters in [Sahai and Waters, 2005], where the size of the public parameters is linear in the number of the attributes of the system or the number of attributes (or features) of a user. More efficient fuzzy IBE [Baek et al., 2007, Furukawa et al., 2008], Attribute-Based Encryption (ABE) [Pirretti et al., 2006] and biometric IBE [Sarier, 2008, 2011b] schemes are achieved with short public parameter size by employing the Random Oracle Model (ROM). Except for the schemes in [Sarier, 2008, 2011b] that work with an ordinary hash function, the main disadvantage of the schemes in [Pirretti et al., 2006, Baek et al., 2007, van Liesdonk, 2007] is the use of the MapToPoint hash function, which is inefficient compared to the ordinary hash functions. To achieve IND-ID-CCA security, the authors of [Sahai and Waters, 2005, Baek et al., 2007, van Liesdonk, 2007] suggest to combine their schemes with well known generic conversion schemes such as Fujisaki-Okamoto [Fujisaki and Okamoto, 1999] or REACT [Okamoto and Pointcheval, 2001]. The only work that considers privacy of biometric attributes in fuzzy IBE is the master thesis of [van Liesdonk, 2007], which adapts the Boneh-Franklin IBE scheme [Boneh and Franklin, 2003] to function as an error tolerant IBE scheme in ROM. The main reason for using Boneh-Franklin IBE is that this scheme is anonymous [Abdalla et al., 2005]. The common property of all these fuzzy/biometric IBE schemes is the use of a number of bilinear pairing computations depending on the size of the receiver's attributes, which affects the efficiency of the system significantly. Recently, other anonymous IBE

schemes [Boneh et al., 2007, Ateniese and Gasti, 2009] for the standard IBE setting (i.e. non-biometric identities) are described, which do not require bilinear pairings and their security is based on the standard quadratic residuosity problem. Besides, these schemes encrypt a message bit by bit, thus they can be used to encrypt short session keys due to the large bandwidth consumption. To achieve IND-ID-CCA security, the schemes can implement the KEM/DEM construction of Bentahar et al. [Bentahar et al., 2008], which takes as input a weakly secure IBE scheme and a hash function to output an IND-ID-CCA secure KEM (Key Encapsulation Mechanism). Finally, the KEM is combined with an IND-CCA secure DEM (Data Encapsulation Mechanism) to obtain an highly secure and efficient hybrid encryption system.

7.1.3 Organization

In section 7.2, we will state the definitions of the primitives that are used in our scheme. In section 7.3, a first attempt for a new construction is given, which provides a weak security level. Next, we analyze the security properties for biometric IBE in section 7.4 and present new generic constructions in section 7.5. Following these results, we also describe a biometric ID-KEM in section 7.6. Finally, we present two applications of our generic constructions in section 7.7 and compare our results to other fuzzy/biometric IBE systems.

7.2 Definitions and Building Blocks

Before stating the necessary definitions, we present some notations. Given a set S, $x \xleftarrow{R} S$ defines the assignment of a uniformly distributed random element from the set S to the variable x. $|S|$ denotes the bit-length of an element in S and μ_i denotes an attribute (or feature) of the biometric feature set w in the universe $\mathbb{U}$ of biometric attributes. Here, ID denotes any identity string such as Name, e-mail address, whereas ID denotes the identity string extracted from biometric information of the user. $\mathcal{ID}$ denotes the identity space, M denotes the message space and C denotes the ciphertext space, whereas $\mathcal{C}$ denotes an error-correcting code with $\mathcal{C}_e$ encoding and $\mathcal{C}_d$ decoding functions.

Definition 7.1. *(γ-uniformity). Let* Π =(Setup, Extract, Encrypt,Decrypt) *be an IBE scheme with space of randomness* COIN*. For a given* $ID \in \{0,1\}^*$*, the corresponding decryption key,* $m \in M$ *and* $c \in C$*, we define* $\gamma(ID, m, c)$ =Pr[$h \xleftarrow{R}$COIN : $c =$ Encrypt$^{\Pi}_{ID}(m; h)$]*. We say that* Π *is* γ*-uniform, if for any* $ID \in \{0,1\}^*$*,* $m \in M$ *and any* $c \in C$*, we have* $\gamma(ID, m, c) \leq \gamma$*.*

For an encryption scheme Π, γ-uniformity is defined as a parameter to evaluate the substantial coin space, i.e. how uniformly in how many large numbers the variants of the encryption of a message occur [Kitagawa et al., 2006]. For instance, e.g. it is $\gamma = 1$ for deterministic encryption and $\gamma = 2^{-k}$ for El Gamal encryption over $\mathbb{Z}_q$ with $k = |q|$.

7.2.1 Robust Sketch and Robust Fuzzy Extractors

As described in the background chapter and in section 6.2.5 of the previous chapter, the main idea of fuzzy sketches is given a public helper data $\mathsf{PAR} = c \oplus b$, one tries to correct the corrupted codeword $\mathsf{PAR} \oplus b' = c \oplus (b \oplus b')$, i.e. the correction is performed by combining the biometrics b' with the public value PAR to obtain the exact biometric template b. In [Boyen et al., 2005], the authors present a new attack for the secure sketches that assume biometrics as secret data. In particular, the presence of an active adversary who maliciously alters the public string PAR leads an adversary even to obtain the secret b' entirely depending on the utilized sketch or fuzzy extractor. This attack can be avoided by using a robust fuzzy extractor, which is resilient to modification of the public value PAR. The generic robust fuzzy sketch described in [Boyen et al., 2005] replaces the value PAR with $\mathsf{PAR}^* = \langle \mathsf{PAR}, H(b, \mathsf{PAR}) \rangle$, where H is a hash function. Although fuzzy/biometric IBE systems assume biometrics as public data, this robust construction can still be used for a different purpose that we show as below. By applying a strong extractor, one can convert any robust sketch to a robust fuzzy extractor.

7.3 A weakly secure Generic Construction

A possible design for an efficient biometric IBE scheme without using bilinear pairings is to combine any IBE scheme Π with an error correcting code $\mathsf{ECC}()$ and a robust sketch. Particularly, given Π=(Setup,$\mathsf{Extract}$, $\mathsf{Encrypt}$,$\mathsf{Decrypt}$) is an $\mathsf{IND\text{-}ID\text{-}CPA}$ secure IBE scheme that encrypts a message (i.e. codeword c) bit by bit, an $\mathsf{ECC}()$ with correction capacity d and a robust sketch of [Boyen et al., 2005] with $\mathsf{PAR}^* = \langle \mathsf{PAR}, H(m, \mathsf{PAR}) \rangle$, the new construction Π'=(Setup',$\mathsf{Extract}'$, $\mathsf{Encrypt}'$,$\mathsf{Decrypt}'$) is described as follows. Figure 7.1 shows an overview of this construction.

- Setup': It is identical to the Setup except for the setup of an $\mathsf{ECC}()$ with correction capacity d and a robust sketch.
- $\mathsf{Extract}'$: Given a user's feature vector $w = (w_1, ...w_n)$ and ms, it returns the corresponding private key set $D_w = (D_1, ...D_n)$ by running $\mathsf{Extract}$ on each w_i.

- $\mathsf{Encrypt}'$: An algorithm that takes as input a similar feature vector w', $\mathsf{Encrypt}$ algorithm of the Π, a message $m \in M$ and a robust sketch, outputs the ciphertext $\langle U, V, W \rangle = \langle \mathsf{Encrypt}^{\Pi}_{w'}(c), \mathsf{PAR}, H(m, \mathsf{PAR}) \rangle$, where $\mathsf{PAR} = c \oplus m$ for a random codeword $c \in \mathcal{C}$.

  Here, $\mathsf{Encrypt}^{\Pi}_{w'}(c)$ denotes the encryption of each bit c_j of the codeword c using the associated biometric feature w'_j individually. For simplicity, we assume that the size of the biometric feature vector, size of the codeword c and the size of the message m are equal. The biometric vector w' of the receiver is attached to the ciphertext as in fuzzy IBE.

- $\mathsf{Decrypt}'$: A deterministic algorithm that given the private key set D_w of the $\mathsf{Decrypt}$ algorithm, an error correcting procedure $\mathsf{ECC}()$ and a ciphertext encrypted with w' such that $d \leq |w \cap w'|$, the algorithm computes $c' = \mathsf{Decrypt}^{\Pi}_{w}(U)$ by decrypting each bit of U using the secret key of the corresponding biometric feature individually. Due to the errors on some of the biometric features of w', we obtain a corrupted codeword, which needs to be corrected via c=$\mathsf{ECC}(c')$. Next, $m = c \oplus V$ is obtained and if $W = H(m, V)$, m is returned, else $\perp$ is returned. Similar to the fuzzy commitment scheme, if w and w' are similar to each other, the codeword c is correctly recovered.

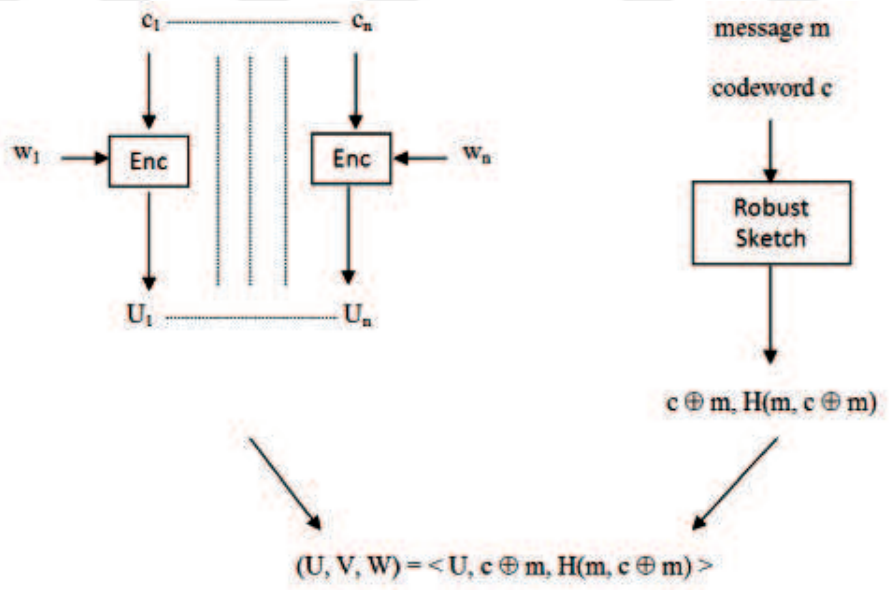

Figure 7.1: A weakly secure generic construction

7.3.1 Entropic Security vs. Indistinguishability

As it is noted in [Dodis and Smith, 2005], semantic security cannot be achieved for fuzzy sketches, when the adversary generates the two strings m_1, m_2 such that $|m_1| = |m_2|$ and thus knows that the challenge ciphertext is the encryption of one of m_1, m_2, the adversary can easily distinguish by computing $m_i \oplus V$ and verifying $W = H(m_i, V)$ from the challenge ciphertext. Thus, fuzzy sketches guarantee entropic security, which is weaker than Shannon security and assumes that the adversary is sufficiently uncertain about the challenge message. Thus, we have to design constructions that provide IND-ID-CCA security. Besides, anonymity, collision attacks and other biometrics-related issues should be considered.

7.4 Security Properties

In addition to the standard security level of (IND-ID-CCA) that an IBE scheme should achieve, biometric IBE schemes have to guarantee the following properties that are particularly important for biometric cryptosystems, since a user could use its biometrics on a number of applications such as identification, authentication, signing, etc. Thus, the traceability of the user's actions should be prevented through the anonymity of the ciphertexts and the secrecy of the identity-biometrics relation.

7.4.1 Anonymity

Informally, Recipient Anonymity (RA) or key privacy means that the adversary must be unable to decide whether a ciphertext was encrypted for a chosen identity, or for a random identity. In other words, an adversary cannot tell who the recipient is by looking at the ciphertext, which could be used to thwart traffic analysis. The formal definition is as follows.

Experiment $Exp^{\text{IBE-RA-CPA}}_{A,\Pi}(l)$
$(ms, params) \leftarrow \text{Setup}(1^l)$
$(ID_0, ID_1, s, m) \leftarrow A(params)$
$b \overset{R}{\leftarrow} \{0,1\}$, $c^* \leftarrow \text{Encrypt}(m, params, ID_b)$
$b' \leftarrow A(s, c^*, params)$
If $b' = b$ return 1 else return 0

The advantage of the attacker A is $Adv^{\text{IBE-RA-CPA}}_{A,\Pi} = |Pr[b' = b] - \frac{1}{2}|$. An IBE scheme Π is said to be IBE-RA-CPA secure if the respective advantage function is negligible for all PPT A.

In order to establish the IBE-RA-CPA/IBE-RA-CCA security of concrete schemes, it is helpful to work with a related notion IBE-RA-RE-CPA/IBE-RA-RE-CCA security. This notion is first defined in [Abdalla et al., 2005], where the only difference between the two notions is the challenge phase. In this phase, the challenger encrypts a random message m' in place of the adversaries choice of message m, hence the acronym RE in IBE-RA-RE-CPA denotes the randomized encryption.

Experiment $Exp_{A,\Pi}^{\text{IBE-RA-RE-CPA}}(l)$
$(ms, params) \leftarrow \text{Setup}(1^l)$
$(ID_0, ID_1, s, m) \leftarrow A(params)$
$b \overset{R}{\leftarrow} \{0,1\}$, $m' \leftarrow M$ s.t. $|m'| = |m|$
$c^* \leftarrow \text{Encrypt}(m', params, ID_b)$
$b' \leftarrow A(s, c^*, params)$
If $b' = b$ return 1 else return 0

In [Abdalla et al., 2005],the notions of IBE-RA-CPA and IBE-RA-RE-CPA are related using a lemma, which is extended in [Paterson and Srinivasan, 2008] for CCA-security.

Lemma 7.1. *Let* Π *be an IBE scheme that is IND-ID-ATK secure and IBE-RA-RE-ATK secure. Then,* Π *is also IBE-RA-ATK secure. Here, ATK*$\in \{$*CPA,CCA*$\}$.

If the ciphertext could be anonymized by anyone using the public key of the recipient, i.e. not just by the encryptor, the encryption scheme is defined as universally anonymous. In current fuzzy IBE systems, the biometric vector w of the receiver is attached to the ciphertext since set overlap is used as the distance metric between the identities w and w' in order to determine the d common features in $|w \cap w'|$ and to bring the ciphertext to the correct order for decryption. Hence, a different system should be designed to achieve anonymity for biometric IBE.

The formal definition of recipient anonymity for biometric IBE (BIBE-RA-CPA) is as follows. Here, w_0, w_1 denote the biometric feature vectors of two different users (i.e. $|w_0 \cap w_1| < d$), d is the error-tolerance and s is the state information. Without the condition of $|w_0 \cap w_1| < d$, the recipient anonymity game cannot be played, as two similar biometrics (i.e. $|w_0 \cap w_1| \geq d$) define the same user.

Experiment $Exp_{A,\Pi}^{\text{BIBE-RA-CPA}}(l)$
$(ms, params) \leftarrow \text{Setup}(1^l)$
$(w_0, w_1, s, m) \leftarrow A(params)$ s.t. $|w_0 \cap w_1| < d$
$b \overset{R}{\leftarrow} \{0,1\}$, $c^* \leftarrow \text{Encrypt}(m, params, w_b)$
$b' \leftarrow A(s, c^*, params)$
If $b' = b$ return 1 else return 0

The advantage of the attacker A is $Adv_{A,\Pi}^{\text{BIBE-RA-CPA}} = |Pr[b' = b] - \frac{1}{2}|$. A biometric

IBE scheme Π is said to be BIBE-RA-CPA secure if the respective advantage function is negligible for all PPT A.

7.4.2 Identity Privacy

For biometric authentication, this notion guarantees the privacy of the sensitive relationship between the user identity (i.e. ID= Name or e-mail address) and its biometrics against a malicious service provider or a malicious database [Bringer et al., 2007b, Tang et al., 2008]. For biometric IBE setting, this notion can be adapted for having privacy even against the trusted authority (PKG) or the encryptor. Thus, identity privacy is a stronger notion than anonymity, namely, identity privacy implies anonymity, which is shown in the following lemma. The privacy of biometrics-identity relation is achieved for many fuzzy IBE systems, which depend only on biometric identities, thus the user does not need to present any document to prove its personal identity ID. However, the fuzzy IBE scheme in [van Liesdonk, 2007] combines the identity ID (i.e. Name, e-mail) of the receiver with his biometric features in order to avoid collusion attacks. This approach is not only against identity privacy but also against the main principle of fuzzy IBE. However, the scheme of [van Liesdonk, 2007] could be corrected using our method described in section 6.2.6 of the previous chapter. The security notion Identity Privacy for biometric IBE (BIBE-IP-CPA) is formally defined as follows:

Experiment $Exp_{A,\Pi}^{\text{BIBE-IP-CPA}}(l)$
$(ms, params) \leftarrow \text{Setup}(1^l)$
$(s, m, ID_0, w_0, ID_1, w_1) \leftarrow A(params)$ s.t. $|w_0 \cap w_1| < d$
$b \xleftarrow{R} \{0,1\}$, $c^* \leftarrow \text{Encrypt}(m, params, w_b, ID_b)$
$b' \leftarrow A(s, c^*, params)$
If $b' = b$ return 1 else return 0

The advantage of the attacker A is $Adv_{A,\Pi}^{\text{BIBE-IP-CPA}} = |Pr[b' = b] - \frac{1}{2}|$. A biometric IBE scheme Π is said to be BIBE-IP-CPA-secure if the respective advantage function is negligible for all PPT A.

Lemma 7.2. *Identity privacy implies anonymity.*

Proof. If an adversary can break the anonymity of biometric IBE, then a simulator can be constructed that can break the identity privacy of biometric IBE.

Given an adversary A breaking the anonymity of biometric IBE, the simulator can run A on the challenge ciphertext c^* generated by randomly picking one of two different identities that are represented by concatenating the identity data to the biometrics data of each user. Since A is able to distinguish the randomly picked identity $ID_b||w_b$

of the recipient from c^*, by separating the result into two parts, the simulator reveals the link between the identity and biometrics. Thus identity privacy is not satisfied. □

For the reverse direction, namely the statement "*Anonymity implies identity privacy*" is only valid against a passive adversary who eavesdrops on the communication channel. However, an attacker -including the sender of the ciphertext or the trusted authorithy issuing the biometric secret keys- who does not know any information about the name or e-mail of the user, has zero advantage in the identity privacy game, as in the case of biometric remote authentication schemes of [Bringer et al., 2007b, Bringer and Chabanne, 2008]. In these biometric systems, the detached (malicious) biometric database has zero advantage in the identity privacy game since it does not have access to the personalized information (i.e. ID, name, e-mail,..) of the users that are stored at the service provider. Similarly, current fuzzy IBE schemes do not take any personalized information as input at any stage of the system.

7.5 Generic Constructions for Biometric IBE

In this section, we describe generic constructions converting any weakly secure IBE scheme that encrypts a message bit by bit into an IND-ID-CCA secure encryption scheme in the error-tolerant setting. For this, we combine a weakly secure IBE scheme Π, an error correcting code ECC() and a generic conversion scheme that preserves the anonymity of Π after conversion, namely, if Π is an anonymous IBE scheme, then the resulting IND-ID-CCA secure biometric IBE scheme is also proven to be anonymous. Since anonymity is an important goal we want to achieve, the sender should not attach the biometric feature set to the ciphertext. As a result, it is impossible for the receiver to determine the set of common features between the biometrics attached to the ciphertext and the (similar) biometrics of the receiver. However, if we employ biometrics that can be ordered/grouped, i.e. biometrics represented as a sequence of n ordered feature points [Ballard et al., 2008, Teoh et al., 2008], such as Iris [Kanade et al., 2009, Bringer et al., 2007b], fingercode [Jain et al., 2000, Tong et al., 2007],face [Li et al., 2006, Ekenel and Stiefelhagen, 2009, Gao et al., 2009, Boehnen et al., 2009, Moreno et al., 2005], online signatures [Igarza et al., 2004], then the receiver is able to make an ordered *element by element* decryption using only his own biometrics. We can observe a suitable representation for our system in the face recognition system of [Moreno et al., 2005], where a feature is zero valued if it cannot be computed because of the non-existence of a region from which it is derived. Besides, the fuzzy commitment construction of [Juels and Wattenberg, 1999] is based on ordered biometric feature vectors. Hence, for non-order features like minutiae of fingerprints, whose components vary and cannot be described as a vector [Zhou and Busch, 2008], our system is not

suitable, if we want to achieve anonymity.

7.5.1 Based on the Fujisaki-Okamoto Conversion

Our first generic construction for biometric IBE is based on the Fujisaki-Okamoto (FO) conversion. Fujisaki and Okamoto proposed a simple conversion scheme called as a hybrid scheme ε^{hy} from weak asymmetric-key encryption (AE) and symmetric-key encryption (SE) schemes into a public-key encryption scheme which is secure in the sense of IND-CCA. Basically, ε^{hy} is defined in [Fujisaki and Okamoto, 1999] as follows.

$$\varepsilon^{hy}(m;\sigma) = \langle AE_{pk}(\sigma; H(\sigma, m)||SE_{G(\sigma)}(m)\rangle$$

In ε^{hy}, σ is picked at random from COIN, where COIN $\in \{0,1\}^*$ is a finite set. H and G are two cryptographic hash functions with H: AKMS x SKMS $\rightarrow$ COIN and G: AKMS $\rightarrow$ SKS, where AKMS denotes asymmetric-key message space, SKMS denotes symmetric-key message space, and SKS is the symmetric-key space. The idea is, first encrypting the redundancy σ with the random coin $H(\sigma, m)$ under public key pk using the probabilistic scheme AE and then encrypting the message under the symmetric key $G(\sigma)$ using the scheme SE. In [Fujisaki and Okamoto, 1999], it is proven that if AE is an one-way encryption scheme, then ε^{hy} is IND-CCA secure in the ROM. However, it is shown that if AE scheme satisfies IND-CPA security, then there is a significant improvement in the security reduction, where IND-CPA implies also one-way encryption [Boneh and Franklin, 2003].

Next, the authors of [Yang et al., 2006],[Kitagawa et al., 2006] describe the FO conversion for standard IBE schemes, which we is extended by [Paterson and Srinivasan, 2008] to multi trusted authority IBE setting. First, we review the Fujisaki-Okamoto (FO) Conversion for standard IBE setting as described in [Yang et al., 2006].

Let Π = (Setup, Extract, Encrypt, Decrypt) is an IND-ID-CPA secure IBE scheme. Define Π' = (Setup$'$, Extract$'$, Encrypt$'$, Decrypt$'$) as a new IBE scheme as below.

- Setup$'$: This is the almost the same Setup algorithm of Π, namely given a security parameter l, the PKG generates the master secret key ms and the public parameters of the system. In addition, H and G are cryptographic hash functions with $H : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \times \{0,1\}^* \rightarrow$ COIN, where COIN $\in \{0,1\}^*$ is a finite set and $G : \{0,1\}^{l_1} \rightarrow \{0,1\}^{l_2}$.

- Extract$'$: This is the same Extract algorithm of Π, namely given a user's identity ID and ms, it returns the corresponding private key D_{ID}.

- $\mathsf{Encrypt}'$: A probabilistic algorithm that takes as input identity ID, $\mathsf{Encrypt}$ algorithm of the Π scheme, a message $m \in \{0,1\}^{l_2}$, outputs the ciphertext $(U,V) = \langle \mathsf{Encrypt}^{\Pi}_{ID}(\sigma); H(\sigma, m, ID) || G(\sigma) \oplus m \rangle$.

- $\mathsf{Decrypt}'$: A deterministic algorithm that given the private key D_{ID} of the $\mathsf{Decrypt}$ algorithm and a ciphertext (U,V) encrypted with ID, it first computes $\sigma = \mathsf{Decrypt}^{\Pi}_{ID}(U)$ and $G(\sigma)$ to obtain $m = G(\sigma) \oplus V$. Finally, $H(\sigma, m, ID)$ is computed for reencryption as $\mathsf{Encrypt}^{\Pi}_{ID}(\sigma; H(\sigma, m, ID))$, thus, the correctness is checked and m is returned.

According to our framework, we present an $\mathsf{IND\text{-}ID\text{-}CCA}$ secure application $\Pi' = (\mathsf{Setup}', \mathsf{Extract}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ that works in error-tolerant IBE setting as follows. Here $c \in \mathcal{C}$ denotes a random codeword and $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ is an $\mathsf{IND\text{-}ID\text{-}CPA}$ secure IBE scheme that encrypts a message bit by bit. Finally, $w = (w_1, ...w_n)$ denotes the feature vector of the user biometrics. For better readibility of the proof, we do not consider collision attacks at this stage, although in section 7.5.2, we replace the biometric w with the collision resistant biometrics BID. Finally, we assume the existence of a code $\mathcal{C}$ that is suitable to the properties of the generic construction.

- Setup': Given a security parameter l, the PKG generates the master secret key ms and the public parameters of the system. Here, H and G are cryptographic hash functions with $H : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \times \{0,1\}^* \rightarrow \mathsf{COIN}$ and $G : \{0,1\}^{l_1} \rightarrow \{0,1\}^{l_2}$.

- $\mathsf{Extract}'$: Given a user's biometric feature vector w and ms, it returns the corresponding private key set D_w by running $\mathsf{Extract}$ on each w_i individually.

- $\mathsf{Encrypt}'$: A probabilistic algorithm that takes as input biometrics w', $\mathsf{Encrypt}$ algorithm of the Π scheme, a message $m \in M$ and a random codeword $c \in \mathcal{C}$, outputs the ciphertext $(U,V) = (\mathsf{Encrypt}^{\Pi}_{w'}(c; H(c, m, w')), G(c) \oplus m)$.

  Specifically, $\mathsf{Encrypt}^{\Pi}_{w'}$ encrypts the message (i.e. the codeword c) bit by bit using the biometric feature vector $w' = w'_1, ..., w'_n$ by computing $\mathsf{Encrypt}^{\Pi}_{w'}(c; H(c, m, w'))$ $= \langle \mathsf{Encrypt}^{\Pi}_{w'_1}(c_1; H(c, m, w'_1)), ..., \mathsf{Encrypt}^{\Pi}_{w'_n}(c_n; H(c, m, w'_n)) \rangle$.

- $\mathsf{Decrypt}'$: A deterministic algorithm that given the private key D_w of the $\mathsf{Decrypt}$ algorithm, an error correcting procedure $\mathsf{ECC}()$ and a ciphertext encrypted with w' such that $|w \cap w'| \geq d$, first computes $c' = \mathsf{Decrypt}^{\Pi}_{w}(U)$ and error corrects c=$\mathsf{ECC}(c')$. Specifically, $\mathsf{Decrypt}^{\Pi}_{w}(U)$ performs the decryption bit by bit as $c' = \langle c'_1, ..., c'_n \rangle = \mathsf{Decrypt}^{\Pi}_{w}(U) = \langle \mathsf{Decrypt}^{\Pi}_{w_1}(U_1), ..., \mathsf{Decrypt}^{\Pi}_{w_n}(U_n) \rangle$. After error correction, $m = G(c) \oplus V$ is obtained. Finally, by computing $H(c, m, w_i)$ and using it in reencryption, the correctness is verified for at least d encryptions (since $|w \cap w'| \geq d$) and m is returned.

Remark 7.1. *For simplicity, we assume that the length l_1 of the codeword c and the size n of the biometric feature set w is equal, i.e. $l_1 = n$. If $n < l_1$, then we obtain a longer feature vector by extracting more features or by providing biometric data from more fingers instead of one fingerprint. In [Boneh et al., 2007], the authors expand the identity ID by computing it as $H(ID, j)$ for $1 \leq j \leq n$ to make it suitable for the encryption of n-bit messages. The same method could be employed to expand the biometric identity of the user. For the case that the underlying IBE scheme is not anonymous, the biometric vector w' of the receiver is attached to the ciphertext. Hence, the biometric trait does not need to be ordered as the receiver with the similar biometrics w can determine the common features easily as in current fuzzy IBE schemes.*

Theorem 7.1. *Let Π be a γ-uniform anonymous IBE scheme with negligable γ. Suppose that the hash functions H, G are random oracles and let A be an IND-ID-CCA adversary which has advantage $\epsilon'(l)$ against anonymous Π' and it runs in time at most $t'(l)$. Suppose A makes at most q_H H-queries, q_G G-queries, q_E Extraction queries and q_D Decryption queries. Suppose that encrypting one message needs time τ. Then there is an OW-ID-CPA adversary R against anonymous Π which has running time $t(l) = O(t'(l) + \tau l_1 q_H)$ and has advantage*

$$\epsilon(l) \geq \frac{1}{q_H + q_G}(2\epsilon'(l) - d\gamma q_D - \frac{q_D}{2^{l_2}})$$

Proof. Given an IND-ID-CCA secure anonymous Π', the goal of the reduction algorithm R is to invert the OW-ID-CPA secure anonymous Π scheme using an adversary A running against Π'.

The challenger of R outputs the public parameters of Π, which is passed by R to the adversary A in order to simulate the setup phase of Π'.

R answers the random oracle and decryption queries of A as follows.

- H-queries: On each new input (c, m, w'_j), R picks a random h_j from the range of H, returns h_j to A and inserts the tuple $(c, m, w'_j, h_j, U'_j, V')$ to the HList. Here, $U'_j = \mathsf{Encrypt}^{\Pi}_{w'_j}(c_j; H(c, m, w'_j))$ and $V' = G(c) \oplus m$, where V' is computed by simulating the G-oracle as below. Here, we basically store the encryption of the j-th bit of the codeword c encrypted with the biometric feature w'_j to be used later during the simulation of the decryption oracle.

- G-queries: On each new input c, R picks a random g from the range of G, returns this value to A and inserts the tuple (c, g) to the GList.

- Private Key Extraction queries: For any identity $\tilde{w}$, the extraction query is passed to the challenger of R and his answer is returned to A.

- Decryption queries: On each new input $(\bar{w}, U, V)$,

  R finds the tuples $(c, m, w'_j, h_j, U'_j, V')$ from the HList such that $V = V'$, $\bar{w_j} = w'_j$ and $U_j = U'_j$ for at least d features of $\bar{w}$.

  R outputs m if the above condition is satisfied, or outputs reject otherwise.

- Challenge: A outputs the challenge identity w^* such that $|w^* \cap \tilde{w}| < d$, two equal length messages (m_0, m_1) on which it wishes to be challenged. R sends w^* to the challenger and receives a ciphertext
  $U^* = \langle \mathsf{Encrypt}^{\Pi}_{w_1^*}(c_1^*; r_1) .... \mathsf{Encrypt}^{\Pi}_{w_n^*}(c_n^*; r_n) \rangle$ and $r_i \in \mathsf{COIN}$. R picks a random $V^* \in \{0,1\}^{l_2}$ and returns A the ciphertext (U^*, V^*).

- Guess: A outputs a guess b'.

After A outputs its guess b', R checks the HList for (c, m, w_j) such that $w_j^* \in w^*$ for at least one feature of w^* or GList for c and returns c to his challenger.

Similar to the computation of the reduction cost presented in [Yang et al., 2006], we first define the following three events:

1. Pr[Succ$A$] the event that A wins the IND-ID-CCA game.

2. Let Ask be the event that algorithm A issues either the hash query $G(c)$ or issues at least one hash query $H(c, *, w'_j)$ for $w'_j \in w$ at some point during the simulation. Here, $*$ denotes any l_2-bit string.

3. Let Fail to be the event that R fails to answer a decryption query correctly at some point during the game, which can occur only when A submits a decryption query (w, U, V), where $U = \{U_j : w_j \in w\} = \{\mathsf{Encrypt}^{\Pi}_{w_j}(c_j; H(c, m, w_j)) : w_j \in w, c_j \in c\}$ without asking $G(c)$ or $H(c, m, w_j)$ for at least d features of w. For the first case, R fails to properly answer each such decryption query with probability at most 2^{-l_2}. And for the second case with probability at most γ^d. Thus, due to the q_D decryption queries, we have $\Pr[\neg\mathsf{Fail}] \leq (1 - \gamma^d - \frac{1}{2^{l_2}})^{q_D} \approx 1 - q_D(\gamma^d + \frac{1}{2^{l_2}})$

Then, similar to [Yang et al., 2006]

$\Pr[\text{Succ}A|\neg\mathsf{Fail}]\Pr[\neg\mathsf{Fail}] \geq \epsilon'(l) + \frac{1}{2} - \Pr[\mathsf{Fail}]$

Since $\Pr[\text{Succ}A|\neg\mathsf{Fail}, \neg\mathsf{Ask}] = \frac{1}{2}$, we also have

$$\begin{aligned}\Pr[\text{Succ}A|\neg\mathsf{Fail}] &= \Pr[\text{Succ}A|\neg\mathsf{Fail} \wedge \mathsf{Ask}] \cdot \Pr[\mathsf{Ask}] + \tfrac{1}{2}(1 - \Pr[\mathsf{Ask}]) \\ &\leq \tfrac{1}{2}\Pr[\mathsf{Ask}] + \tfrac{1}{2}\end{aligned}$$

Hence, we have that $(\frac{1}{2}\Pr[\mathsf{Ask}] + \frac{1}{2})\cdot \Pr[\neg\mathsf{Fail}] \geq \epsilon'(l) + \frac{1}{2} - \Pr[\mathsf{Fail}]$

and therefore, $\Pr[\mathsf{Ask}] \geq 2\epsilon'(l) - \Pr[\mathsf{Fail}]$

As a result,

$$\begin{aligned}\epsilon(l) &\geq \tfrac{1}{q_H+q_G}\Pr[\mathsf{Ask}] \\ &\geq \tfrac{1}{q_H+q_G}(2\epsilon'(l) - (1-(1-q_D(\gamma^d + \tfrac{1}{2^{l_2}})))) \\ &\simeq \tfrac{1}{q_H+q_G}(2\epsilon'(l) - dq_D\gamma - \tfrac{q_D}{2^{l_2}})\end{aligned}$$

R's running time is computed as in [Yang et al., 2006]. First, we consider that R has to run the encryption algorithm of Π at most $l_1 q_H$ times. By adding the time that is required by A we obtain $t(l) = O(t'(l) + \tau l_1 q_H)$.

□

Next, we show that our generic construction preserves the anonymity of the underlying encryption scheme. In this proof, we play the standard game for anonymity, namely, the game for IBE-RA-CPA/IBE-RA-CCA as described in 7.4. An alternative proof can also be described by playing the game of IBE-RA-RE-CPA/IBE-RA-RE-CCA and apply Lemma 7.1. Since the generic construction outputs an IND-ID-CCA secure IBE scheme, it is enough to prove that the generic construction guarantees the notion of IBE-RA-RE-CCA. Again for better readibility of the proof, we do not consider collision attacks on the biometrics at this stage, although in section 7.5.2, we replace the biometric w with the collision resistant biometrics BID.

Theorem 7.2. *Let Π be a γ-uniform anonymous IBE scheme with negligable γ. Suppose that the hash functions H, G are random oracles and let A be an IBE-RA-CCA adversary which has advantage $\epsilon'(l)$ against anonymous Π' and it runs in time at most $t'(l)$. Suppose A makes at most q_H H-queries, q_G G-queries, q_E Extraction queries and q_D Decryption queries. Suppose that encrypting one message needs time τ. Then there is an IBE-RA-CPA adversary R against anonymous Π which has running time $t(l) = O(t(l) + l_1 q_H \cdot \tau)$ and has advantage*

$$\epsilon(l) \geq 2(\frac{\epsilon'(l)+1}{2} - \frac{q_H}{2^{l_1}})(1 - q_D(\gamma^d + \frac{1}{2^{l_2}})) - 1$$

.

Proof. Given an IBE-RA-CCA secure anonymous Π', the goal of the reduction algorithm R is to distinguish the identity of the IBE-RA-CPA secure anonymous Π scheme using an adversary A running against Π'.

The challenger of R outputs the public parameters of Π, which is passed to the adversary A in order to simulate the setup phase of Π'.

R answers the random oracle and decryption queries of A as follows.

- H-queries: Identical to the proof of theorem 7.1.
- G-queries: Identical to the proof of theorem 7.1.
- Private Key Extraction queries: Identical to the proof of theorem 7.1.
- Decryption queries: Identical to the proof of theorem 7.1.
- Challenge: A outputs the challenge identities w^0, w^1 such that $|w^b \cap \tilde{w}| < d$ and $|w^0 \cap w^1| < d$ for $b \in \{0,1\}$ and a message m. R picks a random codeword c and sends (w^0, w^1, c) to the challenger and receives a ciphertext U^*. Here, $U^* = \{U_j : w_j^b \in w^b\} = \{\mathsf{Encrypt}^{\Pi}_{w_j^b}(c_j; r_j) : w_j^b \in w^b\}$ and $r_j \in \mathsf{COIN}$. R simulates the G oracle on input c and obtains $g = G(c)$, and returns A the ciphertext $(U^*, g \oplus m)$. Four sorts of queries are answered as the same as before.
- Guess: A outputs a guess b'.

After the challenge query has been issued, if the adversary A makes H-oracle queries on either (c, m, w_j^0) or (c, m, w_j^1) for at least one feature of w^b, R outputs $b' = 0$ or $b' = 1$, respectively, as its guess for the value of the bit b. If neither hash query is made, then R returns the same bit b' that A outputs to his challenger. R wins if $b' = b$.

We should note that if we play the IBE-RA-RE-CPA/ IBE-RA-RE-CCA game, then the challenge phase of the proof slightly changes. Specifically, A outputs only the challenge identities to be given to R, similarly, R obtains from his challenger an encryption of a random codeword, instead of the codeword c that R has picked at random.

The analysis of the reduction cost is similar to the analysis presented in [Fujisaki and Okamoto, 1999, Kitagawa et al., 2006, Paterson and Srinivasan, 2008]. Let $\Pr[\mathrm{Succ}A]$ be the probability that adversary A outputs a bit $b' = b$. Similarly, let $\Pr[\mathrm{Succ}R]$ be the probability that the reduction R outputs a bit $b' = b$. Let Ask_b be the event that algorithm A issues at least one query $H(c, m, w_j^b)$ for $w_j^b \in w^b$ and let $\mathsf{Ask}_{\bar{b}}$ be the event that algorithm A issues at least one query $H(c, m, w_j^{\bar{b}})$ for $w_j^{\bar{b}} \in w^{\bar{b}}$ at some point during the simulation.

$$\begin{aligned}\Pr[\mathrm{Succ}A] = {} & \Pr[\mathrm{Succ}A|\mathsf{Ask}_b]\Pr[\mathsf{Ask}_b] \\ & + \Pr[\mathrm{Succ}A|(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}]\ \Pr[(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}] \\ & + \Pr[\mathrm{Succ}A|(\neg\mathsf{Ask}_b) \wedge (\neg\mathsf{Ask}_{\bar{b}})]\ \Pr[(\neg\mathsf{Ask}_b) \wedge (\neg\mathsf{Ask}_{\bar{b}})]\end{aligned}$$

Similarly,

$$\begin{aligned}\Pr[\text{Succ}R] = &\Pr[\text{Succ}R|\mathsf{Ask}_b]\ \Pr[\mathsf{Ask}_b] \\ &+ \Pr[\text{Succ}R|(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}]\ \Pr[(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}] \\ &+ \Pr[\text{Succ}R|(\neg\mathsf{Ask}_b) \wedge (\neg\mathsf{Ask}_{\bar{b}})]\ \Pr[(\neg\mathsf{Ask}_b) \wedge (\neg\mathsf{Ask}_{\bar{b}})]\end{aligned}$$

From the conditions of the simulation,

$\Pr[\text{Succ}R|\mathsf{Ask}_b]$=1,$\Pr[\text{Succ}R|(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}]$ =0,
$\Pr[\text{Succ}A|(\neg\mathsf{Ask}_b) \wedge (\neg\mathsf{Ask}_{\bar{b}})]$ =$\Pr[\text{Succ}R|(\neg\mathsf{Ask}_b) \wedge (\neg\mathsf{Ask}_{\bar{b}})]$

Therefore,

$$\begin{aligned}\Pr[\text{Succ}R]\text{-}\Pr[\text{Succ}A] = &\Pr[\mathsf{Ask}_b](1\text{- }\Pr[\text{Succ}A|(\mathsf{Ask}_b]) \\ &+ \Pr[(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}](0\text{-}\Pr[\text{Succ}A|(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}]) \\ &\geq \text{- }\Pr[(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}]\end{aligned}$$

Since, even a computationally unbounded adversary has no information about what the string c (which is uniformly distributed on a set of size 2^{l_1}) and our adversary makes at most q_H queries to the H-oracle, we have

$\Pr[(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}] \leq \frac{q_H}{2^{l_1}}$. Hence,

$$\begin{aligned}\Pr[\text{Succ}R] &\geq \Pr[\text{Succ}A] \text{ - } \Pr[(\neg\mathsf{Ask}_b) \wedge \mathsf{Ask}_{\bar{b}}] \\ &\geq \tfrac{\epsilon'(l)+1}{2} - \tfrac{q_H}{2^{l_1}}\end{aligned}$$

As before, let Fail to be the event that R fails to answer a decryption query correctly at some point during the game, which can occur only when A submits a decryption query (w, U, V), where $U = \{U_j : w_j \in w\} = \{\mathsf{Encrypt}^{\Pi}_{w_j}(c_j; H(c, m, w_j)) : w_j \in w, c_j \in c\}$ without asking $G(c)$ or $H(c, m, w_j)$ for at least d features of w. For the first case, R fails to properly answer each such decryption query with probability at most 2^{-l_2}. And for the second case with probability at most γ^d. Thus, due to the q_D decryption queries, we have $\Pr[\neg\mathsf{Fail}] \leq (1 - \gamma^d - \frac{1}{2^{l_2}})^{q_D} \approx 1 - q_D(\gamma^d + \frac{1}{2^{l_2}})$

Hence, we have,

$$\epsilon(l) \geq 2\Pr[\text{Succ}R]\ \Pr[\neg\mathsf{Fail}] - 1 \geq 2(\tfrac{\epsilon'(l)+1}{2} - \tfrac{q_H}{2^{l_1}})(1 - q_D(\gamma^d + \tfrac{1}{2^{l_2}})) - 1.$$

□

7.5.2 Collision Attacks

As described in section 6.2.6 of the previous chapter, we can use our new method to prevent collision attacks so that different users cannot collude in order to decrypt a ciphertext that should only be decrypted by the real receiver. We note that the method

presented in [Sahai and Waters, 2005, Baek et al., 2007] is not applicable to different fuzzy/biometric IBE systems designed for non-pairing-based cryptographic techniques.

In the biometric cryptosystems such as BIO-IBS [Burnett et al., 2007], the fixed length binary string b is computed using the feature vector and the hash of b is used as the identity ID. Specifically, we use the robust fuzzy extractor to obtain a unique biometric string ID via error correction codes from the biometric template b of the user in such a way that an error tolerance t is allowed. In other words, we will obtain the same biometric string ID even if the fuzzy extractor is applied on a different b' such that $\mathsf{dis}_{\mathcal{H}}$ (b,b')$< t$. Here, dis() is the distance metric used to measure the variation in the biometric reading and t is the error tolerance parameter of the fuzzy extractor.

In the anonymous fuzzy IBE scheme of [van Liesdonk, 2007], collusion attacks are avoided by combining each biometric feature w_i with the identity (i.e. Name,e-mail) of the user. However, this approach is against the nature of fuzzy IBE, where the identities should only consist of the biometric data of the user. Besides, identity privacy is not satisfied despite the anonymity of the scheme. One can correct this fuzzy IBE scheme by using $\mathsf{BID} = \langle H_1(w_1, \mathsf{ID}), ..., H_1(w_n, \mathsf{ID})\rangle$ in the key generation phase. This way, the privacy of biometric-identity relation and the resistance against collusion attacks is maintained. We can combine an ordered feature set w and a unique biometric string obtained from the same/different biometric trait. For instance, we can combine face + iris, where face can described as a set of ordered features and iris can be represented as a 2048 bit string which is already combined with error correction procedures in secure sketch/fuzzy extraction applications in order to have improved accuracy [Bringer et al., 2007a]. Hence, a multi-modal approach for preventing collision attacks has also benefits in security and identification of the users during key generation process. As before, we note that the computation of ID is performed only once and for all.

Based on our generic construction for the error-tolerant setting, we present an anonymous IND-ID-CCA secure application $\Pi' = (\mathsf{Setup}', \mathsf{Extract}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ that prevents collision attacks as follows. Here $c \in \mathcal{C}$ denotes a random codeword and $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ is an IND-ID-CPA secure anonymous IBE scheme that encrypts a message bit by bit such as [Boneh et al., 2007, Ateniese and Gasti, 2009]. Finally, BID denotes the combined feature vector of the user biometrics that is generated based on our new method presented before. Finally, the only difference to our generic construction for error tolerant setting is the replacement of the biometric feature vector w with the collision resistant biometric vector BID.

- Setup': The same as Π' except for an additional fuzzy extractor scheme that is included to the public parameters to avoid collision attacks.
- $\mathsf{Extract}'$: The same as Π' except that each secret key component of the user is computed for the combined biometric features $\mathsf{BID}_j = H_1(w_j, \mathsf{ID})$ as in Sarier

[2008, 2011b] to avoid collision attacks. Here ID can also be extracted from a different biometric trait (i.e. not necessarily from w).

- $\mathsf{Encrypt}'$: A probabilistic algorithm that takes as input the combined biometric feature vector BID', Encrypt algorithm of the Π scheme, a message $m \in M$ and a random codeword $c \in \mathcal{C}$, outputs the ciphertext

$$\langle U, V \rangle = \langle \mathsf{Encrypt}^{\Pi}_{\mathsf{BID}'}(c; H(c, m, \mathsf{BID}')), G(c) \oplus m \rangle$$

.

Specifically, $\mathsf{Encrypt}^{\Pi}_{\mathsf{BID}'}$ encrypts the message (i.e. the codeword c) bit by bit using the collision resistant biometrics $\mathsf{BID}' = (\mathsf{BID}'_1, ..., \mathsf{BID}'_n)$ by computing $\langle \mathsf{Encrypt}^{\Pi}_{\mathsf{BID}'_1}(c_1; H(c, m, \mathsf{BID}'_1)), ..., \mathsf{Encrypt}^{\Pi}_{\mathsf{BID}'_n}(c_n; H(c, m, \mathsf{BID}'_n)) \rangle$, which is denoted by $\mathsf{Encrypt}^{\Pi}_{\mathsf{BID}'}(c; H(c, m, \mathsf{BID}'))$.

- $\mathsf{Decrypt}'$: The same as Π'. The only difference is the computation of $H(c, m, w_i)$ is replaced by $H(c, m, \mathsf{BID}_i)$ before the reencryption due to our new method for biometric identity generation to avoid collision attacks.

By replacing the biometrics w with collision resistant biometrics BID, we uniquely bind each feature of w to the user, thus avoid collision attacks. The proofs remain identitical and the binding procedure is performed by the encryptor only once and for all.

7.5.3 Based on REACT

As it is noted in [Okamoto and Pointcheval, 2001], Fujisaki-Okamoto transformation converts any one-way cryptosystem into a CCA secure encryption scheme, but it is not optimal due to the re-encryption operation during the decryption phase. In [Okamoto and Pointcheval, 2001], an efficient and IND-CCA secure generic conversion scheme with a tight reduction cost is presented, which takes as input a OW-PCA secure encryption scheme and avoids the disadvantages of FO transformation via

$$\varepsilon^{hy}(m; R) = \langle AE_{pk}(R) || SE_{G(R)}(m) || H(R, m, AE_{pk}(R), SE_{G(R)}(m)) \rangle$$

Similar to FO conversion, REACT is also implemented for IBE in [Kitagawa et al., 2006]. When used in biometric IBE setting, one should modify REACT for IBE as

$$\langle U, V, Y \rangle = \langle \mathsf{Encrypt}^{\Pi}_{w'}(c), G(c) \oplus m, H(c, m, U, V) \rangle$$

Thus, the only difference to the FO transformation adapted to the error-tolerant setting occurs in the decryption stage where only one hash computation, i.e. $H(c, m, U, V)$

- G-queries: On each new input c, R picks a random g from the range of G, returns g to A and inserts the tuple (c, g) to the GList.
- H-queries: On each new input (c, m, U, V), R picks a random h from the range of H, returns h to A and inserts the tuple (c, m, U, V, h) to the HList.
- Private Key Extraction queries: For any identity $\tilde{w}$, the extraction query is passed to the challenger of R and his answer is returned to A.
- Decryption queries: Let (w, U, V, Y) be a decryption query issued by A. R responds as follows:

  1. R picks up a tuple (c, m, U, V, h) from the HList such that $Y = h$.
  2. R computes $G(c) \oplus m$ and checks if $V = G(c) \oplus m$. If this holds, R queries (w, c, U) to the PC (plaintext checking) oracle.
  3. If the PC oracle answers "yes", R returns m to A. Otherwise, R outputs "reject".

  As opposed to the previous proof, here, we do not consider error-tolerance for some of the components of the ciphertext, as the decryption oracle checks first whether the hash of all the components that form the ciphertext is queried to the H-oracle. If this tuple is not found in HList, the query is rejected. Due to this property, the proof is the same as the original proof of REACT, as in [Okamoto and Pointcheval, 2001, Kitagawa et al., 2006]. The probability that A comes with a valid ciphertext without querying the H-oracle is 2^{-l_2} and similarly, the probability that A comes with a valid ciphertext without querying the G-oracle is 2^{-l_5}.
- Challenge: A outputs a challenge identity w^* such that $|w^* \cap \tilde{w}| < d$ and two messages on which it wishes to be challenged. R sends w^* to the challenger and receives a ciphertext U^* encrypted using w^*. Next, R generates a l_2-bit random string V^* and a l_5-bit random string Y^*, which are all returned to A. Four sorts of queries are answered as the same as before.
- Guess: A outputs a guess b'.

After A outputs its guess b', R picks all c's which appear in tuples on the Glist and the Hlist. For each c, R queries (w^*, c, U^*) to PC oracle. If PC oracle returns 'yes', R outputs the c as the answer of OW-ID-PCA game.

The advantage of R is identical to the computation described in [Kitagawa et al., 2006], i.e. $\epsilon(l) \geq 2\epsilon'(l) - q_D(\frac{1}{2^{l_2}} + \frac{1}{2^{l_5}})$.

In order to prove that our generic construction preserves anonymity, we should simply modify the challenge phase of the proof similar to the proof presented in [Zhang et al., 2007] as below.

Challenge: A outputs a message m and the challenge identities w^0, w^1 such that $|w^b \cap \tilde{w}| < d$ and $|w^0 \cap w^1| < d$ for $b \in \{0,1\}$. R picks a random codeword c^* and sends (w^0, w^1, c^*) to the challenger and receives a ciphertext U^*. Here, $U^* = \{U_j : w_j^b \in w^b\} = \{\mathsf{Encrypt}^{\Pi}_{w_j^b}(c^*) : w_j^b \in w^b\}$. R simulates the G oracle on input c and obtains $g = G(c^*)$, and returns A the ciphertext $(U^*, g \oplus m, h)$, where $h = H(c^*, m, U^*, g \oplus m)$ by simulating the H oracle. Finally, when A stops and outputs a bit b, R also outputs the same bit.

The advantage of R is identical to the computation described in [Zhang et al., 2007], we review it briefly as below. From the above description, R may reject a correct ciphertext query. Denote this event as Fail1, which happens with probability $(2^{-l_2})^{q_1} + (2^{-l_5})^{q_D}$, where q_1 and q_D is the number of G-oracle queries and decryption queries, respectively. Additionally, R may fail when R is queried on c^* within G-oracle queries. Denote this event as Fail2, which happens with probability at most $(2^{-l_2})^{q_1}$. Furthermore, denote the event R fails in the simulation as FailR. We know if R does not fail,A's advantage is at most $\epsilon'(l)$. Denoting the advantage of R as $\epsilon(l)$, we have

$$\epsilon(l) = \epsilon'(l)\Pr[\neg(\mathsf{FailR}] = \epsilon'(l)\Pr[\neg(\mathsf{Fail1} \wedge \mathsf{Fail2})] \geq \epsilon'(l) - \Pr[\mathsf{Fail1}] \text{ - } \Pr[\mathsf{Fail2}].$$

As a result, $\epsilon'(l) \leq q_1 2^{-l_2} + q_D 2^{-l_5}$.

Again, collision attacks should be avoided by our new method as before, namely by replacing the w with the biometric identity BID as in the previous constructions. □

7.6 A Generic Biometric ID-KEM Construction

A Key Encapsulation Mechanism (KEM) consists of three algorithms: Key generation, encapsulation and decapsulation algorithms, where a KEM outputs a random session key to be used by the Data Encapsulation Mechanism (DEM) in the symmetric encryption in order to achieve efficient encryption of long messages. Current identity-based KEM's [Bentahar et al., 2008] are not suitable for error prone identities, thus we present a generic construction for a biometric ID-KEM Π'=(Setup′,Extract′, Enc′,Dec′) that takes any IBE scheme Π=(Setup,Extract, Encrypt,Decrypt) that encrypts a message bit by bit. Here, H, and G denote cryptographic hash functions with $H : \{0,1\}^n \times \{0,1\}^* \rightarrow \{0,1\}^{l_1}$ and $G : \{0,1\}^n \rightarrow \{0,1\}^{l_2}$.

- Setup′: The same Setup algorithm of Π except for the two additional hash functions H and G.

- $\mathsf{Extract}'$: The same $\mathsf{Extract}$ algorithm of Π.
- Enc': This algorithm takes as input a biometric vector w', $\mathsf{Encrypt}$ algorithm of Π, a random codeword $c \in \mathcal{C}$, it returns $\langle U, K\rangle = \langle \mathsf{Encrypt}^{\Pi}_{w'}(c; H(c,w)), G(c)\rangle$. As before, the encryption is performed bit by bit $\mathsf{Encrypt}^{\Pi}_{w'}(c; H(c,w))$ $= \mathsf{Encrypt}^{\Pi}_{w'_1}(c_1; H(c,w_1)), ..., \mathsf{Encrypt}^{\Pi}_{w'_n}(c_n; H(c,w_n))$. Here, $K = G(c) \in \mathbb{K}_{\mathsf{ID\text{-}KEM}}$ is an encapsulation key from the key space of the ID-KEM, which is used as input to a DEM for the encryption of the actual message.
- Dec': A deterministic algorithm that given the private key D_w of the $\mathsf{Decrypt}$ algorithm, an error correcting procedure $\mathsf{ECC}()$ and a ciphertext (U) encrypted with w' such that $d \leq |w \cap w'|$, it computes $c' = \mathsf{Decrypt}^{\Pi}_{w}(U)$ by decrypting bit by bit and corrects the error via c=$\mathsf{ECC}(c')$. Finally, by computing $H(c, w_i)$ and using it in reencryption, the correctness is checked for at least d encryptions (since $|w \cap w'| \geq d$) and the algorithm returns either the encapsulated key $G(c)$, else $\perp$ is returned.

The security of a biometric ID-KEM is defined identical to the definition given in [Bentahar et al., 2008] using the following game between an adversary and a challenger.

Experiment $\mathsf{IND\text{-}ID\text{-}CCA}(l, \mathsf{ID\text{-}KEM}, \mathrm{A})$
$(params, ms) \leftarrow \mathsf{Setup}(l)$
$(s, w^*) \leftarrow A_1^O(params)$
$(K_0, U^*) \leftarrow \mathsf{Enc}(w^*, params)$
$(K_1) \overset{\mathrm{R}}{\leftarrow} \mathbb{K}_{\mathsf{ID\text{-}KEM}}$
$b' \leftarrow A_2^O(w^*, U^*, s, K_b)$
If $b' = b$ return 1 else return 0

The advantage of the attacker $A = (A_1, A_2)$ is $Adv^{\mathsf{IND\text{-}ID\text{-}CCA}}_{A,\mathsf{ID\text{-}KEM}} = |Pr[b' = b] - \frac{1}{2}|$. Hence, a biometric ID-KEM is $\mathsf{IND\text{-}ID\text{-}CCA}$ secure if the advantage of A is negligible in the security parameter l. Here O represents the oracles that A has access to, the private key extraction and decapsulation oracles.

Theorem 7.4. *Let Π be a γ-uniform anonymous IBE scheme with negligable γ. Suppose that the hash functions H, G are random oracles and let A be an IND-ID-CCA adversary which has advantage $\epsilon'(l)$ against anonymous ID-KEM. Suppose A makes at most q_1 H-queries, q_2 G-queries,q_E Extraction queries and q_D Decapsulation queries. Then there is an OW-ID-CPA adversary R against anonymous Π which has advantage at least $\epsilon'(l) \geq \frac{2\epsilon(l)}{q_1+q_D} - dq_D\gamma$.*

Proof. Given an $\mathsf{IND\text{-}ID\text{-}CCA}$ secure anonymous biometric ID-KEM, the goal of the reduction algorithm R is to invert the $\mathsf{OW\text{-}ID\text{-}CPA}$ secure anonymous IBE scheme using an adversary A running against ID-KEM.

The challenger of R outputs the public parameters of IBE, which is passed to the adversary A in order to simulate the setup phase of ID-KEM.

A responds with the challenge biometric identity w^*, which is relayed to the challenger of R, which returns the encryption U^* of a random message $c^* \in C$ encrypted with w^*. R outputs U^* together with a random key K_0 to simulate the challenge phase of ID-KEM and answers the random oracle and decapsulation queries of A as follows.

1. H-queries: On each new input (c, w_j), R picks random h and g from the ranges of G, H returns h to A. Next, R computes $U_j = \mathsf{Encrypt}^{\Pi}_{w_j}(c_j; H(c, w_j))$ inserts the tuple (c, U_j, h) to the HList and and (c, g) to the GList.

2. G-queries: On each new input (c), R returns a random g and adds the tuple (c, g) to the GList.

3. Private Key Extraction queries: For any identity w, the extraction query is passed to the challenger of R.

4. Decapsulation queries: On each new input (w, U),

   - If $|w \cap w^*| < d$, R runs the private key extraction oracle and answers A as the real decapsulation oracle would.
   - If $|w \cap w^*| \geq d$, R checks $\mathsf{Encrypt}^{\Pi}_{w}(c; H(c, w)) = U$. Since the encryption is performed bit by bit, we actually check whether at least d values of the set $\langle \mathsf{Encrypt}^{\Pi}_{w_1}(c_1; H(c, w_1)), ..., \mathsf{Encrypt}^{\Pi}_{w_n}(c_n; H(c, w_n)) \rangle$ is found in the HList. If we can find at least d such values, the check is successful, then R returns $H_2(c)$. If not, R returns reject.

Finally, A outputs its guess b'. R will pick at random an entry from HList or GList and returns this to the challenger.

Let Fail to be the event that R fails to answer a decapsulation query correctly at some point during the game, which can occur only when A submits a decryption query (w, U), where $U = \{U_j : w_j \in w\} = \{\mathsf{Encrypt}^{\Pi}_{w_j}(c_j; H(c, w_j)) : w_j \in w\, c_j \in c\}$ without asking $H(c, w_j)$ for at least d features of w. R fails to properly answer each such decryption query with probability at most γ^d. Thus, due to the q_D decryption queries, we have $\Pr[\neg\mathsf{Fail}] \leq (1 - \gamma^d)^{q_D} \approx 1 - q_D(\gamma^d)$. Similar to the previous constructions, we use our new method to prevent collision attacks by replacing w with the biometric identity BID, which is resistant to collision attacks. □

As we can se, this ID-KEM construction corresponds to the first part of the FO conversion without the message m, thus, if the underlying IBE scheme is anonymous, we obtain also an CCA secure and anonymous ID-KEM. The proof is identical (except we

remove the message m and the oracle corresponding to the second part of the FO conversion) to our first generic construction based on FO conversion. To avoid repetition, we leave the details to the reader.

7.7 Applications

In this section, we present two concrete instantiations based on the anonymous IBE schemes of [Boneh et al., 2007, Ateniese and Gasti, 2009], which do not require bilinear pairings and encrypt a message bit by bit. Thus, they could be used as an input to our generic constructions with the following modifications to avoid collusion attacks.

7.7.1 Based on the scheme of Boneh et al.

The first space efficient IBE scheme AnonIBE is introduced in [Boneh et al., 2007], which is IND-ID-CPA secure in the standard model based on the difficulty of the Interactive Quadratic Residuosity (IQR) problem and the encryption of a n-bit message results in a single element in $\mathbb{Z}_N$ plus $n+1$ additional bits. Here, N denotes a RSA composite. First, we briefly review the scheme AnonIBE and its main components, the reader is referred to [Boneh et al., 2007] for the details.

Definition 7.2. *Let Q' be a deterministic algorithm that takes as input (N, u, R, S) where $N \in \mathbb{Z}^+$ and $u, R, S \in \mathbb{Z}_N$. The algorithm outputs polynomials $f, \bar{f}, g, \tau \in \mathbb{Z}_N[x]$. We say that Q' is Enhanced IBE Compatible if the following conditions hold [Boneh et al., 2007]:*

- (Condition 1a) If R and S are quadratic residues, then $f(r)g(s)$ is also a quadratic residue for all square roots r of R and s of S.
- (Condition 1b) If uR and S are quadratic residues, then $\bar{f}(\bar{r})g(s)\tau(s)$ is also a quadratic residue for all square roots $\bar{r}$ of uR and s of S.
- (Condition 2a) If R is a quadratic residue, then $f(r)f(-r)S$ is also a quadratic residue for all square roots r of R.
- (Condition 2b) If uR is a quadratic residue, then $\bar{f}(\bar{r})\bar{f}(-\bar{r})S$ is also a quadratic residue for all square roots $\bar{r}$ of uR.
- (Condition 2c) If S is a quadratic residue, then $\tau(s)\tau(-s)u$ is also a quadratic residue for all square roots s of S.

- (Condition 3) τ is independent of R, that is, $Q'(N, u, R_1, S)$ and $Q'(N, u, R_2, S)$ produce the same τ for all N, u, R_1, R_2, S.

An example for Enhanced IBE Compatible Q' is given in [Boneh et al., 2007].

The authors define an efficient anonymous IBE scheme AnonIBE using the Enhanced IBE Compatible Q' as follows.

- Setup: Generate two primes (p, q) and compute $N = pq$, where N is a RSA composite. Select a random $u \xleftarrow{R} J(N)/\mathrm{QR}(N)$. Here, $J(N)$ denotes the set $\{x \in \mathbb{Z}_N : (\frac{x}{N}) = 1\}$, where $(\frac{x}{N})$ is the Jacobi symbol of x in $\mathbb{Z}_N$. Also, $\mathrm{QR}(N)$ is the set of quadratic residues in $J(N)$. The public parameters are $params = (N, u, H)$, where H is a hash function $H : \mathcal{ID} \times [1, n] \to J(N)$. The master key is $msk = (p, q, K)$, namely the factorization of N together with a random key K for a pseudorandom function $F_K : \mathcal{ID} \times [1, n] \to \{0, 1, 2, 3\}$.

- Extract: It takes as input msk, an identity ID and a message length parameter n. The algorithm outputs a private key $D_{ID} = (r_1, ..., r_n)$ for decrypting encryptions of n-bit messages as follows. For j= 1,...,n do:

  1. $R_j \leftarrow H(ID, j) \in J(N)$ and $t \leftarrow F_K(ID, j) \in \{0, 1, 2, 3\}$
  2. let $a \in \{0, 1\}$ such that $u^a R_j \in \mathrm{QR}(N)$
  3. let z_0, z_1, z_2, z_3 be the four square roots of $u^a R_j \in \mathbb{Z}_N$ and set $r_j \leftarrow z_t$

- Encrypt: The encryption algorithm that takes as input the identity ID of the receiver, $params$ and a message $m = m_1...m_n \in \{-1, +1\}^n$. It generates a random $s \in \mathbb{Z}_N$ and sets $S \leftarrow s^2 \bmod N$. Then, $Q'(N, u, 1, S)$ is computed to obtain the polynomial τ and $k \leftarrow (\frac{\tau(s)}{N})$. Here, Q' is a deterministic algorithm that satisfies the properties listed above and we run Q' with inputs (N, u, R_j, S) for $j = 1, ..., n$, $N \in \mathbb{Z}^+$, and $u, R_j, S \in \mathbb{Z}_N$.

  Specifically, for $j = 1, ..., n$ do:

  1. Compute $R_j \leftarrow H(ID, j)$ and run $Q'(N, u, R_j, S)$ to obtain g_j
  2. Compute $e_j = m_j \cdot (\frac{g_j(s)}{N})$

  The ciphertext is $U = (S, k, e)$, where $e = e_1...e_n$.

- Decrypt: The decryption algorithm takes as input the ciphertext U and the private key $D_{ID} = (r_1, ..., r_n)$ and recovers $m' = m'_1...m'_n$ as follows. For $j = 1, ..., n$, set $R_j \leftarrow H(ID, j)$ and run $Q'(N, u, R_j, S)$ to obtain f_j, $\bar{f}_j$

  If $r_j^2 = R_j$ set $m_j \leftarrow e_j \cdot (\frac{f_j(r_j)}{N})$, else if $r_j^2 = uR_j$ set $m_j \leftarrow e_j \cdot k \cdot (\frac{\bar{f}_j(r_j)}{N})$

In order to prevent collision attacks, we slightly modify AnonIBE using our method, where we take the biometric identity as $\mathsf{BID} = (\mathsf{BID}_1, ..., \mathsf{BID}_n) = (w_1||\mathsf{ID}, ..., w_n||\mathsf{ID})$ instead of the ID and thus, we use $H(w_j, \mathsf{ID})$ in place of $H(ID, j)$ for $j = 1, ..., n$. Since the features $w_j \in \mathbb{U}$ are ordered, w_1 represents $j = 1$, w_2 represents $j = 2$, etc. for the particular biometric identity string ID. Since we need to encrypt (or encapsulate) a random codeword c in our generic constructions, the message m that is encrypted in AnonIBE becomes $m = c$. This way, we obtain the following IND-ID-CPA biometric IBE scheme, which is also anonymous if implemented for ordered biometrics.

1. Setup: The same as AnonIBE except for the hash function H, which is taken as $H : \mathcal{BID} \to J(N)$, where $\mathcal{BID} = \mathbb{U} \times \mathcal{ID}$. We assume that the features $w_j \in \mathbb{U}$ are ordered as in [Li et al., 2006, Sutcu et al., 2007, Chang et al., 2006]. The master key is $msk = (p, q, K)$, namely the factorization of N together with a random key K for a pseudorandom function $F_K : \mathbb{U} \times \mathcal{ID} \to \{0, 1, 2, 3\}$.

2. Extract: It takes as input msk, a biometric vector w with length n. The algorithm outputs a private key for biometric identity BID as $D_{\mathsf{BID}} = (r_1, ..., r_n)$ for decrypting encryptions of n-bit messages as follows. For j= 1,...,n do:

   - $R_j \leftarrow H(w_j, \mathsf{ID}) \in J(N)$ and $t \leftarrow F_K\ (w_j, \mathsf{ID}) \in \{0, 1, 2, 3\}$.
   - let $a \in \{0, 1\}$ such that $u^a R_j \in \mathrm{QR}(N)$.
   - let z_0, z_1, z_2, z_3 be the four square roots of $u^a R_j \in \mathbb{Z}_N$ and set $r_j \leftarrow z_t$.

3. Encrypt: The encryption algorithm that takes as input (collision resistant) biometrics BID' of the receiver, $params$ and a codeword $c = c_1...c_n \in \mathcal{C}$. It generates a random $s \in \mathbb{Z}_N$ and sets $S \leftarrow s^2 \bmod N$. Then, $Q'(N, u, 1, S)$ is computed to obtain the polynomial τ and $k \leftarrow (\frac{\tau(s)}{N})$. Here, Q' is a deterministic algorithm that satisfies some properties [Boneh et al., 2007] and takes as inputs (N, u, R_j, S), where $N \in \mathbb{Z}^+$, and $u, R_j, S \in \mathbb{Z}_N$. It outputs polynomials $f_j, \bar{f}_j, g_j, \tau \in \mathbb{Z}_N[x]$. Finally, for $j = 1, ..., n$ do:

   - Compute $R_j \leftarrow H(\ w_j, \mathsf{ID})$ and run $Q'(N, u, R_j, S)$ to obtain g_j.
   - Compute $e_j = c_j \cdot (\frac{g_j(s)}{N})$.

   The ciphertext is $U = (S, k, e)$, where $e = e_1...e_n$.

4. Decrypt: The decryption algorithm takes as input the ciphertext U and the private key $D_{\mathsf{BID}} = (r_1, ..., r_n)$ and recovers $c = c_1...c_n$ as follows. For $j = 1, ..., n$, set $R_j \leftarrow H(w_j,\mathsf{ID})$ and run $Q'(N, u, R_j, S)$ to obtain f_j, $\bar{f}_j$

   If $r_j^2 = R_j$ set $c_j \leftarrow e_j \cdot (\frac{f_j(r_j)}{N})$, else if $r_j^2 = uR_j$ set $c_j \leftarrow e_j \cdot k \cdot (\frac{\bar{f}_j(r_j)}{N})$.

As noted before, the security of the Anonymous IBE depends on the difficulty of the interactive quadratic residuosity (IQR) problem in the standard model and QR problem in ROM. The encryption of a binary string results as a ciphertext of size $log_2 N + n + 1$, where N is a RSA modulus and n is length of c and BID. For simplicity, we assume that the length of the codeword c and the size of the biometric feature set w is equal. Since we cannot change the message size (i.e. the codeword), we have to adapt the identity, if $|w| \leq |c|$. For this, we can apply the solutions discussed in remark 7.1. Since the modified scheme is also secure in the sense of IND-ID-CPA, it is input to one of our generic constructions to obtain either an IND-ID-CCA secure encryption scheme or an IND-ID-CCA secure KEM.

The main drawback of the scheme of [Boneh et al., 2007] is its inefficiency since the complexity is quartic in the security parameter. Recently, Ateniese and Gasti [Ateniese and Gasti, 2009] proposed an efficient and universally anonymous IBE scheme based on the QR assumption in the ROM. Similar to the modification presented above, if the key generation of the scheme in [Ateniese and Gasti, 2009] is adapted for biometric identities, we are able to integrate this modified IBE scheme into one of our generic constructions.

7.7.2 Based on the scheme of Ateniese et al.

The second application of our generic construction is based on the scheme of [Ateniese and Gasti, 2009], whose security relies on the QR assumption in the ROM. Similar to the scheme of [Boneh et al., 2007], an n-bit message (i.e. codeword c) is encrypted bit by bit resulting in a ciphertext of $2n(120+1024)$ bits if necessary optimizations suggested in [Ateniese and Gasti, 2009] are applied. Thus, it could be used as an input to our generic constructions with the following modifications to avoid collision attacks. Let us first review briefly the basic scheme, the reader is referred for the details to [Ateniese and Gasti, 2009].

- Setup: Let H be a full domain hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_N^*[+1]$, where $\mathbb{Z}_N^*[+1]$ denotes the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol $+1$, $k_0$and k security parameters (e.g., $k_0 = 1024$ and $k = 128$). Generate two primes (p, q) and compute $N = pq$, where N is a k_0-bit Blum integer and p, q are two $k_0/2$-bit primes each congruent to 3 modulo 4. The public parameters are $params = (N, H)$ and the master secret key is $msk = (p, q)$.

- Extract: It takes as input msk, and an identity ID. The algorithm outputs a private key $D_{ID} = r$ by computing $a \leftarrow H(ID)$. Thus, the jacobi symbol $(\frac{a}{N}) = +1$. $r \in \mathbb{Z}_N^*$ is chosen such that $r^2 \equiv a \bmod N$ or $r^2 \equiv -a \bmod N$.

- Encrypt: The encryption algorithm that takes as input ID of the receiver, $params$ and a bit $m_j \in \{-1, +1\}$ since we encrypt the message m bit by bit, namely for each bit $m_j \in \{-1, +1\}$ we do the following. For $j = 1, ..., n$,
  - choose at random $t_j, v_j \in \mathbb{Z}_N^*$ such that $(\frac{t_j}{N}) = (\frac{v_j}{N}) = m_j$.
  - compute $(f_j, g_j) = (t_j + \frac{a}{t_j}, v_j - \frac{a}{v_j})$ and select random $T_j, V_j \in \mathbb{Z}_N^*$ and set $Z_j^1 = f_j + T_j$,$Z_j^2 = g_j + V_j$
  - mask the ciphertext using one of the constructions in [Ateniese and Gasti, 2009].

  The encryptor sends the ciphertext $(Z_j^1, T_j^1, ..., T_j^k)$ and $(Z_j^1, V_j^1, ..., V_j^k)$
- Decrypt: On input the ciphertext and D_{ID}, first the recipient derives the intended ciphertext. For $j = 1, ..., n$, the receiver computes
  - One of the two tuples $(Z_j^1, T_j^1, ..., T_j^k)$ or $(Z_j^2, V_j^1, ..., V_j^k)$ is discarded based on whether a or $-a$ is a square. Lets assume we keep the tuple $(Z_j^1, T_j^1, ..., T_j^k)$ and discard the other.
  - In order to decrypt, find the smallest index $1 \leq i \leq k$ such that $GT(a, Z_j^1 - T_j^i, N) = +1$. Here $GT(\cdot)$ denotes the Galbarith's test, which is defined over the public key a as the Jacobi symbol of $GT(a, Z_j^1 - T_j^i, N) = (\frac{(Z_j^1 - T_j^i)^2 - 4a}{N})$
  - Output $(\frac{Z_j^1 - T_j^i + 2r}{N}) = m_j$
  - we run the same procedure above if the second tuple is selected and the first tuple is discarded by replacing a with $-a$, Z_j^1 with Z_j^2, and T_j^i with V_j^i.

In order to prevent collision attacks and due to the structure of our biometric identity, we encrypt each bit of the message (i.e. codeword c), which is n bits long, using the biometric identity vector of size n. Specifically, the identity $a = H(ID)$ that is used to encrypt a single bit m_j is replaced by $a_j = H(w_j, \mathsf{ID})$ for encrypting each bit m_j of the message individually. As opposed to our modified scheme based on [Boneh et al., 2007], the difference between the number of features and the message size does not cause any problem as the message size is not a fixed parameter of the scheme of [Ateniese and Gasti, 2009]. Since each bit of the message is encrypted individually without depending on a fixed message size parameter, if the size of the biometric feature set is less than the size of the message (i.e. the codeword), the remaining bits of the message are individually encrypted by starting again from the first feature till the message is completed.

Remark 7.2. *In [Ateniese and Gasti, 2009], the authors suggest the use of a function $G : \{0, 1\}^* \to \mathbb{Z}_N^*$, which is modeled as a random oracle that maps a uniformly random*

e-bit string α to a random value in $\mathbb{Z}_N^*$. The parameter e must be large enough, e.g., e = 160. G is used to expand a short seed α into a value selected uniformly and independently in $\mathbb{Z}_N^*$. Thus, for the encryption of the bit m_j, a single short seed $\alpha_j \xleftarrow{R} \{0,1\}^e$ (and $\beta_j \xleftarrow{R} \{0,1\}^e$) plus a counter to generate all values T_j^i's (and V_j^i's) is required. This first solution would provide minimal ciphertext expansion, since only the seed α_j must be sent for the encryption of each bit. However, this solution is computationally expensive, hence, the authors suggest a second efficient variant, where for each plaintext, the sender selects a random message identifier MID_m which is sent along with the ciphertext. Also a new global parameter l which is a small positive integer is selected. For bit m_j, the sender computes: $(Z_j^1, \alpha_j^1, ..., \alpha_j^l)$ or $(Z_j^2, \beta_j^1, ..., \beta_j^l)$ where $\alpha_j^i, \beta_j^i \in \{0,1\}^e$, when $i < l$ and $\alpha_j^l, \beta_j^l \in \{0,1\}^{e'}$ for some $e' > e$. The intended ciphertext is derived by the recipient by computing:

$T_j^i = G(MID_m||0||\alpha_j^i||i||j)$ *or* $V_j^i = G(MID_m||1||\beta_j^i||i||j)$ *for* $i < l$

$T_j^i = G(MID_m||0||\alpha_j^i||i||j)$ *or* $V_j^i = G(MID_m||1||\beta_j^i||i||j)$ *for* $i \geq l$

7.8 Comparison

To show the efficiency of our constructions, we will compare our results to the existing IND-CPA secure fuzzy IBE schemes secure in the ROM. In [Ateniese and Gasti, 2009], the authors implement different anonymous IBE schemes to present the average times of encryption of a short session key. Using these values presented in [Ateniese and Gasti, 2009], we compare our results to any pairing based fuzzy IBE system in Table 7.1. For simplicity, we use different variables to represent the approximate times, where x and y denote the encryption and decryption times for Boneh-Franklin IBE [Boneh and Franklin, 2003] scheme implemented for a unique identity such as an e-mail address. Specifically, x is the time to compute two exponentiations within their respective groups if the bilinear pairing is precomputed and y is the time for one pairing computation, which is the dominant operation in terms of computation cost. For fuzzy IBE systems, since the identity is represented as a set of features of size n with $20 < n < 100$ depending on the biometric modality, the required times are computed as multiples of x and y. When compared to the exact times of the scheme [Ateniese and Gasti, 2009] that we implement for our generic construction, the encryption of a message of the same size requires approximately $4x$, whereas the decryption time is again y. Here, k is the size of the message, d is the error tolerance parameter, which can be $10 < n < 50$ depending on the biometric modality. The only disadvantage of this system is the large bandwidth due to the bit by bit encryption, which we analyze in table 7.2. For this, we compare the bandwidth required for pairing based fuzzy IBE systems based on a bilinear pairing group $(\hat{e}, \mathbb{G}, \mathbb{F})$ such that $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{F}$ where we take the suggested

parameters of [Galindo and Garcia, 2009] as $|\mathbb{Z}_q| \approx |\mathbb{G}| \geq 512$ and $|\mathbb{F}| \approx 3072$. Here, we write $|\mathbb{G}|$ to denote the number of bits needed to represent an element in $\mathbb{G}$. Finally, for our construction, a fuzzy extraction procedure FE for encryption and an error correcting procedure ECC for decryption stage is required. Again, the computation of FE is performed only once and for all. As one can note, there is a tradeoff between the bandwidth of our two constructions and their computational efficiency.

Table 7.1: Computational Cost of various IND-CPA secure biometric IBE systems

	Encryption time	Decryption time	Anonymity	Multi-modal Biometrics
Current Pairing based fuzzy IBE Systems†	nx	dy	No	No
Anonymous fuzzy IBE [van Liesdonk, 2007]	ny	ny	Yes	No
Our Construction‡	$4x$ + FE^*	y+ ECC	Yes	Yes

†:for biometric identities [Sahai and Waters, 2005, Pirretti et al., 2006, Baek et al., 2007, Furukawa et al., 2008];
‡:Based on the scheme of [Ateniese and Gasti, 2009];
$*$: FE computed only once and for all.

Table 7.2: Ciphertext size of various biometric IBE systems

Biometric IBE Systems	Bandwidth
Pairing based fuzzy IBE Systems†	$2n512$ + 3072 bits
Our Construction based on [Boneh et al., 2007]	k +1 + 1024 bits
Our Construction based on [Ateniese and Gasti, 2009]	$2k(120 + 1024)$ bits

n: size of the feature set; k: size of the message;
†:for biometric identities [Sahai and Waters, 2005, Pirretti et al., 2006, Baek et al., 2007, Furukawa et al., 2008, van Liesdonk, 2007, Sarier, 2008];

7.9 Conclusion

In this chapter, we present a new design for biometric IBE, which results in a highly secure encryption system preserving the anonymity of the underlying encryption system if implemented for ordered biometrics. In addition to reduced computational costs, the security of our design could be based on stronger (standard) assumptions as opposed to the current fuzzy IBE systems. We note that our system is only designed for biometric applications, however, an interesting future work could be the design of a different

method for preventing collision attacks for IBE systems without depending on pairings, which may lead to generalize our system to attribute based encryption.

Chapter 8

Conclusion

In this thesis, we focused on biometric cryptosystems that are designed according to a realistic security model and that provide a security reduction to guarantee the security notions of this model. Unfortunately, we noticed that very few biometric cryptosystems in the fields of remote authentication, encryption and signature are evaluated from a cryptographic point of view. We start by analyzing the distributed remote authentication schemes in the literature and show that almost all of these schemes have a different security gap despite the security reductions they provide. This is actually due to an inherent weakness in these constructions that require a secure sketch for improved accuracy; the sketch itself leak information to the internal adversaries. We show that the security notions, in particular identity privacy notion cannot be guaranteed if the sketch is stored either in a public database or as encrypted at the server-side, where the components of the server-side are assumed to be malicious in this model. However, if the error-correction procedure is performed at the client side in the setting of a two factor authentication or at least the helper data is stored secretly in the user's smartcard, then the security notions can be achieved for the current protocols.

Generally, for biometric remote authentication, simultaneous attacks against the biometric system can only be prevented by multi-factor solutions. To achieve this goal, we combine basic cryptographic primitives such as homomorphic encryption and zero-knowledge proofs with a tamper-proof smartcard that stores the cancelable biometric transformation parameters. As we pointed out, the concept of "Encryption with Equality Testing" (EET) enables the server to make the authentication decision without using any decryption key. Gap Diffie Hellman groups give rise to a natural application of EET, which we already implemented before the actual introduction of the concept.

The second part of the thesis is devoted to the design of biometric IBE/IBS systems in a stronger security model achieving better efficiency. The main problem with current

fuzzy IBE systems is that they all require the same key generation technique for binding the biometric features to the user, which allows for a limited number of protocols, whose security is based on stronger (bilinear) assumptions. However, by simply binding the biometric features using a fuzzy-extracted identity string -either from the same or a different biometric trait of the user-, a wider class of IBE systems become applicable for biometric identities. The outcome of this tweak was tremendous as it made the constructions rest on IBE schemes without bilinear pairings, and consequently led to more efficient biometric IBE schemes with much smaller decryption cost. For particular biometric modalities, our generic constructions based on error-correcting codes, conversion schemes and weakly secure anonymous IBE schemes preserves the anonymity property of the underlying IBE scheme while upgrading its security.

The immediate prospect of this different key generation method is its extension to other identity-based mechanisms/signatures, which share the various advantages of using biometric identities as public keys. In the future, we expect new applications of biometric cryptosystems including certificateless encryption schemes with biometric public keys, biometric-based key agreement protocols and other multi-factor protocols for security applications.

Bibliography

M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *CRYPTO'05*, volume 3621 of *LNCS*, pages 205–222. Springer, 2005.

A. Adler. Vulnerabilities in biometric encryption systems. In *AVBPA'05*, pages 1100–1109, 2005.

S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In *ASIACRYPT'03*, volume 2894 of *LNCS*, pages 452–473. Springer, 2003.

K. M. Apampa, T. Zhang, G. B. Wills, and D. Argles. Ensuring privacy of biometric factors in multi-factor authentication systems. In *SECRYPT'08*, pages 44–49. INSTICC Press, 2008.

G. Ateniese and P. Gasti. Universally Anonymous IBE Based on the Quadratic Residuosity Assumption. In *CT-RSA'09*, volume 5473 of *LNCS*, pages 32–47. Springer, 2009.

J. Baek, R. Safavi-Naini, and W. Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In *Public Key Cryptography-PKC'05*, volume 3386 of *LNCS*, pages 380–397. Springer, 2005.

J. Baek, W. Susilo, and J. Zhou. New constructions of fuzzy identity-based encryption. In *ACM ASIACCS'07*, pages 368–370. ACM, 2007.

L. Ballard, S. Kamara, F. Monrose, and M. K. Reiter. Towards practical biometric key generation with randomized biometric templates. In *ACM CCS'08*, pages 235–244. ACM, 2008.

M. Barbosa, T. Brouard, S. Cauchie, and S. M. D. Sousa. Secure biometric authentication with improved accuracy. In *ACISP'08*, volume 5107 of *LNCS*, pages 21–36. Springer, 2008.

P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. In *ASIACRYPT'05*, volume 3788 of *LNCS*, pages 515–532. Springer, 2005.

M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM CCS'06*, pages 390–399. ACM, 2006.

M. Bellare and A. Palacio. Towards plaintext-aware public-key encryption without random oracles. In *ASIACRYPT'04*, volume 3329 of *LNCS*, pages 48–62. Springer, 2004.

M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT'94*, pages 92–111, 1994.

M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO'98*, pages 26–45, 1998.

M. Bellare, C. Namprempre, and G. Neven. Security Proofs for Identity-Based Identification and Signature Schemes. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 268–286. Springer, 2004.

K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. *J. Cryptology*, 21(2):178–199, 2008.

A. Bhargav-Spantzel, A. C. Squicciarini, and E. Bertino. Privacy preserving multi-factor authentication with biometrics. In *Digital Identity Management*, pages 63–72. ACM, 2006.

C. Boehnen, T. Peters, and P. J. Flynn. 3d signatures for fast 3d face recognition. In *ICB'09*, volume 5558 of *LNCS*, pages 12–21. Springer, 2009.

D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *EUROCRYPT'04*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

D. Boneh, C. Gentry, and M. Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In *FOCS'07*, pages 647–657. IEEE, 2007.

T. E. Boult, W. J. Scheirer, and R. Woodworth. Revocable fingerprint biotokens: Accuracy and security analysis. In *CVPR'07*. IEEE, 2007.

X. Boyen. Reusable cryptographic fuzzy extractors. In *ACM CCS'04*, pages 82–91. ACM, 2004.

X. Boyen. General *d hoc* encryption from exponent inversion ibe. In *EUROCRYPT'07*, volume 4515 of *LNCS*, pages 394–411. Springer, 2007.

X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In *EUROCRYPT'05*, volume 3494 of *LNCS*, pages 147–163. Springer, 2005.

J. Bringer and H. Chabanne. An authentication protocol with encrypted biometric data. In *AFRICACRYPT'08*, volume 5023 of *LNCS*, pages 109–124. Springer, 2008.

J. Bringer and V. Despiegel. Binary Feature Vector Fingerprint Representation From Minutiae Vicinities. In *BTAS'10*, pages 1–6. IEEE, 2010.

J. Bringer, H. Chabanne, G. D. Cohen, B. Kindarji, and G. Zémor. Optimal Iris Fuzzy Sketches. In *BTAS'07*, pages 1–6. IEEE, 2007a.

J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, and S. Zimmer. An application of the goldwasser-micali cryptosystem to biometric authentication. In *ACISP'07*, volume 4586 of *LNCS*, pages 96–106. Springer, 2007b.

J. Bringer, H. Chabanne, D. Pointcheval, and Q. Tang. Extended private information retrieval and its application in biometrics authentications. In *CANS'07*, volume 4856 of *LNCS*, pages 175–193. Springer, 2007c.

J. Bringer, H. Chabanne, D. Pointcheval, and S. Zimmer. An Application of the Boneh and Shacham Group Signature Scheme to Biometric Authentication. In *IWSEC'08*, volume 5312 of *LNCS*, pages 219–230. Springer, 2008.

A. Burnett, F. Byrne, T. Dowling, and A. Duffy. A Biometric Identity Based Signature Scheme. *International Journal of Network Security*, 5(3):317–326, 2007.

J.L. Cambier, U. Cahn von Seelen, R. Moore, I. Scott, M. Braithwaite, and J. Daugman. Application specific biometric templates. In *IEEE Workshop on Automatic Identification Advanced Technologies*, pages 167–171. IEEE, 2002.

E. Chang and Q. Li. Hiding secret points amidst chaff. In *EUROCRYPT'06*, volume 4004 of *LNCS*, pages 59–72. Springer, 2006.

W. Chang, R. Shen, and F.W. Teo. Finding the original point set hidden among chaff. In *ACM ASIACCS'06*, pages 182–188. ACM, 2006.

C. Chen, R. N. J. Veldhuis, T. A. M. Kevenaar, and A. H. M. Akkermans. Multi-bits biometric string generation based on the likelyhood ratio. In *BTAS'07*, pages 1–6. IEEE, 2007.

L. Chen and Z. Cheng. Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. In *Cryptography and Coding, IMA Int. Conf.*, volume 3796 of *LNCS*, pages 442–459. Springer, 2005.

L. Chen, Z. Cheng, J. Malone-Lee, and N. Smart. Efficient ID-KEM based on the Sakai-Kasahara key construction. *IEE Proceedings Information Security*, 153(1): 19–26, 2006.

T. C. Clancy, N. Kiyavash, and D. J. Lin. Secure smartcard based fingerprint authentication. In *WBMA'03*, pages 45–52. ACM, 2003.

C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding'01*, pages 360–363. Springer, 2001.

R. Cramer, R. Gennaro, and B. Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.

R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan. Bounded cca2-secure encryption. In *ASIACRYPT'07*, volume 4833 of *LNCS*, pages 502–518. Springer, 2007.

J. Daugman. How iris recognition works. *IEEE Trans. Circuits Syst. Video Techn.*, 14 (1):21–30, 2004.

W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *STOC'05*, pages 654–663. ACM, 2005.

Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT'04*, volume 3027 of *LNCS*, pages 523–540. Springer, 2004.

H. K. Ekenel and R. Stiefelhagen. Generic versus salient region-based partitioning for local appearance face recognition. In *ICB'09*, volume 5558 of *LNCS*, pages 367–375. Springer, 2009.

A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.

C. Fontaine and F.Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP J. Inf. Secur.*, 2007(15):1–15, 2007.

E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.

J. Furukawa, N. Attrapadung, R. Sakai, and G. Hanaoka. A Fuzzy ID-Based Encryption Efficient When Error Rate Is Low. In *INDOCRYPT'08*, volume 5365 of *LNCS*, pages 116–129. Springer, 2008.

S.D. Galbarith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. `http://eprint.iacr.org/`.

D. Galindo and F. D. Garcia. A schnorr-like lightweight identity-based signature scheme. In *AFRICACRYPT'09*, volume 5580 of *LNCS*, pages 135–148. Springer, 2009.

T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, 1984.

H. Gao, H. K. Ekenel, and R. Stiefelhagen. Pose normalization for local appearance-based face recognition. In *ICB'09*, volume 5558 of *LNCS*, pages 32–41. Springer, 2009.

C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In *ICALP'05*, volume 3580 of *LNCS*, pages 803–815. Springer, 2005.

S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *STOC '82*, pages 365–377. ACM, 1982.

G. Hanaoka and H. Imai. A generic construction of cca-secure cryptosystems without nizkp for a bounded number of decryption queries. Cryptology ePrint Archive, Report 2006/408, 2006. `http://eprint.iacr.org/`.

R. Heyer. Biometrics technology review 2008. Australian Goverment Department of Defence, Land Operations Division, Defence Science and Technology Organisation DSTO-GD-0538, http://dspace.dsto.defence.gov.au/dspace/bitstream/1947/9704/1/DSTO-GD-053820PR.pdf, 2008.

S. Hirata and K. Takahashi. Cancelable biometrics with perfect secrecy for correlation-based matching. In *ICB'09*, volume 5558 of *LNCS*, pages 868–878. Springer, 2009.

J. J. Igarza, I. Hernaez, I. Goirizelaia, and K. Espinosa. Applying dynamic methods in off-line signature recognition. In *Biometric Technology for Human Identification'04*, volume 5404, pages 418–424. SPIE, 2004.

Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Batch codes and their applications. In *STOC'04*, pages 262–271. ACM, 2004.

Y. Itakura and S. Tsujii. Proposal on a multifactor biometric authentication method based on cryptosystem keys containing biometric signatures. *Int. J. Inf. Sec.*, 4(4): 288–296, 2005.

A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9(5):846–859, 2000.

D. Jao. Co 485/685: Lecture 13. `djao.math.uwaterloo.ca/wiki/images/0/02/Lecture13.pdf`, 2009.

A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, 2006.

A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM CCS'99*, pages 28–36. ACM, 1999.

S. Kanade, D. Petrovska-Delacrétaz, and B. Dorizzi. Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data. In *CVPR'09*, pages 120–127. IEEE, 2009.

J. Katz. *Efficient Cryptographic Protocols Preventing 'Man-in-the-Middle' Attacks*. PhD thesis, COLUMBIA UNIVERSITY, 2002.

D. Khader. Attribute Based Group Signatures. Cryptology ePrint Archive, Report 2007/159, 2007. `http://eprint.iacr.org/`.

E. Kiltz and G. Neven. *Identity-Based Cryptography*, chapter Identity-Based Signatures. IOS Press, 2009.

T. Kitagawa, P. Yang, G. Hanaoka, R. Zhang, H. Watanabe, K. Matsuura, and H. Imai. Generic Transforms to Acquire CCA-Security for Identity Based Encryption: The Cases of FOpkc and REACT. In *ACISP'06*, volume 4058 of *LNCS*, pages 348–359. Springer, 2006.

N. Koblitz and A. Menezes. Intractable problems in cryptography. Cryptology ePrint Archive, Report 2010/290, 2010. `http://eprint.iacr.org/`.

E. Kushilevitz and R. Ostrovsky. Replication is not needed: single database, computationally-private information retrieval. In *FOCS '97*, page 364. IEEE, 1997.

Q. Li, Y. Sutcu, and N. D. Memon. Secure Sketch for Biometric Templates. In *ASIACRYPT'06*, volume 4284 of *LNCS*, pages 99–113. Springer, 2006.

S. Z. Li and A. K. Jain, editors. *Biometric Encryption*. Springer, 2009.

J. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *AVBPA'03*, volume 2688 of *LNCS*, pages 393–402. Springer, 2003.

H. Lipmaa. An oblivious transfer protocol with log-squared communication. In *ISC'05*, volume 3650 of *LNCS*, pages 314–328. Springer, 2005.

J. K. Liu, M. H. Au, and W. Susilo. Self-Generated-Certificate Public Key Cryptography and Certificateless Signature/Encryption Scheme in the Standard Model. In *ACM ASIACCS'07*, pages 273–283. ACM, 2007.

J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Short generic transformation to strongly unforgeable signature in the standard model. In *ESORICS'10*, volume 6345 of *LNCS*, pages 168–181. Springer, 2010.

H. Maji, M. Prabhakaran, and M. Rosulek. Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance. Cryptology ePrint Archive, Report 2008/328, 2008. `http://eprint.iacr.org/`.

P. Mansukhani, S. Tulyakov, and V. Govindaraju. Using support vector machines to eliminate false minutiae matches during fingerprint verification. In *Biometric Technology for Human Identification IV*, volume 6539, pages 1–6. SPIE, 2007.

C.A Melchor and P. Gaborit. A fast private information retrieval protocol. In *ISIT'08*, pages 1848–1852. IEEE, 2008.

P. Mihailescu. The fuzzy vault for fingerprints is vulnerable to brute force attack. *CoRR*, abs/0708.2974, 2007.

P. Mihăilescu, A. Munk, and B. Tams. The fuzzy vault for fingerprints is vulnerable to brute force attack. In *BIOSIG'09*, volume 155 of *LNI*, pages 43–55. GI, 2009.

A.B. Moreno, A. Sanchez, J. Velez, and J. Diaz. Face recognition using 3d local geometrical features: Pca vs. svm. In *ISPA'05*, pages 185 – 190, 2005.

K. Nandakumar and A. K. Jain. Multibiometric Template Security Using Fuzzy Vault. In *BTAS'08*, pages 1–6. IEEE, 2008.

K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007a.

K. Nandakumar, A. Nagar, and A. K. Jain. Hardening fingerprint fuzzy vault using password. In *ICB'07*, volume 4642 of *LNCS*, pages 927–937. Springer, 2007b.

G. Neven, N. Smart, and B. Warinschi. Hash function requirements for schnorr signatures. *Journal of Mathematical Cryptology*, 3(1):69–87, May 2009.

T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform. In *CT-RSA'01*, volume 2020 of *LNCS*, pages 159–175. Springer, 2001.

P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.

K.G. Paterson and S. Srinivasan. Security and anonymity of identity-based encryption with multiple trusted authorities. In *Pairing'08*, volume 5209 of *LNCS*, pages 354–375. Springer, 2008.

K. Peng and F. Bao. Batch range proof for practical small ranges. In *AFRICACRYPT'10*, volume 6055 of *LNCS*, pages 114–130. Springer, 2010.

M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure Attribute-Based Systems. In *ACM CCS'06*, pages 99–112. ACM, 2006.

F. L. Podio and J. S. Dunn. Biometric authentication technology: From the movies to your desktop. Report NIST/Biometric Resource Center, 2001. `http://www.itl.nist.gov/div893/biometrics/Biometricsfromthemovies.pdf`.

D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

H. T. Poon and A. Miri. A collusion attack on the fuzzy vault scheme. *Int'l J. Inf. Sec.*, 1(1):27–34, 2009.

N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

C. Rathgeb. Iris-based biometric cryptosystems. Diplomarbeit, Salzburg University, 2008.

A. Rattani and M. Tistarelli. Robust multi-modal and multi-unit feature level fusion of face and iris biometrics. In *ICB'09*, volume 5558 of *LNCS*, pages 960–969. Springer, 2009.

A. Rattani, D.R. Kisku, M. Bicego, and M. Tistarelli. Feature Level Fusion of Face and Fingerprint Biometrics. In *BTAS'07*, pages 1–6. IEEE, 2007.

R. Rivest. Lecture notes 9: Homomorphic encryption. `web.mit.edu/6.857/OldStuff/Fall01/handouts/L09-homomorphic.ps`, 2001.

R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

A. Ross and A. K. Jain. Multimodal biometrics: An overview. In *EUSIPCO'04*, pages 1221–1224, 2004.

A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS'99*, pages 543–553, 1999.

A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *EUROCRYPT'05*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.

R. Sakai and M. Kasahara. ID based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive, Report 2003/054, 2003. `http://eprint.iacr.org/`.

R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairing. SCIS'00, 2000.

T. Sakashita, Y. Shibata, T. Yamamoto, K. Takahashi, W. Ogata, H. Kikuchi, and M. Nishigaki. A proposal of efficient remote biometric authentication protocol. In *IWSEC'08*, volume 5824 of *LNCS*, pages 212–227. Springer, 2009.

N. D. Sarier. A New Biometric Identity Based Encryption Scheme. In *International Symposium on Trusted Computing - TrustCom'08*, pages 2061–2066. IEEE, 2008.

N. D. Sarier. A New Approach for Biometric Template Storage and Remote Authentication. In *Advances in Biometrics, Third International Conference on Biometrics - ICB'09*, volume 5558 of *LNCS*, pages 909–918. Springer, 2009a.

N. D. Sarier. A Survey of Distributed Biometric Authentication Systems. In *Biometrics and Electronic Signatures - BIOSIG'09*, volume 155 of *LNI*, pages 43–55. GI, 2009b.

N. D. Sarier. Practical Multi-factor Biometric Remote Authentication. In *IEEE Fourth International Conference on Biometrics: Theory, Applications and Systems - BTAS'10*, pages 1–6. IEEE, 2010a.

N. D. Sarier. Improving the Accuracy and Storage Cost in Biometric Remote Authentication Schemes. *J. Network and Computer Applications*, 33(3):268–274, 2010b.

N. D. Sarier. Biometric Identity Based Signature Revisited. In *Revised Selected Papers of EuroPKI'09*, volume 6391 of *LNCS*, pages 271–285. Springer, 2010c.

N. D. Sarier. Generic Constructions of Biometric Identity Based Encryption Systems. In *WISTP'10*, volume 6033 of *LNCS*, pages 90–105. Springer, 2010d. *Best Student Paper Award.*

N. D. Sarier. Security Notions of Biometric Remote Authentication Revisited. In *STM'11*, LNCS. Springer, 2011a. *To appear.*

N. D. Sarier. A New Biometric Identity Based Encryption Scheme Secure Against DoS Attacks. *Security and Communication Networks*, 4(1):23–32, 2011b.

W. J. Scheirer and T. E. Boult. Cracking fuzzy vaults and biometric encryption. In *BSYM'07*, pages 1–6. IEEE, 2007.

W. J. Scheirer and T. E. Boult. Bipartite biotokens: Definition, implementation, and analysis. In *ICB'09*, volume 5558 of *LNCS*, pages 775–785. Springer, 2009.

C. P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3): 161–174, 1991.

C.P. Schnorr and M. Jakobsson. Security of signed elgamal encryption. In *ASIACRYPT'00*, volume 1976 of *LNCS*, pages 73–89. Springer, 2000.

S. F. Shahandashti and R. Safavi-Naini. Threshold Attribute-Based Signatures and Their Application to Anonymous Credential Systems. In *AFRICACRYPT'09*, volume 5580 of *LNCS*, pages 198–216. Springer, 2009.

A. Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.

Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84*, pages 47–53, 1984.

G. Shanqing and Z. Yingpei. Attribute-based Signature Scheme. In *ISA'08*. IEEE, 2008.

K. Simoens, J. Bringer, H. Chabanne, and S. Seys. Analysis of biometric authentication protocols in the blackbox model. *CoRR*, abs/1101.2569, 2011.

N. P. Smart and F. Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Appl. Math.*, 155(4):538–547, 2007.

D. R. Stinson. Some observations on the theory of cryptographic hash functions. *Des. Codes Cryptography*, 38(2):259–277, 2006.

Y. Sutcu, Q. Li, and N. Memon. Secure sketch for biometric templates. In *ASIACRYPT'06*, volume 4284 of *LNCS*, pages 99–113. Springer, 2006.

Y. Sutcu, Q. Li, and N. Memon. How to protect biometric templates. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 1–14. SPIE, 2007.

Q. Tang, J. Bringer, H. Chabanne, and D. Pointcheval. A Formal Study of the Privacy Concerns in Biometric-Based Remote Authentication Schemes. In *ISPEC'08*, volume 4991 of *LNCS*, pages 56–70. Springer, 2008.

A. B. J. Teoh, Y. W. Kuan, and S. Lee. Cancellable biometrics and annotations on biohash. *Pattern Recognition*, 41(6):2034–2044, 2008.

I. Teranishi and W. Ogata. Relationship between standard model plaintext awareness and message hiding. In *ASIACRYPT'06*, volume 4284 of *LNCS*, pages 226–240. Springer, 2006.

C. Tilton. Study report on biometrics in e-authentication. Report INCITS M1/07-0185rev, 2007. `http://standards.incits.org/apps/group_public/download.php/24528/m1070185rev.pdf`.

V. V. T. Tong, H. Sibert, J. Lecoeur, and M. Girault. Biometric fuzzy extractors made practical: A proposal based on fingercodes. In *ICB'09*, volume 4642 of *LNCS*, pages 604–613. Springer, 2007.

Y. Tsiounis and M. Yung. On the security of elgamal based encryption. In *PKC'98*, volume 1431 of *LNCS*, pages 117–134. Springer, 1998.

U. Uludag and A. Jain. Securing fingerprint template: Fuzzy vault with helper data. In *Computer Vision and Pattern Recognition Workshop*. IEEE, 2006.

U. Uludag, S. Pankanti, and A. K. Jain. Fuzzy vault for fingerprints. In *AVBPA'05*, volume 3546 of *LNCS*, pages 310–319. Springer, 2005.

M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C.V. Jawahar. Efficient biometric verification in encrypted domain. In *ICB'09*, volume 5558 of *LNCS*, pages 899–908. Springer, 2009.

P.P. van Liesdonk. Anonymous and Fuzzy Identity-Based Encryption. Master's thesis, Technische Universiteit Eindhoven, 2007.

B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *EUROCRYPT'05*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

Q. Li Y. Sutcu and N. Memon. Secure biometric templates from fingerprint-face features. In *Workshop on Biometrics'07*. IEEE, 2007.

G. Yang, C. Tan, Q. Huang, and D. Wong. Probabilistic public key encryption with equality test. In *CT-RSA'10*, volume 5985 of *LNCS*, pages 119–131. Springer, 2010.

P. Yang, T. Kitagawa, G. Hanaoka, R. Zhang, K. Matsuura, and H. Imai. Applying Fujisaki-Okamoto to Identity-Based Encryption. In *AAECC'06*, volume 3857 of *LNCS*, pages 183–192. Springer, 2006.

P. Yang, Z. Cao, and X. Dong. Fuzzy Identity Based Signature. Cryptology ePrint Archive, Report 2008/002, 2008. `http://eprint.iacr.org/`.

S. Yang and I. Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *ICASSP'05*, pages 609–612. IEEE, 2005.

M. Zhang, B. Yang, W. Zhang, and T. Takagi. Multibiometric Based Secure Encryption and Authentication Scheme with Fuzzy Extractor. *International Journal of Network Security*, 12(1):50–57, 2011.

R. Zhang, G. Hanaoka, and H. Imai. Orthogonality between Key Privacy and Data Privacy, Revisited. In *Inscrypt'07*, volume 4990 of *LNCS*, pages 313–327. Springer, 2007.

X. Zhou and C. Busch. A Novel Privacy Enhancing Algorithm for Biometric System. In *BIOSIG'08*, volume 137 of *LNI*, pages 39–46. GI, 2008.