REPUBLIC OF TÜRKIYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

SMART CARD SECURITY AUTHENTICATIONS USING HOMOMORPHIC ENCRYPTION

Shamam Raed Fadhil AL-TIMEEMI

Master's Thesis

Supervisor

Asst. Prof. Dr. Ayça Kurnaz TÜRKBEN

Istanbul, 2022

SMART CARD SECURITY AUTHENTICATIONS USING HOMOMORPHIC ENCRYPTION

Shamam Raed Fadhıl AL-TIMEEMI

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled SMART CARD SECURITY AUTHENTICATIONS USING HOMOMORPHIC ENCRYPTION prepared by SHAMAM RAED FADHIL AL-TIMEEMI and submitted on 15/12/2022 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

Asst. Prof. Dr. Ayça Kurnaz TÜRKBEN

the Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Ayça Kurnaz TÜRKBEN	Department of Software Engineering, Altinbaş University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altinbaş University	_____
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, Arel University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.
Submission data of the thesis to Institute of Graduate Studies: ____/____/____

I hereby declare that all information and data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Shamam Raed Fadhil AL-TIMEEMI

Signature

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Dr Ayça Kurnaz Türkben for all support during my study.

It's great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

ABSTRACT

SMART CARD SECURITY AUTHENTICATIONS USING HOMOMORPHIC ENCRYPTION

AL-TIMEEMI, Shamam Raed Fadhil

M.Sc, Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Ayça Kurnaz TÜRKBEN

Date: 12/2022

Pages: 75

Nowadays, there are a lot of different systems and platforms, and they all want to be as efficient as possible so that they can provide their customers better service. That persons or computers can trust the data they receive and that it has not been changed is the goal of maintaining communication integrity and authenticity. This may be accomplished by ensuring that the data has not been altered. Utilizing cryptographic methods that are considered standard is necessary in order to maintain integrity. In order to ensure the system's safety, a variety of cryptographic procedures are required. Information that is supposed to be kept private might end up being leaked elsewhere. In this article, we will cover the importance of homomorphic encryption, as well as its uses, applications, and limitations. Theoretical assessments of HE systems including the DGHV, Paillier, BGV, and FHEW are carried out, and an introduction to PKI and encryption approaches based on PKI is also provided. When it comes to the usage of both CPU and memory, the algorithms known as Paillier and RSA, which are used for public-key cryptography, are the most essential. We are able to evaluate the performance of the HE algorithms that we have selected by first installing and then executing the corresponding library software. During the course of the testing method, the RSA public-key encryption library is opened as well as installed in order to evaluate the impact that this library has on the amount of CPU and memory that is used. After that, we will discuss future work that will be triggered by our in-depth investigation of the issue, as well

as new ideas for enhancing higher education that will arise from our findings as a direct consequence of our investigations.

Keywords: HE, RSA, PKI, BGV

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

HE : Homomorphic Encryption

RSA : Rivest Shamir Adelman

HCI : Human Computer Interaction

API : Application Programming Interface

DOS : Denial Of Service

PCI : Payment Card Industry

1. INTRODUCTION

1.1 BACKGROUND

Since Julius Caesar utilized the "Caesar" Cipher in Rome and the Japanese used the "Purple" cipher in WWII, encryption has been an essential element of human life. Using encryption, it is possible to keep sensitive data safe from eavesdroppers who could try to intercept, edit, or create it before it reaches its intended recipient. It is a security technology that can help protect electronic communication and computer programs from unauthorized access. When encrypting, we mean changing the shape of the text, such as employing a code, so that it cannot be deciphered. In the early days of data encryption, there were only two ways for encrypting data: traditional encryption and asymmetric encryption. You may transpose letters by jumbling them together or you can substitute them by replacing them with another letter or number, for example. It is possible to state that transposition was utilized first, followed by the substitution method. Transposing letters has been employed to create the Ceaser cipher, which is one of the earliest ciphers. The use of codes allowed for the later replacement of letters and words. As an example, the Enigma machine built by the German military during World War II employed substitution for ciphering messages and also produced a code book for encrypting the words. Traditional encryption, which only uses transposition and substitution as methods of text encoding, has been in use since antiquity and has aided researchers in the development of more advanced encryption methods like public-key encryption and homomorphic encryption, both of which are covered in the sections that follow.

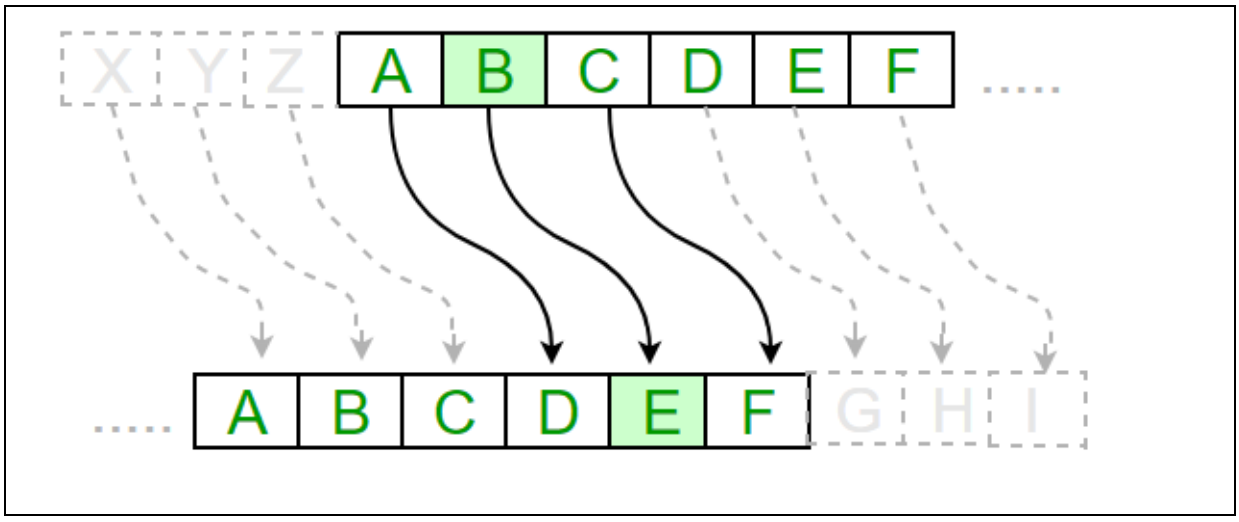

Figure 1.1: Caesar's Cipher in its simplest form

1.2 PROBLEM STATEMENT

Numerous systems and platforms nowadays aim for efficiency in order to provide superior service to their clientele. The purpose of ensuring communication integrity and authenticity is to guarantee that individuals or computers can trust the data they receive and that it has not been altered. Integrity requires the use of standard cryptographic techniques. A number of cryptographic techniques are necessary to secure the system. Secret data may be made public in other places. Consider how a solution will be used in a variety of contexts while looking for one. While sensors are necessary for a wide variety of smart card applications, they lack modern security features. As a result, the procedure should be adaptable.

Attempts at system authentication should also take into account the network architecture of the Information device. In the majority of circumstances, distributed networks outperform centralized networks that suffer from bottlenecks and lack of scalability. As a result, we know that the identification center will be detrimental to users. Without the assistance of a third-party authentication provider, identifying other system users should be possible.

1.3 OBJECTIVES

The objective of this dissertation is to educate people on how smart card security affects data security. Here are a few examples of why encryption may be advantageous. Additionally, this course covers the homomorphic encryption (HE) algorithm and its implementations. To attain the following goals:

i. Give an overview of the security risks associated with smart card, as well as a description of the term smart card.
ii. Discuss whether or not classical encryption is an effective countermeasure against threats and how difficult it is to implement.
iii. In this course, you will learn the fundamentals of homomorphic encryption (HE) and then examine several real-world applications of homomorphic encryption (HE).

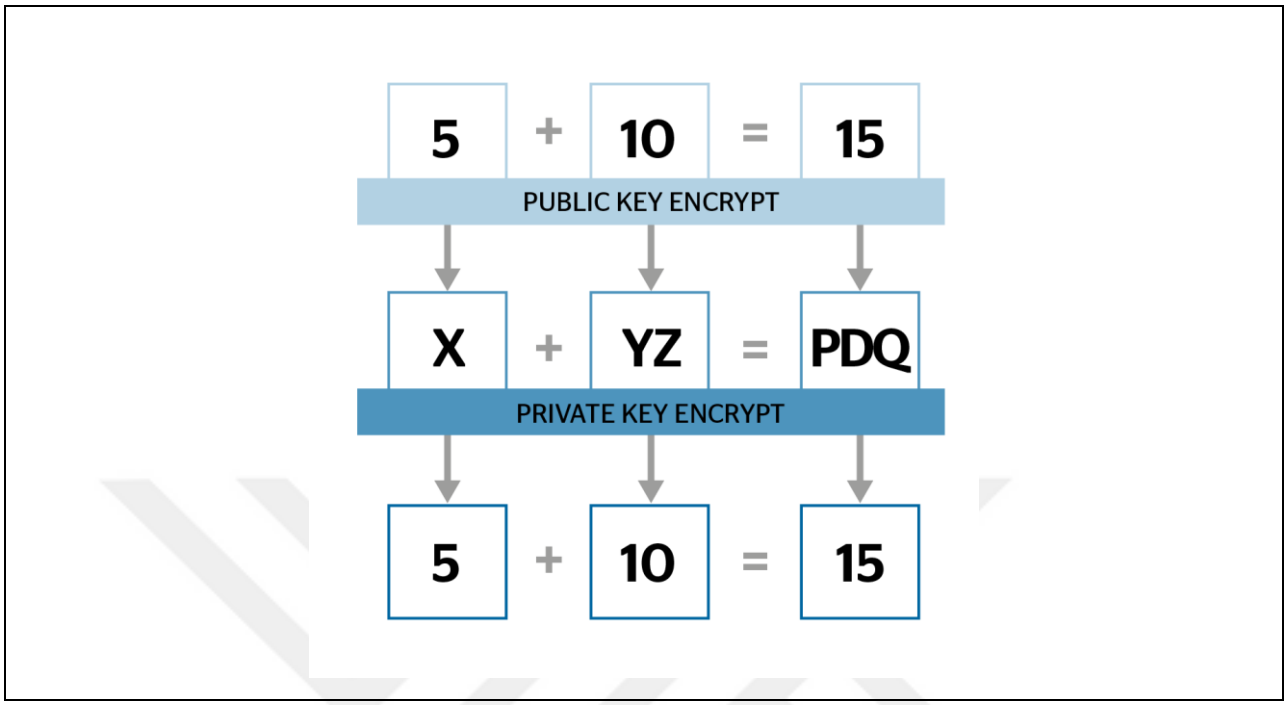

Figure 1.2: HE basic form

1.4 CONTRIBUTION

The researcher's contribution to this study is justified in light of the recognized research gap and outlined research topics. The thesis discusses the relevance, usefulness, applications, and limits of homomorphic encryption. Theoretical evaluations of HE systems such as the DGHV, Paillier, BGV, and FHEW are conducted, as is an introduction to PKI and PKI-based encryption techniques. The Paillier and RSA algorithms (public-key cryptography) are the most important in terms of CPU and memory utilization. By installing and running libraries associated to the HE algorithms we've chosen, we can determine how well they work. Additionally, the RSA public-key encryption library is opened and installed to determine its influence on CPU and memory use throughout the testing procedure. Following that, we will explore future work that will be prompted by our in-depth examination of the problem, as well as new ideas for improving higher education that will result from our discoveries. The researchers seek to attain the following results by responding to the research questions and completing the study's objectives:

i. Understanding Homomorphic Encryption is crucial for any cryptography expert, as is distinguishing it from public and private key encryption.

ii. Practicing with the functionality and implementation of high-level libraries can help you get a better understanding of how they might be used.
iii. Performance and security analysis are utilized to determine the strengths and limitations of four high-efficiency algorithms.
iv. Clinical trials are used to examine and contrast the many HE techniques now accessible.
v. If you have access to operational public-key algorithms, you may examine their influence on CPU and memory use.

1.5 THESIS STRUCTURE

The following is the outline of the thesis: In Section 2, we'll take a look back at some of the smart grid research and implementation that's come before. In Section 3, we provide some context for each of the subparts. In Section 4, we provide a thorough description of our model and its implementation, and in Section 5, we describe our simulations and the data they yield. In Section 6, we sum up our findings and discuss their potential implications.

2. LITERATURE REVIEW

They employed both Rivest Shamir Adelman (RSA) and Cryptosystem (AES) for their ciphering needs (AES), which is a significant departure from earlier methods The implementation of a double-encryption technique ensures the security of data sent between users. The RSA algorithm is used to increase the level of difficulty of the encrypted text, whilst the AES algorithm is utilized to swiftly extract data. When RSA is used for file transmission, the time-consuming process of creating asymmetric keys ensures the system's security. A public key (PK), private key (PB), file identifier (FID), and very large random number (RB) are all generated by the cloud service (RB). To begin, the customer requests a personal assistant from a cloud-based provider. The cloud system supplies the user with the file's PB and ID, which they can subsequently use. The user then uploads the encrypted file to the cloud. When a user wants a file from the cloud service, the user makes a request to the server using the given public key. The cloud service has located and made accessible the requested file in the Cloud Storage System (CSS). The AES algorithm is used to encrypt this data. The symmetric method's secret key, RB, is encrypted using the public key that is used to decode it. When a user requests it, the CSS returns the RB along with the requested file. Symmetric techniques are employed to address the key distribution problem, however how the keys are kept is not described. Cloud service providers will be required to address concerns about data privacy and security as cloud computing technology advances, as a consequence of rising demand. Sachdev et al. [1] encrypted data prior to transmission to the cloud using the AES data security method. AES is a better solution than DES for systems with limited resources, such as 8-bit and 64-bit computers, because to its smaller memory footprint. As long as the key size is more than 128 and a multiple of 32, AES keys are simple to produce and may be utilized without modification with any block of data. Before the user's data is delivered to the cloud service provider for storage, it is encrypted using the AES encryption algorithm. Because the user controls both the data and the key, the data is always valid in any scenario. AES uses less memory and is faster to process than the other existing encryption methods. Due to the fact that this solution requires the user to have a separate physical server, the cost of hardware will rise. By implementing Zhao et alFully .'s Homomorphic Encryption method [2], users would be able to perform calculations on encrypted data without first decrypting it]. The fact that a user may manipulate ciphertext without disclosing the original data increases the security of this technology by hiding the original data.

Only addition and multiplication are done when evaluating completely homomorphic cryptography results. The results of multiplication and addition acquired from the ciphertext must be compared to those obtained from the plain text. Calculating ciphertext is more difficult than computing plain text. As a result, even with a little amount of data, doing such a sophisticated calculation would take an inordinate amount of time. The user's first steps involve logging into the cloud and picking a storage option that matches their desired level of data protection. AES encryption is utilized for private data in the private component, whereas Blowfish encryption is used for public data in the public component. Two data encryption technologies are known for hybrid data storage that may be utilized in either the private or public sectors to assure data security. Numerous encryption techniques, such as the Simplified Advanced Encryption Standard (S-AES), the International Data Encryption Technique (IDEA), and the Blowfish algorithm, may give some amount of security. To provide the maximum degree of security, Blowfish encryption is utilized first, followed by IDEA encryption. After encrypting a file, the Secure Hashing Algorithm (SHA-1) generates an integrity code that may be used to validate the file's integrity. At the beginning of the encrypted file, the SHA-1 algorithm is employed to produce a 16-digit alphanumeric token. Before comparing the SHA-1 token to the one supplied at the time of retrieval, an appropriate authentication method must be undertaken to ensure that the data being retrieved is not damaged. Kaur et al. [3] devised a technique for encrypting user-selectable storage areas. Blowfish encryption is utilized to secure the application's private, AES, and public portions. There are two levels of encryption available in the hybrid area, allowing the user to choose the one that is best suited for their specific circumstances. Although techniques such as IDEA and Blowfish have a lesser degree of security, they are utilized initially, followed by IDEA implementation on the previously encrypted data. When data is securely stored on the cloud, the first step is to retrieve it, followed by a second step including double authentication and integrity verification. This technique is broken into two steps to ensure data security. As a result, this technique ensures the security of both stored and retrieved data. Rani et al. [4] successfully secured users' sensitive data and ciphertext retrieval using public and private key encryption approaches. When you log into the cloud system, the hybrid technique ensures the security of your data. The login information is supplied in plain text as part of this method, but the password information is encrypted using a variety of different algorithms. It is then examined to ensure that the user's credentials match those currently stored in the system. This implies that any anyone who tampers with data may be

identified using the cloud-stored login and password, accelerating the investigation into the root cause of the issue. The Caesar cipher is used to encrypt the password, the username is stored on the server in clear text. The last step is to decrypt the encrypted output using the RSA technique, and then re-encrypt it. The final output is encoded with a system of single-letter replacement. According to maximum security, according to the authors, passwords should be encrypted three times, even though doing so would triple the time required for encryption. Gai et al. [5] suggested a mechanism for workload distribution called Cluster Load Balancing. When implemented in this way, the AES algorithm provides three critical aspects of security: authentication, confidentiality, and integrity. AES encryption technology is used to protect sensitive data. D2ES (Dynamic Data Encryption Strategy) encrypts data on the basis of the privacy scale assigned to it during transmission. The privacy weight specifies that only data with the greatest privacy value will be encrypted, which results in a reduction in encryption time by encrypting only the most sensitive data. After selecting the data package type and its size, the execution time for encrypted and unencrypted data is determined.

3. MATERIALS AND METHODS

3.1 SMART CARD AND JAVA CARDS

INRIA Sophia Antipolis is involved in the OASIS project2, which is developing a Java Card1 application development environment. This thesis describes the environment. Centaur, a tool for creating interactive programming environments, was used to create the Java Card environment. Formal descriptions of language syntax and semantics enable programmers to construct more advanced tools such as structure editors, compilers, and interpreters. [6] A graphical programming environment, for example, may show stacks, heaps, and objects at any stage throughout the interpretation process. Centaur enables the rapid development of high-level abstraction tools. Due to the executable nature of the formal semantics, it can be tested and debugged, which is critical if it is to be used as the basis for proofs. Additionally, access to formal semantics enables the proof of pertinent facts about programs inside the same framework. Security is often the primary priority in smart card applications; as a result, proving capabilities are critical in this environment. Due to the fact that our programming environment has much more complicated components, it is impossible to cover everything in this course.

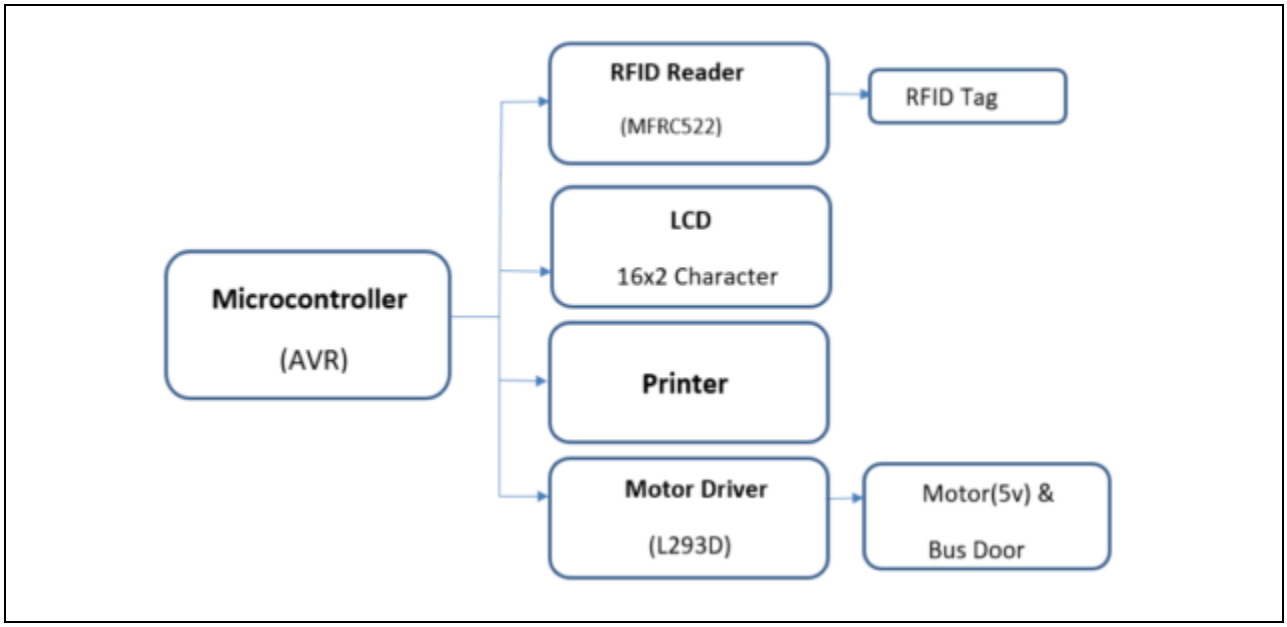

Figure 3.1: INRIA SOPHIA and the oasis project

To maintain the security of sensitive and secret information, it is vital to use security systems that prohibit unauthorized access. A secure element is a tamper-resistant platform used to safely and securely host programs and the sensitive and cryptographic data they contain. Security elements are integrated into everything from computers and autos to TVs, secure SD cards, video game consoles, and mobile phones, as well as linked home appliances and next-generation devices. As smart cards have become increasingly prevalent in daily life, they have risen to become the most extensively utilized method of security. A smart card is a kind of embedded system device that incorporates a microprocessor, memory modules (RAM, flash memory, and EEPROM), serial input/output (I/O) ports, and a data bus. It is completely safe, efficient, and cost-effective. Both the operating system and applications are stored in the EEPROM of the chip. Modern smart cards begin with ROM booting and then go to Flash memory, where the operating system may be continued to be utilized. In this situation, it is stored in the computer's Flash memory. [7] Secure data storage is possible on a smart card, and the card may also offer data security when used in a transaction environment. The availability of hostile attacks on embedded software, which raises significant security concerns, has proved to be a significant impediment to the development of intelligent cards. Business and academics are developing a variety of solutions to protect smart cards against such assaults and to assure the security of transactions. [Bou+11c]. Typically, smart cards have an on-chip memory capacity of 5 kB of RAM, 256 kB of ROM, and 248 bytes of EEPROM. This has a substantial influence on the program's design. The hardware itself is protected by the first layer of security. To evade an internal bus probe, the memory, CPU, and crypto-processors are all merged onto a single chip that also has resin-coated sensors. When a physical assault is detected on the card, such sensors (light sensors, heat sensors, voltage sensors, and so on) are used to temporarily or permanently deactivate the card or its applications. The program provides a second layer of security. It is prohibited to collect or edit sensitive information without first verifying that the activity is permissible. Hackers should be unable to access the smart card's software. It is vital to undertake a comprehensive investigation of all potential attack pathways before installing suitable defences in the form of software.

3.1.1 Smart Card Components

A microprocessor and memory are housed on a credit card-sized plastic card that is often covered under (typically) contact pads. Access to the information contained in this circuit is made simple,

secure, and accurate via the use of smart card-enabled terminals. After establishing themselves in banking and telecommunications, smart cards have expanded into other sectors such as public transportation, healthcare, and retail. When it comes to financial transactions and data storage, the usage of smart cards is accelerating. At the current stage of smart card technology, it is clear that the technology is looking for a suitable application. Dispersion of products is important in order to reach a bigger client base.

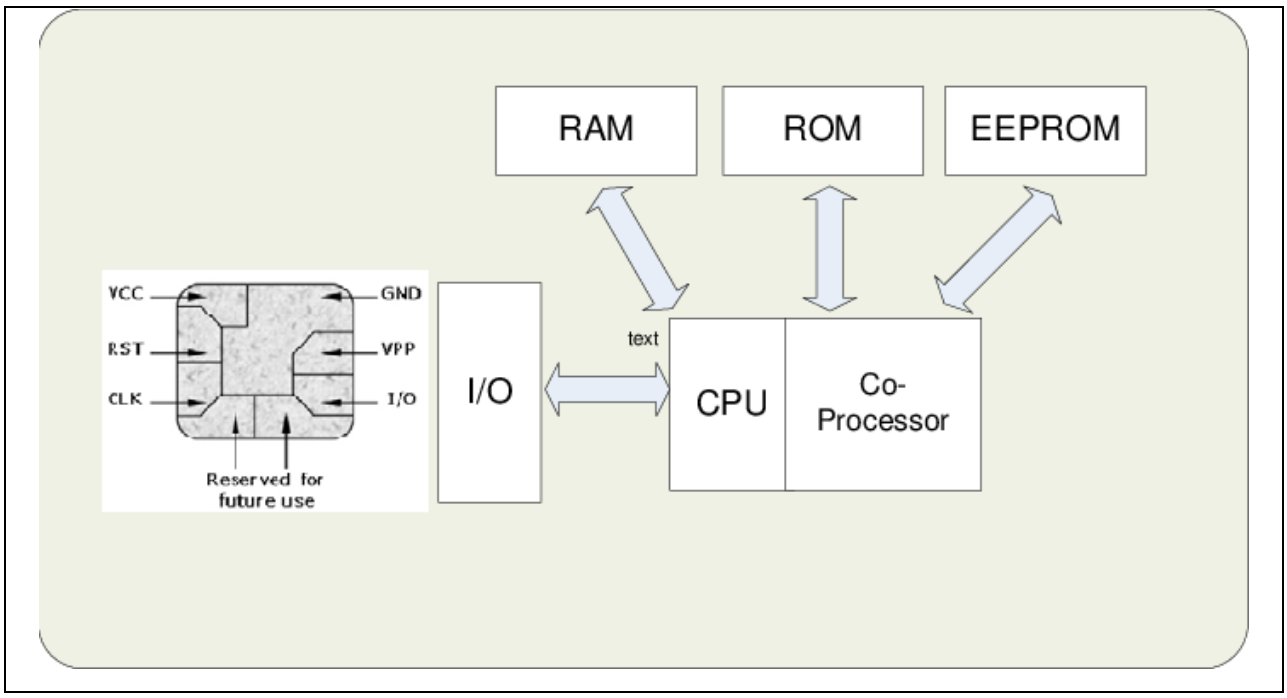

Figure 3.2: Smart card components

Apart from being a multi-purpose smart card, the universal smart card is also known by this name. The word "universal" is used rather than "limited," indicating that the product has a greater range of uses. It is possible to use a universal smart card on any device that supports smart cards and expect it to perform correctly. In the 1970s and 1980s, smart card (intelligent card) systems began to be created, albeit their globalization and potential have not yet been fully realized. While they have gained widespread acceptance and implementation in certain countries, such as France, the technology remains in its infancy in others, such as the United States. Sociotechnical and proprietary obstacles that are causing the system's delayed adoption.

Conclusions from the research are difficult to draw due to the study's broad breadth and the vast number of participating organizations. The breadth of the research was constrained by the small

number of smart card businesses that were able to participate in the case study at the time. As a consequence, the findings lack statistical significance and should be seen as a springboard for further investigation and theory development rather than as a conclusive conclusion. The case study was conducted over a two-year period with a group of Australian small and medium-sized firms (SMEs) active in the smart card market (1996 - 1998). Small and medium-sized businesses (SMEs) were included in the research since they were located in Australia yet functioned independently. This study provides a design framework for the development of smart card applications from the standpoint of small enterprises. Due to its simplicity, this approach to application design may easily be implemented by a small or medium-sized business

3.1.2 Essential Characteristics of Smart Card

Smart cards are standard-sized plastic cards that have integrated circuits to perform a variety of purposes. [8] To conclude, a "smart card" is a piece of plastic that contains an integrated circuit (or a series of them). Smart cards are often referred to as chip cards and integrated circuit (IC) cards. In its simplest form, a smart card is an data-processing and storage integrated circuit. Smart cards are gaining popularity in contemporary culture due to their ease of purchase and usage. As a result of its extensive usage, we've become fairly reliant on it. Numerous new sorts of physical connections have arisen as a result of the introduction of smart cards between a range of agents, including producers and consumers, people and governments, and other organizations. When you create a bank account with a Swedish bank, you will get a bank card equipped with an integrated microprocessor. By using this card, we may bypass lengthy queues at cash registers (kassa) and pay for our purchases at self-service terminals at stores by simply inserting our card and scanning the bar codes for the items we want to buy. Everything that is necessary may be accomplished outside of normal banking hours. In the above scenario, societal acceptability and unchanging demands serve as the impetus for technical growth. In shopping malls, smart cards and customer terminals have been employed to minimize lengthy queues at the pay checkout. Numerous variables contribute to technological advancement, such as raising an application's security, increasing the efficiency of a service, or improving a device's usability. Technology has grown more ingrained in people's everyday lives as it progresses, and this tendency is predicted to continue. Technological improvements have an effect on every aspect of our lives. On the other hand, human-computer interaction (HCI) is becoming an increasingly important component of the

rapid expansion of technological innovation. HCI has grown rapidly in response to this tendency. "Healthcare information technology has effectively become an infinite domain," Rogers continues. Human-computer interaction (HCI) seems to be a large area that spans humans, computers, and human interaction due to its relevance to a varied range of professions.

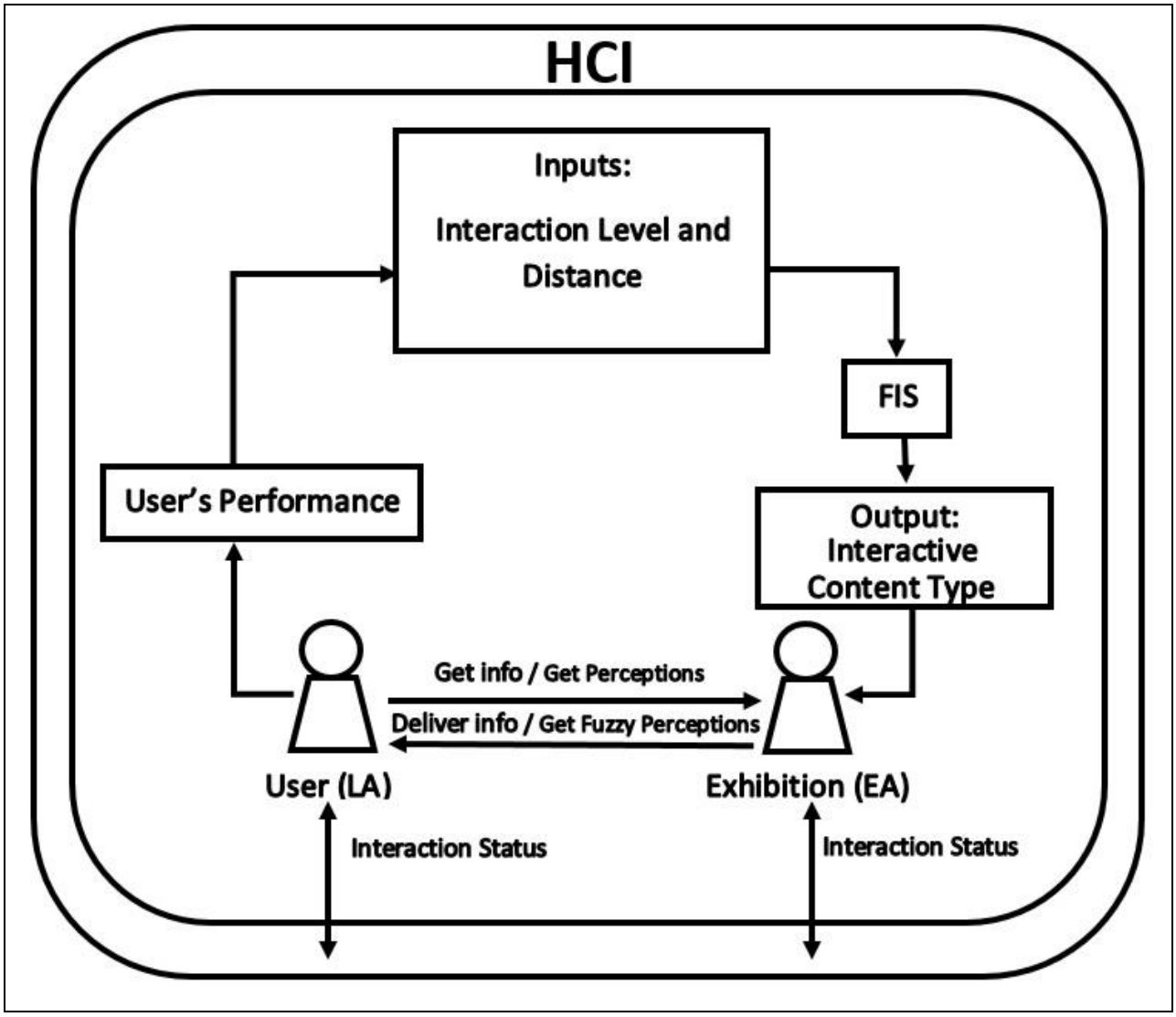

Figure 3.3: Human-computer interaction (HCI)

As computers grow more economical and capable of doing a broader range of functions, we increasingly rely on them in many parts of our lives, including our employment. Human-computer interaction (HCI) is a subfield of ergonomics, which is concerned with the interface between humans and technology. The same is true for smart cards and their ability to assist individuals in accomplishing a range of actions in a number of situations in order to accomplish a variety of

goals. As designers, we understand that design is a deliberate effort to alter the world and ourselves. According to Nelson et al., the design process generates both planned and unintentional consequences. By using human-centred design principles, it is possible to create a smart card that is more helpful to the community. The discipline of design is always concerned with moving and depicting the unknown and unknowable. "In ubicomp contexts, human-computer interaction and interface design are concerned with a big number of computer devices interacting with a large number of other devices," the study continues. Networks of connected systems capable of supporting a diverse set of domain-oriented artifacts are currently being developed in phases. This is also why we need to consider a new HCI's image at the moment. People's needs have changed as a result of technological innovation; they no longer need to go to a market to buy items; instead, they can use their smart cards to make purchases from the comfort of their own homes or on mobile platforms. This feature significantly simplifies and expedites people's lives. With the aid of ubiquitous computing and human-computer interaction (HCI), designers may fulfill user requests by incorporating human-computer interaction (HCI) concepts into information spaces (e.g. iPads, Smartphones). The traditional definitions of "human-computer interaction" (HCI) focus on how humans perceive the capabilities of a specific technology, as well as how machines "know" what other technologies are capable of processing or displaying. To put it another way, the arrival of HCI-enabled embedded technology demonstrates a sea change in the way designers think. They were formerly responsible for developing new technology; however, they are now responsible for integrating several computer technologies into a variety of things in order to serve the public. Numerous technologies and artifacts are being merged to improve results and to stay up with the present degree of technological advancement.

3.2 ENCRYPTION AND SMART CARD SECURITY

There are significant contrasts between secure application development for smart cards and application development for other platforms. Due of their limited computational and memory capabilities, smart cards need power and a clock from a smart card reader[8]. When designing a secure smart card-based application, various security concerns must be made. The detailed description of certain attacks against smart cards and other devices equipped with The next step is a microprocessor with built-in security. Additional considerations must be taken while developing a secure application, but they are more general in nature and are outside the focus of this chapter.

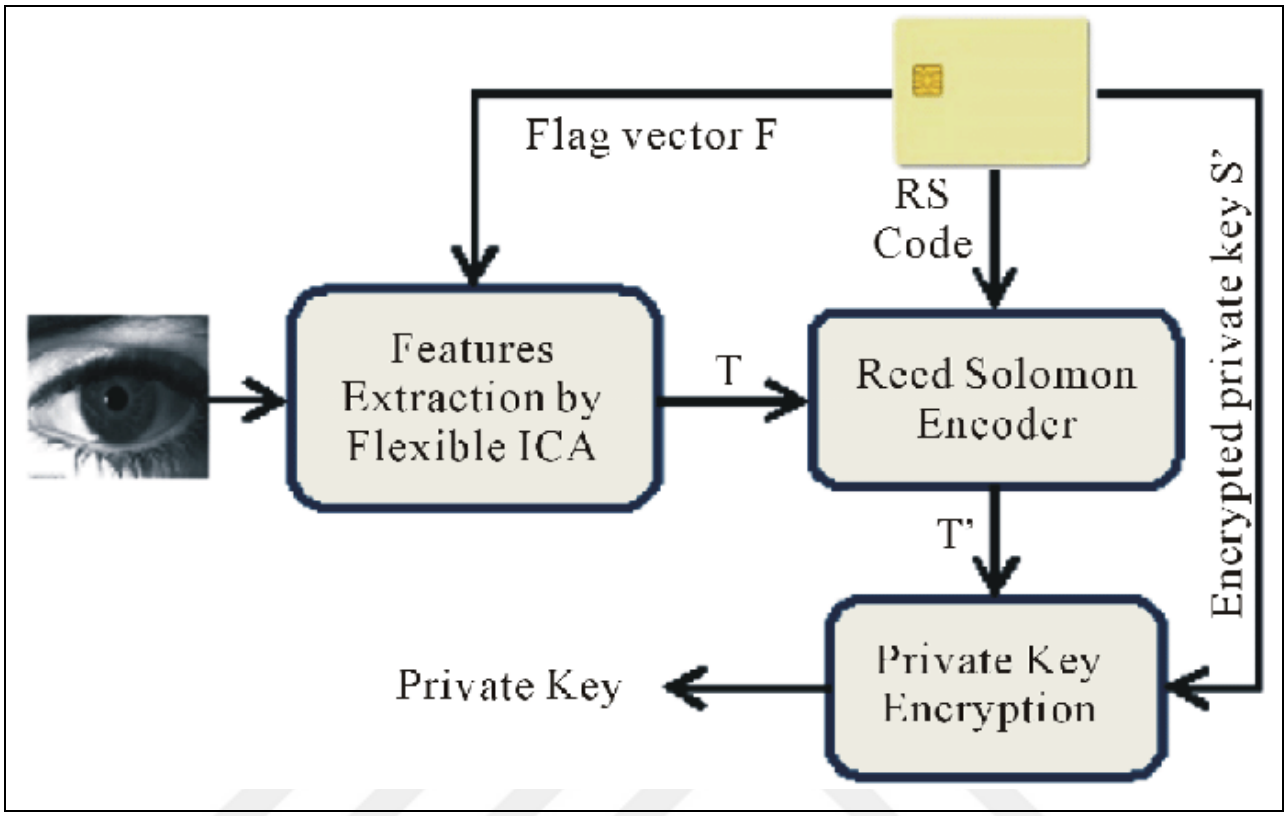

Figure 3.4: Smart card encryption

Security breaches using smart cards may be grouped into three types, each of which is examined. The following are some examples:

i. To be successful, the microprocessor of a smart card must be removed, and the card must be physically abused. At the very least, any secure microprocessor may be vulnerable to such assaults. However, in order to see any impact from these assaults, you will often need to invest in a substantial quantity of pricey equipment and commit significant time to them. As a result, semiconductor makers and students at well-funded colleges bear an outsized share of the burden of invasive assaults. In this example, the probes are put on bus lines that connect the blocks of a chip (a hole in the chip's passivation layer is required for this). This means that an attacker may be able to eavesdrop on data being transferred between blocks. If the intensity of the concentrated ion beam assault is great enough, it may be utilized to damage or produce tracks on the chip's surface. This might be used to reconnect fuses, for example. Despite the fuse, chip designers were able to implement a test mode that enabled them to read and write to all memory areas simultaneously. This mode was disabled as soon as the chip's internal fuse

blew (prior to the chip leaving the manufacturer's facility). This approach is no longer practicable in newer secure microprocessors, since the test circuit is frequently deleted after die separation, making the attack ineffectual. Additional information on intrusive assaults is available.

ii. Semi-invasive assaults need the chip's surface to be exposed. Attackers are now seeking to breach the microprocessor's security without physically damaging it. Electromagnetic emission monitoring using a suitable device, as well as fault introduction utilizing laser and white light, are examples of this kind of attack. [9] The following sections include further information on these assaults. Different sorts of semi-invasive assaults may be classified.

iii. These assaults, usually referred to as non-invasive attacks, make no changes to the secure microprocessor or plastic card under attack. For instance, an attacker may seek to view data that has been leaked during the execution of a given command [10] , or they may attempt to create failures via techniques other than light.' Monitoring the power usage of a CPU and introducing a glitch into the power supply are two examples of this kind of attack. This section delves further into power analysis assaults and fault attacks.

3.3 SMART CARD SECURITY THREATS

Contactless smart cards and other forms of near-field communication (NFC) technology are finding widespread use today. A high degree of security is required for the bulk of these applications. Due to the radio interface, contactless cards are easier to use than magnetic stripe cards; nevertheless, this convenience comes at a cost. Certain apps, such as access control programs designed by non-security specialists, may include weaknesses despite the fact that these technologies are successful. In contrast to software development, [11] secure hardware creation is both costly and time demanding. Security-sensitive systems often use highly secure protocols, some of which have been submitted to formal verification. The chain's weakest link is the software implementation, which is often developed more rapidly than the hardware and is not subjected to rigorous testing prior to release. In terms of overall efficacy, software implementation is just as critical as the other components of the system. Security issues may occur as a consequence of improper protocol implementation, which necessitates the usage of secure hardware in an acceptable manner, which is very difficult. It is essential for the specialists who create these systems to understand the benefits and drawbacks of each kind of card in order for the system to

work properly. Contactless communication systems may benefit from an automated method to identifying potential security flaws. [10] Certain security incidents may be undetected if security processes are designed and reviewed informally. When combined with protocol fault detection, formal verification methods provide a systematic approach to identifying protocol flaws. The research is motivated by the growing adoption of new contactless technologies, as well as the creation of a diverse range of applications by non-security professionals. Both attackers and developers are pursuing vulnerabilities in contactless technology systems because to the widespread use of contactless technology worldwide and the potential for large financial benefit from compromising such a system.

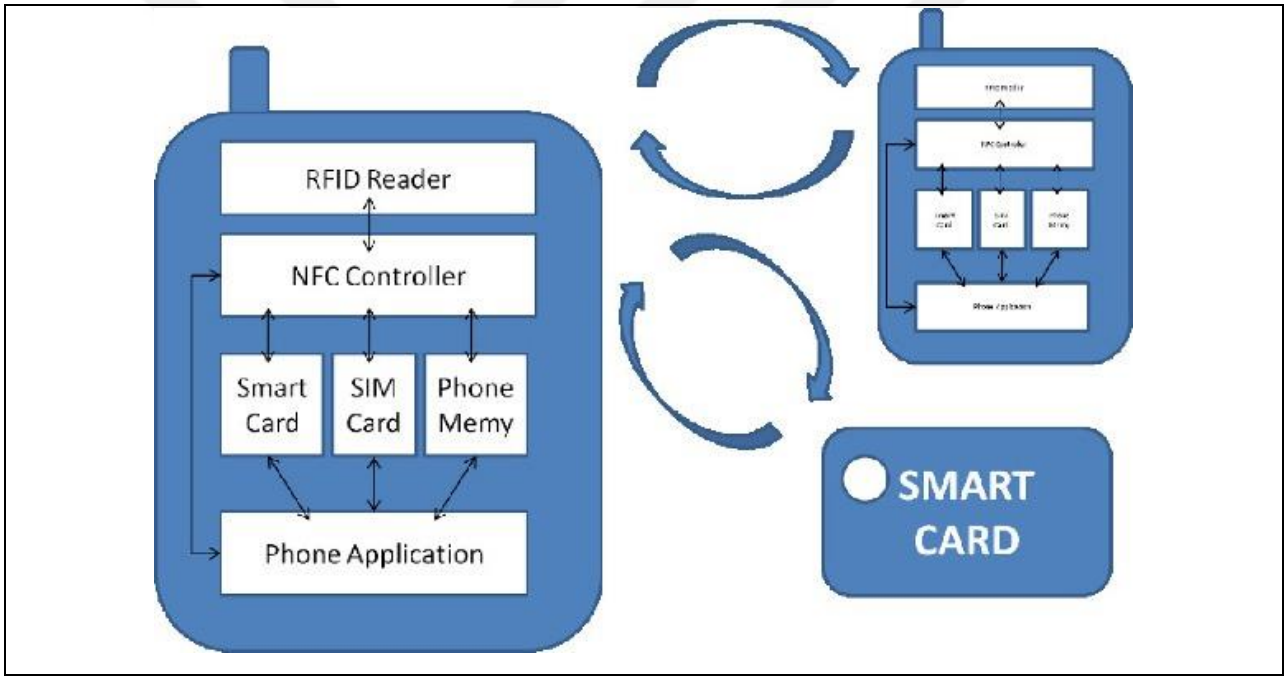

Figure 3.5: Near-field communication (NFC) in smart card

3.3.1 Insecure Application Programming Interfaces

According to a statement made by Facebook in September 2018, a vulnerability in the application programming interface (API) code for the View As function resulted in the compromise of the data of fifty (50 million) users. Cambridge Analytica obtained information on around 80 million Facebook users between 2013 and 2015 by exploiting a security flaw in Facebook's application programming interface (API). In 2018, an unauthenticated API endpoint exposed the data of around 37 million Panera Bread customers, and an unprotected Venmo API, a mobile payment

service owned by PayPal, exposed hundreds of millions of transactions. Over 143 million individuals were impacted by the most serious kind of data breach, which happened as a result of a flaw in the architecture used to build an API for credit reporting business Equifax. Numerous firms' security personnel were utterly oblivious of breaches that occurred months, if not years, prior to their discovery. According to a data security study, including security measures lengthens the time required to use APIs. Additionally, the poll indicated that three-quarters of security professionals failed to build adequate security mechanisms for access to the APIs being developed. New application programming interfaces (APIs) are generated and delivered on a yearly basis. According to predictions, the cloud API market will generate more than $1.7 billion in revenue on its own by 2026. There are an average of 26.7 vulnerabilities per API among the 363 APIs utilized by a medium-sized firm. Despite developments in network security, such as the adoption of zero trust security and the usage of DevOps principles, API vulnerabilities exist. Despite the fact that APIs continue to utilize unsafe coding methods, APIs are built for speed and utility, rather than layering on security as code is written. As a consequence, hackers may be able to access the cryptographic keys of a system through a backdoor. According to the firm, hackers who get access to the encrypted digital source code management system GitHub often attack Amazon's online services and other networks. Because system developers often provide configuration files, user data, and corporate assets in their source code, hackers may easily obtain and abuse these credentials. In the Panera Bread data breach, the API was not authenticated, allowing hostile parties to get access to the code and extract the last four digits of credit card information that was saved in plain text, enabling them to conduct fraud.

Among other measures, traditional methods of network security include attack prevention via the use of perimeters, trusted users, and firewalls. Historically, firewalls have been delivered pre-configured with a set of rules governing their usage and maintenance. Throughout the prior decade, hackers targeted a business's network infrastructure in order to steal critical data. Client PCs and mobile devices may now access the company's external cloud servers, which are gaining popularity as more firms migrate away from internal servers and toward external cloud servers. Because hackers no longer have access to classic attack strategies, they have developed more complicated approaches that are more broadly distributed and destructive. Despite major advancements in technology over the last two decades, workers and authorized users may access company data through the web or mobile apps from a wide range of electronic gadgets, including personal

computers (desktop and laptop), tablets, and mobile phones. Due to the growth of online applications and application programming interfaces (APIs) that enable users to access data through a mix of cloud applications and local apps, it has become simpler for hackers to perform email phishing attacks against these web applications and APIs. Despite the fact that no data packets are originally permitted to enter the network, data breaches do occur in zero trust situations. API flaws are to blame for these data leaks. Due to the enormous number of systems in need of upgrades, it's challenging to deliver them quickly. Additionally, APIs for non-visible-to-the-user apps rely on third-party backend code, which makes them vulnerable to attack if the code is hacked.

3.3.2 Malicious Insiders

Over 80% of businesses now utilize cloud-based services, and the majority of governments have begun to centralize their citizens' data in massive data centres, while individuals are increasingly using the cloud to archive their personal data. These factors all have a role in the ease with which data theft happens. Numerous external cyber-attacks, including unauthorized network access and denial of service (DoS) attacks, as well as computer viruses, Trojan Horses, and Worms, preoccupy the minds of the majority of government and business decision-makers. Their IT budget is devoted to safeguarding these networks against external attacks, accounting for 10% of overall spending. Intruders pose immense hazards both inside and outside, but the harm produced by an intruder is far greater than the damage caused by an intruder. Individuals who have gotten access to an organization's data pose a much greater threat to security than any other kind of threat. Despite this, the risk represented by insiders continues to be undervalued and disregarded. Insider attacks are the costliest kind of information security breach, costing an average of £115,000, according to a recent Ponemon Institute LLC analysis. To put it another way, the insider has direct access to information about and access to their organization's financial resources. As a result of this individual's confidence in the organization, the organization has provided him or her approved access, allowing him or her to circumvent any physical or technological security obstacles. However, the frequency of events involving insider threats has increased significantly. According to a recent Ponemon Institute study, 88% of information technology experts anticipate that insider risk would either remain constant or rise over the next two years.

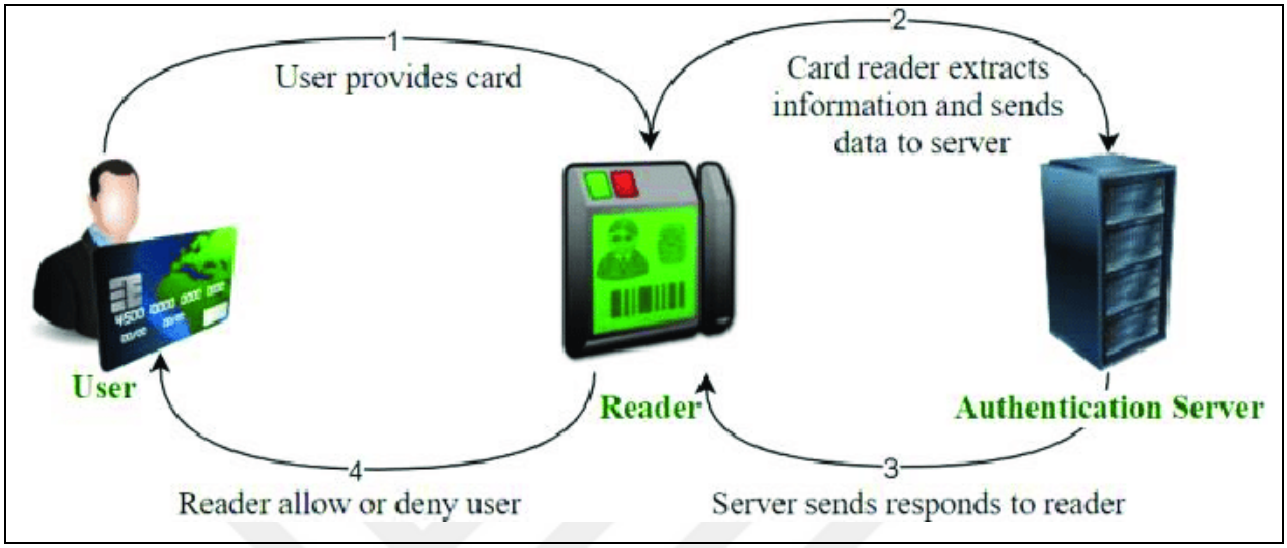

Figure 3.6: malicious insiders in smart card

The vast majority of these occurrences go unreported and unpunished, either because to a lack of evidence to prosecute the insider or because the organization is worried about its reputation and the potential for unwanted publicity if legal action is taken against the insider. Demonstrates how to deal with internal dangers. Insider threat issues have been intensively studied over the past decade in an effort to find a single answer to the present issue of security data breaches. Security research has shown a rise in insider threat breaches over the last three years, with a large event impact on businesses as a consequence of insiders' activity. This pattern is predicted to persist in the next years. Despite significant advancements in technological protections, academics have put a heavy focus on the non-technical components of insider threat avoidance in their studies. Because they have access to sensitive information, privilege users are the weakest link in an organization's security. Due to their mastery of technical controls and processes, this group of individuals is able to overcome current technological restrictions. This is why the issue statement for this project is to develop a novel strategy for assisting businesses in mitigating the risk of insider threat by using a variety of diverse strategies.

3.3.3 Shared Technology Issues

As a consequence, the current generation of youngsters is being raised in a technologically advanced culture. Children's attention is piqued by a rising number of social media apps and

websites, and the universe of these resources is always expanding. Mobile phones and laptop computers, for example, [12] are omnipresent in today's culture, and it's easy to see why. A generation of children is being raised in an atmosphere that views technology as a means to an end rather than an end in and of itself. Numerous students devote hours of their days to popular games such as Facebook, MySpace, World of Warcraft, and Sim City..... According to data, students are more likely to use computers to study in the classroom these days. Teachers are making major efforts to keep up with rapid technological advancements in order to better integrate their students' interests and academic learning in the classroom. Today's teachers under enormous pressure to provide the best educational experience possible to pupils in the twenty-first century. As part of these expectations, educators must ensure that pupils have the technical and informational abilities necessary to thrive in today's rapidly growing and digitally driven world. Educators are always on the lookout for new digital tools that will enable them to serve their students more effectively. Despite overwhelming praise for its ability to enhance student learning, technology usage in the classroom has also been challenged as a potentially harmful tool. Education is becoming more accessible to all people as a result of contemporary information technologies. Each learner, however, need a support structure in order to accomplish their lifelong learning goals. The success of students is contingent upon the engagement of parents, friends, and supervisors who are also educators and coaches. Proponents of school-to-work transitions arrange support and understanding to ensure that students learn the knowledge, skills, attitudes, and habits necessary for success in all aspects of their lives. Technology integration in the classroom has a beneficial effect on kids' lives. As advantageous as integrating technology into the classroom has been, it is not without problems. It is conceivable that technology use will increase learning by inspiring pupils to be more engaged and driven. To ensure the success of construction and engagement, learners must be provided the chance to make choices. They must also be exposed to learning that occurs in meaningful contexts in which they participate actively." Is it enough to offset the disadvantages? Researchers observed that computer usage was connected with changes in basic psychomotor and cognitive abilities in a study that examined the effect on motor skills. This category includes computers, electronic organizers, navigation systems, and other related gadgets. This may have a detrimental effect on pupils' academic success. Along with demonstrating the benefits and disadvantages of technology, future study should include a thorough examination of the causes that lead to those benefits and drawbacks. We surveyed K-12 instructors to ascertain

their thoughts on the advantages and cons of incorporating technology into education and its effect on student learning. [13] The findings were summarized in a report. Apart from the internet and computer-mediated tools, no other kind of technology was considered for this research. As a teacher, you must be aware of how technology affects student learning since it has the ability to aid or hinder students' growth. The researchers performed this study to provide instructors with knowledge on the most effective teaching approaches. Instruction geared on supporting children in achieving academic achievement may be beneficial to such students. Parents of kids may be able to have a better understanding of how technology affects their children's academic progress by using it. The outcomes of this study will add to the body of information already available on the topic of technology integration in the classroom.

3.3.4 Data Loss or Leakage

In general, "data loss" refers to the disappearance of data from any kind of storage device. This is an issue that affects everyone who regularly uses a computer. When data is physically or logically removed from an organization, whether purposefully or inadvertently, and is not retrieved, data loss occurs. Data loss has established itself as a key issue in today's business world, one that must be addressed by the enterprises that create the data. [14] Data leakage occurs when the privacy and security of sensitive data are jeopardized. The term "data exfiltration" refers to the unlawful transportation of information from inside an organization to a remote location. In contrast to a breach of confidentiality caused by a data leak, data loss occurs when a file is accidentally deleted, or a computer system malfunctions. When it comes to data breaches, both terms may be used to characterize the scenario, which has emerged as one of the most serious issues confronting organizations in recent years. DLP is a computer security word that refers to the process of identifying, monitoring, and protecting data when it is in use, in motion, or at rest. Taking Countermeasures Against Data Loss/Leakage (DLP) To detect sensitive data included in DLP-compliant files, network communications and deep content analysis are combined. DLP is primarily intended to safeguard information assets while interfering as little as possible with a corporation's or organization's operations. To prevent undesirable events from occurring, protections have been implemented.

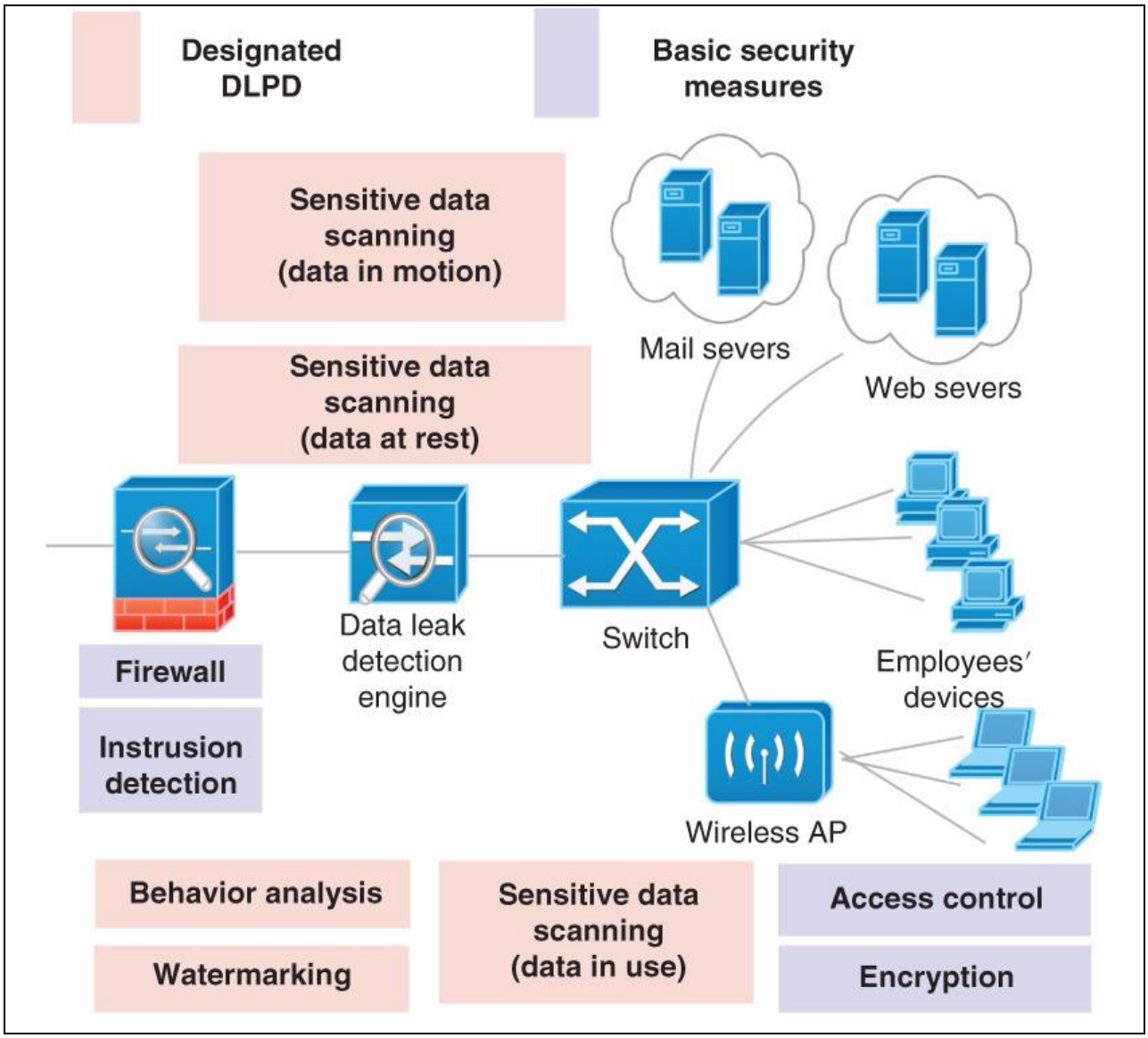

Figure 3.7: Countermeasures Against Data Loss/Leakage (DLP)

Data loss prevention (DLP) may also be used to enhance data management processes and, in certain situations, to save compliance costs. Protecting sensitive information is feasible via the implementation of systems capable of detecting and preventing illegal access.[15] The phrase "data leak prevention" is used interchangeably with data leak detection, Eliminating Data Loss, Detecting and Preventing Data Loss, content monitoring, content filtering, information protection and control, and extrusion prevention system. According to the study's conclusions, data loss and leakage may be avoided by the deployment of DLP technology, which addresses both "data loss"

and "data leaking." We will examine DLP technology in the context of a huge organization in this case study.

What is the DLP program's objective?

i. The necessity to adhere to PCI (Payment Card Industry) security standards while processing credit cards is an example of a necessary compliance and regulatory requirement.
ii. Follow these principles to keep sensitive information about consumers, businesses, and employees out of the wrong hands.
iii. This capability is global in scope due to central configuration and enforcement.

DLP may be utilized in a variety of ways, including the following:

i. There are several classifications for sensitive assets.
ii. Sensitive asset audits are conducted.
iii. Audits of the identity and access management systems used by the company
iv. Encrypting your assets is an excellent approach to safeguard them.
v. Granting EDRM (Enterprise Digital Rights Management) authorization to critical assets is a complicated procedure.

3.4 ACCOUNT OR SERVICE HIJACKING

Cloud computing has altered the landscape of information technology, as well as the way companies and government agency's function. It fundamentally alters the game. [16] Cloud computing has been a major success because to its scalability and advantages such as coherence, high availability, and economies of scale. On the other hand, as new threats and vulnerabilities have emerged in line with technical advancements, the cloud computing environment has developed into a battlefield for more complex security concerns. These attacks cast doubt on the effectiveness of security measures in place to protect the information technology infrastructure. When it comes to preserving IT infrastructure, it's critical to remember that the most precious asset is information. Almost 60% of firms worldwide have begun to use cloud computing services, and the pace of cloud migration in developed nations is accelerating. For example, in New Zealand, the number of small and medium-sized businesses that have migrated to the cloud has increased by about 175% in the previous year. On the other side, migrating data from a centrally hosted

server to the cloud increases the possibility of data loss due to network failure. Prior to migrating IT infrastructure to the cloud, it is necessary to undertake a thorough risk assessment and adopt mitigation measures. To ensure that cloud computing's promises and threats are realized to their full potential, the most modern and effective security protocols must be in place. When it comes to defending against the most unique and intricate attacks, the efficiency of a security solution is decided by how closely it conforms to best practices. Cloud computing's proclivity for missing or neglecting critical IT information is the most serious security issue related with cloud computing. Even though cloud technologies provide low-cost and rapid services, their utilization may compromise the security of IT operations. Cloud computing may provide several benefits to enterprises, but only if properly deployed and enforced. Effective security standards must be devised and adhered to if firms are to reap the advantages of cloud computing while simultaneously safeguarding their intellectual property. Small and medium-sized organizations (SMEs), in particular, have been reluctant to understand the critical nature of cyber hazards for a number of reasons. This is due to their failure to accurately detect and quantify cyber-attack-related damages. As a result of recent cybercrime and cyberterrorism incidents, every business today has a cybersecurity strategy in place and is on the lookout for more cost-effective cyber security solutions. Open source and freeware solutions have entered a sector historically controlled by proprietary software from huge firms like as IBM, Cisco, Norton, and others, ushering in a new era in the growth of cyber security software. When it comes to risk management in a cloud computing environment, the most crucial component is an awareness of the nature of security threats. Following current events and gathering expert insights on vulnerabilities and threats is critical for making sound risk management choices about cloud adoption strategies. Keeping up on current events and following expert opinions on vulnerabilities and threats is crucial for risk management. [17] It is critical to create and execute well-structured security measures in order to combat cyber-attacks and assist the company in quickly resuming regular operations. The New York Times website was taken down for almost six hours after an account hijacking assault, making it one of the most publicized security breaches in history. We've created this post to provide our findings from our incident investigation into the New York Times outage (NYT). To further avoid similar occurrences in the future, we provide our customers and partners with incident prevention tools and an integrated incident prevention methodology.

3.5 UNKNOWN RISK PROFILE

Given the breadth of threats, it is critical to consider risk and resilience while designing a strategy for system security. Paradigms may be considered of as intellectual frameworks or modes of thought in this way, and they can be classed accordingly. Reduce the possibility and size of future losses is a frequent goal in the risk paradigm. The resilience paradigm is concerned with enhancing a system's capacity to absorb, adapt to, and recover from shocks while maintaining critical functions and capabilities. The purpose of this research is to examine how these two paradigms may be applied to risks that are unknown, unquantifiable, systemic, or improbable but catastrophic in character. Numerous studies have shown that gaining resilience is preferable than taking a risk on anything when presented with such challenges. These publications provide light on risk and resilience research and demonstrate how resilience research has benefited the risk analysis community as a consequence of its results. When it comes to unknown, unquantifiable, systemic, and improbable/catastrophic risks, the study may have been too eager to highlight resilience over risk. Both paradigms, according to the researchers, are well-suited to coping with such threats. While paradigms may be incapable of fully resolving these challenges, this does not always mean that the paradigms are incorrect.

Risk and resilience may be defined in several ways. Kaplan and Garrick (1981) describe risk as the triple threat of threats, their chance of occurrence, and the size of their repercussions if they do occur. Risk management requires identifying and analyzing hazards, as well as calculating their likelihood and size. It is also critical to develop ways for mitigating all of these issues. Along with the probability of loss, the risk paradigm may take the possibility of gain into account, albeit this is less usual. According to a widely accepted definition of resilience, "the capacity to anticipate and prepare for, absorb and recover from, and effectively adjust to tough circumstances" is an acquired feature. Maintaining the integrity of vital system activities is important, even if this requires modifying other features to changing conditions brought about by threats. This is the resilience paradigm's ultimate goal. It has been stated that when four characteristics are present, the risk paradigm is inappropriate, and that the resilience paradigm is more suitable:

i. At the moment, there are no recognized threats. While doing risk analysis without hazard identification is challenging, resilience strategies need a more adaptable approach to planning for the unpredictable (emphasis original).

ii. There are no straightforward formulae for determining the probability and size of a risk. For example, a risk-based approach highlights the need of being aware of the potential of unforeseen damage even when dangers are detected and avoided (emphasis original). Risk encompasses a greater range than resilience, [18] which is crucial when risk cannot be defined.

iii. External threats are described as systemic when they have repercussions that influence more than one region of the system or when they have ramifications that affect other connected systems. Resilience engineering, by focusing on system capabilities rather than individual components, discovers critical system features that benefit stakeholders and society as a whole. Consider the following:

iv. Unlikely or catastrophic threats to our safety. Alternatively, you may claim that "conventional risk analysis methodologies have been shown to be inadequate in a number of known occurrences with a low probability of occurrence but catastrophic repercussions." In terms of risk management, resilience is defined as an effort to "minimize the consequences of failure, but with more frequent failures and shorter recovery periods," whereas risk management is defined as an effort to "minimize the likelihood of failure, albeit with rare catastrophic outcomes and lengthy recovery periods."

3.6 CRYPTOGRAPHY IN THE SMART CARD

As technology progresses, the complexity of the programs that operate on it increases proportionately. Security is the process of safeguarding valuable items from theft, loss, or change. The term "data security" has an influence on everyone's daily activities since it encompasses a wide range of applications and affects everyone Concerns about data security have increased as technology has rapidly expanded into virtually every transaction, from parking meters to national defines. Three types of authentications are employed in the field of security:

i. a piece of personal data, such as a password or pin

ii. any of several portable electronic devices (like a SecureID card).

iii. a unique biometric identifier. A plastic card, sometimes referred to as a smart card, is a kind of chip card.

includes a computer chip for the purpose of storing and transmitting data. The card's microprocessor stores and analyses this data, which may be associated with either value or

material. [19] Although France developed smart card technology first, consumers were hesitant to adopt it due to a lack of technical infrastructure and a high cost. Users' personal information could only be saved to a limited degree on these cards. The advancement of smart card technology is contingent upon users' trust that their personal information will not be sold or shared with others. To do this, formal techniques for specifying and verifying smart card applications may be utilized. Initially, smart cards included just protections that prevented them from being refilled. Today's smart cards may be reused and carry a large amount of data due to their high transaction rates. These smart cards have microprocessors that enable them to perform on-card operations. It is capable of recognizing, authenticating, storing, and processing applications. Because most smart cards are constructed utilizing cryptographic techniques, they are very important in the commercial security market. The ability of smart cards to conduct cryptographic algorithms inside their own circuitry provides them a substantial edge over magnetic stripe cards. Consequently, the user's personal information (such as PIN codes or keys) is completely contained inside the bounds of the tamper-resistant silicon chip, providing the system's greatest degree of security. Smart cards are now utilized in a range of sectors, including telecommunications, banking (for credit and debit cards), healthcare, audio-visual, identity management, transportation, and access control. Cryptographic computations are enabled via secret keys included in the non-volatile memory of these cards. These secret keys are often taken from tamper-resistant cards during smart card attacks in order to modify the card's contents and complete an authorized transaction. In these instances, it would be desirable to use an authentication technique. When a PIN is put into a smart card that has been approved, it can only work. After smart card authentication, a time-stamped certificate is signed that transferred specific authority from the user to another party. The number of password authentication techniques continues to rise at a breakneck pace. Because they are used on distant networks, remote password authentication systems are also password authentication schemes. Due to the distance between login points and a remote password authentication system, communications between them are insecure. Password guessing attacks come in two flavors: online and offline. The attacker uses a succession of guessed passwords to circumvent the server's verification procedure in an online password guessing attack. Offline password attacks perform iterative guesses of a password by intercepting password-related communications between a user and a server. The attacker then verifies if his estimate was correct. When these problems are at risk, it is critical to prioritize security and privacy. To be adaptable, an organization must be able

to adjust rapidly and unobtrusively to changing situations. File access, dangerous applets, communication protocols, and weak encrypted protocols are all vulnerable to attacks thanks to hidden commands, parameter poisoning, and buffer overflows. are just a few of the logical flaws that may be exploited in smart cards. Side-Channel attacks on smart cards are defined as assaults that monitor a variety of metrics, including power consumption, electromagnetic radiation, time, and voltage, in order to ascertain the behaviour and features of smart card transactions. This signifies that the fundamental functionality of the Cryptographic Smart Card security mechanism has been judged obsolete. The above evidence demonstrated unequivocally that any smart card security mechanism may be broken. However, there is a general estimate of the cost of breaking into the system, that needs to be significantly less than the value of the information being secured by the system. Two-Factor Authentication requires the user to submit both a physical device (a smart card) and a PIN (which may represent a secret key or password) (a PIN). [20] As a result of needing both a physical and a non-physical element, user authentication is much more secure. Single-factor authentication (i.e., using something you know, such as a password or personal identification number) is less secure than two-factor authentication, since it depends on usernames and passwords, both of which are easily accessible (PIN). Two-factor authentication makes it more difficult for fraudsters to steal a user's online identity. To utilize two-factor authentication, the user must both know and own something. Most two-factor authentication solutions should include a registration, login, and verification phase. The user and server exchange a secret during the registration procedure, which is stored on the user's smartcard.

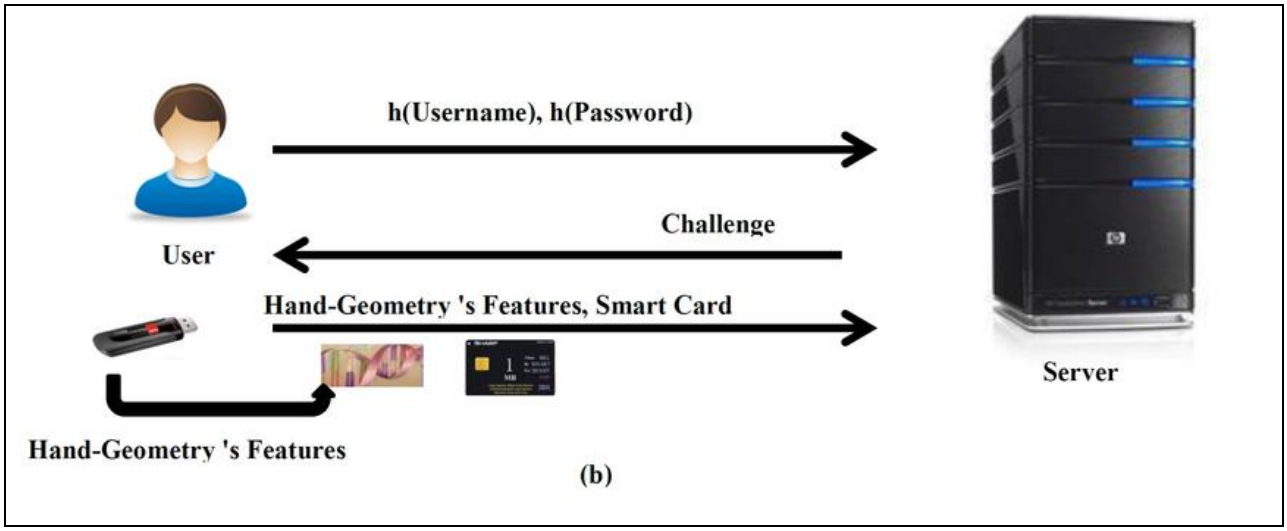

Figure 3.8: PIN hijacking in a smart card

3.7 ENCRYPTION AS A THREAT COUNTERMEASURE IN SMART CARDS

ICCs, which are cryptographic hardware with an integrated circuit that may be used for data storage and administration, provide a hermetic environment for these functions. Due to their physical properties, the hardware and data it stores are safeguarded from unauthorized use and cannot be tampered with. With smart cards, it is possible to securely store, change, and preserve critical information. They are used by a wide range of software programs because they are a tried-and-true means of providing security protection. While Jurgen Dethloff and Helmut Grotrupp received the first patent for an integrated circuit card in 1968, Roland Moreno, dubbed the "Father of the Microchip," invented the first smart card in 1974. [21] Since their inception in the early 1990s, smart cards have expanded in popularity and applicability across a broad range of industries. Turkish drivers' licenses were provided with smart cards for the first time in 1987, while Denmark released the first electronic prepaid card in 1992. In 1994, the consortium of Europay, MasterCard, and Visa (often referred to as EMV) published the first official specification for bank-issued smart cards, opening the path for broad deployment. As a consequence, smart-card design and capabilities have branched and developed through time to provide low-cost solutions for intricate scenarios including both single- and multi-application smartcards, resulting in the creation of low-cost solutions for complex circumstances. As a result, smart-card applications cover a broad range of functions, from basic user identification to data security and financial processing. It is used in a wide number of commercial applications, including identifying badges, telecommunications, access control systems, and banking. Indeed, they've grown so ubiquitous in our lives that they've been elevated to the status of a commodity in their own right. This list includes a variety of applications that all use smart cards in some form. These applications involve the creation, storage, and processing of data, as well as the execution of fundamental operations on sensitive data. Smart cards are utilized in a wide variety of applications, from simple commercial transactions to highly personal and user-centric transactions. Smart cards have already established themselves as the de facto industry standard for ensured security and will do so in the future. Similarly, to CERN, the European Organization for Nuclear Research, users of Facebook's social networking site will be able to validate their identity using smart cards when logging in. Smart cards continue to be used largely for data access and user authentication, and this trend is anticipated to continue. The processing of data has become more dependent on the use of smart cards, which are capable of executing very sensitive tasks. Because smart cards not only store and

provide access to static data, but also have the potential to alter it without being detected, they are the only method to implement protocols with a higher level of security. [22] To expand the use and versatility of smart cards, it is necessary to define a set of common procedures and security criteria that all smart cards must comply to. As a consequence, a platform is created that enables any developer to produce applications without having to construct their own smart cards, thereby establishing an industry standard. Due to the fact that all programs make use of the same smart-card interface, advancements in that technology may have an effect on them all simultaneously. Establishing a baseline is a difficult task, given the unique features of smart cards, including their limited memory and computational capabilities and their dependency on an external device (the smart-card reader) for functioning. Each application that utilizes smart cards must account for card and reader to be able to be accurately described and used properly. Maintaining secrecy has always been a significant issue when working with sensitive data. Throughout the development and deployment of sensitive data management systems, computational security has grown as a disciplined strategy that enables the testing and validation of a wide number of procedures. The phrase "computational security" encompasses work on threat modelling, reasoning, and standards. Despite the fact that it is publicly available, the "security by obscurity" strategy, in which commercial manufacturers offer security by disregarding standard adherence and concealing the underlying implementation details, is widespread practice. To conduct this kind of assault on the implementation, the attacker would need to dedicate considerable time to reverse engineering and analysing it, among other things. Although security experts and academics have often criticized this strategy, they argue that it does not offer participants with a safe environment.

3.8 HOMOMORPHIC ENCRYPTION

The purpose of this thesis is to use homomorphic encoding to protect text documents in order to develop a cloud-based text search engine. Cloud computing is a way of offering conventional computing services such as servers, storage, databases, networking, and software, as well as, more recently, artificial intelligence (AI) through the Internet, in distant data centres, via the Internet and cloud computing. Because cloud services are available to anybody and are billed on a per-use basis, they may be a less expensive option for individuals and businesses of all sizes, depending on their needs. By using homomorphic encryption, a ciphering approach, any data may be encrypted and decoded. When Homomorphic Encryption is used, third parties (such as cloud

service providers) may instantaneously apply the functionality to encrypted data without disclosing any personal or commercial information. As with other public encryption systems, the Homomorphic encryption system encrypts data using a public key, similar to how other public encryption systems function. On the other hand, only individuals who hold the appropriate private key are able to decrypt the encrypted data. That being said, what sets this encryption apart from others is its support for algebraic operations and the ability of users to perform a range of mathematical operations on the encrypted data. Cloud storage provides several benefits to cloud customers, including cheap pricing, high scalability, and easy-to-manage platforms. While it is true that the usage of third-party cloud providers, particularly low-cost ones, has revolutionized the way individuals and organizations store data, it has also created concerns among certain cloud users, prompting some corporations to build their own private clouds. This, however, makes no difference to the problem of data security.

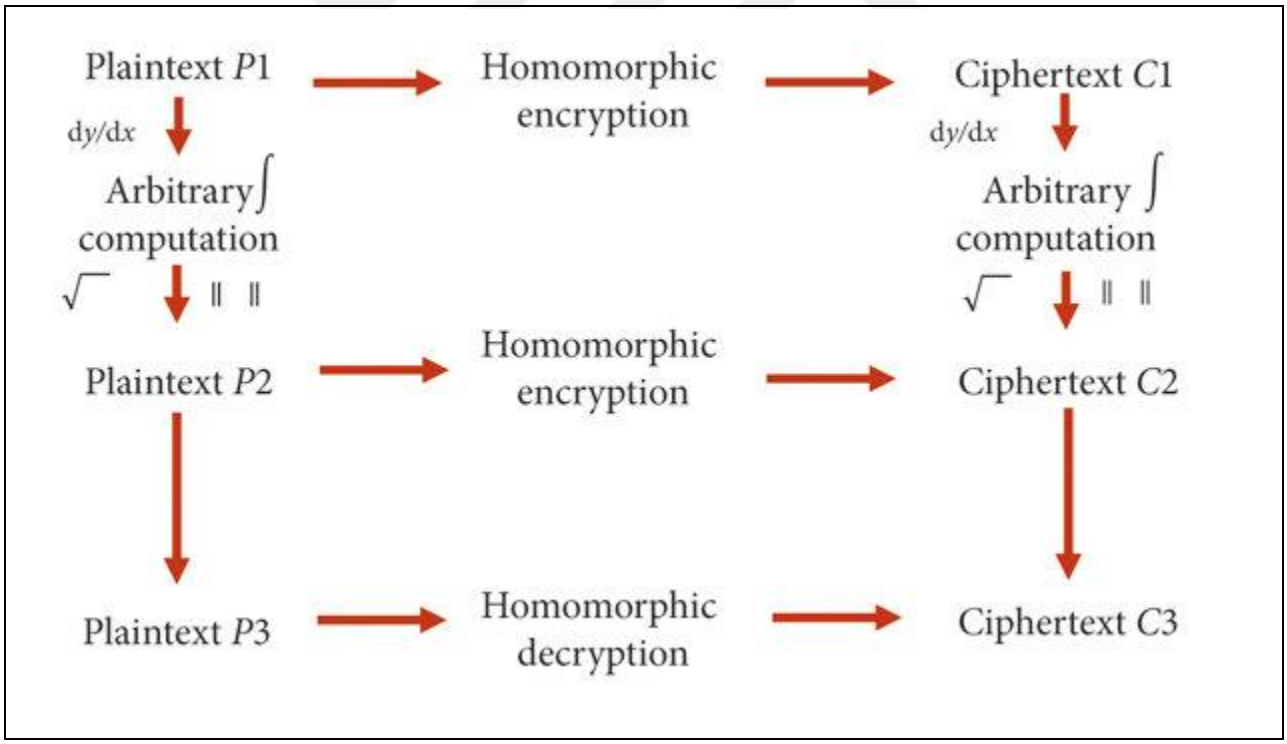

Figure 3.9: Homomorphic encryption in smart card

As a result of these worries, many organizations are forced to rely on a tried-and-true technique of data storage that has existed for decades: encrypted storage. Encrypted storage was first intended to protect data from illegal access or theft. As a consequence of the development cloud computing,

which allows for the secure remote storage of both content and data, the internet today stores not just material for sharing but also a large amount of personal data that requires limited access and privacy protection. In regard to the safekeeping of sensitive information online, striking a balance between data security and accessibility is becoming more critical. Cloud computing will reach its full potential only with the introduction of safe online data management systems. Because they guarantee that personal data is safely delivered to the server and that unauthorized individuals are prevented from accessing the data, access control and secure data transmission are the pillars of traditional privacy measures for online personal data. Once the data reaches the server, it is decrypted and made accessible to users, who can then take use of services like search and data summarization. Untrustworthy or hostile service providers or hackers may get access to a user's sensitive information, hence raising the danger of identity theft. A system administrator may be able to see personal picture albums that are publicly accessible in an unencrypted manner. If the data in an encrypted database is encrypted using traditional cryptographic ciphers, processing and extracting the data from the database becomes challenging for the server. Thus, information retrieval from encrypted databases must be improved if users' privacy is to be preserved without jeopardizing the availability or value of the material being returned.

3.8.1 Partially Homomorphic Encryption

Through the use of homomorphic encryption algorithms, [23] it is possible to safely conduct tasks that can be performed on encrypted data without decrypting it. Another way of saying it is that the homomorphic encryption of a group (G), where m1 and m2 are the encryption keys, must satisfy the condition that if one possesses two encrypted c1 = Ek (m1, Ek(m2)), one can simply calculate EK (m1 M2) without decrypting the c1 and c2 encryptions. Homomorphic encryption algorithms, for example, are necessary for retrieving private information, electronic voting, multi-party computation, and cloud computing, among other uses. Numerous distinct problems benefit from totally homomorphic encryption systems that allow for just two operations on the underlying algebraic structure, addition and multiplication, in comparison to typical encryption methods. Rivest, Adleman, and Dertouzos pioneered the study of homomorphic encryption immediately after the creation of RSA, coining the term "privacy homomorphism" to characterize their work. Multiplication homomorphic encryption on the basis of RSA is a robust homomorphic encryption technique that may be utilized in a wide range of applications. ElGamal is an asymmetric

multiplicative homomorphic cryptography technique that makes use of asymmetric multiplicative homomorphic cryptography. This kind of encryption is referred to as "additive homomorphic encryption." Gentry pioneered the concept of a completely homomorphic encryption system. As a result of this finding, a variety of completely homomorphic encryption algorithms were eventually created. The complexity of certain challenges is crucial for the security of homomorphic encryption systems, since it dictates their level of security. For a number of practical reasons, the adoption of homomorphic encryption algorithms that are bit-by-bit is rare (such as storage and transmission efficiency). Delegating computation is critical in many client-server systems, and homomorphic encryption methods enable this to be done safely and securely (e.g., cloud computing). [24] Due to the restricted processing capability of the clients and the need to keep communication costs low, client-server encryption methods that are computationally "light" (e.g., over finite fields or rings) are desired. Due to the inadequacy of current structures, even if homomorphic encryption systems of this kind are invented, their complexity will make them impossible to use in real-world applications. However, when the ciphertexts in Domingo-completely Ferrer's homomorphic encryption technique are multiplied over polynomial rings, the ciphertext expands in proportion to the number of ciphertexts. Due to the usage of well-known plaintexts, DomingoFerrer's technique is made worthless. Armknecht and Sadeghi created an additive homomorphic encryption technique based on Reed-Solomon code that requires just a small number of multiplications, providing additional security. In comparison, Armknecht's technique has a flaw in that it may result in improper decoding of the message if the number of mistakes reaches a critical level. Ring learning with mistakes serves as the basis for a fully symmetric symmetric absolutely homomorphic encryption scheme. According to Boneh and Lipton, every completely homomorphic encryption system over ring Zn may be cracked in sub-exponential time under certain number theoretic assumptions. According to this concept, a cryptanalyst who does not have prior knowledge of the secret key may decrypt any ciphertext in an exponentially short amount of time. Maurer and Raub later extended Boneh et alwork .'s to finite fields with tiny characteristic values, which they dubbed the Boneh-Raub expansion. It is conceivable that cipher-only assaults may undermine completely homomorphic encryption schemes over finite fields or rings.

3.8.2 Fully Homomorphic Encryption Schemes

Constructing a cryptographic approach that is completely homomorphic has been a long-standing open topic in the cryptography field. Rivest, Adleman, and Dertouzous proposed this notion, which was first referred to as a privacy homomorphism, a few years after Rivest, Shamir, and Adleman invented the RSA encryption method. It is possible to easily get Q I + I mod N = (Q + I mod N, an encrypted version of the product of the original plaintexts, using the multiplicatively homomorphic encryption method known as RSA. Thus, what may be done with a completely homomorphic encryption scheme: an encryption scheme E with an efficient algorithm EvaluateE that, given any valid public key pk, any circuit C (not only a circuit of multiplication gates, as in RSA), and any ciphertext I " EncryptE pk," returns a ciphertext that is entirely homomorphic to the original.

$$\psi \leftarrow \mathsf{Evaluate}_{\mathcal{E}}(\mathrm{pk}, C, \psi_1, \ldots, \psi_t) \ ,$$

Is pk capable of securely encrypting C (1, t)? And, without the decryption key, you have entire flexibility to act on encrypted data in any manner you see fit (query it, write into it, or do anything else that can be suitably described as a circuit). The panel discussed the possibility of using private data banks. It is conceivable for encrypted data to be stored on a server in which the user lacks confidence. After this, it may contact the server for information about the data, and the server will answer with an encrypted response that the user may decode using the EvaluateE method. [25] Rather than returning all encrypted data to the user for processing independently, which would be a simple solution, we want the server to respond unambiguously in this instance. Since then, cryptography experts have compiled a thorough list of "killer" applications for completely homomorphic encryption. There has been no physical structure up to this point [26] .

4. PROPOSED METHOD

In this chapter, we compare the smart card's security features with those of the java card in an effort to better grasp the java card's security procedure, We next discuss some of the security and fault tolerance problems with the smart card system, as well as A number of the assaults that take advantage of these flaws.

4.1 JAVA CARD SECURITY

All the Smart card security mechanisms, such as communication verification and key encryption, that make up a java card contribute to its overall security. Applet firewall and bytecode verification are two examples of Java card security features. Java's built-in safeguards against grammatical and syntax errors [2]. See the diagram below for an overview of the overarching plan for protecting data exchange across the various java card parts.

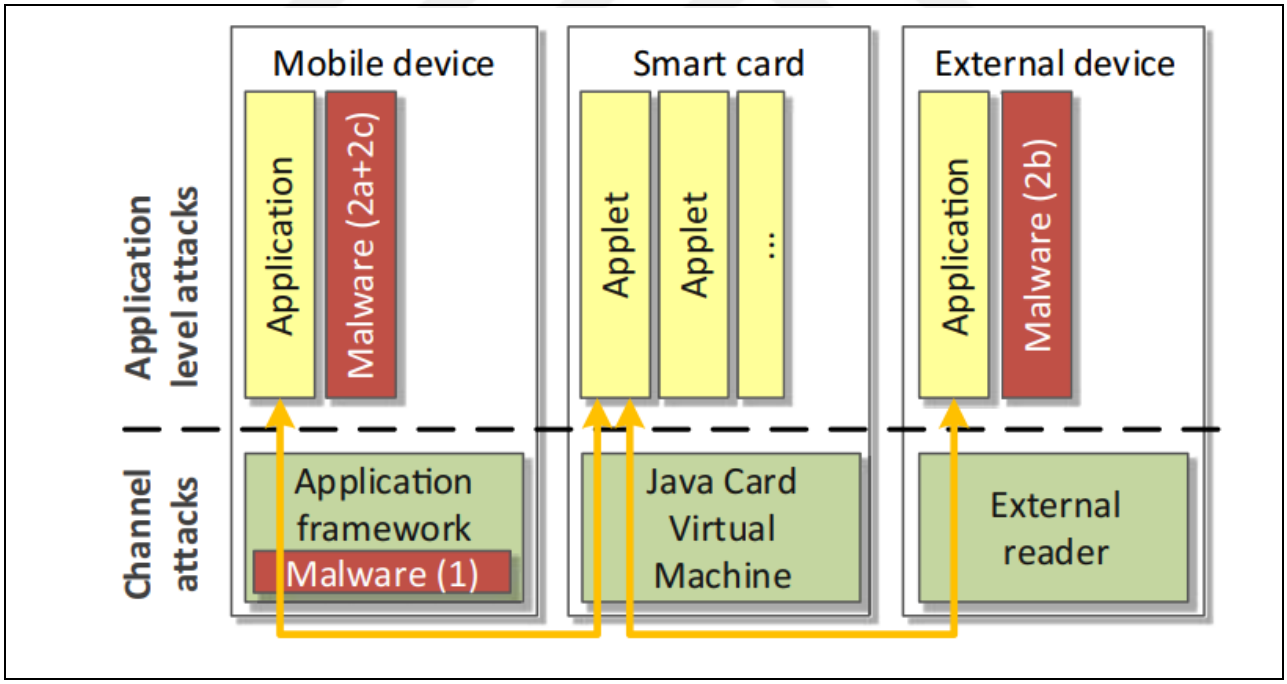

Figure 4.1: Secure channel between applet and java card

This helps to secure an program execution environment and lessens vulnerability to malware interfering with the execution of objects on the smart card . The java card security system suggested a firewall between applications that communicate with one another on the same card through a shared object.

4.1.1 Java Card Applet Security

Applets written in the java programming language can now be deployed to smart cards and run instantly, much like web browser plugins. Since each applet is stored in its own transparent and secure section of the java card's memory, and yet they are able to communicate with one another using a universally accessible medium; the Java card provides a secure environment for the successful implementation of the applets through a strict set of firewall requirements that allows for fine-grained management of data passing between applets. See the diagram below for a breakdown of the cap files' contents:

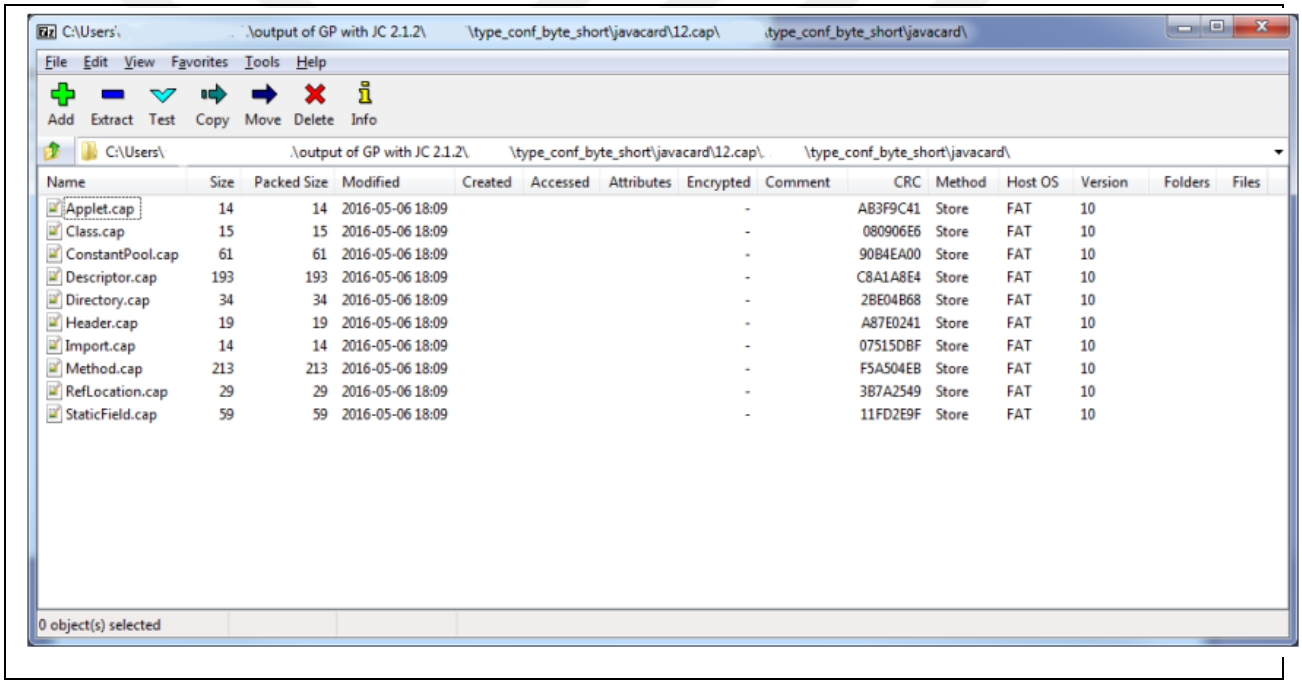

Figure 4.2: CAP file contents

4.1.2 Java Language Security

Java's security features, when combined with those of the Java Card API, have made it possible to execute a wide variety of secure applications quickly and intuitively modern object-oriented programming techniques. In contrast, the security features of traditional smart cards carrying out application code in either assembly or C allows for an assessment of the safety measures the speed of execution into account. We may summarize Java's built-in security features like this. Through the use of public, protected, and private name spaces, it establishes who has access to the pooled

data. Since It's a type of programming language called "object-oriented.", it also allows for the encapsulation of data within a procedure and the provision of several entry points to the same set of data for various kinds of applications.

i. By encapsulating data in a shared location, it's possible to re-use previously proven, effective code.

ii. Compile and runtime error check code at runtime to catch typos and inconsistencies.

iii. Provides a formal definition for the java bytecode language, which is the compiled form of Java.

iv. Unlike other languages like C and C++, pointer-related security vulnerabilities are not an issue.

v. It does not permit using array indices outside of those specifically allocated areas of memory.

vi. Allows users to see where their memory is being allocated, therefore eliminating the possibility of a mistakenly reused memory area.

4.1.3 Smart Card Security

Many people underestimate how difficult it is to ensure the safety of a Smart card, which is in fact a multi-faceted challenge with equally intricate solutions. However, there are vulnerabilities in particular layers of security . To get around the drawbacks of using multiple cryptographic algorithms that each require their own dedicated chip and negatively impact other areas of the card's performance, we integrated the precautions taken by the smart card system with those of the java card system. After integrating the java card's security features, the whole operation of the smart card security system is depicted in figure 4.3.

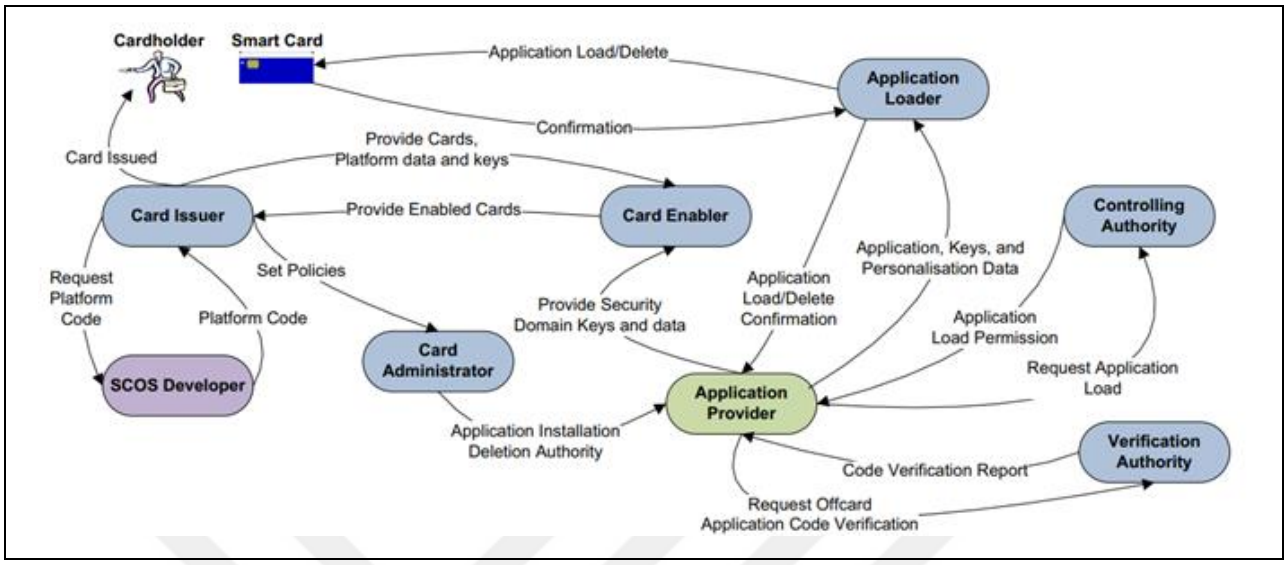

Figure 4.3: Generic platform of smart card

Figure 4.3 explains how the java card's overall security mechanisms can independent and fragmented, suggesting each applet can execute tasks independently while still communicating with one another and the system's other components over a secure channel.

4.2 SECURITY CHALLENGES

Access to an array's indices is constrained by type mismatch checking, which in turn uses the space actually for that array, was the primary reason for developing the Java Card. To improve the security of smart cards, this was done. In addition, unlike C and C++, pointer arithmetic is not supported, and access to classes, methods, and fields is strictly regulated regardless of the descriptor. Java cards are vulnerable to attacks that aim to exploit their design flaws, which can be hardware, software, or a combination of the two. One such flaw is the java card's limited EPROM memory, which limits the amount of code that can be executed (fetched and decoded) during the bytecode interpretation process (see Figures 4.4 and 4.5).

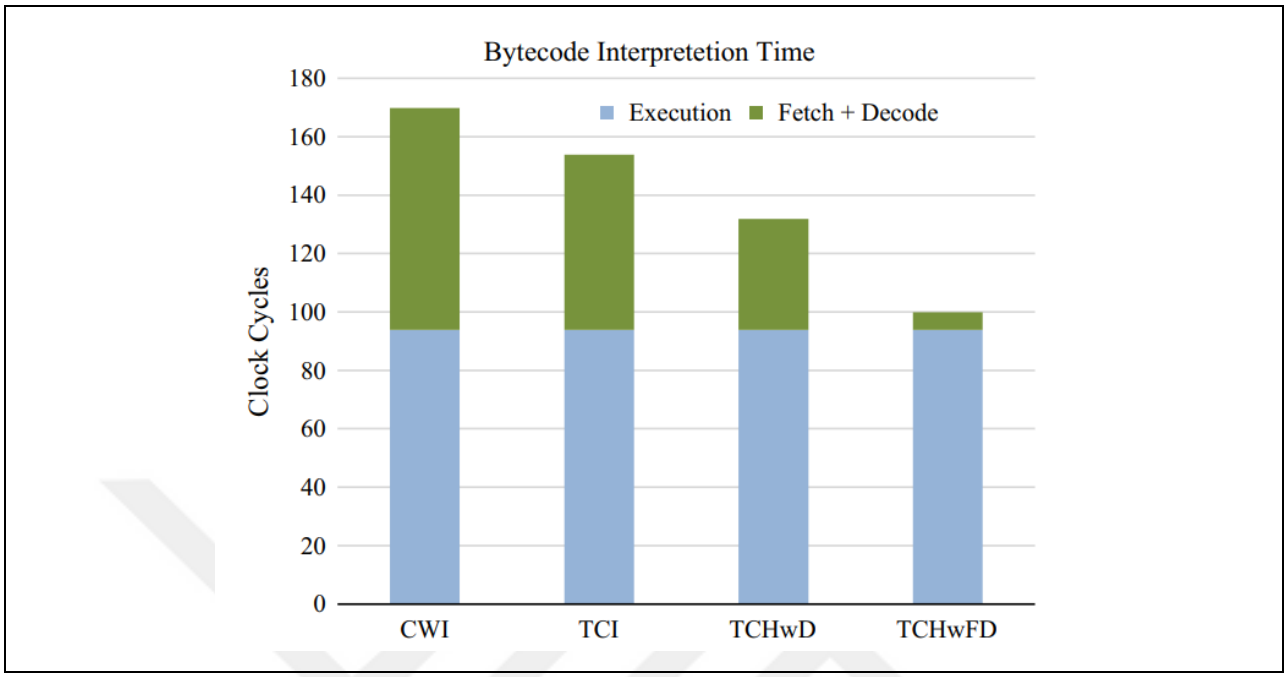

Figure 4.4: Time required for the code execution in java card

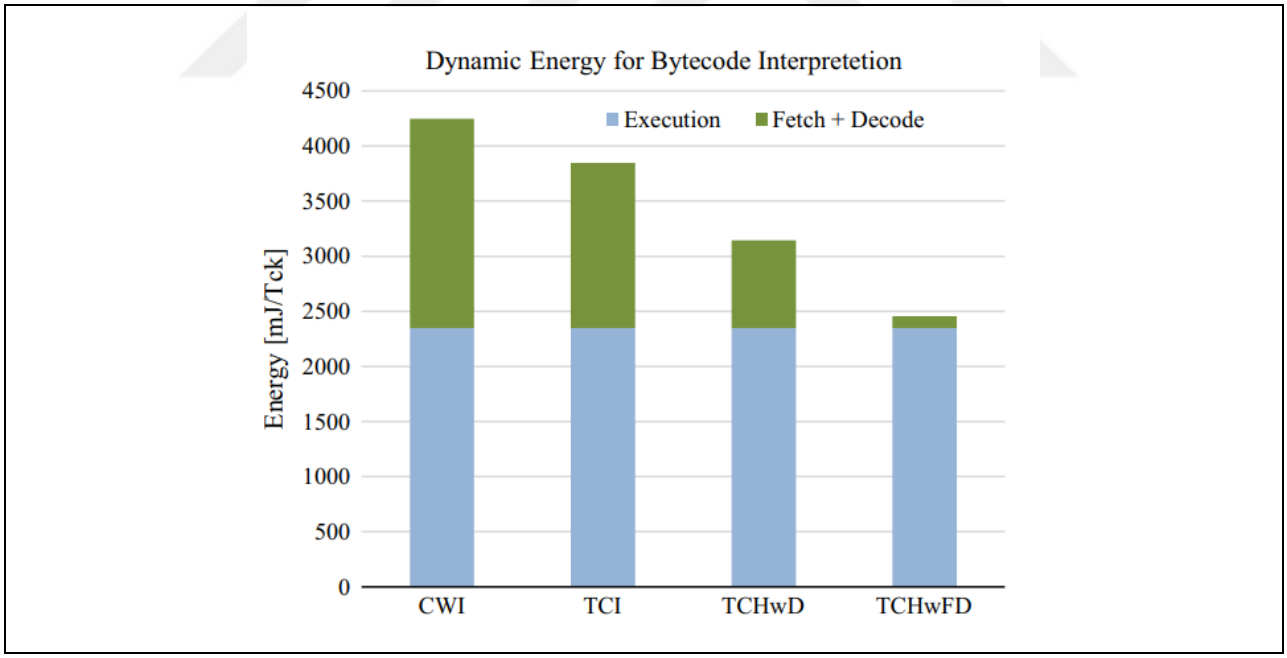

Figure 4.5: Energy required for the code execution in java card

BCV, or Bytecode Validator, Verifier, and Validator, the GP's Worldwide Network , Moreover, using Java Card Firewall are some additional precautions that have been taken. Because these safeguards protect sensitive data, which is the primary target of these attacks, we will discuss some

of the most common software and hardware attacks on the Java card and the Java programming language in this section. These attacks can be divided into two categories: software and hardware:

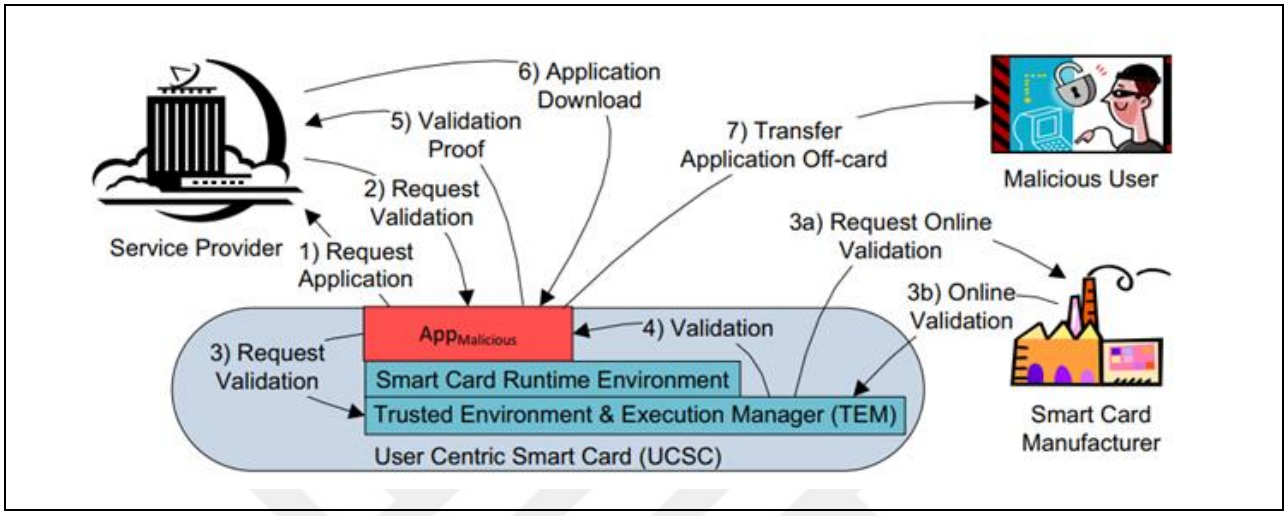

Figure 4.6: Generic attacks scheme

4.2.1 Hardware Attacks

To malfunction the smart card into granting unwanted access or revealing important information saved within the card, a hardware attack is necessary Hardware attacks are costly because they require the use of high-end, complex equipment that is typically only available to government agencies or large corporations Hardware attacks can take the form of electromagnetic or Power-Trace Analysis; they can also include attacks that aim to alter the functioning state of the card by mean code execution; these attacks can involve laser beams or changes to the card's power and clock. Both hardware and software attacks target the Java card's flaws, such as its limited runtime security checks as a result of its due to a combination of factors, including the delay inherent in bytecode verification BCV and applet configuration, several security measures must be implemented. the limited memory space dedicated for each applet, which makes it transport which applet is responsible for which job and what is stored in that applet, as shown in figure 4.6 below:

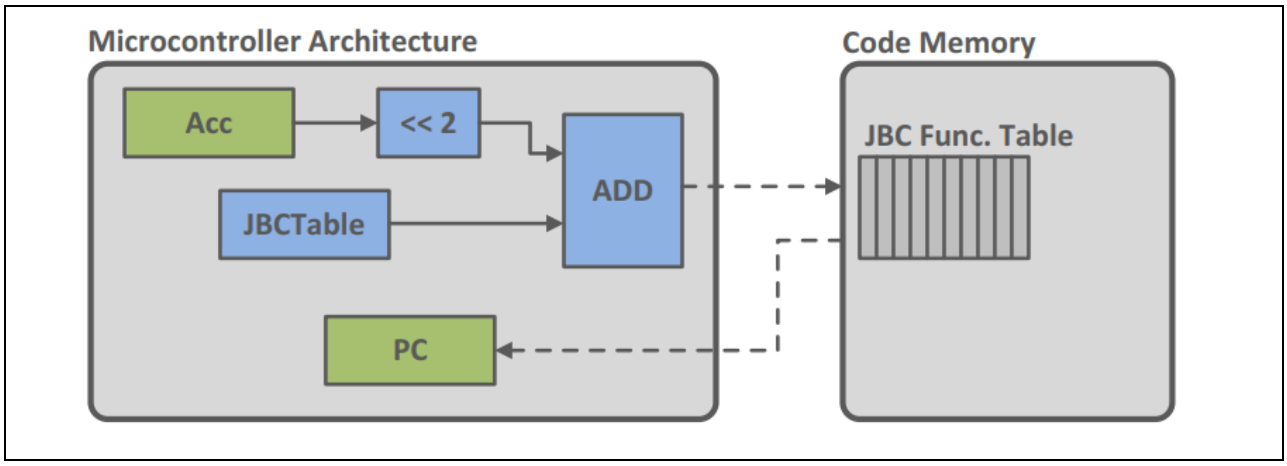

Figure 4.7: Micro controlling the Java card Memory through a hardware piece

4.2.2 Software Attacks

Attacks that modify a CAP file to carry out malicious tasks, installing a third-party applet that gains access to the memory, or simply carrying out a brute-force threat to guess the PIN or password are all examples of logical attacks that aim to use the vulnerabilities of the java card constraints and gain unauthorized entry to the card's memory typically, these attacks are carried out by replacing legitimate code in a CAP file with malicious code, installing malicious third-party applets with memory access, or employing brute-force techniques to guess the PIN or password. One feature of java card security measures is that the applets in the java car operate independently from each other, as shown in the figure below. This is because almost all of these attacks rely on the attacker being able to install custom applets and CAP files.

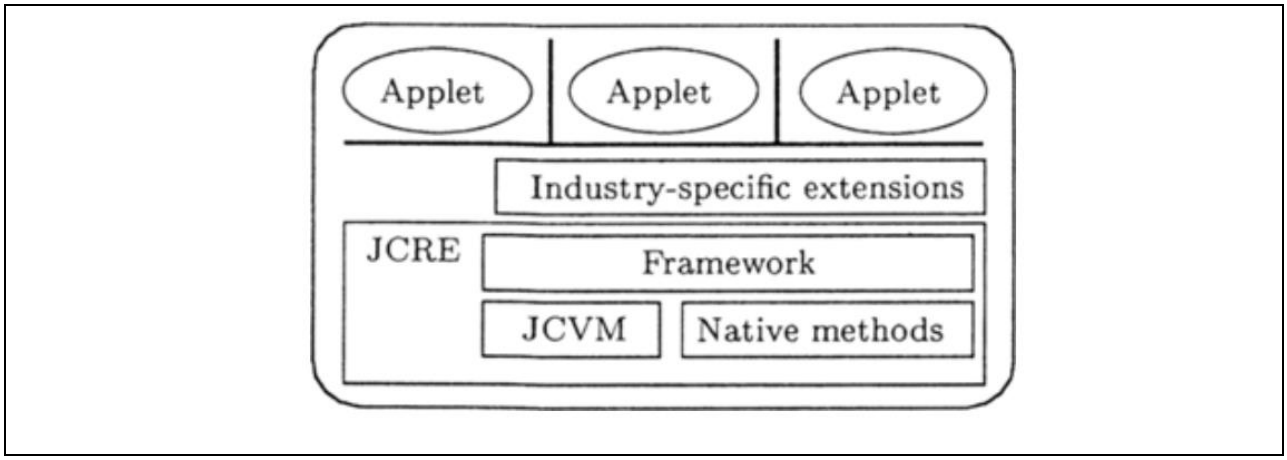

Figure 4.8: Java card Applet's framework

Although hostile applets that carry out CFT attacks are typically rejected by the verifier due to the fact that they were not generated by a lawful complier, ill-formed applets are still a problem.[40] Unless the card has a proven method of avoiding BCV fault injection, the card is suspect, these applets may execute risky tasks.

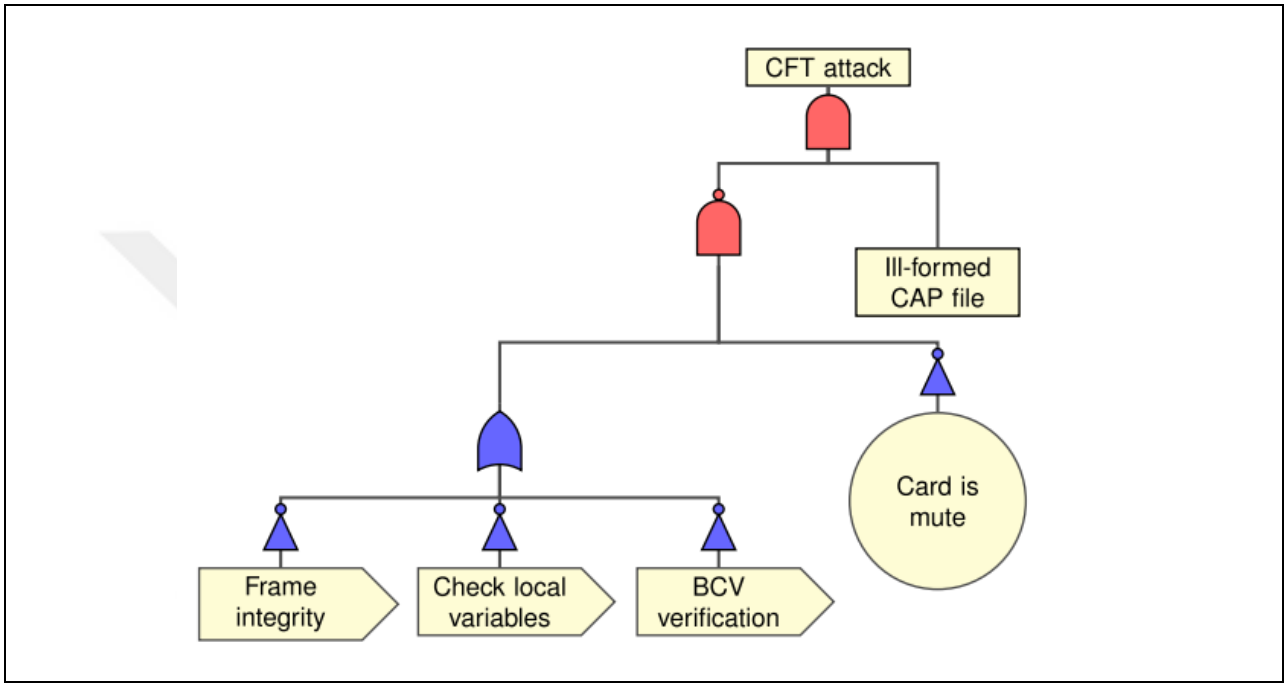

Figure 4.9: CFT attack by ill formed applets

Some examples of elementary logical attacks based on malformed applets are:

i. To refer to an illegally short value.

ii. Below is some code that demonstrates casting an illegal class object to an array.

```
1  static byte[] ba = {0x00, 0x01, 0x02};
2  static short[] sa = (short[]) ptr( addr( ba ) );
3
4  public static short addr(Object o) {
5      return o;
6  }
7
8  public static Object prt(short addr) {
9      return addr;
10 }
```

i. Misuse of the Java card's transaction mechanism, as illustrated by the code below:

```
1
2  short[] trArrS;      // class variable
3  ...
4      short[] localArrS;  // local variable
5      JCSystem.beginTransaction();
6          trArrS = new short[1];
7          localArrS = trArrS;
8      JCSystem.abortTransaction();
```

ii. Modifying array metadata in the manner depicted in the following diagram:

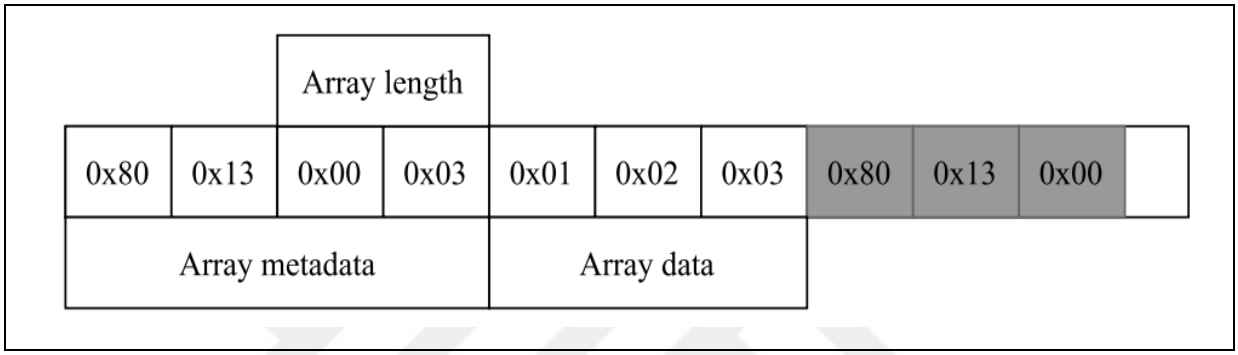

Figure 4.10: Implementation of the array manipulation attack

iii. Binary incompatibility can lead to type misunderstanding, as demonstrated in the following code:

```
1
2  public static short[] returnRef( short[] ref ) {
3      return ref;
4  }
```

iv. Underflow in the stack due to a lack of checks, as depicted in the following diagram:

```
1      dup_x 64; // 64 = 0100 0000, m = 4, n = 0
2      sstore_1;
3      sstore_2;
4      sstore_3;
5      sstore 4;
```

v. To edit an AID applet, simply dump its memory and edit the code as described below:

```
0x00: 0xA0 0x00 0x00 0x00 0xEE 0x01 0x02 0x00
0x08: 0x0E 0x00 0x02 0x1B 0x02 0x00 0x02 0x07
.........................................
0xE8: 0x07 0xAC 0x00 0x00 0x00 0x08 0xB7 0x00
0xF0: 0x07 0xA0 0x00 0x00 0x00 0xEE 0x01 0x02
0xF8: 0x8D 0x84
```

i. Manipulation of CAP files through the use of corrupted applets.

ii. By running malicious code and taking advantage of type confusion, a complete memory dump can be obtained .

iii. The ability to run any program at will.

iv. Creating duplicates of the card's applets, as depicted in the image below:

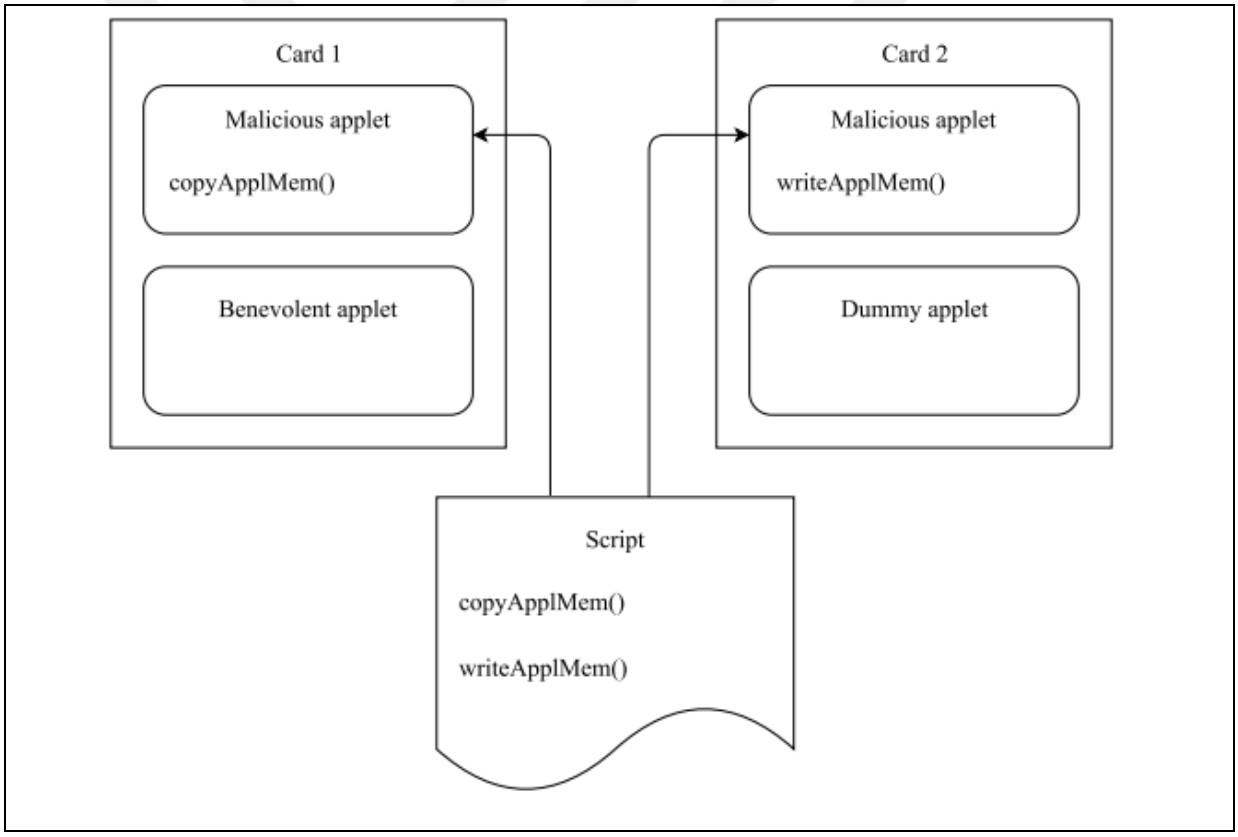

Figure 4.11: Cloning of an installed applet

vi. Buffer array access outside of allowed bounds, as in the following code:

```
1   public static short addr( byte[] ptr ) {
2      return (short)ptr;
3    }
4
5    public static byte[] ptr( short addr ) {
6      return null;
7    }
```

While a firewall of applets, such as the one described in the code below, can be used to protect against this type of attack:

```
1
2  public class App extends Applet {
3      byte[] buffClassCopy;
4      ...
5      public void process(APDU apdu) {
6          byte[] buffLocalCopy;
7          byte[] buffer = apdu.getBuffer();
8          buffLocalCopy = buffer; // allowed by the firewall
9          buffClassCopy = buffer; // forbidden
10     }
11 }
```

vii. The Return of the Ownership Rollback The next illustration depicts the brute-force method typically used to create a pin try counter.

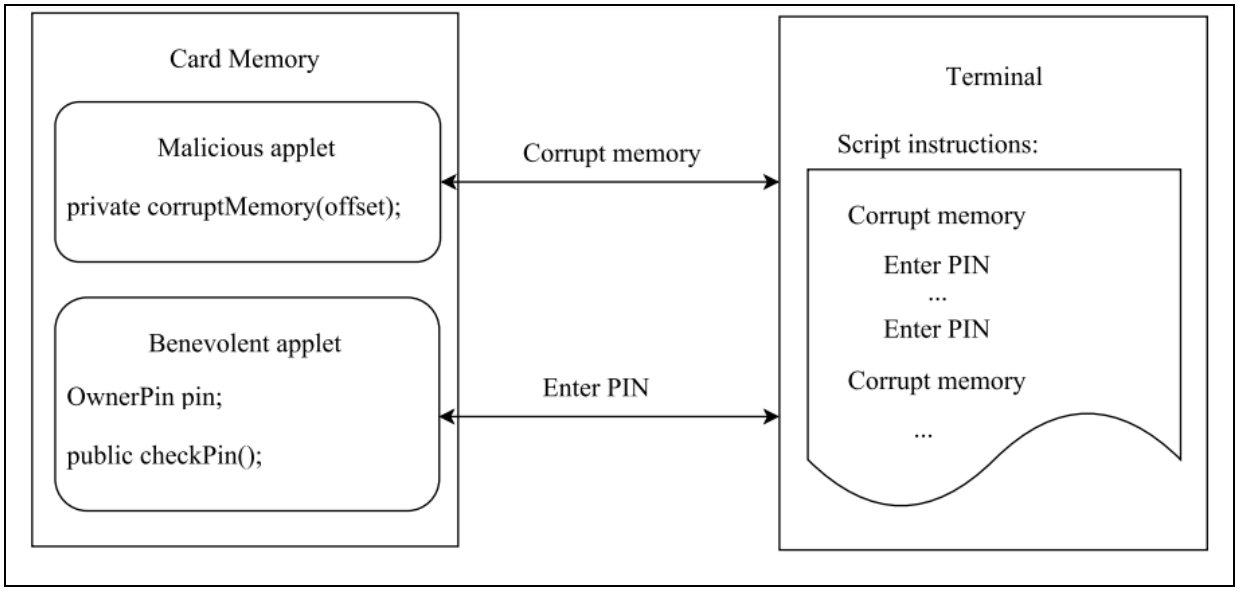

Figure 4.12: Implementation of Owner Pin attack

viii. Owner Pin memory space instance overcome ECB encryption by observing the location of the PIN, as depicted in the following figure.

PIN	Corresponding encryption
0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77	0x00 0x90 0xA9 0x6D 0x91 0xC3 0xB0 0x6A 0x39 0xC9 0xA9 0x6D 0x91 0xC3 0xB0 0x6A 0x39 0xC9 0x63 0xBB
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77	0x00 0x90 0x2B 0xB1 0x9D 0x21 0x84 0x1D 0xF2 0x7A 0xA9 0x6D 0x91 0xC3 0xB0 0x6A 0x39 0xC9 0x1A 0x26
0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	0x00 0x90 0xA9 0x6D 0x91 0xC3 0xB0 0x6A 0x39 0xC9 0x2B 0xB1 0x9D 0x21 0x84 0x1D 0xF2 0x7A 0x71 0x35

Figure 4.13: Memory encryption on the PIN

ix. Retrieving Owner Pin as a plain text by decrypting the Owner Pin as shown in the figure below:

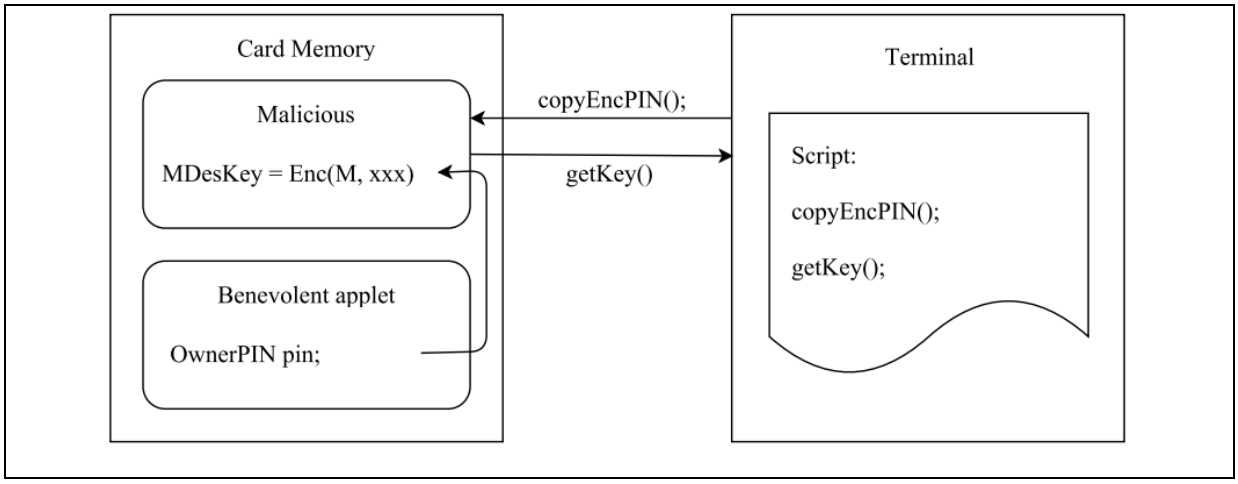

Figure 4.14: Decrypting the Owner Pin

x. As illustrated in the image below, the PIN encryption's DES key must be changed.

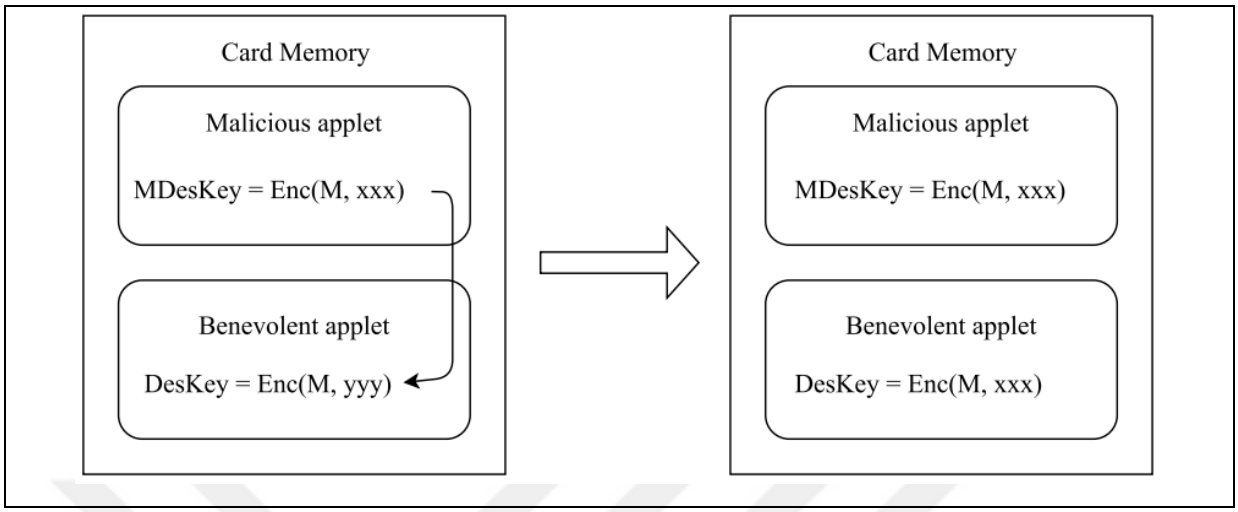

Figure 4.15: Replacing the DES key

xi. Launch an attack by cracking the DES key, as depicted in the figure below:

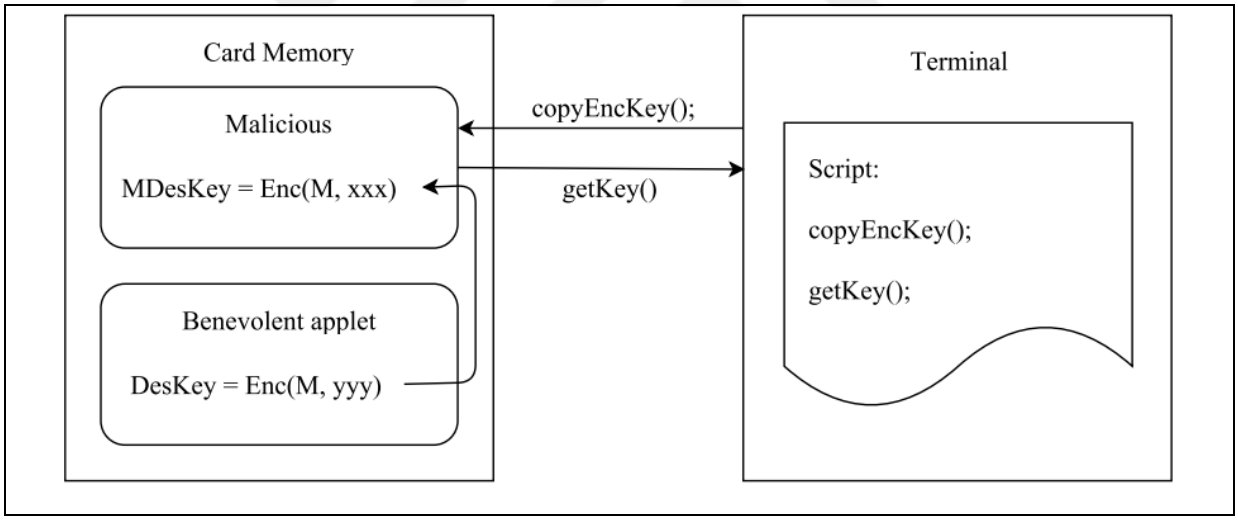

Figure 4.16: Decrypting the DES key

A. BRUTE-FORCE ATTACKS

Malicious applets can use the concept of brute force to access private information stored on a java card by repeatedly trying to enter the correct PIN or crypto key into the card's keypad until they succeed. This is known as a "brute force attack.

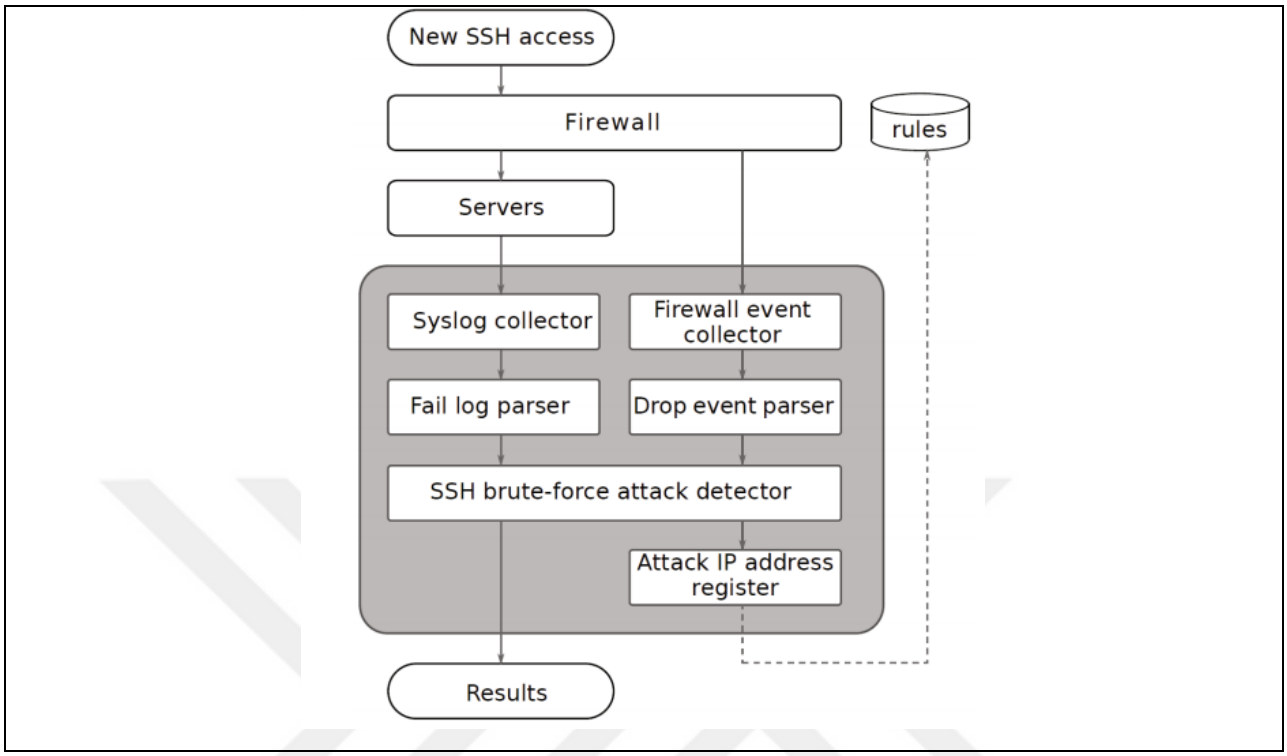

Figure 4.17: Brute force attack detection method

These attacks will always succeed if the owner does not implement a runtime counter, regardless of whether or not the PIN's try counter is protected by a runtime counter. The four bytes in the data structure that represent the maximum number of failed attempts are easily identifiable, but it is difficult to locate the memory location where the try counter is stored . One of the drawbacks of reducing the maximum allowed failed login attempts is that it leaves the system open to denial-of-service assaults, as depicted in Figure 2. DoS.

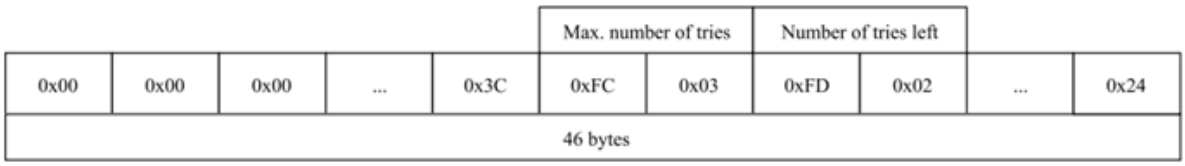

Figure 4.18: The 4-digit try counter stored in the memory space

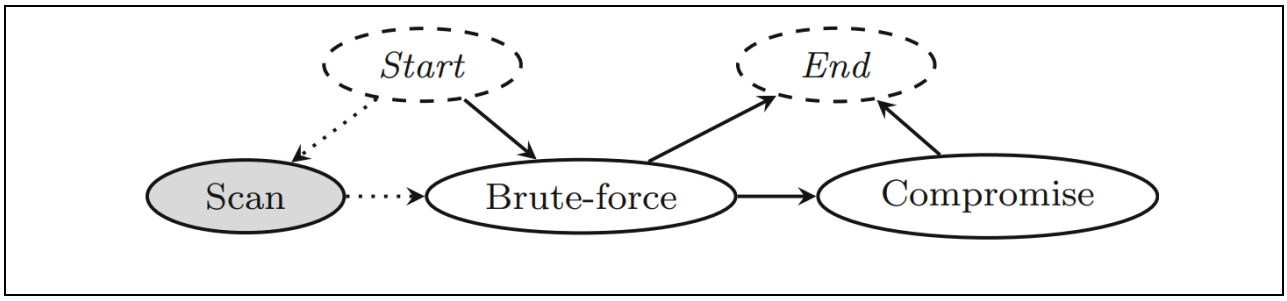

Figure 4.19: General implementation of the Brute force attack

B. DOS ATTACKS

i. These attacks exploit security measures such as "try counters," which limit the number of data or "tries" that can be sent to a system and render it inaccessible to its original users. When a system is overloaded with data, its original users are no longer able to access it. DoS attacks fall into two distinct categories:

ii. The most common type of distributed denial of service attack is known as a flooding DoS attack, and its goal is to slow down or stop a system from providing service by bombarding it with redundant and unnecessary data the service

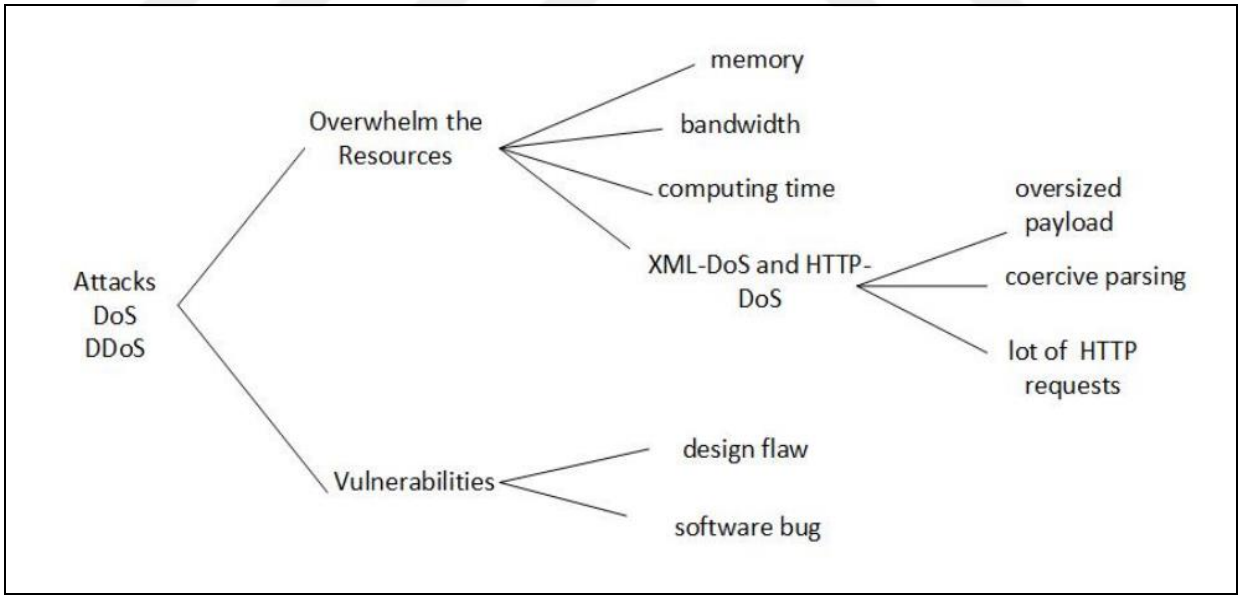

Figure 4.20: General implementation of the flood DoS attack

iii. Some types of DoS assaults are designed to cause the system to crash, such as "crash DoS" attacks.

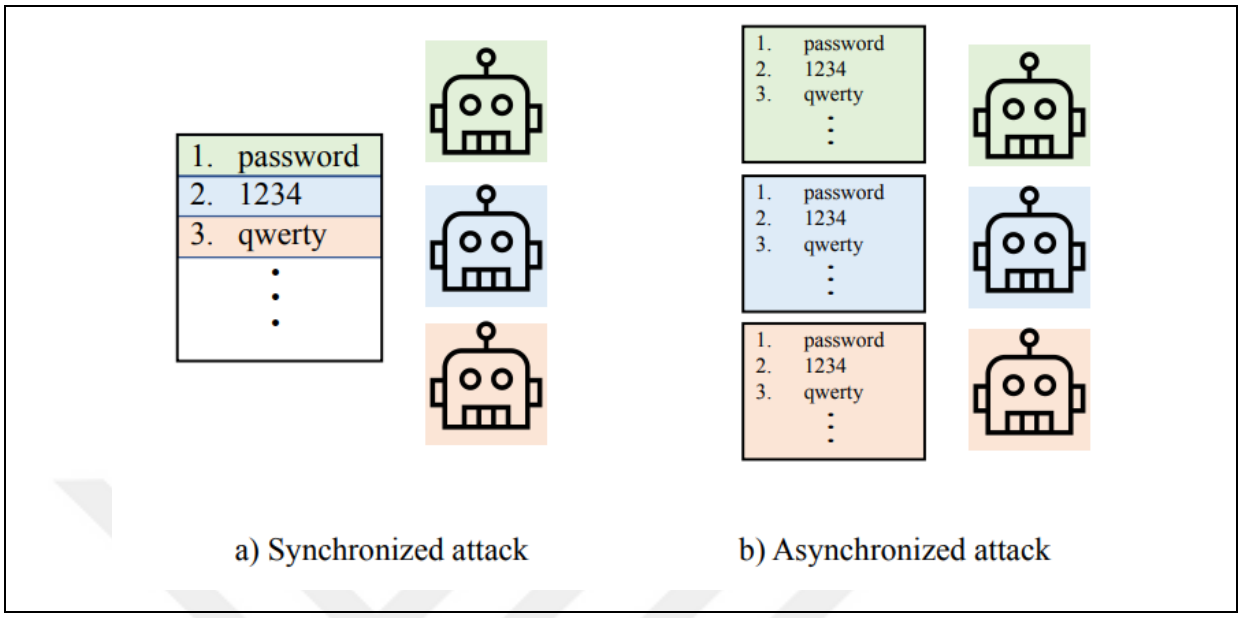

Figure 4.21: Where (a) is the synchronized crash DoS attack and (b) is the asynchronized crash DoS attack

We simulate many java card manufacturing strategies utilizing the output obtained via the Java Virtual Machine and afterwards putting it to good use of these attacks on several cards with varying settings to demonstrate the superiority of our proposed approach over the default configurations of these card manufacturers. We will go into greater depth about our approach in the following section.

4.3 PROPOSED COUNTERMEASURE

To counter brute force and denial of service attacks on Java cards, we suggest a homomorphic encryption-based technique, which we shall detail in this chapter , Before putting our method into action, we will walk over an overview of the java card security system and the fundamentals of homomorphic encryption.

4.3.1 Java Card Security Components

For reasons already discussed, the java card's security features are based on protections built into the Java language and operating system used by smart cards; indeed, the java card was created for the express purpose of bolstering the security of the smart card Therefore, the java card has certain security features that its forerunner, the smart card, lacked.

i. Transaction Atomicity: This is enforced by the JCRE, means that any modifications to transactions in applets must occur after the transaction has finished executing normally.

ii. The applet firewall: With the help of a firewall scheme, which the Java card implements between the individual applets, which specifies a lifetime for the applet's installation and processing and allows the individual applets to communicate with one another through a common object, as illustrated in the image below., the Java platform provides a safe environment in which to execute code.

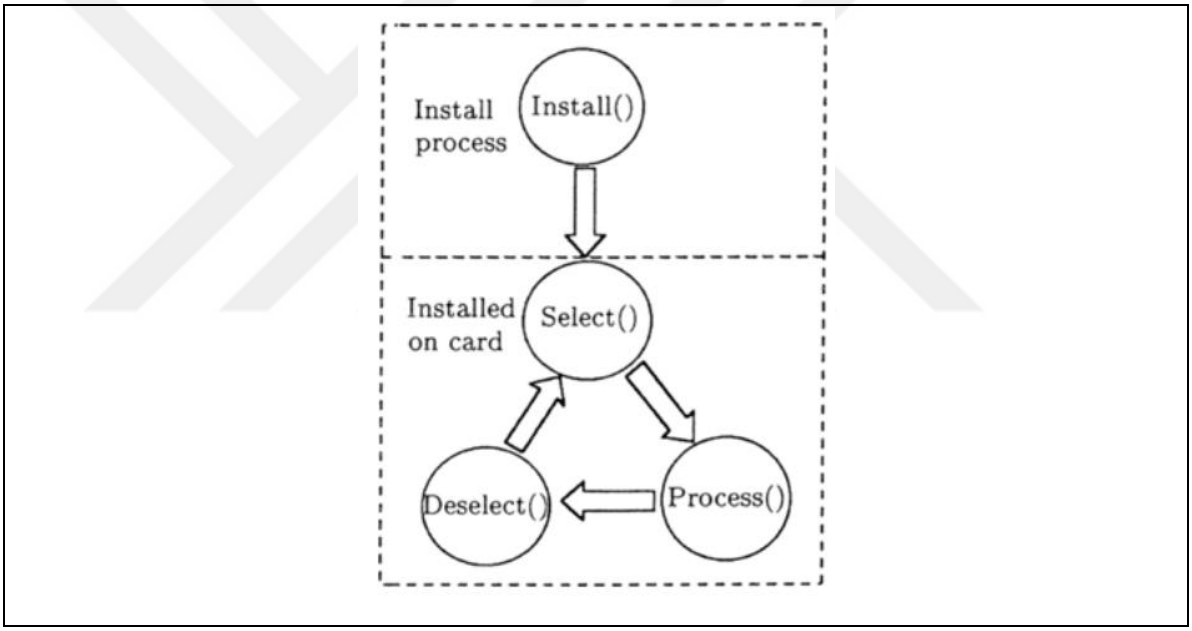

Figure 4.22: Java card Applet lifetime

iii. Cryptography classes: Cryptographic functions including encryption, decryption, PIN management, message digits, and key interfaces can all be implemented on a Java card via pre-installed cryptographic packages.

iv. The allocation of memory space for the java applets: Which can be in volatile or no-volatile memory, is an example of a security gap that needs to be filled as part of the java card's memory management, a crucial component of the java card system.

```
1  public class CA extends Applet
2  {
3      short s = 42; // persistent
4      short[] ss = new short[8];  // persistent
5      byte[] bs = {0x19, (byte)0x84 }; // persistent
6      byte[] ts;  // persistent
7      ts = JCSystem.makeTransientByteArray // content is volatile
8              ((short)8, JCSystem.CLEAR_ON_RESET );
9
10     void foo() {
11         short[] f = { 1, 2 };    // volatile
12         byte v = (byte) 0x42;    // volatile
13     }
14 }
```

The java card virtual machine is activated each time a job is requested. The JCVM allocates a java frame and pushes it into the stack; malformed applets can use this flaw to avoid bytecode verification by invoking illegitimate java frames, as seen in the following image.

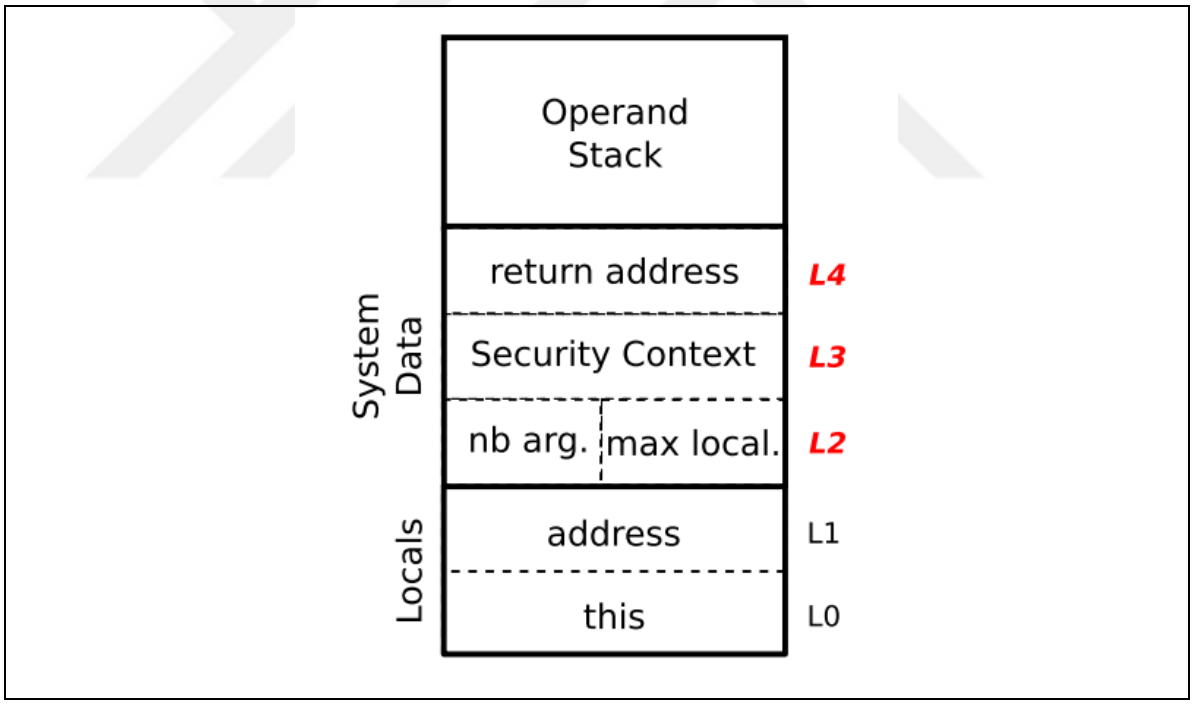

Figure 4.23: Contents of the Java card frame

The pin Tries Remaining () function returns the portion of the random number generated for unsuccessful PIN entry attempts. The function Owner PIN () determines the maximum number of PIN code entry attempts permitted. If an attacker is able to reach the maximum value of the random number, we will provide them with bogus information relevant to the type of information they are

attempting to access but unrelated to any data stored in the smart grid. The following general code demonstrates this point:

```
If (pinTriesRemaining == -1) {
System.out.println("Access Granted")
 System.out.println("Fake Cardholder Name");
 System.out.println("Fake Account Number");
 System.out.println("Fake Expiry Date");
 System.out.println("Fake Security Code");
      } else {
         System.out.println("Try again");
      }
```

Pin Tries Remaining () is a function that generates a random number based on the number of failed attempts to enter the PIN into one of the PIN fields. The number of times a user is permitted to attempt to enter a PIN is governed by the Owner PIN function (). An attacker who reaches the maximum value of the random number will be provided with a bogus data set that has no relation to any data stored in the smart grid. Examining the code in the subsequent example will demonstrate this point:

```
        if(pinTriesRemaining
== -1){
                return true;
              }else{
                return false;
```

```
        }

    }
```

As an additional layer of protection against a brute-force attack on a four-digit PIN, there are smart cards that limit memory access and have a try counter that operates in real time. Using an applet, this type of attack can be cracked in under fifteen minutes.

As depicted in the following diagram, using a PIN applet loaded in an ATM simulator and an online Java tool called "Brute-force," we create a way to launch a brute-force assault on a number of simulated java cards. These configurations simulate various manufacturing processes and security measures:

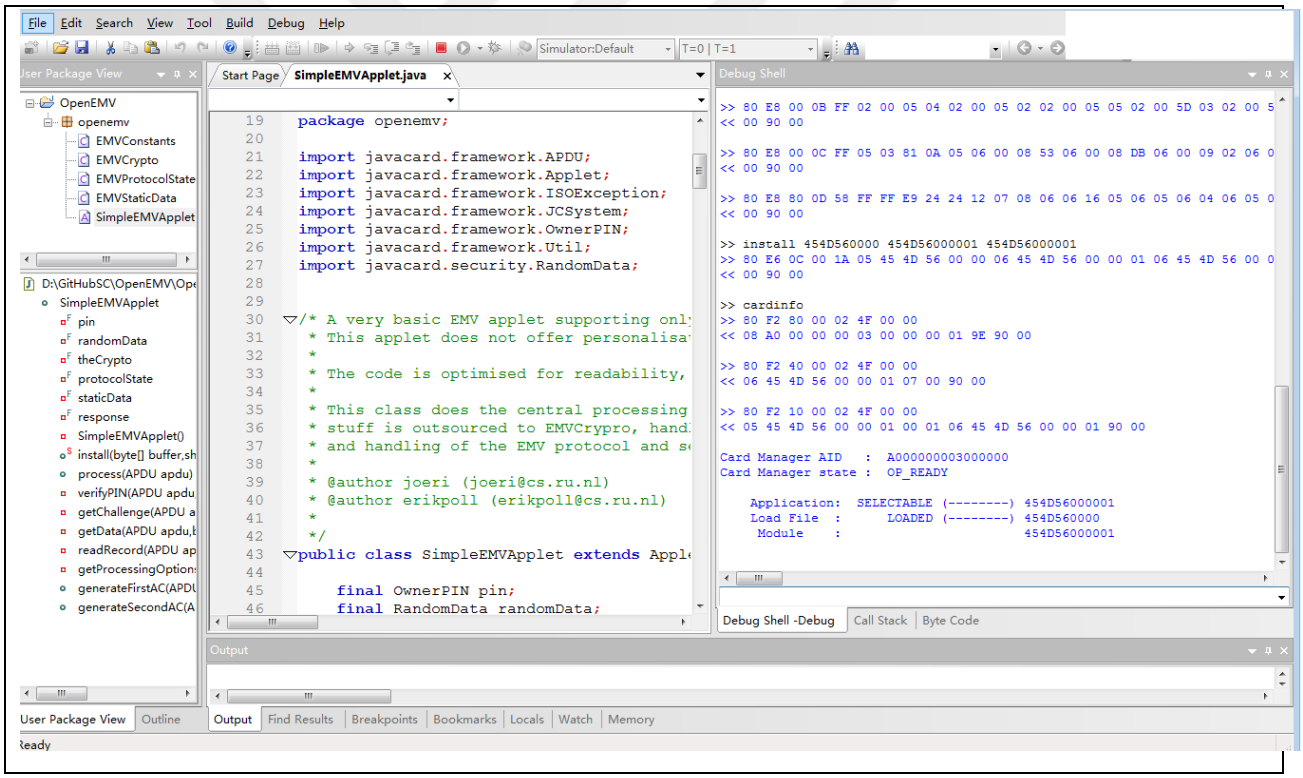

Figure 4.24: Simple PIN verification applet in java card simulator

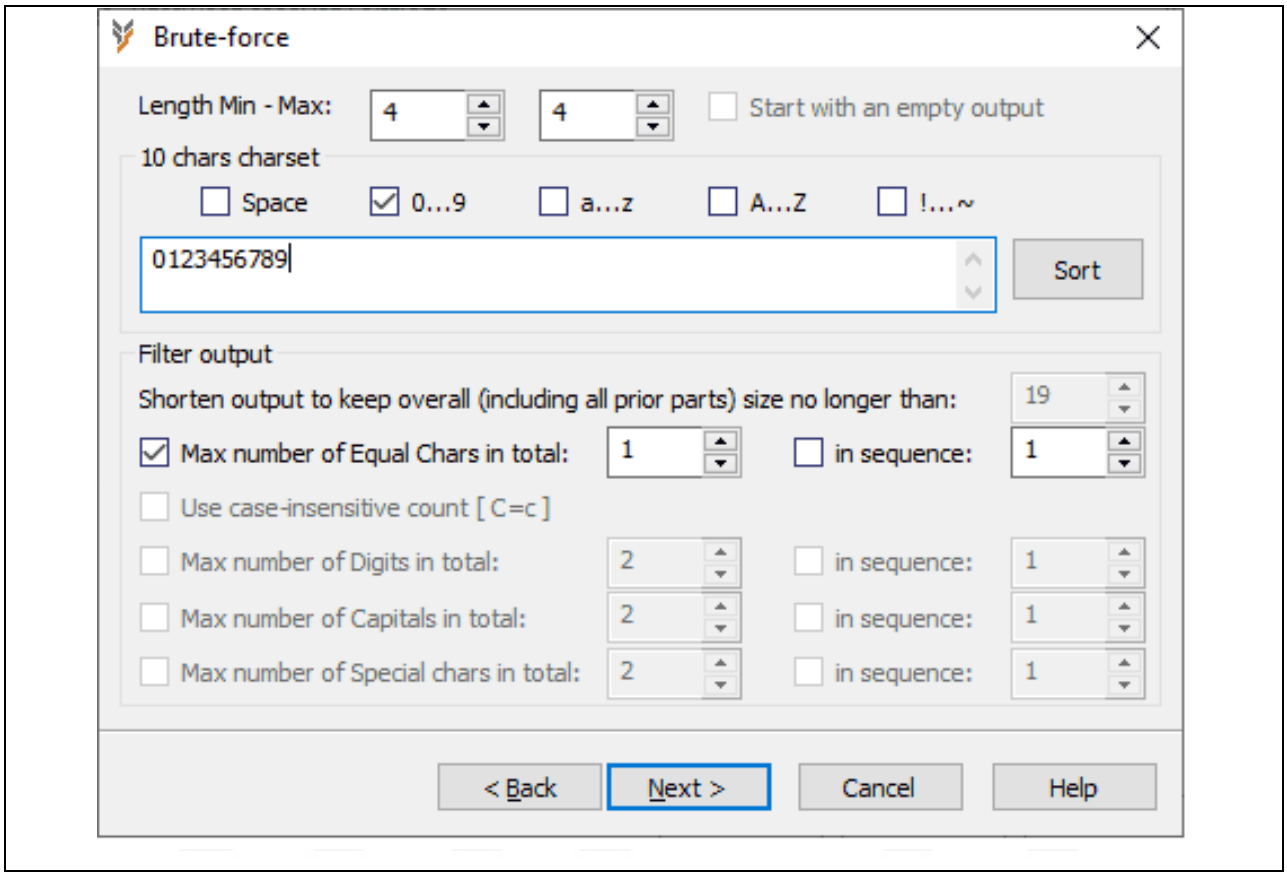

Figure 4.25: An online Java tool name “Brute-Force”

In order to mimic a brute-force attack on an ATM PIN, we have implemented a simulator for such attacks and given the attacker a maximum of four attempts. We trick the attacker into thinking the smart card contains information they may use by presenting them with sham data and transactions but are completely unrelated to it when the attacker reaches the try counter's maximum, and we keep a log of the transaction when the attacker transfers funds to his or her personal account, as shown in the figures below:

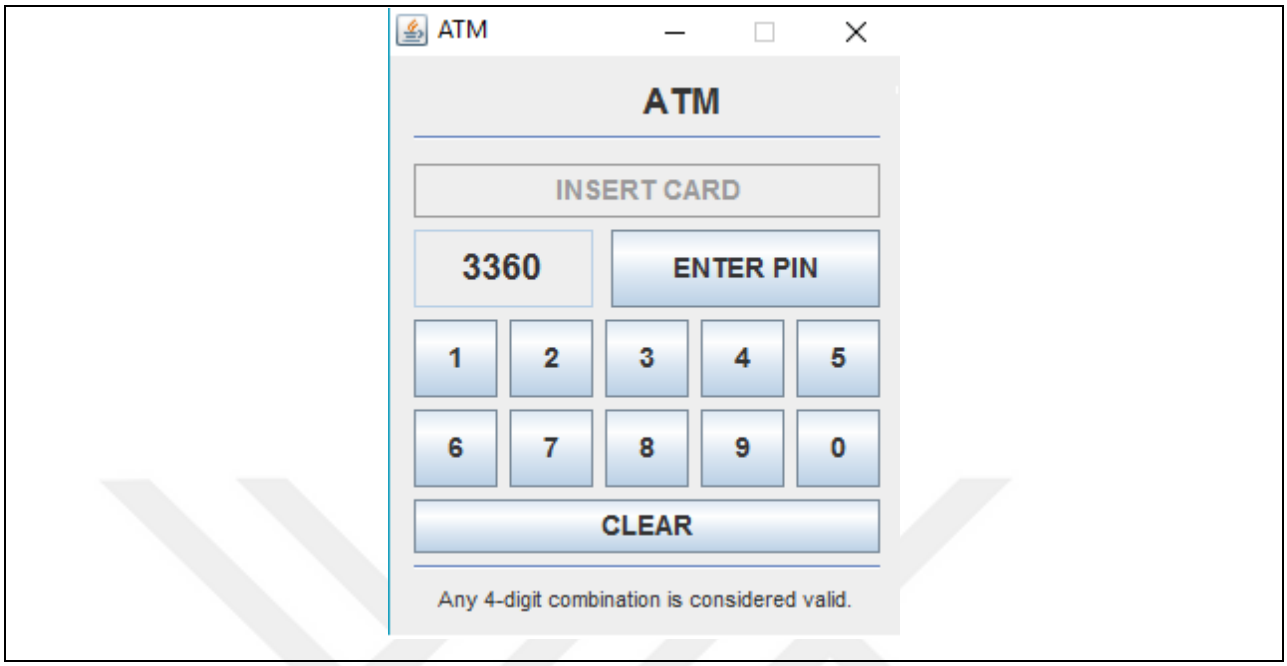

Figure 4.26: PIN ATM simulator

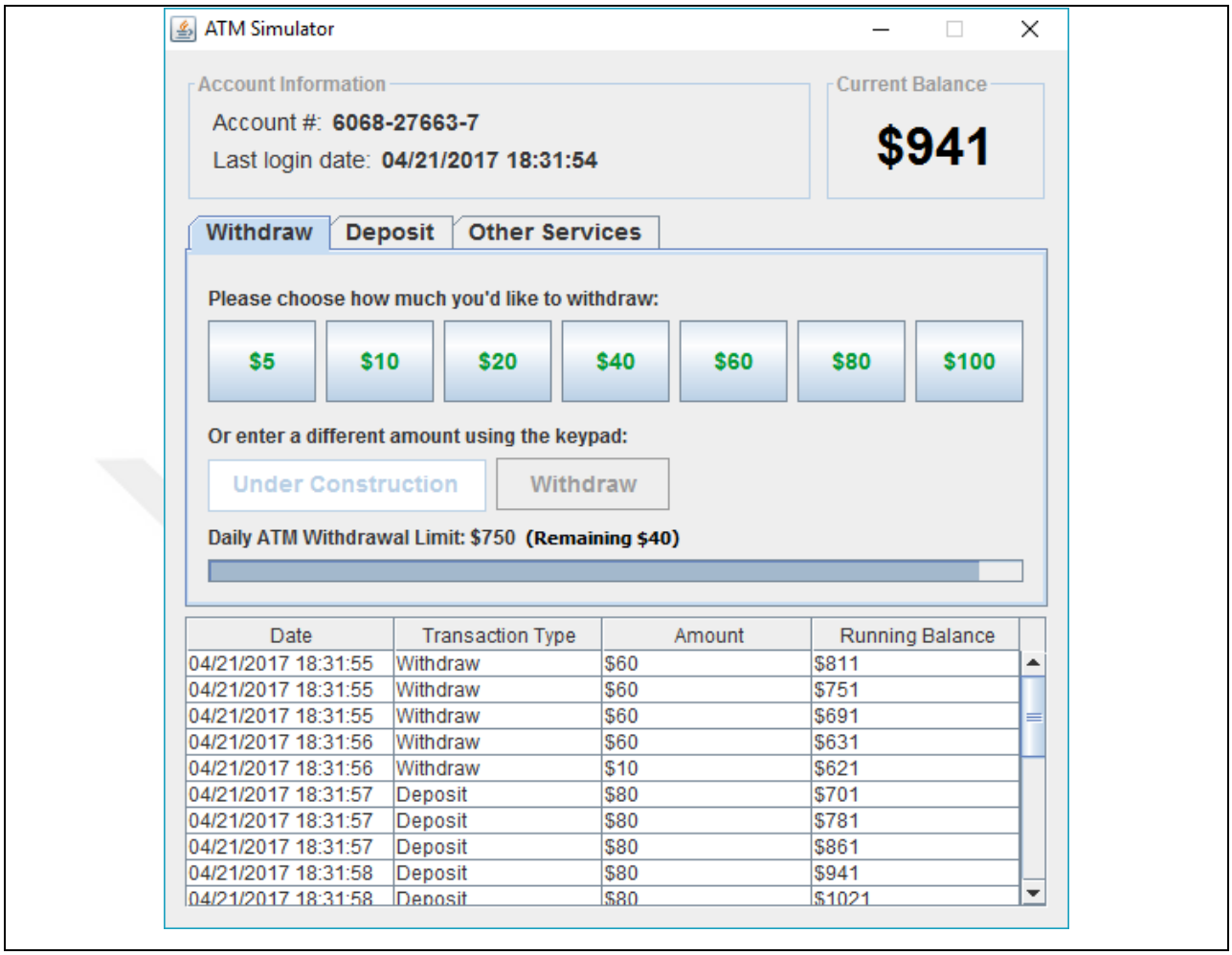

Figure 4.27: Fake information presented to the attacker in order to stop the attack

As can be seen in the graphic below, the ATM simulator responds to withdrawal attempts with a notice indicating a system fault rather than the transaction itself, hiding any potential fraud protection mechanism.

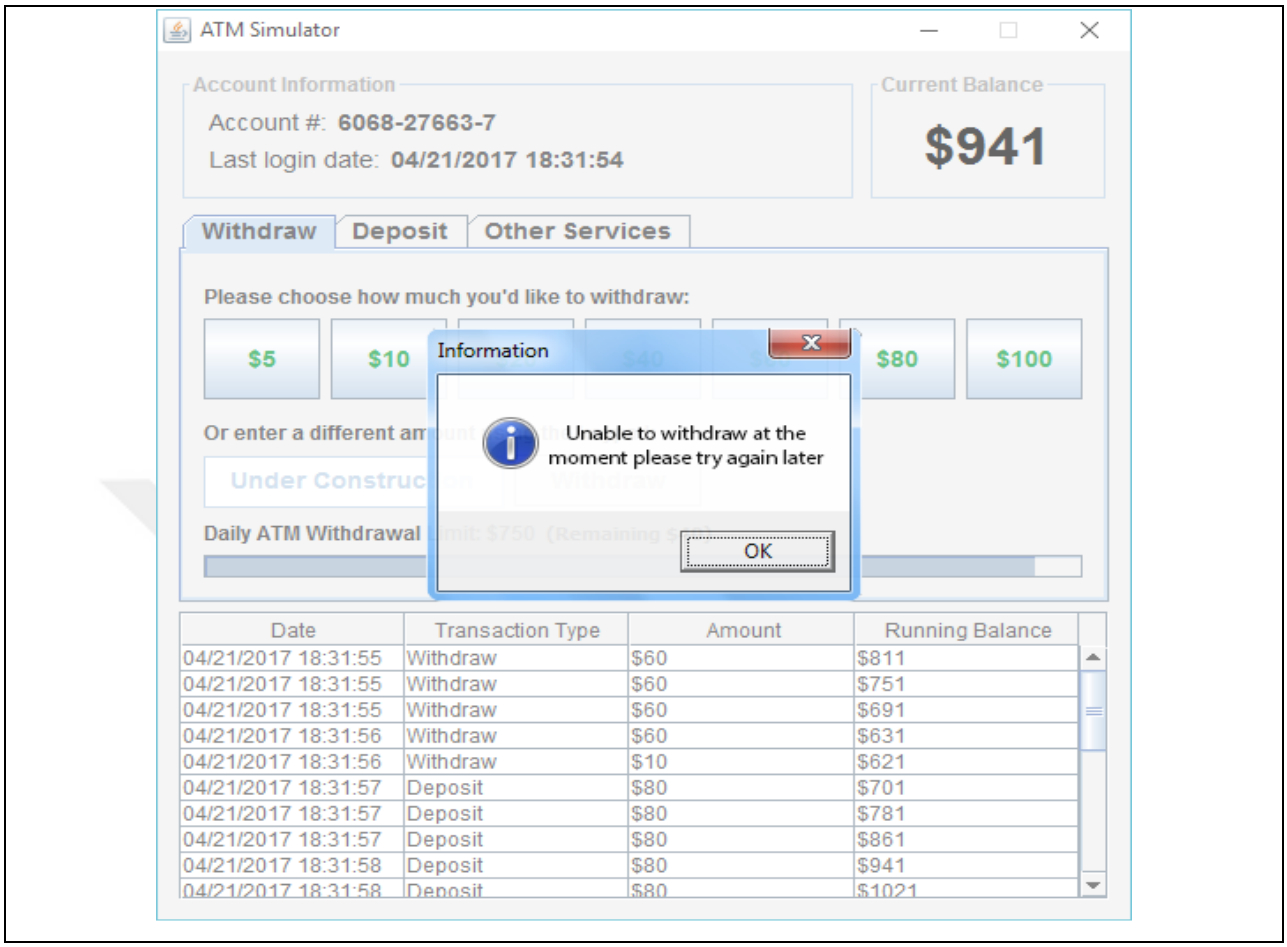

Figure 4.28: Fake message indicating an error in the money withdraw process

Table 4.1: Comparison of the performance to other work sin the literature

author	method	Duration	Type of attack	No. preventions	No. detections
Abha et.al	AES	12 ms	DDoS	500	708
Khanezaei	RSA	33ml	DoS	317	1097
Kumar	MD5	41s	NjRat	423	983
Hussain	CBSRE	10ms	Brute Force	163	642
Talpur,	SNS	23 s	DoS	435	1002
Proposed	HE	10ms	DDoS	653	2105

5. CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The notion of Internet of Things There have only been a handful of real studies demonstrating practical and implementable Internet of Things security solutions (IoT). Existing cryptographic primitives and algorithms are evaluated on a variety of Internet of Things (IoT) devices in order to determine the degree of their performance limits and memory needs. In the modern day, symmetric ciphers and hash functions may be created rapidly utilizing a small number of devices. These activities can be completed in less than one millisecond using microcontrollers with less than one kilobyte of RAM. Internet of Things-related apps and services have the ability to use modular operations and asymmetric cryptography. 4 kilobytes of random-access memory (RAM) and 10 kilobytes of store memory are required for the devices to function. For computationally constrained devices, such as MSP microcontrollers and smart cards, an RSA signature created with a 2048-bit private key might entail a delay of more than one hundred milliseconds. Apps that need to verify data and transfer it in real time, for instance, are unable to use activities that need a great deal of processing power. Regarding the Internet of Things (IoT), academics are investigating new privacy protection strategies. These techniques include k-anonymity (data privacy), attribute-based signatures (ABR), and attribute-based encryption (AEE). Proof of knowledge and bilinear pairing strategies, as well as public key cryptography, are just a few of the many methods we've identified for protecting the privacy of users. Due to the low RAM available on many of the devices, it may be challenging to implement these privacy answers for the IoT problem set. Some methods of cryptography, such bilinear pairings, modular exponentiation, and point multiplication, often have significant memory and RAM requirements. This is due to the fact that they need other cryptographic libraries to conduct these actions. On ARM devices, pairing is the cryptographic process that consumes the most time and resources. On an ARM system running at 2.26 GHz, a single pairing operation using 175-bit curves takes around two and a half seconds to execute. On devices such as chip cards and ARM processors, it takes tens of milliseconds to several seconds to execute the stages homomorphic encryption group signature schemes attribute-based signature schemes attribute-based encryption and signature-cryptography. These are examples of approaches that protect privacy. For the creation of secure and private Internet of Things applications, a solution that does not rely on time-consuming bilinear pairing methods, provides

short signatures, and can be rapidly deployed even on devices with limited memory resources is needed. [53] There are a number of Internet of Things (IoT) applications where non-pairing attribute-based signature techniques seem to be a feasible means of preserving privacy. Chip cards and SAM modules are two examples of restricted devices that might be used to implement these ideas. However, both approaches need a tamper-proof module, and the signatures they generate may be rather big.

5.2 FUTURE WORK

The unique notion of homomorphic encryption allows the results of operations performed on encrypted data to be disclosed without revealing the plaintext information on which the operations were performed. This safeguards the data from compromise. Within the scope of this work, homomorphic encryption methods are examined. Specify its multiplicative and additive properties and expound on its qualities. Concerns about the security of cloud-stored data must also be addressed. As consumers of cloud computing, we rely on homomorphic encryption as a fundamental notion for protecting our private information. The secret key that may be used to decrypt our computations is nonetheless saved on the cloud server. This permits the server to continue to reside in the cloud. As a result, the Homomorphic approach has the potential to be used to the issue of data and computation security. This method safeguards both the privacy of user information and the data itself. The advantages include greater server utilization, better data security, and enhanced data privacy. The data owner is relieved of the need to do calculations

REFERENCES

[1] G. Ramachandra, M. Iftikhar, and F. A. Khan, “A comprehensive survey on security in cloud computing,” Procedia Comput. Sci., vol. 110, pp. 465–472, 2017.

[2] “Achieving fine-grained access control for secure data sharing on cloud servers,” Concurr. Comput. Pract. Exp., vol. 23, no. 12, pp. 1443–1464, 2011.

[3] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, “Profit-aware application placement for integrated fog–cloud computing environments,” J. Parallel Distrib. Comput., vol. 135, pp. 177–190, 2020.

[4] H. A. K. Al-Shqeerat, F. M. Al-Shrouf, M. R. Hassan, and H. Fajraoui, “Cloud computing security challenges in higher educational institutions-A survey,” Int. J. Comput. Appl., vol. 161, no. 6, pp. 22–29, 2017.

[5] M. Tebaa, S. El Hajji, and A. El Ghazi, “Homomorphic encryption method applied to Cloud Computing,” in 2012 National Days of Network Security and Systems, 2012, pp. 86–89.

[6] B. R. Purushothama and B. B. Amberker, “Efficient query processing on outsourced encrypted data in cloud with privacy preservation,” in 2012 International Symposium on Cloud and Services Computing, 2012, pp. 88–95.

[7] R. V. Rao and K. Selvamani, “Data security challenges and its solutions in cloud computing,” Procedia Comput. Sci., vol. 48, pp. 204–209, 2015.

[8] D. Hyseni, A. Luma, B. Selimi, and B. Cico, “The proposed model to increase security of sensitive data in cloud computing,” Int. J. Adv. Comput. Sci. Appl, vol. 9, no. 2, pp. 203–210, 2018.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in 2010 Proceedings IEEE INFOCOM, 2010, pp. 1–9.

[10] M. M. P. Mr, C. A. Dhote, and D. H. S. Mr, Homomorphic encryption for security of cloud data, vol. 79. Elsevier, 2016.

[11] M. Kaur and R. Singh, “Implementing encryption algorithms to enhance data security of cloud in cloud computing,” Int. J. Comput. Appl., vol. 70, no. 18, 2013.

[12] N. K. Murthy and R. Selvam, “Security issues and challenges in cloud computing,” Int. Adv. Res. J. Sci. Eng. Technol, vol. 2, p. 12, 2015.

[13] Z. Kartit et al., “Applying encryption algorithm for data security in cloud storage,” in International Symposium on Ubiquitous Networking, 2015, pp. 141–154.

[14] Y. Li, K. Gai, L. Qiu, M. Qiu, and H. Zhao, “Intelligent cryptography approach for secure distributed big data storage in cloud computing,” Inf. Sci. (Ny)., vol. 387, pp. 103–115, 2017.

[15] P. Naga Hemanth, N. Abhinay Raj, and N. Yadav, “Secure data transfer by implementing mixed algorithms,” in Recent findings in intelligent computing techniques, Springer, 2019, pp. 79–85.

[16] M. B. Qureshi et al., “Encryption Techniques for Smart Systems Data Security Offloaded to the Cloud,” Symmetry (Basel)., vol. 14, no. 4, p. 695, 2022.

[17] K. V Pradeep, V. Vijayakumar, and V. Subramaniyaswamy, “An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment,” J. Comput. Networks Commun., vol. 2019, 2019.

[18] S. Narayan, M. Gagné, and R. Safavi-Naini, “Privacy preserving EHR system using attribute-based infrastructure,” in Proceedings of the 2010 ACM workshop on Cloud computing security workshop, 2010, pp. 47–52.

[19] A. K. Dubey, A. K. Dubey, M. Namdev, and S. S. Shrivastava, “Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment,” in 2012 CSI Sixth International Conference on Software Engineering (CONSEG), 2012, pp. 1–8.

[20] N. Khanezaei and Z. M. Hanapi, “A framework based on RSA and AES encryption algorithms for cloud computing services,” in 2014 IEEE Conference on Systems, Process and Control (ICSPC 2014), 2014, pp. 58–62.

[21] K. Gai, M. Qiu, H. Zhao, and J. Xiong, “Privacy-aware adaptive data encryption strategy of big data in cloud computing,” in 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), 2016, pp. 273–278.

[22] Avijit, S. Paul, Rm. T. Goswami, and S. Nath, “Cloud computing security issues & challenges: A review,” in 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1–5.

[23] S. Rani and A. Gangal, “Cloud security with encryption using hybrid algorithm and secured endpoints,” Int. J. Comput. Sci. Inf. Technol., vol. 3, pp. 4302–4304, 2012.

[24] E. M. Mohamed, H. S. Abdelkader, and S. El-Etriby, “Enhanced data security model for cloud computing,” in 2012 8th International Conference on Informatics and Systems (INFOS), 2012, pp. CC–12.

[25] R. Kaur and R. P. Singh, “Enhanced cloud computing security and integrity verification via novel encryption techniques,” in 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014, pp. 1227–1233.

[26] F. Zhao, C. Li, and C. F. Liu, “A cloud computing security solution based on fully homomorphic encryption,” in 16th international conference on advanced communication technology, 2014, pp. 485–488.

[27] X. Boyen and B. Waters, “Full-domain subgroup hiding and constant-size group signatures,” in International workshop on public key cryptography, 2007, pp. 1–15.

[28] J. López, A. Maña, and A. Muñoz, “A secure and auto-configurable environment for mobile agents in ubiquitous computing scenarios,” in International Conference on Ubiquitous Intelligence and Computing, 2006, pp. 977–987.

[29] A. Muñoz, A. Maña, and J. González, “Dynamic security properties monitoring architecture for cloud computing,” in Security Engineering for Cloud Computing: Approaches and Tools, IGI Global, 2013, pp. 1–18.

[30] S. Hussain et al., “A lightweight and formally secure certificate based signcryption with proxy re-encryption (CBSRE) for Internet of Things enabled smart grid,” IEEE Access, vol. 8, pp. 93230–93248, 2020.

[31] M. Uddin, A. Khalique, A. K. Jumani, S. S. Ullah, and S. Hussain, “Next-generation blockchain-enabled virtualized cloud security solutions: review and open challenges,” Electronics, vol. 10, no. 20, p. 2493, 2021.

[32] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, “EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 9, pp. 1621–1631, 2012.

[33] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” in Annual international cryptology conference, 2004, pp. 41–55.

[34] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the Decryption of {ABE} Ciphertexts,” 2011.

[35] X.-J. Lin, L. Sun, and H. Qu, “Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications,” Comput. Secur., vol. 48, pp. 142–149, 2015.

[36] B. T. Rao, “A study on data storage security issues in cloud computing,” Procedia Comput. Sci., vol. 92, pp. 128–135, 2016.

[37] S. Sicari, L. A. Grieco, G. Boggia, and A. Coen-Porisini, “DyDAP: A dynamic data aggregation scheme for privacy aware wireless sensor networks,” J. Syst. Softw., vol. 85, no. 1, pp. 152–166, 2012.

[38] K. El Defrawy and G. Tsudik, “Prism: Privacy-friendly routing in suspicious manets (and vanets),” in 2008 IEEE International Conference on Network Protocols, 2008, pp. 258–267.

[39] J.-S. Coron, D. Naccache, and M. Tibouchi, “Public key compression and modulus switching for fully homomorphic encryption over the integers,” in Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2012, pp. 446–464.

[40] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89–98.

[41] A. Wasef and X. Shen, “Efficient group signature scheme supporting batch verification for securing vehicular networks,” in 2010 IEEE International Conference on Communications, 2010, pp. 1–5.

[42] C. Paquin and G. Zaverucha, “U-prove cryptographic specification v1. 1,” Tech. Report, Microsoft Corp., 2011.

[43] D. Chaum and E. van Heyst, “Group signatures,” in Workshop on the Theory and Application of of Cryptographic Techniques, 1991, pp. 257–265.

[44] F. Li, Z. Zheng, and C. Jin, “Secure and efficient data transmission in the Internet of Things,” Telecommun. Syst., vol. 62, no. 1, pp. 111–122, 2016.

[45] B. Libert, T. Peters, and M. Yung, “Group signatures with almost-for-free revocation,” Annual Cryptology Conference. Springer, pp. 571–589, 2012.

[46] A. Alcaide, E. Palomar, J. Montero-Castillo, and A. Ribagorda, “Anonymous authentication for privacy-preserving IoT target-driven applications,” Comput. Secur., vol. 37, pp. 111–123, 2013.

[47] J. Sen, “Secure and Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks,” Cryptogr. Secur. Comput., vol. 2012, no. 3, pp. 133–164, 2012.

[48] J. Camenisch and E. Van Herreweghen, “Design and implementation of the idemix anonymous credential system,” in Proceedings of the 9th ACM Conference on Computer and Communications Security, 2002, pp. 21–30.

[49] M. S. H. Talpur, M. Z. A. Bhuiyan, and G. Wang, “Shared–node IoT network architecture with ubiquitous homomorphic encryption for healthcare monitoring,” Int. J. Embed. Syst., vol. 7, no. 1, pp. 43–54, 2015.

[50] D. S. AbdElminaam, “Improving the security of cloud computing by building new hybrid cryptography algorithms,” Int. J. Electron. Inf. Eng., vol. 8, no. 1, pp. 40–48, 2018.

[51] S. Hussain, S. S. Ullah, M. Uddin, J. Iqbal, and C.-L. Chen, “A comprehensive survey on signcryption security mechanisms in wireless body area networks,” Sensors, vol. 22, no. 3, p. 1072, 2022.

[52] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, “TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things,” IEEE Trans. Mob. Comput., 2021.

[53] D. Das, “Secure cloud computing algorithm using homomorphic encryption and multi-party computation,” in 2018 International Conference on Information Networking (ICOIN), 2018, pp. 391–396.