T.C.
USKUDAR UNIVERSITY
INSTITUTE OF SCIENCE

DEPARTMENT OF CYBER SECURITY

MASTER'S DEGREE PROGRAM OF CYBER SECURITY

MASTER'S DEGREE THESIS

SECURING PEER-TO-PEER COMMUNICATION BASED ON TCP/IP STRUCTURE AND MODIFIED DIFFIE-HELLMAN MODEL

Albaraa I. A. Salama

Thesis Advisor

Dr. Ihab Elaff

ISTANBUL-2023

T.C.
USKUDAR UNIVERSITY
INSTITUTE OF SCIENCE

DEPARTMENT OF CYBER SECURITY
MASTER'S DEGREE PROGRAM OF CYBER SECURITY

MASTER'S DEGREE THESIS

SECURING PEER-TO-PEER COMMUNICATION BASED ON TCP/IP STRUCTURE AND MODIFIED DIFFIE-HELLMAN MODEL

Albaraa I. A. Salama

Thesis Advisor

Dr. Ihab Elaff

ISTANBUL-2023

ABSTRACT

Securing peer-to-peer communication based on TCP/IP structure and modified Diffie-Hellman model

TCP/IP protokolündeki mevcut zayıflıklara rağmen evrensel olarak benimsenmeyi başardı ve bu da onu öngörülebilir gelecekte yeri doldurulamaz hale getirdi. Bu makale, Diffie-Hellman'ın geliştirilmiş bir versiyonunu ve AES algoritmasını dahil ederek TCP/IP modelinin güvenliğini arttırmayı amaçlamaktadır. Bu geliştirilmiş Diffie-Hellman sürümü, Diffie-Hellman'ın paylaşılan gizli anahtarını AES algoritması için giriş anahtarı olarak kullanarak genel değerlerin korunmasını sağlar. Bu değişiklikler, model için yüksek güvenliği garanti ederken, TCP başlığındaki mevcut ek yükleri yeniden kullanarak ve veri alanını ek bilgilerle genişleterek ek işlem süresini en aza indirir.

Keywords: modified Diffie-hellman, protected key, peer-to-peer, security

ABSTRACT

Securing peer-to-peer communication based on TCP/IP structure and modified Diffie-Hellman model

Despite the existing weaknesses within the TCP/IP protocol, it has achieved universal adoption, making it irreplaceable for the foreseeable future. This paper aims to enhance the security of the TCP/IP model by including an enhanced version of Diffie-Hellman, along with AES algorithm. This enhanced Diffie-Hellman version ensures the protection of public values, utilizing the shared secret key of Diffie-Hellman as the input key for the AES algorithm. These modifications guarantee heightened security for the model, all while minimizing additional transaction time by reusing existing overheads within the TCP header and expanding the data field with supplementary information.

Keywords: modified Diffie-hellman, protected key, peer-to-peer, security

THANKS TO

I would like to extend my heartfelt appreciation to my dedicated advisor, Dr. Ihab Elaff, for their invaluable guidance and unwavering support throughout my thesis journey. Their expertise and mentorship were instrumental in shaping the quality of my research, and their patience and encouragement were truly motivating. I am deeply grateful for their constructive feedback and for believing in my abilities.

I also want to express my profound gratitude to my family for their constant encouragement, love, and understanding. Their unwavering support has been my rock, and I couldn't have accomplished this without them.

This thesis is a result of the collective support and encouragement of these wonderful individuals, and I am truly fortunate to have each of you in my life.

FORM OF DECLARATION

Herewith I declare, that I obtained all the information and documents in this study within the framework of academic rules, presented all visual, auditory, and written information and results in accordance with scientific ethics, did not falsify the data I used, referred to the sources I used in accordance with scientific norms, that my thesis was original except in the cases cited, produced by me and written in accordance with the Thesis Writing Guide of Uskudar University Institute of Sciences.

Date

Student's Name and SURNAME

Signature

CONTENTS

INDEX OF TABLES

Page

INDEX OF FIGURES

Page

INDEX OF IMAGERY AND ABBREVIATIONS

DES : Data Encryption Standard

AES : Advanced Encryption Standard

PIN : Personal Identification Number

DH : Diffie-Hellman

RSA : Rivest, Shamir, and Adleman

LAN : Local Area Network

MAN : Metropolitan Area Network

WAN : Wide Area Network

PAN : Personal Area Network

WLAN: Wireless Local Area Network

AP : Access Point

SAN : Storage Area Network

CAN : Campus Area Network

EPN : Enterprise Private Network

VPN : Virtual Private Network

MAC : Media Access Control

TCP : Transaction Control Protocol

IP : Internet Protocol

HTTP : Hypertext Transfer Protocol

FTP : File Transfer Protocol

SMTP : Simple Message Transfer Protocol

DNS : Domain Name System

DHCP : Dynamic Host Configuration Protocol

UDP : User Datagram Protocol

ICMP : Internet Control Message Protocol

IGP : Internet Gateway Protocol

EGP : Exterior Gateway Protocol

BGP : Border Gateway Protocol

SLIP : Serial Line Internet Protocol

PPP : Point-to-point Protocol

MTU : Maximum Transmission Unit

MSS : Maximum Segment Size

ALSP : Application Layer Security Protocol

1. CRYPTOGRAPHY

1.1. Introduction

Using the shortest way, and as what Hans Delfs, Helmut Knebl said in their book, Introduction to Cryptography: Principles and Applications, Cryptography is the science of keeping secrets secret (Delfs & Knebl, 2016, p. 22). Using a little bit more words, Cryptography is the practice and study of mathematical techniques for securing communications, systems and distributed computations in the presence of third parties to allow only the sender and intended recipient of a message to view its contents.

As Arto Salomaa says in his book, Public-key Cryptography, Cryptography is continuing struggle between two worlds, there is the world of legal communications, there is also the dark world of the enemy who illegally tries to intercept the messages and do all kinds of vicious things. For people in the legal world, it is desirable that the enemy understands very little of the messages. The enemy, on the other hand, would like to have easily understandable messages (Salomaa, 2013).

Unfortunately we are living in a world that we need to protect ourselves and our belongings from the dark world members, cryptography from a long time till now, is providing new technologies in order to protect every protectable asset

1.2. History

As Cryptology has a fascinating history that spans ancient times to the modern era. It flourished in ancient Greece and Rome, but was lost in the West until the Renaissance, while thriving in the Arabic world. The Arabs introduced frequency analysis, a powerful cryptanalysis tool. During the Renaissance, cryptology experienced a revival in the West, focusing on the monoalphabetic substitution cipher and frequency analysis techniques. The period from 1500 to the mid-18th century witnessed the emergence of modern nations, increased use of codes and ciphers, and the creation of the unbreakable Vigenère cipher.

The American Revolution marked a pivotal point for cryptology, as the new nation had to create its own secret writing systems. Both the Americans and the British lacked sophistication in cryptography and steganography, leading to a learning process filled with successes and failures. The advent of technology in the 19th century brought significant advancements, including the telegraph and the American Civil War, which fueled rapid development in cipher systems. The breaking of the supposedly unbreakable Vigenère cipher was a notable achievement. World War I introduced wireless telegraphy and machine cryptography, leading to the establishment of formal cryptanalytic organizations and the contributions of figures like Herbert Yardley and William Friedman.

Between the World Wars, Yardley and Friedman played instrumental roles in establishing permanent cryptographers and cryptanalysts, ensuring the United States had a solid cryptologic infrastructure. The era also witnessed the rise of electromechanical cipher machines, such as Enigma and M-134C/SIGABA, which played a crucial role in cryptography's progression into the computer age. World War II intensified efforts to break sophisticated cipher machines.

The contributions of William Friedman, Lester Hill, and Claude Shannon propelled cryptology into the mathematical and statistical realm. This paved the way for new algorithms in digital encryption. The Data Encryption Standard (DES), and the Advanced Encryption Standard (AES), played significant roles in the 20th century. Public-key cryptography emerged as a solution to the key exchange problem, revolutionizing the field by dividing keys into public and private parts. This enabled secure communication without the need for a secure key distribution method (Dooley, 2018).

1.3. The purpose of cryptography

There are some specific security requirements within the context of any application-to-application communication that the security should provide for protecting whatever needs to be protected within unsecured channels such as confidentiality, integrity, authentication, and non-repudiation (Sharma & Gupta, 2017).

- Confidentiality: ensuring that no one can read the message except the intended receiver, by, preserving authorized restrictions on information access and disclosures.
- Integrity: to make the receiver be sure that the received message has not been altered in any way from the original. How the sender sent the message, how the receiver received it.
- Authentication: the process of providing one's identity to assure that the communicating entity is the one that it claims to be, like the server needs to know exactly who is accessing their information or site, and a client needs to know that the server is system it claims to be.

  Authentication three factors
  - Something you know: Password, personal identification number (PIN), address.
  - Something you have: Token such as a bank card.
  - Something you are: Fingerprints and voice recognition.
- Non-repudiation: to prove that the sender is really sent this message.

1.4. Basic terminology

The scientific study of any discipline must be built upon rigorous definitions arising from fundamental concepts. Here is a list of terms that you should know for better understanding and a more clean reading (AspEncrypt.com - Crypto 101: Basic Terminology, n.d.).

- Cryptography is the science of making communications secure.
- Cryptoanalysis is the science of breaking ciphertext without knowing the key.
- Cryptology is considered to be a branch of mathematics that deals with both cryptography and cryptoanalysis.
- Plaintext is an understandable raw text message.
- Ciphertext is a data stream which looks like a meaningless and random sequence of bits.
- Encryption is converting the plaintext into a ciphertext.
- Decryption is converting back the ciphertext into plaintext again.
- Key is a piece of information that is used in encryption and decryption algorithms.
- Symmetric algorithm is an algorithm that uses only one key to encrypt and decrypt ciphertext. In this algorithm, both of the sender and the receiver should agree on a key before exchanging messages securely.
- Stream cipher are used in some symmetric algorithms that works with 1 bit or 1 byte of plaintext at a time.
- Block cipher are used in another algorithms that works on blocks of bits at a time.
- Asymmetric algorithm is an algorithm that uses a pair of keys (public and private) for encryption and decryption. As the public key is public, anyone can

use it in order to encrypt messages, but only the holder of the private key could decrypt it.

- Hash is a one-way function that converts a message into a fixed-size string of characters, it is usually unique to the original message.
- Digital Signature is a method to ensure the authenticity and integrity of a digital message by using both encryption and hashing.

1.5. Types of cryptographic algorithms

There are several ways of classifying cryptographic algorithm, the most famous categorizing methodology is based on the number of keys that are needed for the encryption and decryption.

In hash functions, there is no need for a key since the plaintext is not recoverable from the ciphertext. For the symmetric encryption, a single private key is needed, but for the asymmetric encryption, there are a pair of keys, a private key and a public one (Kessler, 2003).

In this section, the symmetric and the asymmetric methods are going to be discussed.

1.5.1. Symmetric encryption

This is the simplest kind of encryption that involves only one secret key to cipher and decipher information.

The key is like a special tool that turns a regular message called "plaintext" into a secret code that nobody can understand unless they have the key. This process is called "encryption," and the secret code is called the "cipher text. (Stinson & Paterson, 2023).

The encryption key and the decryption key are the same in most of symmetric algorithms. These algorithms, also known as secret-key algorithms, single-key algorithm, or one-key algorithms. These algorithms require that both of the sender and the receiver to agree on a key before they can communicate securely. DES, AES, Blowfish, RC4, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithms are AES-128, AES-192, and AES-256 (Schneier, 1996).

The main disadvantage of the symmetric key encryption is that all parties involved have to exchange the key used to encrypt the data before they can decrypt it.

1.5.2. Asymmetric encryption

Asymmetric encryption or public key cryptography, which is a relatively new method, compared to symmetric encryption. Asymmetric encryption uses two keys to encrypt a plain text.

A public key would be used to encrypt the plaintext and a private key would be used to decrypt the cipher. Note that everyone could know the public key, whereas only one person could know the private key (Stinson & Paterson, 2023).

Asymmetric encryption is mostly used in day-to-day communication channels. Especially over the internet. Popular asymmetric key encryption algorithm includes Diffie-Hellman, RSA, EIGamal, DSA, Elliptic curve techniques, and PKC5.

1.5.3. Data encryption standard (DES)

The DES algorithm considered to be symmetric algorithm which can be described as follows. 64-bits length plaintext and 56-bits length key; much longer plaintext amounts are processed as in blocks of 64-bit length.

Decryption with DES is considered to be as the same as the encryption process but in reversed manner, from k16, k15, until k1 is used on the final iteration (Stallings, 2015).

The DES algorithm work flow is going to be described as in three major steps, firstly by processing the key, and finally by performing the inverse permutation.

- Process the key

  In order to process the key, and as the first step, the user provides a 64-bit key, it can be entered directly or the key could be the result of hashing something.

  Each byte includes a parity bit, specifically the least significant bit (the 8th bit), which serves the purpose of ensuring correct parity. To maintain proper parity in a key, every byte should have an odd number of '1' bits.

  As the second step, the algorithm calculates the key schedule by perform a permutation on the 64-bit key in order to reduce the key size to 56-bit key. The key schedule depends on the parity bits to be discarded.

  Look at figure 1.1 to see how the permuted choice 1 block converted a 64-bit key to 56-bit key by discarding the parity bits.

Input Key							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Permuted Choice One (PC-1)						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Figure 1.1: Key sample and the permuted choice 1 block

Notice that the first bit, 'bit 1' (the most significant bit), in the permuted block corresponds to 'bit 57' in the original key, 'bit 2' aligns with 'bit 49,' and so forth.

In the third step, the DES algorithm splits the permuted key into two parts, the first part ‘C0’ is the first 28-bits, and the rest 28-bit as the second part ‘D0’.

Finally, as the fourth step of processing the key, it is calculating or generating all the needed sub key for the entire encryption process (16 sub keys). Each sub key will be used in each iteration of the encryption/decryption process. The process of generating each iteration sub key is done by two steps

- Based on the left shift schedule, conduct either one or two shifts for both Ci-1 and Di-1 to obtain Ci and Di. The required number of left shifts depends on the schedule illustrated in Figure 1.2.

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure 2.2: Left shift schedule

– Permute the concatenation between Ci Di. This step, and after using the permuted choice 2 block shown in figure 1.3, the DES will produce the iteration key result which is 48-bit long.

Permuted Choice Two (PC-2)							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Figure 3.3: Permuted choice 2 block

- Process the 64-bit data block

The user's intended message for transmission will be segmented into 64-bit data blocks. If a data block's length is less than 64 bits, it must be suitably padded to meet the 64-bit requirement for the application. Once the data block has been adjusted to a 64-bit format, proceed with the initial permutation using the table shown in Figure 1.4.

Initial Permutation (IP)							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Figure 4.4: Initial Permutation block

Then, splitting the result into two halves, the first 32-bit half is called ‘L0’, and the second 32-bit half is called ‘R0’. Finally, applying the 16 rounds starting with i=0. The process of figuring each round is done by eight steps.

– Expand the 32-bit Ri-1 into 48-bit.

The expansion is done according to the bit selection function using expansion permutation table, the expansion permutation table is shown in figure 1.5.

Expansion Permutation					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Figure 5.5: Expansion permutation table

– Xi=E(Ri-1)XOR Ki

The Xi shown in the previous equation is the 48-bit result data block, E(Ri-1) is the expanded right half of the data block, and the Ki is the sub key generated previously (iteration key).

– Xi into eight 6-bit blocks (S-Boxes)

In figure 1.6, there is 8 blocks, each block is 6-bits long, and these blocks together is called the S-Boxes. Bits 1-6 are B1, bits 7-12 are B2… bits 43-48 are B8.

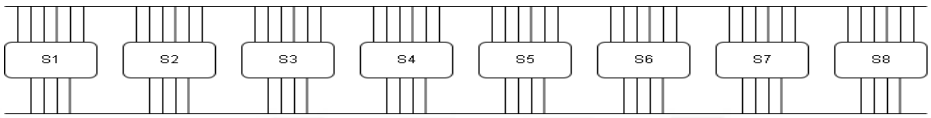

Figure 6.6: S-boxes block

– Substitute the values found in the S-Boxes for all Bj

S1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	13	15	4	2	11	6	7	12	0	5	14	9

S6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Figure 7.7: The results from s1 till s8

Start with j = 1 to 8. All values in the S-boxes should be considered 4 bits wide, then take the 1st and 6th bits of Bj together as 2-bit value and call it m, indicating the row in Sj to look in the substitution.

Take the 2nd through 5th bits of Bj together as 4-bit value and call it n indicating the column in Sj to find the substitution. Finally replace Bj with Sj [m] [n], loop back to 1 until all 8 blocks have been replaced.

Ex: B1 = 011101 n = (01)2, m = (1110)2

m	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	n
S1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	00
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	01
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	10
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	11

Figure 8.8: Example result

– Permute the concatenation of B1 through B8 as indicated below to P.

Permutation Function							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Figure 9.9: Permutation function

– Find Ri = Li-1 XOR P

– Assign Ki = Ri-1

– Loop back from step 4 until K16 has been applied

- Perform the inverse permutation on the block R16 L16

  As the Final step for the DES to complete its encryption process, left and right blocks are swapped.

1.5.4. Advanced encryption standard (AES)

The previously mentioned standard, DES, lost its efficacy in ensuring security. The increasing computational power of modern computers rendered the algorithm insufficient for security purposes. In 1998, a specialized computer known as the DES cracker successfully broke the DES algorithm in under three days. This computer was purpose-built for the sole purpose of cracking DES. (Selent, D., 2010).

November 26, 2001, marked the announcement of the Federal Information Processing Standards Publication 197, which introduced a standardized version of the Rijndael algorithm as the new encryption standard. This standard was named the Advanced Encryption Standard (AES) and remains the prevailing encryption standard today. (FIPS, 2001).

AES employs a block size of 128 bits and offers key lengths of 128, 192, or 256 bits (Stallings, 2015). In the following section, we will delve into the AES workflow, which can be broken down into three broad steps.

- Round 0

  In the first round, the AES algorithm will use the original key for the encryption process, the AES output after the first round will be the state matrix XOR the round key (in this round, the key to be used is the original key).

- Round 1 to N-1

  Each round from round 1 till round N-1 has to go through 4 steps named as byte substitution, shift rows, mix columns, and adding round key.

  – Byte substitution

  Byte substitution is done by combining the AES S-box matrix with the state matrix figured in first round. AES S-box is shown in figure 10.

AES S-Box																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	3W	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	62	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 10.10: AES S-box

  – Shift rows

  Here, the AES algorithm shifts each row within the state matrix depending on the row number, row 1 is shifted by 0, row 2 is shifted by 1, and so on. Look at figure 11 that explains the shifting mechanism.

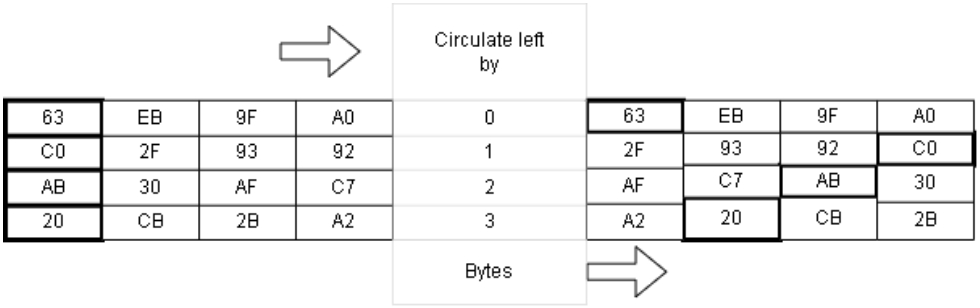

Figure 11.11: Row shifting depending on the row number

– Mix columns

In this step, matrix multiplication will be used, first row from the first matrix multiplied by first column from second matrix. As the result, each byte is replaced by a value depending on all 4 bytes in the column. Look at figure 1.12 to understand how the matrix multiplication works, and underneath, there is an equation explains the multiplication process.

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

X

63	EB	9F	A0
2F	93	92	C0
AF	C7	AB	30
A2	20	CB	2B

Figure 12.12: Matrix multiplication

Notice that bytes are treated as a polynomials rather than numbers. The following example equations describes how bytes are represented as polynomials for the previous matrixes.

(02 × 63) ⊕ (03 × 2F) ⊕ (01 × AF) ⊕ (01 × A2)
02 = 0000 0000 = X
63 = 0110 0011 = X^6 + X^5 + X + 1
02 × 63 = X^7 + X^6 + X^2 + X = 1100 0110

03 × 2F = X^6 + X^5 + X^4 + 1 = 0111 0001
01 ×AF = X^7 + X^5 + X^3 + X^2 + X + 1 = 1010 1111
01 × A2 = X^7 + X^5 + X = 1010 0010

As the final step in mixing columns, the AES algorithm does an XOR for the result provided from matrix multiplication. The following equation describes the XOR operation for the previous result.

02×63=1100 0110
03×2F=0111 0001
⊕
01×AF=1010 1111
01×A2=1010 0010

=BA

– Add round key

In the last step, for figuring the next round key, the AES algorithm does a 3 more steps for the previously preserved result (State Matrix ⊕ Round1 key).

- Divide the state key into four parts, $W[0]$ till $W[3]$, each part contains 4 bytes.
  $W[0] = (54, 68, 61, 74)$
  $W[1] = (73, 20, 6D, 79)$

$W[2] = (20,4B,75,6E)$
$W[3] = (67.20.46,75)$

- Calculate $G(W[3])$
  Initially, the AES algorithm performs a circular left shift on$W[3] = (12,46,75,67)$, followed by byte substitution using the AES S-box, which yields $(B7,5A,9D,85)$. Lastly, a round constant, $G(W[3]) + (01,00,00,00)$ is added, resulting in $(B6,5A,9D,85)$.

- Figure $W[4]$ till $W[7]$.
  $W[4] = W[0] \oplus \mathrm{G(W[3])} = (\mathrm{E2,32,FC,F1})$
  $W[5] = W[4] \oplus \mathrm{W[1]} = (91,12,91,88)$
  $W[6] = W[5] \oplus W[2] = (B1,59,E4,E6)$
  $W[7] = W[6] \oplus W[3] = (D6,79,A2,93)$
  $KEY = E2\ 32\ FC\ F1\ 91\ 12\ 91\ 88\ B1\ 59\ E4\ E6\ D6\ 79\ A2\ 93$

- Round N
  The last round is a little bit different from the previous rounds, there is no mixing columns here, there are byte substitution, shifting rows, and adding the last round key (K10).

1.5.5. Diffie-Hellman

Whitfield Diffie and Martin Hellman discovered what is now known as the Diffie-Hellman (DH) algorithm in 1976.which is a method for securely exchanging a shared secret key between two parties over an untrusted network.

The shared secret is important between two parties who may not have ever communicated previously, so that they can encrypt their communications easily (Carts, 2001).

The Diffie-Hellman key exchange flow is going to be described as in couple of steps from agreeing on large prime numbers till calculating the shared secret key. Look at figure 1.13 to see the Diffie-Hellman key exchange diagram.

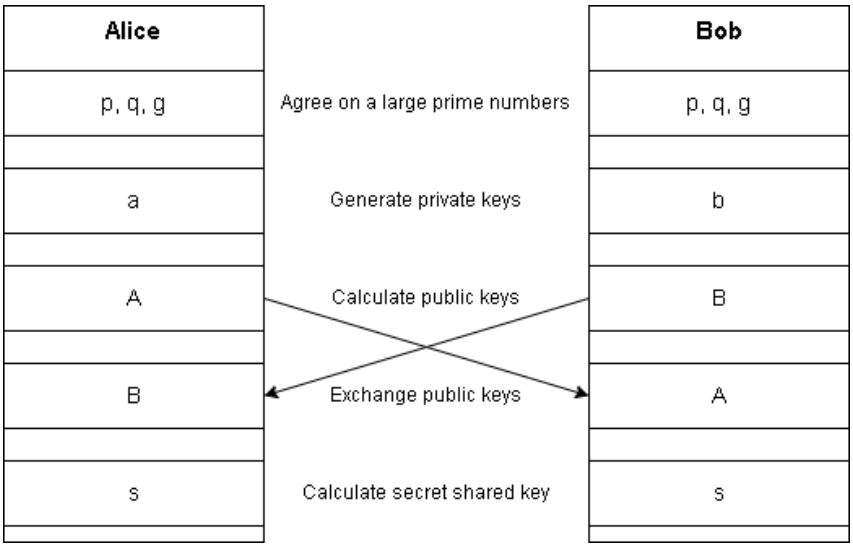

Figure 13.13: Diffie-Hellman key exchange diagram

As shown in figure 1.13, there are two parties want to exchange keys in order to calculate a secret shared key. The Diffie-Hellman key exchange process (Rescorla, 1999) goes through 5 steps. In this section, the key exchange flow will be described.

- Agreeing on the large primes

  Both parties agree to use a large prime number ‘p’, another large prime ‘q’, and a generator ‘g’ where g=h^((p-1)/q) mod p, h is any integer with 1<h<p-1.

- Generating private keys

  Each party randomly generate its own private key, for the first party ‘Alice’, ‘a’ is the private key, For ‘Bob’, the second party private key would be ‘b’.

- Calculating public keys

  Using the previously mentioned keys, each party calculates its own public key using the following formula, for Alice, the formula would be $A = g^a mod\ p$, for Bob, $B = g^b mod\ p$ is the suitable formula.

- Exchanging public keys

  Each party shares her public key with the other party, Alice sends his public key ‘A’ to Bob, Bob does the same by sending his public key ‘B’ to Alice. In this step, in case of eavesdropper existing, these public keys would be known for him.

- Calculating the secret shared key

  Both parties can compute the shared secret key 's' using the other party's public key and their own private key, following the formula outlined below. Taking Alice as an example, the formula is expressed as $s = B^a mod\ p$.

1.5.6. RSA

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman introduced a cryptographic algorithm known as RSA. RSA is a public-key cryptosystem and also supports digital signatures. It was inspired by the earlier work of Diffie and Hellman from several years prior. In RSA, public keys are used for encryption, while private keys are used for decryption. This means that only the owner of the private key can decrypt the cipher (Milanov, 2009). The RSA algorithm is illustrated in Figure 1.14 (Stallings, 2015).

Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

Figure 14.14: The RSA algorithm

Also in (Goshwe, 2013), a detailed description for the RSA algorithm is provided in a couple of steps.

- Select two large prime numbers, p and q, and using both of prime numbers, it will be easy to calculate the modulus $n = pq$.
- Select the public exponent ‘e’ that is relatively prime to the product$(p-1)(q-1)$.
- Calculate the private key ‘d’ using the following equation$(ed-1)/((p-1)p-1)$.
- The public key is the number pair (n,e). I is extremely hard to figure the private number ‘d’ from the modulus ‘n’ and the exponent ‘e’ if p and q are large enough.
- Encrypting the message ‘M’ using the public key creates the ciphertext ‘C’, the encryption is done by using the following equation $C = M^e mod\ n$.
- The receiver then now can decrypt the ciphertext using the private key depending on following equation $M = C^d mod\ n$.

2. NETWORKING

2.1. Introduction

Computer network is a collection of large amount of computers which they are connected by a single technology. Computers are considered to be connected if they are able to make communications (Wetherall and Tanenbaum, 2013).

For an effective communication within a computer network, three essential elements must be in place. Firstly, there must exist two distinct entities: the sender and the receiver. These entities should possess content or information to exchange between them. Secondly, there should be a transmission medium or channel connecting these entities, facilitating the transfer of the shareable items from the sender to the receiver. Finally, a well-defined set of communication rules or protocols agreed upon by the entities plays a crucial role. When these components are combined, they form the foundational structure of communication within the network (Kizza, 2013).

In this chapter, network types, network topologies, network devices, and network protocols will be briefly described.

2.2. Types of network

Computer networks play a critical role in modern communication and information exchange. They enable computers and other devices to share resources, exchange data, and collaborate in real-time.

With several types of computer networks available, each with its distinct characteristics and use cases. Figure 2.1 shows the network types.

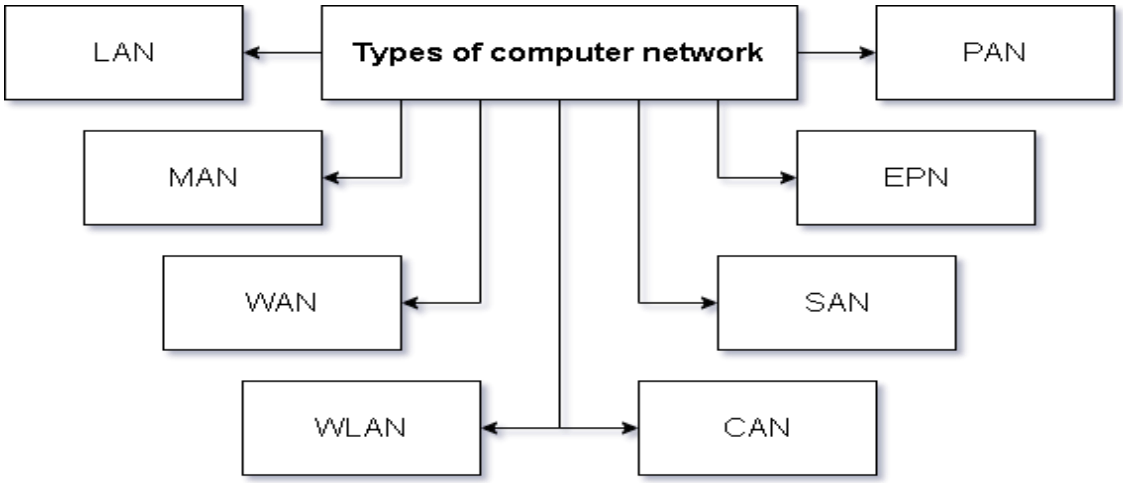

Figure 2.1: Network types

As figure 2.1 shows, there are several types of computer networks. In this sections each network type is going to be illustrated.

- Local Area Network (LAN)

  LANs are popular among small and medium-sized organizations due to their ease of setup, maintenance, and security. They can be established using various networking technologies, such as Wi-Fi, Bluetooth, or Ethernet.

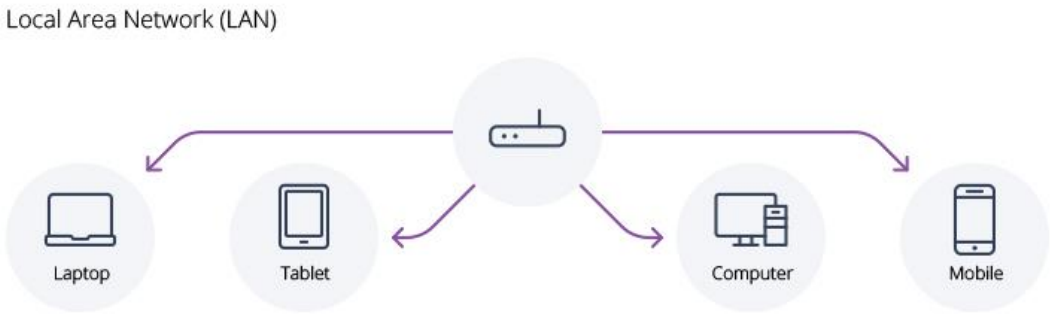

Figure 2.2: Local area network (LAN) (Petryschuk & Petryschuk, 2023)

One of the key advantages of a LAN is its speed, as data transfer rates are typically much faster than those of WANs. Additionally, LANs are more affordable than other network types, as they require minimal cabling and can be easily expanded to accommodate new devices. Furthermore, LANs

are more secure than WANs, as they are typically confined to a single location and access to the network can be restricted through the implementation of security protocols like firewalls and access controls (Tanenbaum & Wetherall, 2013, p. 19).

- Metropolitan Area Network (MAN)

  MAN network covers a larger geographical area than a LAN but is smaller in scale compared to a WAN. Usually, a MAN interconnects computers and devices within a city or town and is frequently employed by institutions operating across multiple locations, such as local governments, hospitals, or universities.

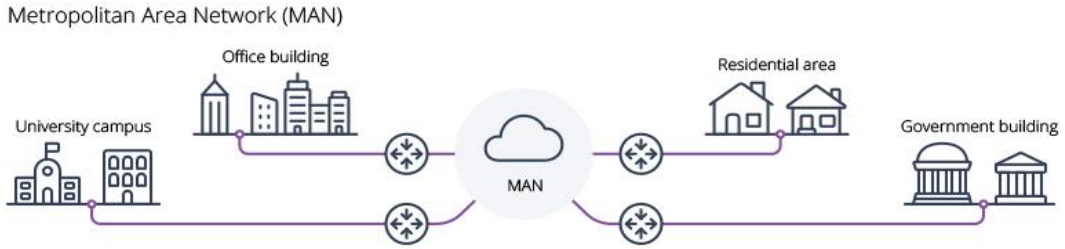

Figure 2.3: Metropolitan area network (MAN) (Petryschuk & Petryschuk, 2023)

Various networking technologies can be employed to establish a MAN, including fiber-optic cables, microwave links, or Wi-Fi. Like LANs, MANs can provide users with high-speed connectivity, security, and reliability. However, setting up and maintaining a MAN can be more complex and costly than a LAN.

Despite its higher setup and maintenance costs, a MAN can be an ideal networking solution for organizations that require high-speed connectivity and reliable communication across multiple locations in a geographic area (Tanenbaum & Wetherall, 2013, p. 23).

- Wide Area Network (WAN)

  The A WAN, short for Wide Area Network, is a network that connects computers and devices over a vast geographic area, such as a continent, or even the world. Its primary purpose is to connect multiple LANs or other networks across long distances. This can be achieved using various technologies, including leased lines, satellite links, or the internet. WAN

users can access resources and applications from anywhere in the world, making it an essential tool for businesses and organizations with global reach.

Figure 2.4: Wide area network (WAN) (Petryschuk & Petryschuk, 2023)

While offering advantages such as security, reliability, and scalability, a WAN is also more expensive and complex to set up than other types of computer networking. However, the cost and complexity are often justified by the significant benefits that a WAN provides, including the ability to connect remote locations and enable communication and data exchange on a global scale (Tanenbaum & Wetherall, 2013, p. 23).

- Personal Area Network (PAN)

  A PAN is a type of network that enables communication between personal devices, such as laptops, and smartphones within a short range. The primary purpose of a PAN is to enable data transfer between personal devices without the need for external networks.

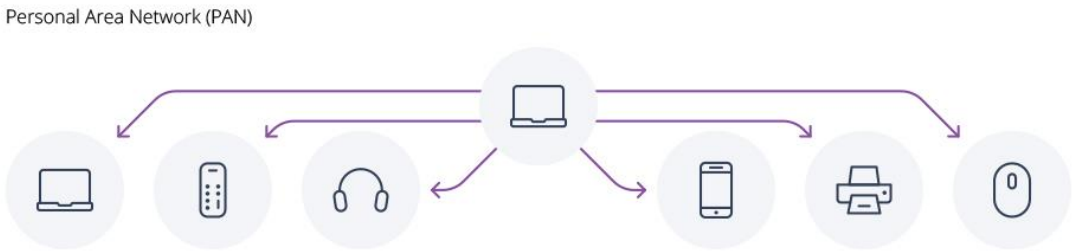

Figure 2.5: Personal area network (PAN) (Petryschuk & Petryschuk, 2023)

A PAN is becoming very popular in modern technology, especially in smart homes and wearable devices, enabling users to control and interact with their devices in a more intuitive and convenient way (Tanenbaum & Wetherall, 2013, p. 18).

- Wireless Local Area Network (WLAN)

  A WLAN is a type of LAN that uses wireless technology, such as Wi-Fi, to connect devices to the network without the need for physical cables. By using wireless signals to transmit data between devices, WLANs offer greater flexibility to network users by allowing them to access resources and applications from anywhere within the range of the wireless network.

  Each computer is equipped with a radio modem and an antenna for communication with other computers. Each computer communicates with a device known as an 'AP' (Access Point), wireless router, or base station. This device serves as an intermediary, relaying packets between the wireless computers and facilitating their connection to the internet. (Tanenbaum & Wetherall, 2013, p. 19).

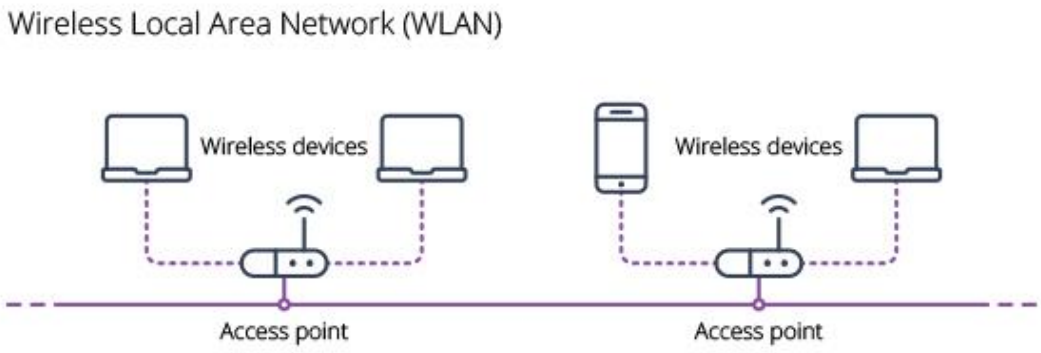

Figure 2.6: Wireless local area network (WLAN) (Petryschuk & Petryschuk, 2023)

Despite their many benefits, WLANs can also present security risks, as wireless signals can be intercepted by unauthorized users. Therefore, implementing security protocols, such as encryption and access control, is essential to ensure the privacy and security of the WLAN.

- Storage Area Network (SAN)

  A SAN is a specialized type of network that is designed to connect storage devices, such as disk arrays to servers. The primary purpose of a SAN is to enable servers to access shared storage resources quickly and efficiently, enabling faster data transfer, backup, and recovery operations (Telikepalli & Drwiega & Yan, 2004).

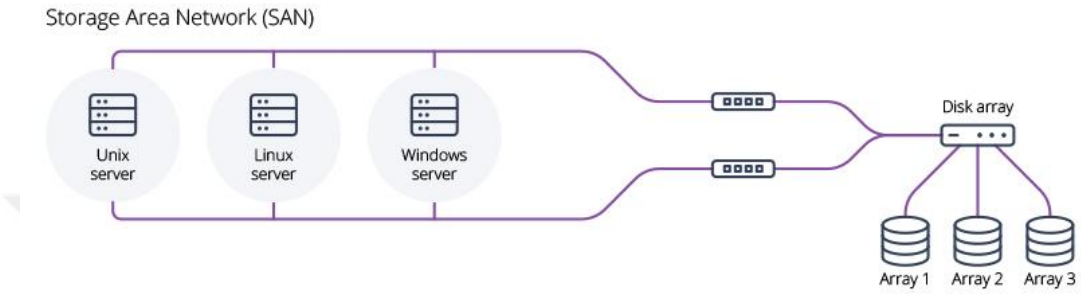

Figure 2.7: Storage area network (SAN) (Petryschuk & Petryschuk, 2023)

- Campus Area Network (CAN)

  A CAN is a network that is used for connecting multiple LANs within a university campus or other large institution.

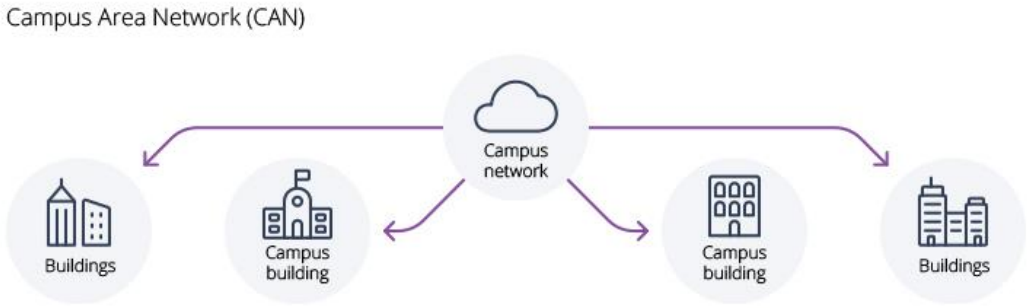

Figure 2.8: Campus area network (CAN) (Petryschuk & Petryschuk, 2023)

- Enterprise Private Network (EPN)

  An EPN is a private network that is used by an organization for its internal communication needs. It is typically more secure and reliable than public networks like the internet.

Figure 2.9: Enterprise private network (EPN) (Petryschuk & Petryschuk, 2023)

2.3. Network topologies

It describes how devices and nodes in a network are connected and how they communicate with one another. There are several types of network topologies, including:

- Bus topology

  Bus Topology is a network configuration where every computer and network device links to a single cable. This forms a multi-point connection and a less robust topology because a backbone failure can lead to a network breakdown. When a sending node transmits a signal, all devices on the same bus receive it. This topology offers easy installation and demands less cabling compared to other arrangements. It suits small networks and is effective in a peer-to-peer setting. Furthermore, if one device fails, it doesn't impact the rest of the network.

  However, the bus topology is considered to be slow when it comes to data distribution. The network could be disrupted if the main cable is damaged, and security can be a concern since the bus topology transmits data in broadcast mode. As a result, this topology is primarily used in small offices or home networks (GeeksforGeeks, 2023).

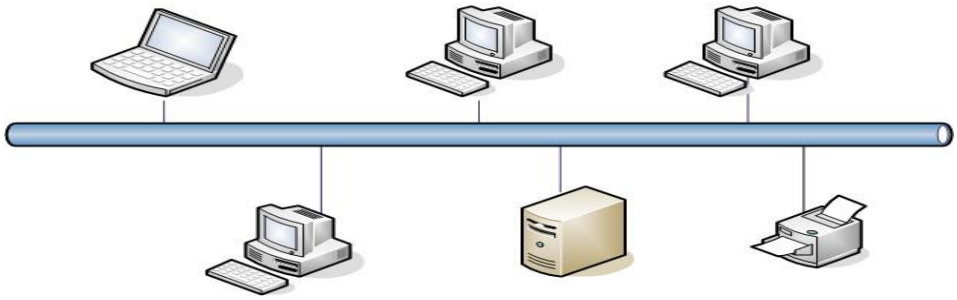

Figure 2.10: Bus topology (Petryschuk & Petryschuk, 2023)

- Star topology

  The star topology derives its name from its star-like shape, where each device is connected to a central hub or switch. All communication between devices is routed through this central hub or switch, which functions as a traffic controller for the network. One of the main advantages of this topology is that if a single device fails, it does not impact the entire network, offering greater reliability than other topologies. Furthermore, adding new devices to the network is simple, new devices are simply connected to the central hub or switch.

  However, expanding the network by adding new devices can be expensive, particularly for larger networks, as each device requires a separate cable running to the central hub or switch. Additionally, if the central hub or switch malfunctions, the entire network could be affected, resulting in network downtime. The star topology remains a popular choice for larger, high-speed networks where reliability and performance are critical (GeeksforGeeks, 2023).

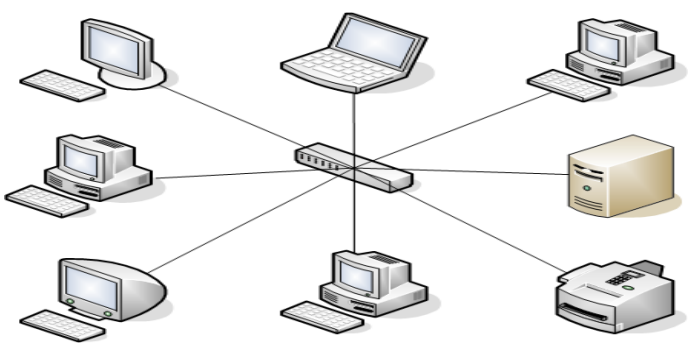

Figure 2.11: Start topology (Petryschuk & Petryschuk, 2023)

- Ring topology

  In a ring topology, devices are interconnected in a circular formation, each connected to exactly two neighboring devices, with each device connecting to the next until the final device links back to the first, forming a continuous closed loop for data transmission. One of the main benefits of this topology is that it offers an even distribution of network traffic, and if one device or cable fails, the network can continue functioning as data can travel in both directions around the ring, which increases the topology's reliability.

  However, if the network traffic increases, data transmission may slow down. Also adding or removing devices from the network can require to reconfigure the entire network. Additionally, a malfunctioning device or broken cable can result in network downtime (GeeksforGeeks, 2023).

  As a result, the ring topology may not be the best choice for large networks due to the possible latency and the difficulties associated with adding or removing devices. It is suitable for LANs and MANs (Alexander, 2017).

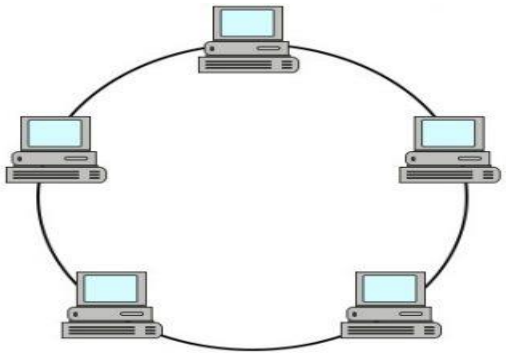

Figure 2.12: Ring topology (Andrea, 2020)

- Mesh topology

  In a mesh topology, every device is linked to every other device within the network, creating multiple redundant paths for data transmission. Which improves reliability and fault tolerance. If a device fails or a cable is

damaged, data can still be transmitted along other paths, preventing network downtime. One of the main benefits of this topology is that new devices can be added to the network easily. However, it may be more expensive and complex than other topologies.

Therefore, the mesh topology is an ideal choice for large networks where reliability and fault tolerance are essential (Alexander, 2017).

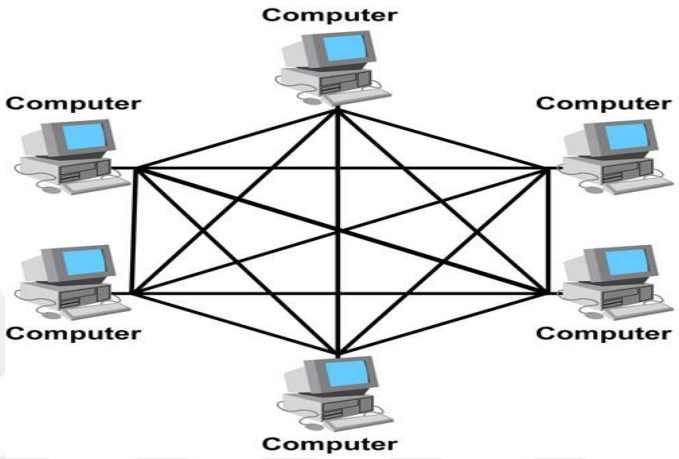

Figure 2.13: Mesh topology (Andrea, 2020)

- Tree topology

  In a tree topology, devices are arranged in a hierarchical structure, with some devices acting as parent nodes and others as child nodes, resembling a tree. Tree topology are considered to be scalable while adding new nodes to the network is easy, and it's easy to manage the network, monitor the performance. However, the network could fail if the root node was corrupted, and it is expensive to build a tree topology due to the large number of nodes are involved.

  As a result, the tree topologies are commonly used in wide area networks (WANs) and local area networks (LANs) because they offer a high level of scalability and can easily accommodate growth (Alexander, 2017).

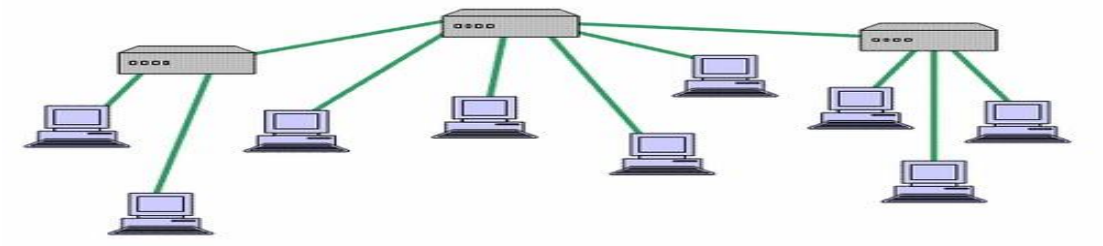

Figure 2.14: Tree topology (Andrea, 2020)

2.4. Network Devices

There are various types of network devices used in computer networking, each serving a specific function to enable communication and data transfer between devices. Some of the most common network devices are:

- Router

  A router is a network device that connects multiple networks together and forwards data packets between them by defining the most efficient path for data to travel. Routers uses logical addresses to forward data packets, such as IP addresses. In case of incoming data packets, they make decisions based on the destination address to determine the best path for the packets to travel.

  Routers has the ability to connect different types of networks, such as Ethernet, Wi-Fi, and cellular networks. This enables devices on different networks to communicate with each other, even if they use different protocols or technologies. Routers also can be used in a variety of settings, from home networks to large networks. They can be connected to modems to enable access to the internet, and also they can be used to create virtual private networks (VPNs) to securely connect remote networks.

- Hub

  A hub is a basic computer network device that connects multiple devices together to form a network. A hub is like a repeater, except that while a repeater have only two connectors, a hub can have many more; that is, it repeats a signal over many cables as opposed to just one.

The hub is considered to be a central point where all devices on the network are connected to it. The hub receives data from a node and broadcast it to all the connected devices on the network, but only the intended node can process the data.

Hubs are commonly used in small networks, due to the congestion and the reduced network performance, especially as the number of nodes grows. (Donahue, 2007, p. 7).

- Switch

  A switch is a network device that connects multiple devices on a local network and enables them to communicate with each other. It operates at the data link layer and uses the device's MAC address to forward data packets to the correct destination device.
  In contrast, a switch maintains a record of which devices are connected to its ports and forwards frames exclusively to the intended devices.

  Switches are available in various sizes, ranging from small desktop switches with a few ports to large rack-mounted switches with hundreds of ports. The switch has the ability to reduce network congestion by crating dedicated communication paths between devices, providing faster and reliable network performance (Donahue, 2007, p. 10).

- Modem

  A modem is a device that connects a computer or network to the internet by converting digital signals from a computer or other digital device into analog signals that can be transmitted over telephone or cable lines, and then converting it back into a digital signals that the computer can understand. Additionally, depending on the type of internet connection being used, there are different types of modems. For example, dial-up, modems are used to connect to the internet over telephone line, cable modems are used to connect to the internet over a cable television line, and wireless modems are used to transmit and receive data using radio frequencies.

Modems are an important component of many computer networks, as they allow devices to connect to the internet and communicate with each other over long distances (Donahue, 2007).

- Firewall

  A firewall is a network security tool that supervises and filters specific types of traffic as it enters or exits your network, relying on predefined security rules. This is done to safeguard the network against unauthorized access or cyberattacks, although this is not always the case. Firewalls are commonly deployed when connecting networks to external entities that are not deemed trustworthy.

  A firewall could be a hardware which they are physical devices that are placed between a network and the internet to filter network traffic, and it could be a software which they are programs that are installed on individual computers or network servers to monitor network traffic.

  Firewalls can be configured to filter traffic based on a variety of criteria, including IP addresses, port numbers, and types of traffic. They can also be configured to block traffic from specific websites or domains (Donahue, 2007, p. 361).

- Access point

  An access point is a hardware device that enables wireless devices to connect to a wired network. The access point transmits wireless signals to devices that are within its range. The devices can connect to the network by sending and receiving data through the access point.

  Access point can be used to extend the range of a wireless network or to create a new wireless network. It's commonly used in home, business networks, and public places (Donahue, 2007).

- Repeater

  A repeater is a networking device that is used to extend the range of a wireless or wired network by amplifying or regenerating signals. It receives a signal and then retransmits it. Repeaters are commonly used in places where a single access point or network switch may not provide adequate coverage (Donahue, 2007).

- Bridge

  Bridge connects two or more local area networks (LANs) together and enables them to communicate with each other. It forwards data packets between networks based on their Media Access Control (MAC) addresses. Bridges can also be used to connect different types of networks together, such as Ethernet and Wi-Fi.

  In this case, the bridge acts as a wireless access point, allowing wireless devices to connect to the wired network (Donahue, 2007, p. 66).

2.5. Network Protocols

From the beginning of human life, we have constantly engaged in essential activities such as breathing, eating, drinking, and communicating. Communication has always been crucial for humans, as we cannot survive without knowledge of our surroundings, the ability to express our needs, and the capacity to search for ideas and information.

In modern times, the vast majority of people use the internet to communicate, share ideas, ask questions, and stay informed about what is happening around the world. The need for communication through the internet has led to the development of protocols to manage these communications. The most well-known and widely used protocol for internet communication is TCP/IP.

TCP/IP uses a client/server model of communication, in which a user requests a service from another user who provides it. This model tells us that TCP/IP is a point-to-point communication system, enabling communication between a point on the network and another point (TCP/IP Illustrated, n.d.).

TCP/IP is a two-layer program consisting of a higher layer and a lower layer. These layers serve as the communication protocols between interconnected network devices on the internet. The Transmission Control Protocol (TCP) is responsible for assembling messages into packets and sending them to other devices, as well as receiving packets and reassembling them into their original messages. On the other hand, the Internet Protocol (IP) handles the routing of packets to their intended destinations (TCP/IP Illustrated, n.d.).

In this section, some TCP/IP protocols (TCP/IP Illustrated, n.d.) will be mentioned, and finally, the work flow for some layers will be described.

2.5.1. Protocols used within TCP/IP model layer

In the beginning, and starting with the top layer, the application layer, a plenty of protocols could be selected to be used depending on the suitable protocol to gain the communication goals.

As the application layer, one of the most well-known protocols is Hypertext Transfer Protocol (HTTP), which facilitates web browsing, server communication, and file downloads, File Transfer Protocol (FTP) which is used for transferring files over internet. Telnet, which is a protocol that allows you to log on to remote computers, Simple Mail Transfer Protocol (SMTP), used for sending and receiving emails, Domain Name System (DNS) is used to map domain names to their corresponding IP addresses, and finally to be mentioned, Dynamic Host Configuration Protocol (DHCP) is a network protocol utilized to automatically allocate IP addresses to devices on a network.

Within the second layer, the transport layer, there are two well-known protocols to used, Transmit ion Control Protocol (TCP), and User Datagram Protocol (UDP). Later on, in this section, the model layer workflow will be briefly describe the TCP protocol workflow.

In the network layer, the following protocols could be used, Internet Control Message Protocol (ICMP), Internet Gateway Protocol (IGP), Exterior Gateway Protocol (EGP), and finally to be mentioned, the Border Gateway Protocol (BGP).

Finally, within the link layer, Point-to-point Protocol (PPP), and Serial Line Internet Protocol (SLIP) are examples for protocols that could be used within the link layer.

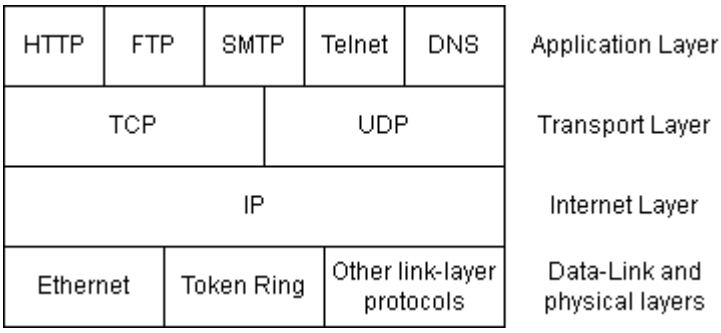

Figure 2.15: TCP/IP protocols

2.5.2. Model layers work flow

The TCP/IP protocol consists of couple of layers, each layer has a way of handing the received data from an application or from a previous layer. In this section, the workflow for each TCP/IP layer is described (TCP/IP Illustrated, n.d.).

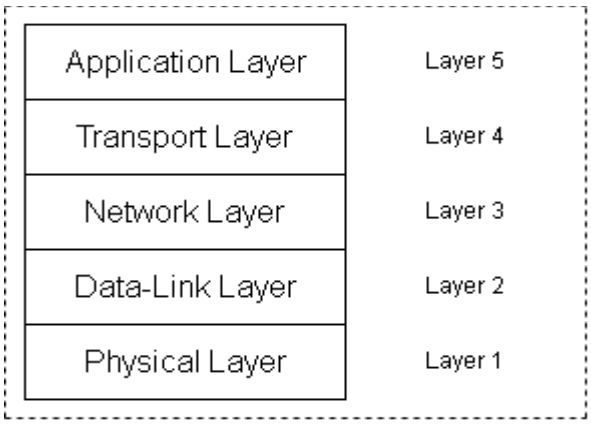

Figure 2.16: TCP/IP model layers

- Application layer

  The application layer serves as a standardized interface for enabling communication among applications over the internet. When an application

desires to transmit data over the network, it must first send the data to the transport layer. There are various ways for an application to transmit data to the TCP/IP application layer, with one popular approach being the sockets API. This API provides a comprehensive set of functions, such as socket (), connect (), send (), recv (), and close (), which allow applications to establish connections and exchange data with other applications. The application layer formats the data in a way that is appropriate for the application protocol being used. For example, if the application is using the HTTP protocol, the data would be structured as an HTTP message. Once the data has been formatted, the application layer proceeds to transfer it to the transport layer by invoking the relevant transport layer protocol function, such as TCP or UDP.

- Transport layer

  When the transport layer receives data from the application layer, it adds a header to the data that contains important information such as the source and destination port numbers, sequence number, and acknowledgment number. The resulting unit, which consists of the header and the data, is called a segment. If the size of the data exceeds the maximum transmission unit (MTU) of the network, the transport layer may divide the data into smaller segments. The maximum size of each segment is determined by the Maximum Segment Size (MSS) value that was negotiated during the TCP handshake. Additionally, during the segmentation process, the transport layer adds sequence numbers and checksums to each segment to ensure the integrity of the data during transmission. Finally, the transport layer passes the segments to the network layer.

- Network layer

  The network layer encapsulate the received segments by adding a network layer header to each segment using the Internet Protocol (IP), forming a new unit called a packet. The network layer header includes information such as the source and destination IP addresses, and other routing information. Then the IP forwards the packets to the appropriate network

based on the routing information in the packet header. After the network layer has determined the appropriate next hop router, if the destination is located on the same network, the network layer will pass the packet down to the link layer. However, if the destination is located on a different network, the network layer will take charge of transferring the packets to the intended destination.

- Link layer

  The Link layer or the data link layer adds the physical addressing information, it adds a header and a trailer to each packet, including the physical address such as MAC addresses. Then the link layer sends the data over physical network like wires or wireless.

2.5.3. TCP/IP packet

IP (Internet Protocol) is provided for transmitting datagrams (blocks of data) from source host to destination host, each host is identified by a fixed length address.

IP intended work doesn't only relies on transmitting datagrams from source to destination, it also provides fragmentation and reassembling long datagrams in order to send them through small packet networks. (RFC 791: Internet Protocol, 1981b).

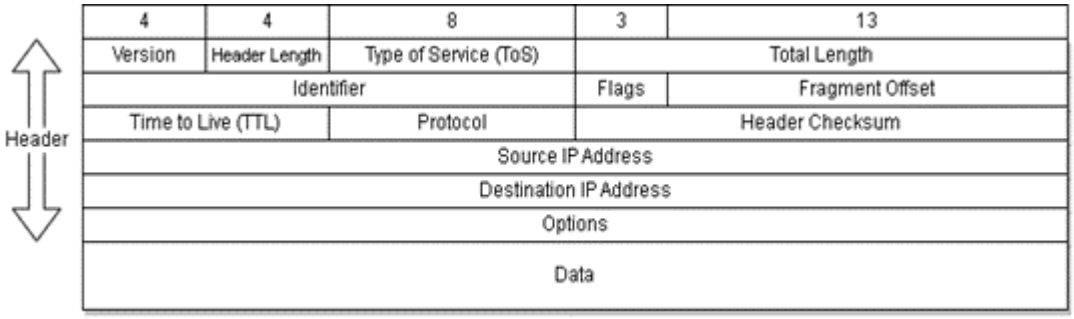

Figure 2.17: IPv4 packet format

As in figure 2.5.3 (IPV4 Packet Format - S1720, S2700, S5700, and S6720 V200R011C10 Configuration Guide - IP Service - Huawei, n.d.-b), the IPv4 datagram consists of a header and a data field. The TCP module would provide the addresses and other parameters in the header as an argument. The header consists of:

- Version (4 bits): This field indicates whether the IP protocol being used is IPv4 or IPv6

  .
- Header Length (4 bits): It specifies the length of the IPv4 header, indicating how long the header section of the IP packet is.

- Type of Service (ToS - 8 bits): This field describes the type of service requested for the packet. It's mainly relevant in a differentiated service model.

- Total Length (16 bits): This value tells us the total size of the IP packet, including both the header and the data.

- Identification (16 bits): IPv4 devices use a counter to keep track of the number of IP datagrams they send. The counter value increases by one each time a datagram is sent, and this value is placed in the identification field.

- Flags (3 bits): Only the rightmost two bits are used. The rightmost bit indicates whether this datagram is the last data fragment (1) or not (0). The middle bit is the fragmentation flag, with 1 indicating that the datagram cannot be fragmented and 0 indicating that it can be fragmented.

- Fragment Offset (13 bits): This field specifies the position or offset of a data fragment within a packet.

- Time to Live (TTL - 8 bits): TTL represents the lifespan of a datagram on a network and is measured in the number of hops it can make before being discarded.

- Protocol (8 bits): This field tells us the type of higher-level protocol being carried within the datagram, such as TCP, UDP, or ICMP.

- Header Checksum (16 bits): Devices calculate a checksum for the IP header when receiving a datagram. If this checksum is 0, it indicates that the header hasn't been modified, and the datagram can be retained. Note that this checksum covers only the header, not the data.

- Source IP Address (32 bits): This field specifies the IPv4 address of the sender, indicating where the packet originated.

- Destination IP Address (32 bits): This field specifies the IPv4 address of the intended receiver, indicating where the packet should be delivered

3. SECURING PEER-TO-PEER COMMUNICATION BASED ON TCP/IP STRUCTURE AND MODIFIED DIFFIE-HELLMAN MODEL

3.1. Introduction

Since a short time, people used to exchange messages using different applications, most of these applications depends on TCP/IP protocol for message exchanging over the internet as a main purpose.

Nowadays, sending or receiving messages is not the only concern to face, there are many concerns to look for. Securing the message itself is one of the most concerns to put an eye on, no one should be able to see or to understand the message except the intended one. In order to achieve full security for the messages, there were the need for cryptography arises.

In a general words, TCP/IP techniques prepares the message to be sent using a couple of layers, each layer add its own header for each message that comes through, then it passes it to the next layer till it reaches the lowest network interface layer (Kowalczyk, 2020).

On the other hand, cryptography assures the security, integrity, and the availability of the message, and also authenticate the involved parties. All that and more just to make the message be secured while travelling through the internet (Stallings, 2015).

Due to the importance of securing channels and travelling messages, a huge number of papers have been published to enhance and discuss the security of TCP/IP protocol.

In (Kumar & Karthikeyan, 2011), Kumar, and Karthikeyan saw that there is a need for security mechanism to be added to the architecture of TCP/IP protocol suit that includes a layer called security layer, which guarantees security to the Application layer using a protocol application layer security protocol (ALSP).

In (Ahsan & Kundur, 2002), Ahsan and Kundur proposed data hiding scenarios using stego-algorithm that considered as an extension to the work on (Handel & Sandford, 1996), (Wolf, 1989), which is more practical and robust since they are based

on redundancies and multiple interpretations of process strategies of the internet protocol, As in their first scenario, they used the flag field located in the TCP/IP packet header in order to make use of its values in data hiding.

In (Handel & Sandford, 1996), Theodore, Handel and Maxwell proposed to use the unused six bits in the IP packet (figure 29) which they are in between data offset byte and the urgent pointer. These six bits, combined with two bits from the IP header provides 1 byte of hidden data per packet transmit.

The main objective of this research is to provide an untraditional security mechanism based on modifying the use of TCP/IP packet fields, also, avoiding some overheads such as session key, and authentication has been taken into consideration to improve communication speed side by side with security.

3.2. TCP header overview

The data field within an IP packet consists of the actual payload or information being transmitted over the Internet Protocol (IP). The structure and content of the data field depend on the higher-level protocol indicated in the "Protocol" field of the IP header. The two most common transport layer protocols that use the IP packet as their underlying unit are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

As this paper tries to alter and modify the TCP/IP packet data field (TCP header) in order to enhance the security, in the remaining of this section, let's give a brief description about the TCP header, look at figure 3.2.2 (BYJU'S Exam Prep, 2022) that shows the TCP header format.

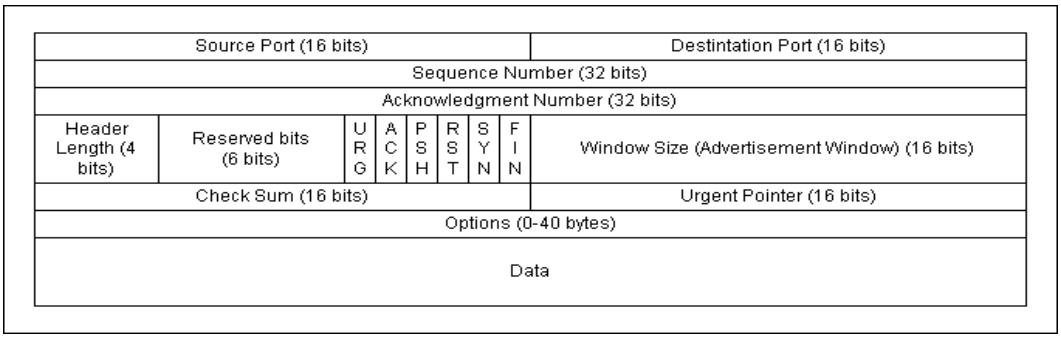

Figure 3.1: TCP header format

- Source Port (16 bits), the application program port number in the sender's host.

- Destination Port (16 bits), the application program port number in the receiver's host.

- Sequence Number (32 bits), each byte in the TCP segment has a unique sequence number that is assigned by the TCP. The sequence number is used to keep track of every byte sent by the sender's host (IBM Documentation, n.d.-b).

- Acknowledgment Number (32 bits) is a counter to keep track of every byte that has been received.

- Header Length (4 bits) contains only a number that that the receiver host uses to calculate the total number of bytes within the TCP header, it is also very important so that the receiver could figure where the data portion begins (Administrator, n.d.-b).

- Control Flag (6 bits) is divided into 6 fields, each field is a 1 bit field, URG (Urgent pointer), ACK (Acknowledgment), PSH (Push the data without buffering), RST (Reset segment control), SYN (Synchronize), and FIN (Finish).

- Window Size (16 bits) is used for flow control, it tells how many bytes the sender can send, and how many bytes the receiver can receive without acknowledgment.

- Check Sum (16 bits) is an error detection mechanism, the sender adds a checksum number called CRC checksum to the checksum field, the receiver could reject the data if the CRC check fails.

3.3. Proposed TCP header structure

The existing TCP/IP Protocol suite architecture doesn't have any specific security mechanism for application layer, which is major setback (Kumar & Karthikeyan, 2011). In (Kumar & Karthikeyan, 2011), they proposed a new layer as a security layer called ALSP, we propose that the security becomes within the TCP header itself.

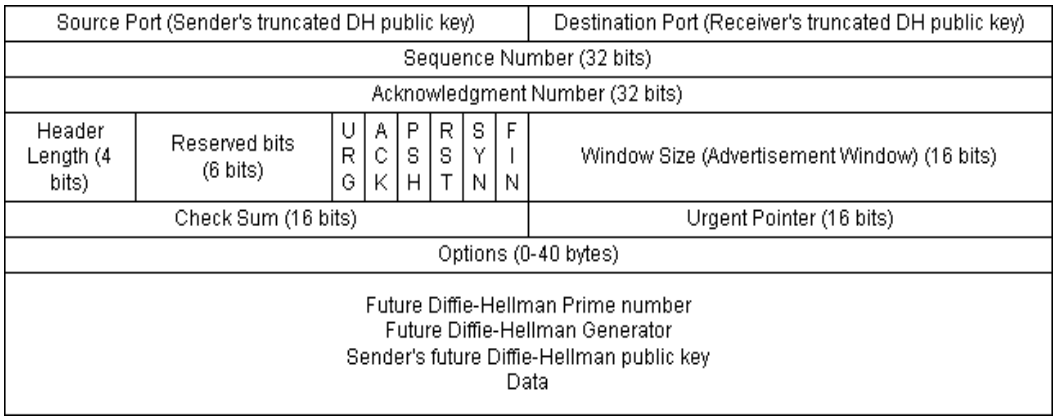

Figure 3.2: Proposed TCP header format

The In the proposed system, we altered the TCP header structure, it was designed in a way to provide a powerful security for the ports, and the data, also, mechanisms for exchanging Diffie-Hellman keys that makes the Diffie-Hellman public keys to become protected.

First, altering the way of using ports. The source port becomes the sender's host truncated Diffie-Hellman public key 'A' (16 bits), also, for the destination port, it becomes the receiver's host truncated Diffie-Hellman public key 'B' (16 bits).

The rest of TCP header fields till the data field stays the same as they were, which they are as following: Sequence number, Acknowledgment number, Header length, Reversed bits, Control flags, Window size, Checksum, Urgent pointer, and Options.

Finally, some space from the data field were reserved for future Diffie-Hellman values in order to be used to calculate future Diffie-Hellman shared keys , these values are the Diffie-Hellman prime number 'p', Diffie-Hellman generator 'g', and the sender's Diffie-Hellman protected key 'A'. These values along with the data represent the new data field.

The new data field is encrypted using AES encryption algorithm in order to secure not only the data, also to secure the public Diffie-Hellman values, in this way, the Diffie-Hellman values are no longer considered to be public, it is protected now, meaning that only the sender and the receiver will know these values. Even if there was a man on the middle, there is no chance to possess the protected Diffie-Hellman values or to read the message.

3.4. Modified Diffie-Hellman

In Chapter 1, the Diffie-Hellman model was discussed, specifically focusing on the exchange of Diffie-Hellman values between Alice and Bob. For a more comprehensive understanding of how Alice and Bob execute this exchange, please refer to Figure 1.13. In the secure communication flow, during the initial cycle, the well-known Diffie-Hellman model is employed with the assistance of a server to facilitate the generation and exchange of Diffie-Hellman values between Alice and Bob.

However, in subsequent cycles, including the second cycle and beyond, a modified version of the Diffie-Hellman model is utilized for the exchange of values between Alice and Bob. For a visual representation of this modified Diffie-Hellman model, kindly look at Figure 3.4.1, which provides an illustrated diagram for better comprehension.

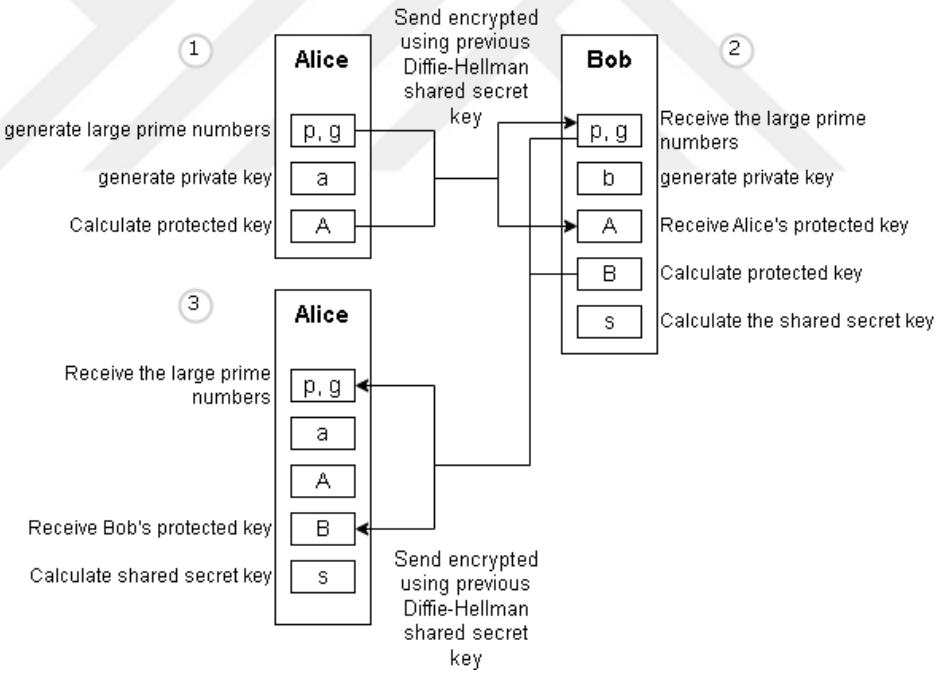

Figure 3.3: Modified Diffie-Hellman model

As shown in Figure 3.4.1, this modified Diffie-Hellman model has three distinct steps. It's important to note that for the modified Diffie-Hellman model to operate effectively, an encryption algorithm must be in use between Alice and Bob.

In the first step, as illustrated in Figure 3.4.1, Alice initiates the process by generating a large prime number 'p,' a generator 'g,' and her private key 'a.' Using these generated values, Alice calculates her protected key 'A.' Subsequently, Alice combines the prime number 'p,' the generator 'g,' and her protected key 'A' and encrypts this composite value for transmission to Bob.

Moving on to step number 2, Bob decrypts the values received from Alice, revealing the prime number 'p,' the generator 'g,' and Alice's protected key 'A.' Bob then generates his own protected key 'b,' which allows him to calculate his protected key 'B.' At this point, Bob can compute the secret shared key 's,' signifying that he now possesses all the Diffie-Hellman values. Notably, Alice does not yet have all the Diffie-Hellman values, prompting Bob to resend the prime number 'p,' the generator 'g,' and his protected key 'B,' all encrypted back to Alice.

As illustrated in step 3, after decrypting Bob's message, Alice utilizes Bob's protected key 'B' to generate the shared secret key 's,' thus affirming that she now possesses all the Diffie-Hellman values.

The key distinction between this version of Diffie-Hellman and the original Diffie-Hellman lies in the protection of all Diffie-Hellman values. This approach ensures that there is no opportunity for an eavesdropper to deduce a protected key, enhancing the security of the communication process.

3.5. Secured communication flow

In this section, the communication flow will be fully described in all the scenarios, from the sending party to the receiving one, the process of receiving the packet, resending from the receiver party to the senders party, and finally, the process of receiving the packet again, but this time as the sender party. We will assume that the sender party is “Alice”, and the receiving party is “Bob”.

- From Alice to Bob

  Look at figure 3.4.1 for an illustrated diagram that describes the secured communication flow from Alice to Bob.

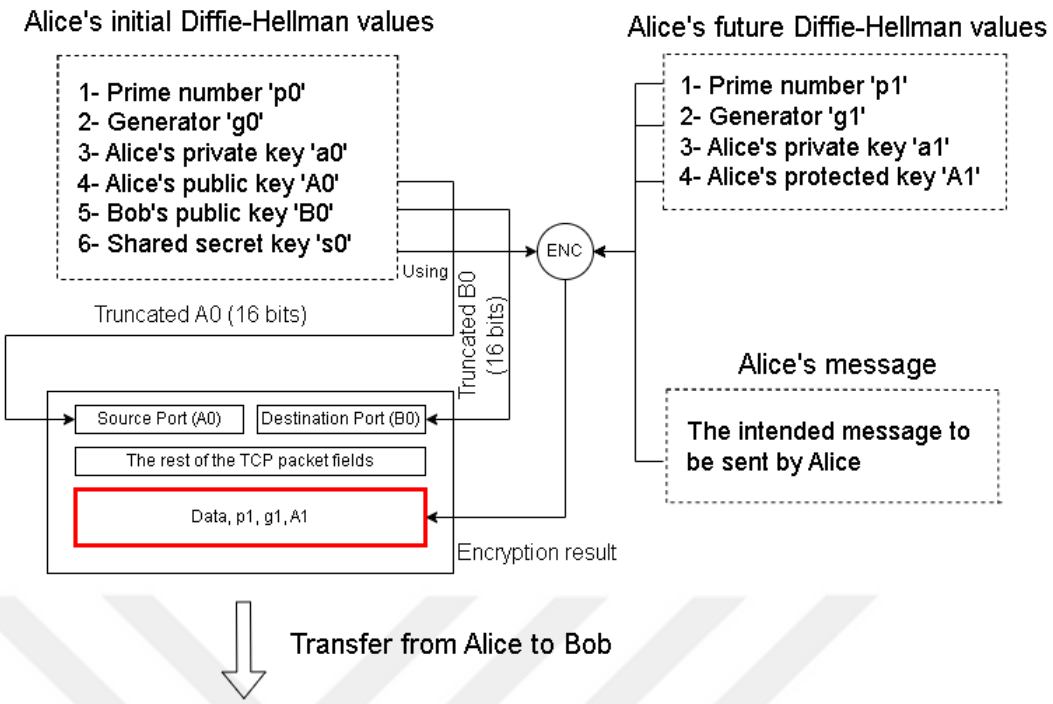

Figure 3.4: Secured communication flow from Alice to Bob

Alice's initial Diffie-Hellman values are the values that Alice has received from the server as discussed previously in Diffie-Hellman section in chapter 1.

Alice's Diffie-Hellman public key 'A0' is going to be truncated to fit into the source port field, Bob's Diffie-Hellman public key is also going to be truncated to fit into the destination port field.
Both the source port and destination port is a 16 bits long, there were the need for truncation came from, also, the truncation process is very important so that in case of an eavesdropper existence, learning the source and destination port will not lead to figuring Alice's and Bob's Diffie-Hellman public keys.

The previously mentioned technique of truncating the public values is considered to be the beginning of converting the public keys into protected keys.

Alice then starts generating a future Diffie-Hellman values from his own, without using any help from the server, Alice generates a prime number, a generator, a private key, and a protected key. Notice that we didn't say a public key, instead, we said a protected key due to that only Alice and Bob are going to be able to poses it, the process of protecting the public keys is going to be discussed in the next step of encrypting the packet.

Those future Diffie-Hellman values are going to be used in future transactions between Alice and Bob.

Now, while al the needed steps for encryption are done, Alice encrypts her future prime number 'p1', future generator 'g1', future protected key 'A1', and the message 'data'. The encryption algorithm to be used is AES encryption algorithm.

For the encryption algorithm to be used, initial Diffie-Hellman shared secret key 's0' is going to be used as the input key, the key long could be 16 bytes, 24 bytes, and 32 bytes long. Finally, the encryption process guarantees that only Alice and Bob could read the message and the future Diffie-Hellman key, this level of security ensures to convert the public Diffie-Hellman values to be protected.

- Bob

  Bob, and after receiving the packet shown in figure 6 from Alice, Bob compares his truncated Diffie-Hellman public key with the packet destination port, also compares Alice's Diffie-Hellman public key with the packet source port. After successfully comparing the source and destination sources, then Bob uses the Diffie-Hellman shared secret key 's0' to decrypt the packet. The values after decrypting the packet are the future prime number 'p1', the generator 'g1', Alice's protected key 'A1', and the message.

  Bob now adds the received Diffie-Hellman values to his future Diffie-Hellman values and generates a new Diffie-Hellman private key 'b', and

calculates a new Diffie-Hellman protected key ‘B1’, with theses values available together, Bob now is able to calculate the new Diffie-Hellman secret shared key ‘s1’. Look at figure 3.4.2 for an illustrated diagram.

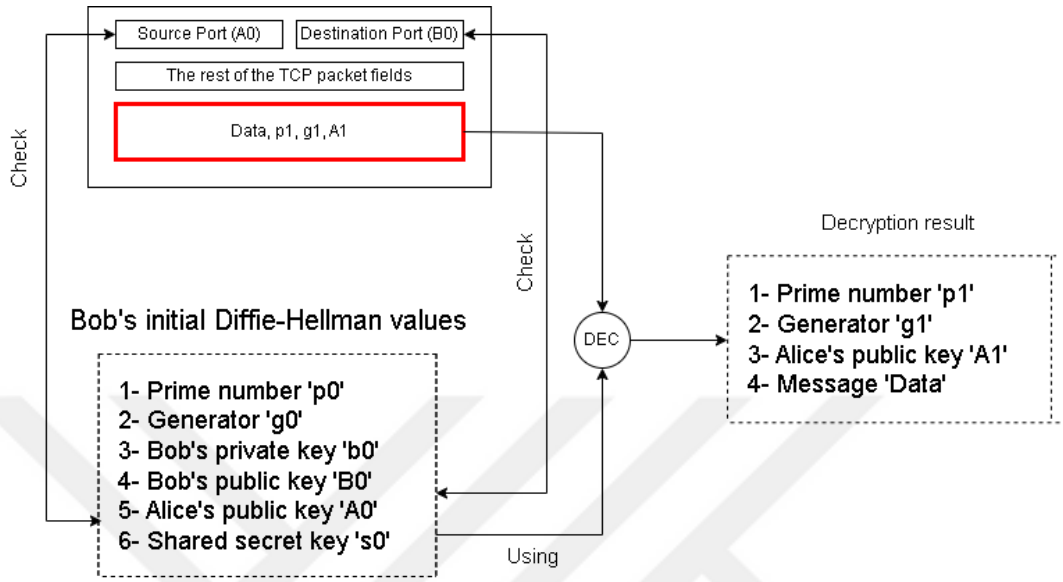

Figure 3.5: Secured communication flow when Bob receives the packet

- From Bob to Alice

  Look at figure 3.4.3 for an illustrated diagram that describes the secured communication flow from Bob to Alice.

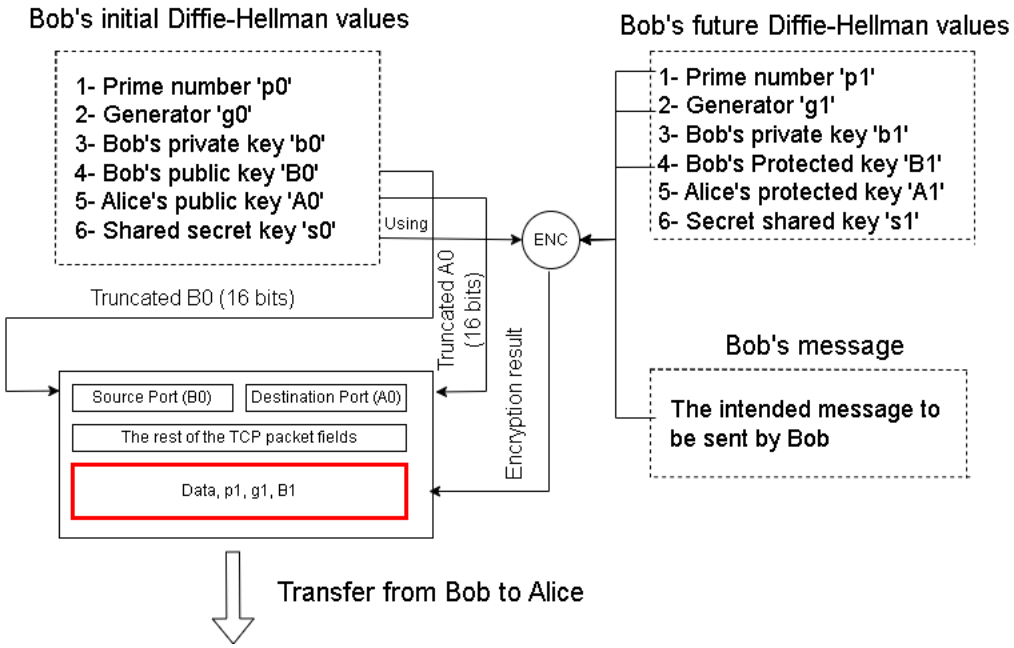

Figure 3.6: Secured communication flow from Bob to Alice

The previous figure is too close to the process of sending from Alice to Bob in figure 32, but there are a few differences as they will be discussed in this section.

The source and destination ports are changed to suit this transaction, as this packet is from Bob, the source port should be Bob's truncated Diffie-Hellman public key 'B0', and as Alice is the receiver, her truncated Diffie-Hellman public key 'A0' is the destination port.

But, the main difference is within the future Diffie-Hellman values that Bob possesses. As in figure 32, Alice only had 4 Diffie-Hellman values, but within this case, Bob has all the 6 values due to receiving Alice's public key from Alice which helped of calculating the new shared secret key 's1'.

- Alice

  As described in figure 33, Alice does the same process that Bob had to do from checking the source and destination ports, decrypting the data field and possessing the new initial Diffie-Hellman values 'p1', 'g1', and Bob's protected key 'B1' till completing her future Diffie-Hellman values depending on the received values.

- Future transaction

  Now, after the first transaction is done, and the session didn't shut down, Alice and Bob replaces their future Diffie-Hellman values with the initial Diffie-Hellman values from the server. Alice and Bob are no longer need any help from the server to generate Diffie-Hellman values as they both exchange their protected values to generate their own secret shared keys. The new Diffie-Hellman values become the initial Diffie-Hellman values. Look at figure 3.4.4 that illustrates the system behavior with Diffie-Hellman values in each transaction.

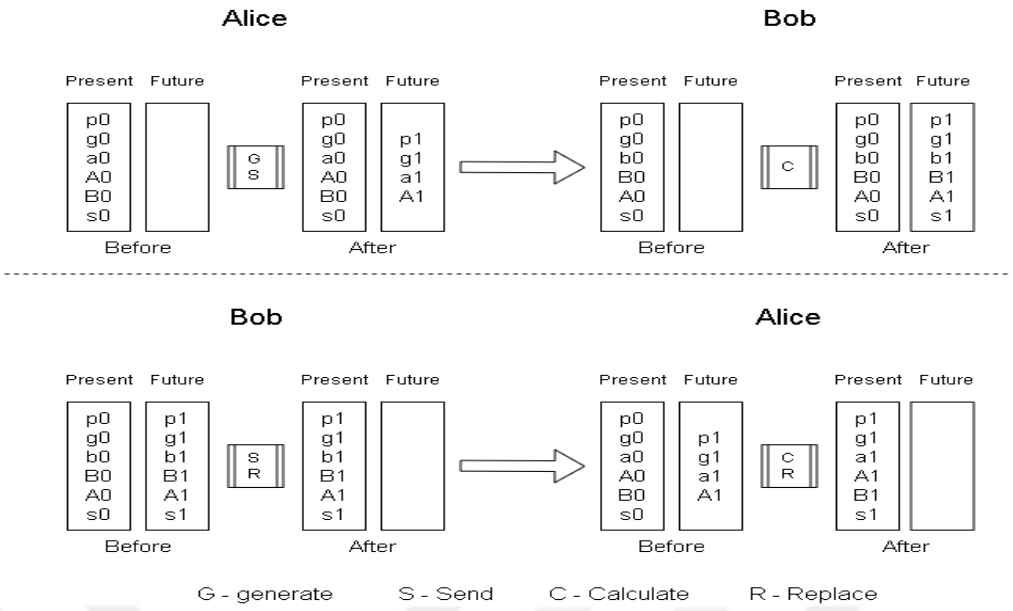

Figure 3.7: The system dealing with modified Diffie-Hellman values

As shown in figure 3.4.4, the present field describes the Diffie-Hellman values that are in use for source port, destination port, and for encryption/ decryption. But, the future filed is made for generating/calculating the future Diffie-Hellman values in order to send them along with the message to the destination in order to use them in future communications.

Also, in figure 3.4.4, we can notice that the beginning state for Alice, and Bob are almost the same as the end state, the only difference is that the Diffie-Hellman values are different.

In case of session shut down due to session time out or a fault in checking the source and destination ports, both Alice and Bob are required to reuse the server to generate Diffie-Hellman values and start from ‘s0’ to encrypt their messages.

4. FINDINGS

4.1. Results

The Performance includes security and time, it is the vital part of the TCP/IP Protocol suite. Several performance metrics are used to evaluate the performance of the encryption algorithms such as Encryption and Decryption time for the AES algorithm. To demonstrate the performance for the proposed architecture, a series of simulation runs are performed on a variety of set of data. Table 1 shows the data that are collected from all the simulations were made in all scenarios.

Table 1: Simulation results

#	State	key size	Encryption/Decryption time	Equivalent time	Total time	Time ratio
1	Sending	0				
	Receiving				1.461	0
2	Sending	16	1.425	2.85		
	Receiving		4.739	9.478	6.164	3.219
3	Sending	24	1.636	2.454		
	Receiving		5.978	8.967	7.614	4.211
4	Sending	32	2.327	2.327		
	Receiving		6.969	6.969	9.296	5.344

In table 1, state column describes the transaction state. Sending state is the state of sending packets from the sending party to the receiving one. The key size is the AES key length, it could be 16 bytes, 24 bytes, and 32 bytes. Transaction time is the time of encrypting the packet within the sending party, and decrypting the packet within the receiving party.

Notice that in table 1, the first row's key size is 0, there is no such key size, the first row describes a normal transaction with a very simple encryption/decryption algorithm just to compare with our work performance. Look at figure 4.1.1 that shows a chart which illustrates the differences between all the scenarios.

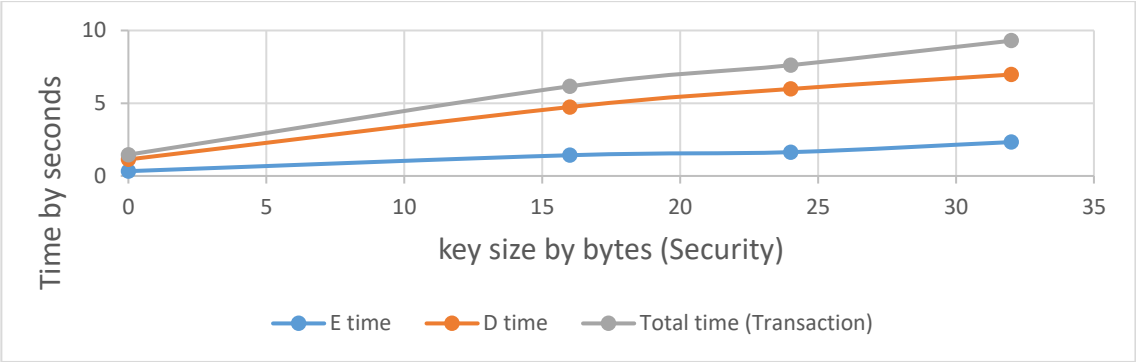

Figure 4.1: Transaction time and security

5. DISCUSSION

In the course of our study, we conducted several simulations to evaluate different security configurations. Table 1 provides an overview of the simulation results. In the first row, we measured the transmission time without any encryption or decryption, resulting in a baseline measurement. In the second row, we introduced a basic level of security by employing a 16-byte encryption key, which yielded a latency of 3.219 seconds. Row 3 exhibited a latency of 4.211 seconds with a heightened level of security. Finally, in the last simulation, a latency of 5.344 seconds was observed.

To gain a better understanding of these results, Figure 4.1.1 illustrates the relationship between transaction time and performance. It becomes evident that the optimal balance between security and transaction time is achieved with a 24-byte encryption key for use in the AES algorithm, resulting in a latency of 4.211 seconds. This configuration represents the most suitable choice for our application due to the more security that the system gains in comparison with the low latency time comparing with when we used 16-byte long key.

6. CONCLUSION

In conclusion, we must acknowledge that despite the existing weaknesses within the TCP/IP protocol, it has achieved universal adoption, making it irreplaceable for the foreseeable future. This paper aims to enhance the security of the TCP/IP model by including an enhanced version of Diffie-Hellman, along with AES algorithm. This enhanced Diffie-Hellman version ensures the protection of public values, utilizing the shared secret key of Diffie-Hellman as the input key for the AES algorithm. These modifications guarantee heightened security for the model, all while minimizing additional transaction time by reusing existing overheads within the TCP header and expanding the data field with supplementary information.

RESOURCES

Administrator. (n.d.-b). TCP Header Anaylsis - Section 3: TCP Header Length Analysis. https://www.firewall.cx/networking/network-protocols/tcp-udp-protocol/tcp-header-analysis.html

Advanced Encryption Standard (AES). FIPS. November 23, 2001.

Ahsan, K., & Kundur, D. (2002, December). Practical data hiding in TCP/IP. In Proc. Workshop on Multimedia Security at ACM Multimedia (Vol. 2, No. 7, pp. 1-8). New York: ACM Press.

Alexander. (2017). The various types of network topologies. Swiss Network Solutions - Swissns GmbH. https://www.swissns.ch/site/2017/06/the-various-types-of-network-topologies/

Andrea, H. (2020). Compare and Contrast Network Topologies (Star, Mesh, Bus, Hybrid etc).Networks Training. https://www.networkstraining.com/compare-and-contrast-network-topologies/

AspEncrypt.com - Crypto 101: Basic Terminology. (n.d.). http://www.aspencrypt.com/crypto101_terminology.html

BYJU'S Exam Prep. (2022, September 29). TCP Header - Definition, Diagram, format [GATE Notes]. Gradeup. https://byjusexamprep.com/tcp-header-i

Carts, D. A. (2001). A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols. SANS Institute.

Delfs, H. H., & Knebl, H. H. (2016, 6 21). Introduction to cryptography: principles and applications. Choice Reviews Online, 53(11), 53-4834. doi:10.5860/choice.195379.

Donahue, G. A. (2007). Network Warrior. y O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Dooley, J. F. (2018). History of cryptography and cryptanalysis codes, ciphers, and their algorithms. Galesburg, IL, USA: Springer.

GeeksforGeeks. (2023). Types of Network Topology. GeeksforGeeks.

Goshwe, N. Y. (2013). Data encryption and decryption using RSA algorithm in a network environment. International Journal of Computer Science and Network Security (IJCSNS), 13(7), 9. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (accessed March, 15, 2010).

IBM documentation. (n.d.-b). https://www.ibm.com/docs/en/zos-basic-skills?topic=4-transmission-control-protocol-tcp

IPV4 Packet Format - S1720, S2700, S5700, and S6720 V200R011C10 Configuration Guide - IP Service - Huawei.(n.d.). https://support.huawei.com/enterprise/en/doc/EDOC1000178170/dd76ea1f/ipv4-packet-format

Kessler, G. C. (2003). An Overview of Cryptography. Gary C. Kessler.

Kizza, J. M., Kizza, W., & Wheeler. (2013). Guide to computer network security (Vol. 8). Berlin: Springer.

Kowalczyk, C. (2020). TCP/IP Protocols | Cryptography. Crypto-i.

Kumar, A., & Karthikeyan, S. (2011). Security model for TCP/IP protocol suite. Journal of Advances in Information Technology, 2(2). doi:10.4304/jait.2.2.87-91t

Kumar, A., & Karthikeyan, S. (2011). Security model for TCP/IP protocol suite. Journal of Advances in Information Technology, 2(2). doi:10.4304/jait.2.2.87-91t

M. Wolf, “Covert channels in LAN protocols,” Proceedings of the Workshop on Local Area Network Security (LANSEC’89), pp. 91 – 102, 1989

Milanov, E. (2009). The RSA algorithm. RSA laboratories, 1-11.

Petryschuk, S., & Petryschuk, S. (2023). 11 Types of networks: Understanding the differences. Auvik. https://www.auvik.com/franklyit/blog/types-of-networks/

Rescorla, E. (1999). Diffie-Hellman Key Agreement Method. https://doi.org/10.17487/rfc2631

RFC 791: Internet Protocol. (1981, September 1). IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc791#section-3.1

Salomaa, A. A. (2013). Public-Key Cryptography. Springer Science & Business Media

Schneier, B. B. (1996). Applied Cryptography. John Wiley & Sons.

Selent, D. (2010). Advanced encryption standard. Rivier Academic Journal, 6(2), 1-14.

Sharma, S., & Gupta, Y. (2017). Study on cryptography and techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2(1), 249-252.

Stallings, W. (2015). Computer security principles and practice.

Stinson, D. R., & Paterson, M. B. (2023). Cryptography: Theory and practice. 6000 Broken Sound Parkway NW, Suite 300: Chapman & Hall/CRC.

T. Handel and M.Sandford., "Hiding data in the OSI network model," (Cambridge, U.K.), First International Workshop on Information Hiding, May-June 1996.

T. Handel and M.Sandford., "Hiding data in the OSI network model," (Cambridge, U.K.), First International Workshop on Information Hiding, May-June 1996.

TCP/IP illustrated. (n.d.). Google Books. https://books.google.ps/books?hl=en&lr=&id=a23OAn5i8R0C&oi=fnd&pg=PR9&dq=TCP/IP+Illustrated,+Volume+1&ots=R9kuASole0&sig=yQDK6uNF1VPzS6SaqSpaSCvNuYM&redir_esc=y#v=onepage&q&f=true

Telikepalli, R., Drwiega, T., & Yan, J. (2004). Storage area network extension solutions and their performance assessment. IEEE Communications Magazine, 42(4), 56-63.

Wetherall, D. J., & Tanenbaum, A. S. (2013). Computer networks. Pearson Education.