CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Key Exchange, Encryption Decryption and Proxy Re-encryption using Pseudoinverses

by

Mariam Shoukat

A thesis submitted in partial fulfillment for the degree of Master of Philosophy

in the

Faculty of Computing
Department of Mathematics

2020

To my parents, husband, brothers and sister for their support and love.

CERTIFICATE OF APPROVAL

Key Exchange, Encryption Decryption and Proxy Re-encryption using Pseudoinverses

by

Mariam Shoukat

(MMT173026)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr Aisha Rafiq	IST, Islamabad
(b)	Internal Examiner	Dr. Muhammad Afzal	CUST, Islamabad
(c)	Supervisor	Dr. Samina Batul	CUST, Islamabad

Thesis Supervisor
Dr. Samina Batul
December, 2020

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
December, 2020

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
December, 2020

Author's Declaration

I, **Mariam Shoukat** hereby state that my M. Phil thesis titled "**Key Exchange, Encryption Decryption and Proxy Re-encryption using Pseudoinverses**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my M. Phil Degree.

(Mariam Shoukat)
Registration No: MMT173026

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled **Key Exchange, Encryption Decryption and Proxy Re-encryption using Pseudoinverses** is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of M. Phil Degree, the University reserves the right to withdraw/revoke my M. Phil degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Mariam Shoukat)

Registration No: MMT173026

Acknowledgements

First of all, I would like to thanks **Allah Almighty** for his countless blessings in my life. He has gifted me a loving family and excellent teachers. He supports me in every path of life. I would like to express my special thanks to my supervisor **Dr. Samina Batul** for her motivation. I am extremely gratefull for her assistance and guidence throughout my theisis project, I will pay special thanks to her for the devotions she paid and gave me the most fruitful endeavor of my life. I have appreciated the guidance of my supervisor and feeling proud to be student of such a great teacher. She is truely epitome of a facilitator, who helped me alot in completing the project within the limited time frame.

I am also obliged to my respected teacher **Dr. Rashid Ali** for his unforgettable support and guidence throughout this journey, through his guidence I came to know about so many new things, I am really thankful to him. Also, many thanks are due to all teachers of CUST Islamabad Dr.Muhammad Sagheer, Dr.Abdul Rehman Kashif, Dr.Shafqat Hussain, Dr. M. Afzal, Dr. Samina Batul, Dr. Dur-e-Shewar Sagheer and Dr. Rashid Ali for always listening me and giving the word of encouragement.

I am greatful to My father **Shoukat Ali** and My mother **Salma Nasreen** for their prayers, love and motivation. I would like to thanks my husband **Sohaib Dilawer** brothers **Nauman Shoukat, Arslan Shoukat** and sisters **Memona Shoukat, Arooj Satti, haidi eve** for their support in completing my degree program. They supported and encouraged me throughout my life. I would like to thanks my all family members for their continuous support and patience during my research work.

I also feel honored and thankful to have such supporting friends for helping me survive all the stress this year and not letting me give up.

Finally, I am obliged to all people who have shared their knowledge and supported me all along.

(Mariam Shoukat)

Registration No: MMT173026

Abstract

ElGamal like encryption and decryption using pseudoinverses is truly fascinating with the great hope of advancing performance and security for high-end applications. It provides a high level of safety measures. A known-plaintext attack is mounted on ElGamal like encryption/ decryption and proxy re-encryption scheme. As a result, the common session key involved in the encryption of plaintext is found. The ElGamal like encryption/ decryption and proxy re-encryption scheme is modified by using pseudoinverses. The suggested improvement is multiplication of pseudoinverse of shared secret key both in encryption and decryption. This results in the improvement of security of the ElGamal like encryption and proxy re-encryption scheme, the system of equation become non-linear, which would be strong against the known plaintext attack. The working principle is based on the pseudoinverse chosen by the communicating parties to secure key exchange, encryption, and decryption. The projected approach is exclusively based on the circulant matrices. Detailed examples of application of the proposed schemes for key exchange protocol are presented. Further, security analysis of the modified scheme is left for future work.

Contents

List of Figures

List of Tables

Abbreviations

2DES	Double Data Encryption Standard
3DES	Triple Data Encryption Standard
AES	Advance Encryption Standard
DES	Data Encryption Standard
DLP	Discrete Logarithm Problem
DH	Diffie-Hellman
ECC	Elliptic Curve Cryptography
GF	Galois Field
IFP	Integer Factorization Problem
MITM	Man in the Middle Attack
MATE	Man in the End Attack
PKC	Public Key Cryptography
PSBC	Proxy Server of User B and User C

Symbols

$\mathbf{C}$	Ciphertext
$\mathbf{D}$	Decryption Algorithm
$\mathbf{E}$	Encryption Algorithm
$\mathbb{F}$	Field
$\mathbb{G}$	Group
$\mathbf{K}$	Key
M_R	Matrix Ring
$\mathbb{N}$	Natural Numbers
$\mathbf{P}$	Plaintext or Message
$\mathbf{R}$	Ring
$\mathbb{R}$	Real Numbers
$\mathbb{Z}_n$	Set of Integers
$\mathbb{Z}_p$	Set of Integers modulo prime p

Chapter 1

Introduction

Due to rapid development in the area of information technology, a secure commercial and private communication is necessary. Therefore, the faster and more efficient methods enable people to protect their valuable information. Adversaries are also there to hack this secret information. The field of cryptography has played a vital role for the secure transformation of important information between two or more people. The main purpose of cryptography is to send information between participants in such a way that the threats from adversaries can be avoided.

1.1 Background

There is a major need of secure channel for wireless networking and secret communication for decades, since the advancement of communication technology is influencing the development of more reliable authentic cryptosystems. Over 2000 years, shift ciphers based on alphabets have been used. Later on, many ciphers were introduced for sending codes or secret messages. For example, mono-alphabetical cipher [1], playfair cipher [2], four square cipher [1] and Hill ciphers [3] of different orders etc. With the passage of time resistance to these cryptosystems has been introduced, and there has been numerous attacks applicable on them. Cryptography [4] actually gives us tools to conceal the sensitive information and transmit it confidentially over the susceptible communication channel. For this purpose cryptography gives us basic structure known as cryptosystem. This system has five major components named as plaintext, encryption algorithm, decryption algorithm, ciphertext and key. Purpose of cryptography is not only encryption and decryption but to provide safety for information and data. Cryptography gives data confidentiality, authenticity, availability and integrity [5].

If we discuss about the security aspects, we note that the security of symmetric or private key encryption schemes relies on the number of communication parties involved in cryptosystem, because one shared secret key is used for encryption and decryption. Handling these shared keys are easy for few communicating parties, but it is very difficult to manage the shared key when there is large increasing number of communicating parties.

In symmetric (private) key cryptography, only a single key is utilized for both the data encryption and decryption. Both parties have to share the key with each other for encryption and decryption. Data Encryption Standard (DES) [6], Double Data Encryption Standard (2DES) [7], Triple Data Encryption Standard (3DES) [8], Advanced Encryption Standard (AES) [9].

In 1976, Diffie and Hellman [10] proposed new idea in cryptography and this concept was known as public key cryptography, and it is based on using two keys (private and public). This concept helped to overcome the problems and weaknesses in secret key cryptography, many of public key cryptosystems were specified as RSA [11], Diffie-Hellman key exchange protocol, ElGamal public key cryptosystem [5, 12] and discrete logarithm problem [13] are considered secure. All the said schemes, systems and methods used some number theoretical and pure algebraic structures. For example, in the field of cryptography, many applications of groups are discussed. Especially, we can say that RSA [11] generally depends upon the structure of finite commutative groups, and it works on invertible elements (units) of $\mathbb{Z}_n$ such that $n = pq$, where p and q are randomly large prime numbers. However, the hard problem is to find these primes p_1 and p_2, because it depends on the factorization problem known as Integer Factorization Problem [14].

1.2 Proxy Re-encryption

Proxy re-encryption basically delegates the decryption process to a third party by re-encrypting the ciphertext. Proxy re-encryption has become an important tool in digital rights management schemes in cloud computing. In [15] the main contribution is the application of circulant matrices in a new way for bidirectional proxy re-encryption and decryption. It involves matrix multiplication and inversion in $\mathbb{Z}_p$.

Xion et al. [16] proposed an application that is called " atomic proxy re-encryption", in which a semi trusted proxy converts a ciphertext for Alice into ciphertext for Bob without seeing the underlying plaintext. This fast and secure re-encryption has become popular as a method for managing encrypted file system. Proxy re-encryption (PRE) [17] allows (semi-trusted) proxy to transform an encryption of M under Alice public key into another encryption of the same message under Bob's public key. The proxy, however, cannot learn the original message and in this way the privacy of both parties can be maintained. Some of the applications of proxy re-encryption are e-mail forwarding, securing distributed file system and digital rights management.

Cryptanalysis is the art and science of breaking cryptosystems. Cryptanalysis has huge importance because without it one cannot determine whether the scheme is really secure or not. In this research the cryptanalysis of proxy re-encryption has been done by applying known plaintext attack on encryption scheme.

1.3 Current Research

In this research, we have discussed **Diffie-Hellman type key exchange, ElGamal like encryption/decryption and proxy re-encryption using circulant matrices discussed in** [15] . It was proposed using circulant matrices from $\mathbb{Z}_p$. The private keys of both parties are chosen as circulant matrices. The inverses are found out by using pseudoinverses. Moreover, they used proxy server for re-encyrption of encrypted data. Our research comprise the following tasks.

1. The cryptanalysis of ElGamal like encryption/decryption and proxy re-encryption is performed by known plaintext attack.
2. Using the left modular multiplication of pseudoinverse of common session key K, we have given a modification of the [15] based on circulant matrices.
3. We have applied a known plaintext attack on the encryption scheme as well as on proxy re-encryption using matrix inversion method, as a result of this attack, we are successful to get the shared secret key involved in encryption/decryption.
4. In this modification the main focus was to harden the feasibility of known plaintext attack and brute force attack.
5. The security analysis of the modified scheme and future work is suggested. The modified scheme seems to be resistant against the attack on the orignal scheme of Rajarama et. al [15]. In fact, this type of attack when applied on the modified scheme results in a complex system of non-linear algebraic homogenious equations in many unknowns. The new system seems to be difficult to solve to best of our knowledge and the hardness of such cryptanalysis depends on the paremeters of the scheme such as the order of matrices involve and size of the field $\mathbb{Z}_p$. Therefore, the detailed analysis of the scheme is left for future work.

1.4 Thesis Layout

The composition of the rest of thesis is as follows:

1. In **Chapter 2**, we will explain the fundamental ideas and definition of cryptography. The Mathematical background, Algebra of matrices and Toeplitz matrices are discussed.
2. In **Chapter 3**, Key exchange, encryption/eecryption and proxy re-encryption is discussed.

3. In **Chapter 4**, Cryptanalysis of the encryption/ decryption scheme given in [15] is presented. Furthermore, to ellaborate the concepts on scheme examples are given. To show the cryptanalysis effectiveness on a larger scale an example of order 7×8 is presented.

4. In **Chapter 5**, The modification of scheme introduced by Rajarama is presented. In this suggested work, we have multiplied left modular pseudoinverse of key with the original encryption scheme. The analysis of the scheme and future work is suggested.

Chapter 2

Preliminaries

In this charter, we will discuss mathematical background, some cryptographic hard problems, basic definitions and examples related to the thesis.

2.1 Cryptography [5]

Cryptography is the science of secure communication between two parties in the presence of malicious entity over the public channel. It is a method of protecting information and communications through the use of codes. More particularly, cryptography is about the construction and analysis of protocols that block hackers to access secret messages. This entire process of secure communication is carried out by the help of a system named as cryptosystem. This system consists of five components named as plaintext, ciphertext, encryption algorithm, decryption algorithm and the key. Plaintext is the original message whereas, the encrypted message is called ciphertext. The plaintext is concealed by ciphertext via the encryption algorithm. The ciphertext is retrieved back to plaintext by the receiver or an authenticated person via the decryption algorithm. Both sender and receiver use a secret key to encrypt the original message. The whole security of this cryptosystem is based on the key security, otherwise the secrecy is compromised. The components of ceyptosystem are given below:

1. **P** = Plaintext
2. **C** = Ciphertext
3. **E** = Encryption Algorithm
4. **D** = Decryption Algorithm
5. **K** = key

There are two types cryptosystem that are illustrated in the next section.

2.1.1 Symmetric Key Cryptosystem [18]

In this method, sender and receiver share a common secret key for both encryption and decryption, which is known to the adversary. As a single secret key is used for both Algorithms, it is called a **secret key cryptography**.

Definition 2.1.1. [19]
"Symmetric key encryption is a type of encryption in which sender and reciever shares only one key (secret key) for both encryption and decryption. This scheme consist of a map

$$E : K \times M \to C$$

such that for each $k \in K$, the map

$$\begin{gathered} E_k : M \to C \\ m \to E(k, m) \end{gathered}$$

is invertible."
Where, $m \in M$ is a plaintext (also called messages). C is a ciphertext and the elements $k \in K$ are the keys. E_k is called the encryption function with respect to the key k. The decryption function is an inverse function and represented as:
$D_k := (E_k)^{-1}$.

The efficient algorithms to compute encryption algorithm and decryption algorithm exist.

As defined in [20, 21], Symmetric key schemes are classified either as **stream cipher**, or **block cipher**.
This scheme is useful because it is faster, easy to implement and requires less computer resources. But the main drawback of this cryptosystem is key distribution and its authentication.
Model of symmetric key cryptography is shown in the FIGURE 2.1

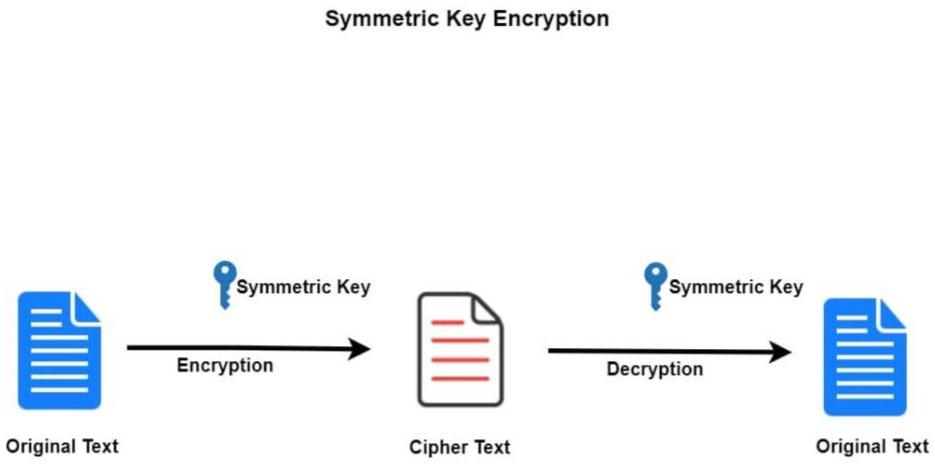


FIGURE 2.1: Symmetric Key

Data Encryption Standard (DES), Double Data Encryption Standard (2DES), Triple Data Encryption Standard (3DES) and Advanced Encryption Standard (AES) are the examples of symmetric key cryptography.
Since, secret key is to be shared among each party involved in communication, it serves the main disadvantage of symmetric key cryptography. Electronic communication used for this purpose is not secure way of exchanging keys because anyone can trap communication channels. The only protected ways of switching keys would be exchanging them personally but it could be a difficult task.

2.1.2 Asymmetric Key Cryptography[22]

Definition 2.1.2. Asymmetric key cryptography also known as public key cryptography is a cryptographic scheme in which pairs of keys are used that are: public keys, which may be distributed widely and private keys, which are only known to the owner. The generation of these keys depends on cryptographic algorithms based on mathematical problems to produce one-way functions.

The model of assymetric key cryptography is shown in the FIGURE 2.2

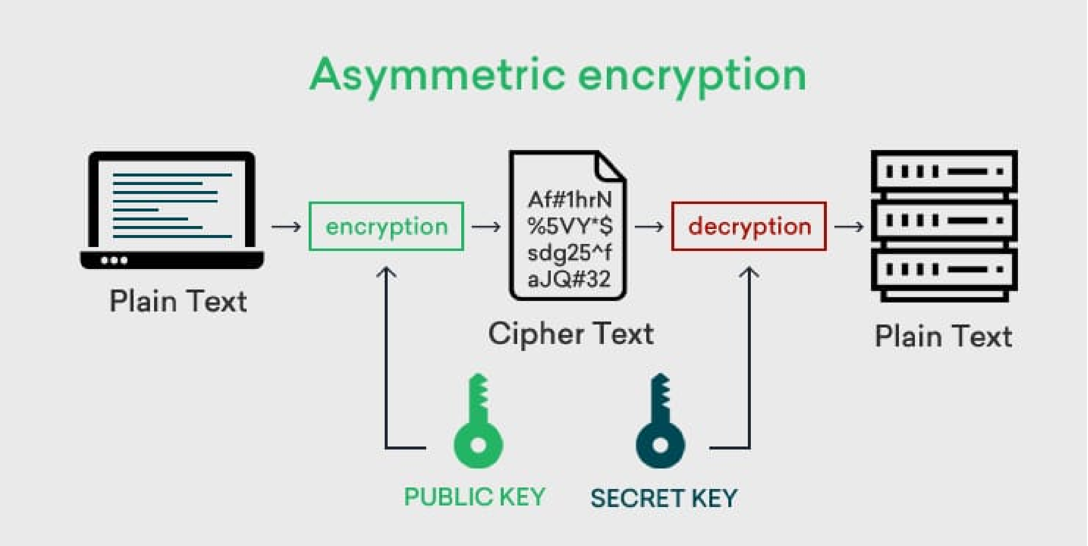


FIGURE 2.2: Asymmetric Key

RSA cryptosystem, ElGamal cryptosystem are examples of asymmetric key cryptography.

2.2 Cryptographic Applications [23]

The basic use of cryptography is transmitting the encrypted communication between us and another system. The most obvious objective of cryptography is not providing confidentiality only, but it also gives the best solutions to other problems. The applications of cryptography are given below,

2.2.1 Confidentiality

Confidentiality means to keep the information secret from unauthorized parties.

2.2.2 Data Integrity

Data integrity refers to maintenance and security of the data, so that it could not be altered during transmission by unauthorized user.

2.2.3 Authentication

This is a service which refers to the identification. The parties those are initiating a communication should identify each other.

2.2.4 Non-repudiation

This provides protection against denial by one of the entities involved in a communication of having participated in all or a part of the communication.

2.3 Mathemtical Background

In this section, we will recall some tools that are used in the thesis.

Definition 2.3.1. [5]
“A group G sometimes denoted by $(G, \cdot)$ is a set of elements with a binary operation denoted by “·” that associates to each ordered pair (a, b) of elements in G an element $(a \cdot b)$ in G such that the following axioms are obeyed.

1. **Closure**: If a and b belongs to G, then $a \cdot b$ is also in G.
2. **Associative**: $a \cdot (b \cdot c)$=$(a \cdot b) \cdot c$ for all a, b, c in G.

3. **Identity element**: There is an element e in G such tha $a \cdot e = e \cdot a$ for all a in G.
4. **Inverse element**: For each a in G, there is an element $a^{'}$ in G such that $a \cdot a^{'} = a^{'} \cdot a = e$."

The definition of a group is illustrated by following examples.

Example 2.3.1.

1. Set of integers $\mathbb{Z}$ is group with respect to addition of integers.
2. General linear group of order n that is $GL_n(R)$ is group of invertible matrices under matrix multiplication.
3. Set of natural numbers $\mathbb{N}$ is not a group under multiplication.

Definition 2.3.2. [24]
A group G is called as abelian if it satisfies the following additional condition:
Commutative: $m.n = n.m$ for all m, n in G.

Following are the examples of abelian group

Example 2.3.2.
Sets $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Q}$ are abelian under addition.

d by ab)

Definition 2.3.3. A non zero set **R** with two binary operations, addition (denoted by $p+q$) and multiplication (denoted by pq, such that for all p, q, r in **R** is known as ring if it satisfies the following axioms:

1. $p + q = q + p$.
2. $(p + q) + r = p + (q + r)$.
3. There is an element 0 in **R** such that $p + 0 = p$.

4. There is an element $-p$ in $\mathbf{R}$ such that $p+(-p)=0$.

5. $p(qr)=(pq)r$.

6. $p(q+r)=pq+pr$ and $(q+r)p=qp+rp$.

If the commutative property with respect to multiplication holds, that is for $p,q \in \mathbf{R}$, $pq=qp$ then such ring is called commutative ring. The examples of ring is given as follows:

Example 2.3.3.

1. $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$ defines a ring under usual addition and multiplication.

2. $M_n(\mathbb{R})$ set of all $n \times n$ matrices over the ring $\mathbb{R}$ is also a ring under addition and multiplication .

3. If p is a prime than the set $\mathbb{Z}_p$ of integer mod p is a ring.

4. Set of odd integer is not a ring because it does not satisfies closure property under multiplication.

Definition 2.3.4. A non empty set X is said to form a semigroup under the binary operation"*", if it satisfies the following two properties:

1. Closure law with respect to "*".

2. Associative law with respect to "*".

Definition 2.3.5. [1]
"A set S together with two binary operation addition and multiplication is called the semiring if it satisfies the following conditions:

1. S is semigroup under addition.

2. S is semigroup under multiplication.

3. Multiplication is distributive over addition in either side. That is, for all $u,v,w \in S$ we have,

$$u(v+w) = (uv) + (uw)$$
$$(u+v)w = (uw) + (vw)$$

Following are the examples of semiring.

Example 2.3.4.

1. Every ring is a semiring therefore set of integers $\mathbb{Z}$, rational number $\mathbb{Q}$, real number $\mathbb{R}$ and complex number $\mathbb{C}$ all are semirings.
2. Set of whole number $\mathbb{W}$ is a semiring.
3. Set of all non-negative integers, non-negative rational numbers and non-negative real numbers are examples of semiring.

If the commutative property with respect to multiplication holds, then such semiring is called as commutative semiring.

$$ab = ba \qquad \forall\ a, b \in S.$$

Definition 2.3.6. [25] A ring R is said to be non-commutative if it does not hold commutative property with respect to multiplication, that is for u and $v \in R$, such that $uv \neq vu$.

Example 2.3.5. [25]
The set $M_2(Z)$ of 2×2 matrices with integer enteries is a non commutative ring with unity $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Definition 2.3.7. [26]
"A nonempty set $(F, +, \cdot)$ together with binary operations \ + " and \ $\cdot$ " is called field F, if the following properties hold.

1. F is abelian under addition.
2. F forms an abelian group under multiplication (only nonzero elements).

3. Multiplication is distributed over addition in F."

Example 2.3.6.

1. Set of real and complex numbers with usual addition and multiplication forms a field.
2. Set of integers $\mathbb{Z}$ is not a field as there are no multiplicative inverses in $\mathbb{Z}$.
3. The set of all $n \times n$ matrices with entries of real numbers under the traditional matrix addition and multiplication forms a field.

Definition 2.3.8. [27]

"The general linear group of order n over any field F(such as the complex numbers) or a ring R(such as the ring of integers) is the set of $n \times n$ invertible matrices with enteries from F or (R) with matrix multiplication as a group operation. Typical notation is $GL_n(F)$, $GL(n, F)$ or simply $GL(n)$, if the field is understood."

Definition 2.3.9. [28]

"A finite field whose order is the form of p^n, where n is any integer and p is prime number is called Galois Field denoted by $GF(p^n)$. In Galois field, elements are defined as

$GF(p^n) = (0, 1, 2, ..., p-1) \cup (p, p+1, p+2, ..., p+p-1) \cup (p^2, p^2+1, p^2+2, ..., p^2+p-1) \cup .... \cup (p^{n-1}, p^{n-1}+1, p^{n-1}+2, ..., p^{n-1}+p-1)$.

The order of Galois field is given by p^n while p is characteristics of field and the degree of the polynomials in $GF(p^n)$ is less than n, while coefficients is at most $p-1$".

Example 2.3.7.

$GF(2^3)$=$(0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1)$ consist $2^3 = 8$, elements where each of the polynomials have degree less than 3 and coefficients are less than 2.

Definition 2.3.10. [29]

A field that contains finite number of elements is known as finite field.

Following are the examples of finite field

Example 2.3.8.

1. $\mathbb{Z}$ under mod p where p is prime is a field.
2. Galois fields are finite field. For example $GF(5^2)$, $GF(5^3)$ and $GF(3)$.

Example 2.3.9.

Finite field $\mathbb{F}_2$ i.e., {0,1} with addition and multiplication is defined in TABLE 2.1 and TABLE 2.2 given below.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

TABLE 2.1: Addition

.	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

TABLE 2.2: Multiplication

2.4 Cryptographic Hard Problems [30]

In this section, we will explain some of cryptographic hard problems which are related to our thesis as given below.

1. **One-way function**

   A function h from Y to Z is called a one-way function if $h(y)$ is easy to compute for all $y \in Y$ but for necessarily all elements $z \in Image(h)$, it figures impracticablly infeasible to find any $y \in Y$ such that $h(y) = z$. In other words, it is the function which is easy to compute on every input but hard to invert it.

2. **Trapdoor one-way function**

   A trapdoor one-way function is a type of one-way function $h : Y \to Z$ with the additional property that given some extra information (called trapdoor information) it becomes feasible to find for any given $z \in Image(h)$, an $y \in Y$ such that $h(y) = z$.

Definition 2.4.1. (Discrete Logarithm Problem) [31]

"Given $x, y \in \mathbb{Z}_p$ such that

$x^n = y \bmod p$

then finding n is known as discrete logarithm problem."

Definition 2.4.2. (Integer Factorization Problem) [31]

An integer factorization problem is defined as, let m be a given number and $m \in \mathbb{Z}$, the problem of decomposition of m to the product of prime p_β and q_β such that.

$$m = p_\beta q_\beta$$

Definition 2.4.3. (Symmetrical Decomposition Problem) [32]

"Given $a, b \in \mathbb{G}$ and $m, n \in \mathbb{Z}$, find $x \in \mathbb{G}$ such that

$$b = x^m.a.x^n$$

then finding x is known as symmetrical decomposition problem".

Definition 2.4.4. (Conjugacy Search Problem) [33]

" Let $\mathbb{G}$ be a group and $x, y \in \mathbb{G}$, whether or not they represent conjugate element of $\mathbb{G}$. That is, the problem is to determine whether there exist an element z of $\mathbb{G}$ such that $y = zxz^{-1}$ is known as Conjugacy Search Problem".

Definition 2.4.5. (Matrix Decomposition Problem)

It is a factorization of matrix into a product of matrices. For A, $B \in \mathbb{Z}_n$, such that

$$\text{AX= B.}$$

Then finding $X \in \mathbb{Z}_n$ is known as matrix decomposition problem.

2.5 Algebra of Matrices [23]

In this section we will discuss rules of addition, multiplication, subtraction, multiplication by a scalar, determinants and inversion of matrices.

Definition 2.5.1. A rectangular array arranged in n rows and m columns in a square bracket is called an $n \times m$ matrix over a ring R and is given as

$$K = \begin{pmatrix} k_{11} & k_{12} & . & . & . & k_{1m} \\ k_{21} & k_{22} & . & . & . & k_{2m} \\ . & . & . & . & . & . \\ k_{n1} & k_{n2} & . & . & . & k_{nm} \end{pmatrix}$$

Matrices are usually denoted by using capital letters such as A, B and K etc. The abbreviated notation of matrix K is given as: $K = [k_{ij}]_{n \times m}$, where k_{ij} denotes the entry in the i^{th} row and j^{th} column of the matrix. The matrix which has n rows and m columns is called "rectangular matrix " of order $n \times m$ and if $m = n$, then K is known as "square matrix ". If each element of diagonal is an element $k \in R$ in a square matrix then it is known as "scalar matrix " of order n, and is written as:

$$\begin{pmatrix} k & 0 & . & . & . & 0 \\ 0 & k & . & . & . & 0 \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & k \end{pmatrix}$$

Definition 2.5.2. Let us consider an $n \times m$ matrix $C = [c_{ij}]$ and $D = [d_{ij}]$ of order $n \times m$ over a ring $\mathbb{R}$, we can define addition of matrices as follows,

$$C + D = [c_{ij} + d_{ij}]$$

The order of $C + D$ is $n \times m$.

Remark 1. Set of all $n \times m$ matrices over a ring R makes an abelian group under addition addition defined for matrices.

Definition 2.5.3. Let A be an $n \times m$ matrix and $k \in R$, then we can define nultiplication of matrix by a sacalar as follows,

$$kA = [ka_{ij}] = [a_{ij}k] = Ak.$$

Definition 2.5.4. If $C = [c_{ij}]$ *and* $D = [d_{ij}]$,

then,

$$P = CD = [c_{ij}][d_{ij}],$$

$$P = [p_{ij}],$$

where,

$$[p_{ij}] = c_{i1}d_{1j} + c_{i2}d_{2j} + \cdots + c_{in}d_{nj}.$$

Remark 2. In general, matrix multiplication does not commute with each other.

2.6 Toeplitz Matrices

In this section we will explain the definitions of Toeplitz matrices, circulant matrix, properties of circulant matrix with the help of examples.

Definition 2.6.1. (Toeplitz Matrices)
"Teoplitz matrix is defined as a matrix, in which each declining diagonal from left to right is constant is called a Toeplitz matrix [34]. It is also known as diagonal-constant matrix and it is named after the German mathematician Otto Toeplitz". A Toeplitz matrix is not always a square matrix. If the $(i, j)^{th}$ element of K is denoted $K_{i,j}$, then we have

$$K_{i,j} = K_{i+1,j+1} = c_{i-j}.$$

The general representation of a teoplitz matrix is given below, the first row of a matrix is taken as $(c_0\ c_1\ c_2\ c_3\ c_4)$, successive row is given as $(c_5\ c_0\ c_1\ c_2\ c_3)$ and so on. A 5×5 teoplitz matrix is given below,

$$A=\begin{pmatrix} c_0 & c_1 & c_2 & c_3 & c_4 \\ c_5 & c_0 & c_1 & c_2 & c_3 \\ c_6 & c_5 & c_0 & c_1 & c_2 \\ c_7 & c_6 & c_5 & c_0 & c_1 \\ c_8 & c_7 & c_6 & c_5 & c_0 \end{pmatrix}$$

2.6.1 The Circulant Matrices [23, 35]

A circulant matrix is a special kind of Teoplitz matrix. It is a square matrix in which each row vector is rotated one element to the right, in this matrix only first row is given, and the successive rows are obtained by cyclically right shifting the present row by one element. Thus the j^{th} row of the circulant matrix of size $n \times n$ is obtained by cyclically right shifting the $(j-1)^{th}$ row by one position, for $j = 2\, to\, n$, given the first row. Let the first row be the row vector, $[c(1), c(2)........c(n)]$. Then after right circular shift of first row by one element, the second row is obtained as $[c(n), c(1)........c(n-1)]$, similarly this process will go on further and each row is determined by right circular shift of a preceeding row. The generalized form circulant matrix C is given as:

$$C = \begin{pmatrix} c(1) & c(2) & . & . & . & c(n) \\ c(n) & c(1) & . & . & . & c(n-1) \\ . & . & . & . & . & . \\ c(2) & c(3) & . & . & . & c(1) \end{pmatrix}$$

2.6.2 Properties of Circulant Matrices

The properties of circulant matrices [23] are given as follows:

1. The circulant matrices, hold a surprising property that the complete matrix can be determined by a single row.

2. Circulant matrices are always the square matrices.

3. The most important property of circulant matrices is that, they are multiplicatively commutative.

4. The inverse of non singular circulant matrices is again circulant.

5. The product and sum of two circulant matrices is again circulant.

6. The rank of $n \times n$ circulant matrix is n.

The multiplying and squaring algorithms of circulant matrices [23] is much faster than the same size of a finite field, it is one dimmensional item used by its first coloumn or row. While on the other hand matrix is two dimmensional item. For example circulant matrix C of order 2 is

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{11} \end{pmatrix}$$

can be stored as $(a_{11} \ \ a_{12})$.
The second row is just the circulant shift of the first row. Note that,

$$A{=}\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{11} \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{11} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}^2 + a_{12}^2 & 2a_{11}a_{12} \\ 2a_{11}a_{12} & a_{11}^2 + a_{12}^2 \end{pmatrix}$$

Hence multiplication of A by itself can be efficiently computed. Therefore computation cost for squaring circulant matrix is much less than that of squaring non circulant matrices. Precisely one has to compute the result of single row or coloumn and the rest of rows or coloumn are just a circulant shift. Let us define a representer polynomial for the circulant matrix C as,

$$\phi(C) = c_0 + c_1x + c_2x^2 + ... + c_{n-1}x^{n-1}$$

Under matrix multiplication and addition circulants become commutative ring which is isomorphic to

$$R = \mathbb{F}[x]/(x^{n-1})$$

following are the easily deducible characterizitions of operations of addition and multiplication of circulant matrices in corresponding polynomial operations.

1. **Addition**

   Let us consider two representre polynomials of circulant matrices C and D over a field $\mathbb{F}$ as

   $$\phi(A) = a_0 + a_1 x + a_2 x^2 + ... + a_{n-1}x^{n-1}, \tag{2.1}$$

   $$\phi(B) = b_0 + b_1 x + b_2 x^2 + ... + b_{n-1}x^{n-1}. \tag{2.2}$$

   Then addition of Equation 2.1 and Equation 2.2 is defined as:

   $$\phi(A+B) = \phi(A) + \phi(B)$$

   Where,

   $$\phi(A+B) = a_0 + a_1 x + a_2 x^2 + ... + a_{n-1}x^{n-1} + b_0 + b_1 x + b_2 x^2 + ... + b_{n-1}x^{n-1}$$

   $$= (a_0 + b_0) + (a_1 + b_1)x + (a_2 + b_2)x^2 + ... + (a_{n-1} + b_{n-1})x^{n-1}.$$

2. **Multiplication**

   The multiplication of Equations (2.1) and (2.2) is defined as:

   $$\phi(AB) = \phi(A) \times \phi(B),$$

   $$\phi(A) \cdot \phi(B) = a_0 + a_1 x + a_2 x^2 + ... + a_{n-1}x^{n-1} \cdot b_0 + b_1 x + b_2 x^2 + ... + b_{n-1}x^{n-1}$$

$$\phi(P) = p_0 + p_1x + p_2x^2 + ... + p_{n-1}x^{n-1} \tag{2.3}$$

The main aim of multiplication is to find p_k for $k = 0, 1, 2, ..., n-1$, here we note that if

$$x_ix_j = \sum_{k=1}^{n} l_{ij}^k x^k.$$

We can define $n \times n$ matrix L_k as $[l^kij]ij$ and it follows as $c_k = AL_kB^l$.

2.7 Modular Arithmetic [23]

The process of executing arithmetic in a finite set of integers is called as a modular arithmetic. Let us give a brief definition of the modulo operation: Let us consider the set of integers $\mathbb{Z}$ and let $a, x, k \in \mathbb{Z}$ and $k \leq 0$. We write

$$a \equiv x \bmod k. \tag{2.4}$$

If m divides $b - r$. Where, r is known as remainder and m is known as modulus. The remainder r is chosen such that

$$0 \leq x \leq k - 1.$$

Usually, we select x from $0 \leq x \leq k - 1$. Consequently, the element of an equivalent class we use does not effect mathematically. By applying division algorithm repeatedly, we can find out the greatest common divisor (gcd) of two positive integers a and b. Above described method is called Euclidean algorithm which is defined as follows:

2.7.1 Eucledian Algorithm [23, 36]

The efficient method to compute the greatest common divisor (GCD) of two integers (numbers) is discussed above and termed as eucledian algorithm, that is the highest number that divides both of the integers, the name is given after the name of ancient Greek mathematician Euclid. This algorithm is one of the oldest algorithm in common use. It is helpful to reduce fractions into simplest form. This algorithm is a part of many other cryptographic and number-theoratic calculations. The division in modular arithmetic can be performed with the help of eucledian algorithm.

The Algorithm

Algorithm 2.7.1. "**Input:** Two positive integers c and d

Output: $GCD(c, d)$

```
1. $M \leftarrow c$; $N \leftarrow d$
2. If $N = 0$ return $M = GCD(c, d)$
3. If $T\ \ = M \bmod N$
4. $M \leftarrow N$
5. $N \leftarrow T$
6. Go to Step 2."
```

2.7.2 Modular Multiplicative Inverse [37, 38]

Extended Eucledian Algorithm

In this section we will explain how to find multiplicative inverses modulo some integer n.

Definition 2.7.1. Given any two integer r and s, the problem is to find an integer t such $r \cdot t \equiv 1 \bmod s$ and $r^{-1} \equiv t \bmod s$, where $1 \leq t \leq s - 1$.

The multiplicative inverse of r mod s are relatively prime that is, $gcd(r, m) = 1$.

Algorithm 2.5.1 (Multiplicative inverse in finite field)

To find the multiplicative inverse in $\mathbb{Z}_p$, we can implement Euclidean Algorithm in the computer algebra system ApCoCoA.

Following is the method of finding the inverse of r mod s.

Input: An integer r and an irreducible integer s.

Output: r^{-1} mod s.

1. Initialize six integers W_i and V_i for $i = 1, 2, 3$ as
   $(V_1, V_2, V_3) = (1, 0, m)$
   $(W_1, W_2, W_3) = (0, 1, r)$
2. If W_3=0, return V_3=gcd(r, s); no inverse of r exist in mod s
3. If W_3=1 then return $W_3 = \text{gcd }(r, s)$ and $W_2 = r^{-1}$ mod s
4. Now divide V_3 by W_3 and find the quotient Q when V_3 is divided by W_3
5. Set $(P_1, P_2, P_3) = ((V_1 - QW_1), (V_2 - QW_2), (V_3 - QW_3))$
6. Set $(V_1, V_2, V_3) = (W_1, W_2, W_3)$
7. Set $(W_1, W_2, W_3) = (P_1, P_2, P_3)$
8. Go to step (ii)

Example of modular inverse is given in the following TABLE 2.3.

Q	A_1	A_2	A_3	B_1	B_2	B_3
	1	0	23	0	1	5
4	0	1	5	1	19	3
1	1	19	3	22	5	2
1	22	5	2	2	14	1

TABLE 2.3: Modular Inverse

The inverse of 5 is 14 under modulo prime 23 as shown below:

2.8 Cryptanalysis

The art of scrutinizing cryptographic schemes is called cryptanalysis, that includes the understanding of working of these schemes and examining them that how these schemes can be broken. In other words we can say that to find the defects in the implementation rather than algorithms. To understand the cryptography practicably, cryptanalysis is a very important part because it gives the deep knowledge about the encryption functions and also its weaknesses which exists in their implementations. During past times, cryptanalyst only attacks to get the key which involves in the encryption algorithm instead of decrypting a message. But now the main concern of a cryptanalyst has been transferred from solving ciphers and investigating the technique used in the encryption to rather solving difficult mathematical problems, to determine the efficient computationally effective method of investigating a ciphertext. Throughout the years on cryptographic protocols and primitives, various types of attacks have been recognized. How an attacker mount these attacks are classified as follows: An attack in which attacker only observe the communication channel is known as the "passive attack". In this attack adversary just threatens the data confidentiality.

In this attack, attacker tries to add, delete or change the transmission on the channel in some other way is known as "active attack". Adversary threatens authentication, confidentiality and data integrity as well. To deduce the plaintext from ciphertext, an active attack is divided into more specialized attacks which are described in the next section.

2.8.1 Algebraic Attack [39]

The main idea of alegebraic attacks is to deduce the secret key by solving nonlinear equations involving message, ciphertext and key bits.

2.8.2 Attacks on Encryption Schemes

We will discuss cryptographic attacks given in [23], these attacks are normally categorised into different types that identifies the kind of information that the cryptanalyst has accessible to mount an algebraic attack. The basic aim of cryptanalyst is to be capable of decrypting new pieces of ciphertext, in all cases without having any additional information. The idea situation for an attacker is to extract secret key.

- **Total break Attack**

In this approach, attacker's main aim is to unveil the secret key or to model another fake key so that he can decrypt the system successfully.

- **Single break Attack**

In this approach, attacker attempts to retrieve plaintext by using the available knowledge on public forum. There are many known cryptographic attacks one can found in the literature, some of them are discussed here.

2.8.3 Ciphertext Only Attacks

This is the type of an algebraic attack in which an adversary knows the encrypted text and has some of the knowledge of encryption technique of scheme and he tries to unveil the original text. For this purpose he will use occurance of frequency of characters or any other. The retrievel of corresponding plaintext makes cipher only text successful.

2.8.4 Known Plaintext Attacks

This is the kind of algebraic attack in which attacker has the knowledge of some of the ciphertext as well as its corresponding plaintext. On the basis of this knowledge, he tries to attemp all the logical attempts to recover the key back that is used in encryption function or makes a logical algorithm to decode mpore ciphertexts.

2.8.5 Chosen Plaintext Attacks

This is a type of attack, in which attacker choses the random plaintext and attempts to obtain ciphertext. Now he will use the pair of plaintext and ciphertext to recover the secret key.

2.8.6 Chosen Ciphertext Attacks

In this type of attack, the attacker will choose the ciphertext, and he attempts to recover the corresponding plaintext or tries to obtain as much information as he can to hack the shared secret key used in encryption of the scheme.

2.8.7 Man in the Middle Attacks

In this type of attack, attacker stays in between the two secretly communicating parties and tries to hack the communication from both ends.

To attempt man in the middle attack, attacker chooses two dummy keys and start the communication with first party by using one of keys and when he establish this channel with first party, he obtains the coded text and tries to decrypt with his own keys. Then he encrypts or altered the received message using his keys and transmits this to second party, when second party approaches him and establish communication he dercypts their encrypted information using his keys. In this way one can interrupt the whole communication by hiding its real identity from both ends and compromise the security of the system.

2.8.8 Man at the End Attack

One of the form of active attack in security of a communication channel found is a Man at the end attack, which is somewhat similar to man in the middle attack. As in this attack, the malicious entity has a control over device which allows him to amend or remove the message sent from one side of communication channel. As the adversary is a human, therefore has much abilities of a human mind. Although attacker has sanction and limitless access to the gadget and this results in all security protections to go in vain for a specific period of time. Timing has a great role in this attack as attacker must have to reciprocate and establish the traffic of message the legitmate one. The need for a timing advantage make this attack more difficult to be implemented.

2.8.9 Brute Force Attack

In this type of attack, attacker tries to use every possible key in order to guess the original message from ciphertext. With larger key extent, this attack is not feasible anymore.

2.9 Hill Cipher Encryption [40]

In 1929, Laster Hill developed a hill cipher. The encryption algorithm takes n successive plaintext letters and substitutes for them n ciphertext letters. The substitution is determined by n linear equations in which each linear equation is assigned a numerical value ($a = 0$, $b = 1$,....$z = 26$). The base of Hill cipher is matrix multiplication, for example, if $m = 3$, the system can be described as follows:

$$\begin{aligned} K_1 &= (C_{11}P_1{+}C_{12}P_2{+}C_{13}P_3) \bmod 27 \\ K_2 &= (C_{21}P_1{+}C_{22}P_2{+}C_{23}P_3) \bmod 27 \\ K_3 &= (C_{31}P_1{+}C_{32}P_2{+}C_{33}P_3) \bmod 27 \end{aligned}$$

This can be further expressed as:

$$K = CP$$

where K and P are coloumn vectors of lenght 3, where K ia a ciphertext and P is representing the plaintext and C is 3×3 matrix, which is the encryption key. All operations are performed under mod 27 here.

Decryption requires the inverse of matrix K. The inverse C^{-1} of matrix C is defined by the following equation.

$$CC^{-1} = I, \text{ where } I \text{ is an identity matrix.}$$

Note: The inverse of a matrix doesnot always exist, when it does it satisfies the proceeding equation.

C^{-1} is computed to the ciphertext, and the plaintext is recovered. In general terms we can write as follows:
Encryption: $K = Ck(P) = K_p$
Decryption: $P = Dk(C) = K - 1C = K - 1K_p = P$.

2.10 Diffie-Hellman Key Exchange Protocol[5]

Ralph Merkle gave the idea of public key protocols, afterwards Diffie and Martin Hellman proposed this idea. Over the public networks, DH is used to transfer keys safely. This key sharing not only support two parties but more than that. DH is highly useful primitive because shared secret key can be helpful to establish a session key secretly that is used in number of different symmetric cryptosystems. The effectiveness of DH depends on the difficulty of computing discrete logrithms. Briefly we can define discrete logrithm in the following way. Primitive root of a prime number q as one whose power modulo q generate all the integers from 1 to $q - 1$. That is if a is the primitive root of the prime number q, then the numbers

$$k \bmod q,\ k^2 \bmod q,\ ...,\ k^{q-1} \bmod q$$

are distinct and consist of integers from 1 to $q - 1$ in some permutation. For any integer a and k primitive root k of prime number q, we can find a unique exponent such that

$$a = k^i \bmod p \qquad 0 \leq i \leq (q - 1)$$

The exponent i is reffered to as discrete logrithm of a for the base k mod q.

2.10.1 The Diffie-Hellman Key Exchange Algorithm

For this scheme, there are two publically known numbers: a prime number q and an integer a this is a primitive root of q. Suppose the user A and B wish to

exchange a key. User A selects a random integer $X_A \leq q$ and computes $Y_A = a^{X_A} \mod q$. Similarly user B independently selects a random integer $X_B \leq q$ and computes $Y_B = a^{X_B} \mod q$. Each side keeps the X value private and make Y value availabe publicly to the other side.
User A computes the key as $K = (Y_B)^{X_A} \mod q$ and user B computes the key as $K = (Y_A)^{X_B} \mod q$. These two calculations produce the identical results.
The result is that two sides have exchanged a secret value. Furthermore, because X_A and X_B are private, an adversary only has the following ingredients to work with: q, α, Y_A and Y_B. Thus, the adversary is forced to take a discrete logrithm to determine the key. For example, To determine the private key of user B, and adversary must compute.

$$X_B = d \ \log_{\alpha,q}(Y_B)$$

The adversary can then calculate the key K in the same manner as user B calculates it.

Example 2.10.1.

Suppose $\alpha = 6$ and $q = 17$ Alice: choose a secret integer $X_A = 5$
Bob: choose a secret integer $X_B = 7$
Alice computes $Y_A = 6^5 \mod 17 = 7$
Bob computes $Y_B = 6^7 \ \text{mod}17 = 14$
Now after the exchange of public key both will compute the common secret key as follows:
Alice: $K_A = (Y_B)^{X_A} = 14^5 \mod 7 = 12$
Bob: $K_B = (Y_A)^{X_B} = 7^7 \mod 7 = 12$

Example 2.10.2.

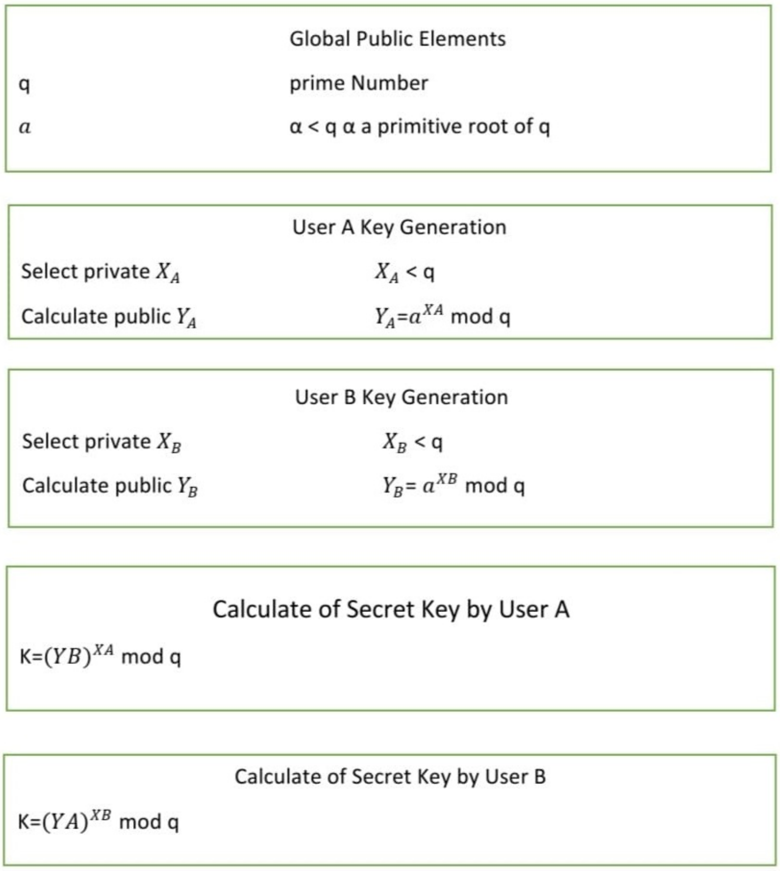
Suppose $\alpha = 9$ and $q = 19$ Alice: choose a secret integer $X_A = 7$
Bob: choose a secret integer $X_B = 5$
Alice computes $Y_A = 9^7 \mod 19 = 4$
Bob computes $Y_B = 9^5 \ \text{mod}19 = 16$
Now after the exchange of public key both will compute the common secret key

as follows:

Alice: $K_A = (Y_B)^{X_A} = 16^7 \bmod 19 = 17$

Bob: $K_B = (Y_A)^{X_B} = 4^5 \bmod 19 = 17$

Key agreement protocol between two users is shown in the FIGURE 2.3.



Global Public Elements

q	prime Number
α	$\alpha < q$ α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = a^{XA} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = a^{XB} \bmod q$

Calculate of Secret Key by User A

$K = (YB)^{XA} \bmod q$

Calculate of Secret Key by User B

$K = (YA)^{XB} \bmod q$

FIGURE 2.3: The Diffie-Hellman key Exchange Algorithm

2.11 The ElGamal Cryptosystem [41]

In 1985, a public key cryptosystem is presented by Taher ElGamal. In this type of encryption, the Diffie-Hellman protocol is utilized so that it can be used as an encryption decryption algorithm. In this cryptosystem, the decryption key is kept private while encryption key is public. The underlying mathematical hard problem of this public key cryptosystem is discrete logarithm problem. For sufficiently large prime modulus, ElGamal cryptosystem is considered to be secure.

Global Parameters

"A large prime p (atleast 512 bits) and generator of multiplicative group g mod p. Alice generates the public/private key pair as follows:

Key Generation

1. Alice chooses any random integer b such that, $b \in 1, 2, ...p - 2$, and computes $A = g^b \bmod p$.

2. The public key of Alice is
$$(p, g, A),$$
and private key is b.

Encryption

Bob encrypts the plaintext m and sends to Alice

1. Bob gets Alice authentic public key (p, g, A).

2. Bob represents the plaintext as integers m in the range $0, 1, 2, ..., p - 2$.

3. Then he selects any random integer k,
$$k \in 1, 2, ...p - 2.$$

4. Bob computes
$$c_1 = g^k \bmod p, \text{ and}$$
$$c_2 = \text{m } A^k \bmod p.$$

5. Finally Bob sends ciphertext
$$C = (c_1, c_2) \text{ to Alice."}$$

Decryption

“Alice receives encrypted message C from Bob and follows the following steps to get the original plaintext/message m.

1. Alice uses her private key b to compute

$$y = c_1^b \bmod p$$

2. Finally Alice finds the plaintext m by computing

$$m = y^{-1}c_2 \bmod p.”$$

All the algorithms such as, RSA, Diffie-Hellman and ElGamal are based on number theory (commutative groups). It is necessary to move towards the development of new cryptosystems, that are thought to be secured on a quantum computer as on a conventional computer (machine). The conjugacy search problem (CSP) is a generalization of discrete logrithm problem DLP. The basic difference is that, DLP is defined on integers while CSP is defined on groups. ElGamal suggested braid groups as platform because CSP is meaningful in such problems.

Chapter 3

Key Exchange, Encryption Decryption and Proxy Re-encryption

In this chapter we will discuss"Deffie-Hellman type key exchange, ElGamal like encryption/ decryption and proxy re-encryption using circulant matrices" presented by C. Rajarama et al. in three steps.

1. In first step, Deffie-Hellman type key exchange protocol will be discussed with the help of an example.

2. ElGamal like encryption/decryption.

3. In the last step proxy re-encryption will be discussed.

3.1 Introduction

The most popular key exchange technique over an unsecure channel is Deffie-Hellman key agreement protocol that is discussed in [15, 41]. In this scheme, the integer matrices are being used as parameters of a cryptosystem to make the size of keys effective and large with smaller sized integers as the elements of the key matrices. The elements of matrices belongs to finite field Z_p where all the enteries are integers ranging from 0 to $(p-1)$ and all the arithmetic operations are carried out with respect to moduolo p, where p is a prime number. In this scheme the Elgamal encryption/decryption is a public key cryptosystem where the cipher text has two components and private keys used are circulant matrices.

This scheme includes matrix multiplication and inversion of square as well as rectangular matrices. Inverse of rectangular matrices are obtained by using pseudoinverse which is discussed in the following section.

3.2 Pseudoinverse

Notation

In the following discussion, the following conventions are adopted that are given in [42, 43].

- $\mathbb{K}$ will denote one of the fields of real or complex numbers, denoted $\mathbb{R}$, $\mathbb{C}$ respectively. The vector space $m \times n$ matrices over $\mathbb{K}$ is denoted by $\mathbb{K}_{m\times n}$.
- For $A \in \mathbb{K}_{m\times n}$, A^T and A^* denote the transpose and Hermitian transpose (also called conjugate transpose) respectively. If $\mathbb{K} = \mathbb{R}$, then $A^* = A^T$.
- Finally, for any positive integer n, $I_n \in \mathbb{K}_{n\times n}$ denotes the $n \times n$ identity matrix.

Definition 3.2.1. For $A \in \mathbb{K}_{m\times n}$, a pseudoinverse of A is defined in [44] as a matrix $A^\dagger \in \mathbb{K}_{m\times n}$ satisfying all the following four axioms, known as the Moore-Penrose conditions.

1. $AA^\dagger A = A$
2. $A^\dagger AA^\dagger = A^\dagger$
3. $(AA^\dagger)^* = AA^\dagger$

4. $(A^{\dagger}A)^{*} = A^{\dagger}A$

$A^{\dagger}$ exists for any matrix A, but when the A has full rank (that is, the rank of A is $\min(m, n)$), then $A^{\dagger}$ can be expressed as a simple algebraic formula, as given below:

1. If A has linearly independent columns (and thus matrix $A^{*}A$ is invertible), $A^{\dagger}$ can be computed as $A^{\dagger} = (A^{*}A)^{-1})A^{*}$.
   This particular pseudoinverse constitutes a left inverse, since in this case $A^{\dagger}A = I$.

2. If A has linearly independent rows (matrix AA^{*} is invertible), $A^{\dagger}$ can be computed as
   $A^{\dagger} = A^{*}(AA^{*})^{-1}$.
   This is a right inverse, as $AA^{\dagger} = I$.

3.2.1 Properties of Pseudoinverse

Existence and uniqueness [45, 46]

The pseudoinverse exists, and it is unique: for any matrix A, there is precisely one matrix $A^{\dagger}$, that satisfies the four properties of the definition 3.2.1.

A matrix satisfying the first condition of the definition is known as generalised inverse. If the matrix also satisfies the second condition, it is called a generalised reflexive inverse. Generalized inverses always exist but are not in general unique. Uniquness is a consequence of the last two conditions.

Pseudoinverse is proven to be unique and defined for the matrices with real or complex entries both. It is also known as generalized inverse.

Basic properties [47]

- If A has real enteries so does $A^{\dagger}$.
- If A is invertible, its pseudoinverse is its inverse. That is $A^{\dagger} = A^{-1}$.
- The pseudoinverse of pseudoinverse is the original matrix: $(A^{\dagger})^{\dagger}) = A$.

- Pseudoinversion commutes with transposition, conjugation and taking the conjugate transpose.
  $(A^T)^\dagger = (A^\dagger)^T, \;\; (\overline{A})^\dagger = \overline{A^\dagger}, \;\; (A^*)^\dagger = (A^\dagger)^*.$

- The pseudoinverse of a scalar multiple of A is the reciprocal multiple of $A^\dagger$: $(\alpha A)^\dagger = \alpha^{-1} A^\dagger$ for $\alpha \neq 0$.

Identities [48]

The following identities can be used to cancel certain subexpressions or expand expressions involving pseudoinverses.

$$\begin{aligned} A^\dagger &= A^\dagger A^{\dagger *} A^* \\ &= A^* A^{\dagger *} A^\dagger \\ A &= A^{*\dagger} A^* A \\ &= A A^* A^{\dagger *} \\ A^* &= A^* A A^\dagger \\ &= A^\dagger A A^*. \end{aligned}$$

Since, for invertible matrices the pseudoinverse equals the usual inverse, only examples of non-invertible matrices are considered below.

Example 3.2.1. [49]

- For $A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, the pseudoinverse is $A^\dagger = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$.

- For $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, the pseudoinverse is $A^\dagger = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \end{pmatrix}$.

- For $A = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}$, the pseudoinverse is $A^\dagger = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 \end{pmatrix}$.

- For $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$, the pseudoinverse is $A^\dagger = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$.

- For $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, the pseudoinverse is $A^{\dagger} = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}$.

3.3 Key Exchange Protocol by C. Rajarama1 et al. [15]

1. **Private Keys**

   Private keys of user A and user B are A and B respectively which are circulant matrices of size $(n \times n)$. Matrices A and B belong to $GL(n, p)$. The elements of the first rows of A and B are chosen so that the rank of both A and B is n. The generator matrix for this DH system is G, which is a rectangular matrix of size $(n-1) \times n$. The elements of G belongs to $\mathbb{Z}_p$. The elements of the generator matrix G are so chosen that the Rank(G) is $(n-1)$. That is, rank $(G) = (n-1)$.

2. **Public Key of User A**

   The public key of user A is denoted by matrix U and it is generated as

   $$U = GA$$

   The size of U is $((n-1) \times n) \times (n \times n) = (n-1) \times n$. By knowing U and G, the private key A cannot be determined, because the left modular multiplicative inverse of G does not exist. In our scheme, G and A are so chosen that the rank of $U = GA$ is $(n-1)$.

3. **Public Key of User B**

   The public key of user B is denoted by matrix V, and is generated by computing B with generator matrix.

   $$V = GB.$$

The size of V is $((n-1)\times n)\times(n\times n) = (n-1)\times n$. By knowing V and G, the private key B cannot be determined, because the left modular multiplicative inverse of G does not exist. In our scheme, G and A are so chosen that the rank of $V = GB$ is $(n-1)$.

4. **Shared Secret Key of User** A

   User A sends matrix U to user B and user B sends matrix V to user A over the unsecured channel. User A calculates the common secret key K_A as given below,

   $$K_A = VA.$$

   The size of K_A is$((n-1)\times n)\times(n\times n) = (n-1)\times n$.

5. **Shared Secret Key of user** B

   User B calculates the common key K_B by computing its private key with the public key of user A as follows,

   $$K_B = UB.$$

   The size of K_B is $((n-1)\times n)\times(n\times n) = (n-1)\times n$.

Example 3.3.1. Let $n = 4$. The value of p is taken as 23. Matrix G be the generator matrix, order of G is 3×4 .A and B are private keys of user A and B respectively, the order of A and B is 4×4.

$$G = \begin{pmatrix} 10 & 4 & 11 & 3 \\ 7 & 9 & 11 & 10 \\ 3 & 6 & 8 & 0 \end{pmatrix}, A = \begin{pmatrix} 10 & 1 & 3 & 3 \\ 3 & 10 & 1 & 3 \\ 3 & 3 & 10 & 1 \\ 1 & 3 & 3 & 10 \end{pmatrix} \text{ and } B = \begin{pmatrix} 3 & 15 & 6 & 3 \\ 3 & 3 & 15 & 6 \\ 6 & 3 & 3 & 15 \\ 15 & 6 & 3 & 3 \end{pmatrix}$$

U and V are calculated as given below:

$$U = GA.$$

$$U = \begin{pmatrix} 10 & 4 & 11 & 3 \\ 7 & 9 & 11 & 10 \\ 3 & 6 & 8 & 0 \end{pmatrix} \begin{pmatrix} 10 & 1 & 3 & 3 \\ 3 & 10 & 1 & 3 \\ 3 & 3 & 10 & 1 \\ 1 & 3 & 3 & 10 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 148 & 92 & 153 & 83 \\ 140 & 160 & 170 & 159 \\ 72 & 87 & 95 & 35 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 10 & 0 & 15 & 14 \\ 2 & 22 & 9 & 8 \\ 3 & 18 & 3 & 12 \end{pmatrix} \bmod 23$$

Since, V=GB.

$$V = \begin{pmatrix} 10 & 4 & 11 & 3 \\ 7 & 9 & 11 & 10 \\ 3 & 6 & 8 & 0 \end{pmatrix} \begin{pmatrix} 3 & 15 & 6 & 3 \\ 3 & 3 & 15 & 6 \\ 6 & 3 & 3 & 15 \\ 15 & 6 & 3 & 3 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 153 & 213 & 162 & 228 \\ 264 & 225 & 240 & 270 \\ 75 & 87 & 132 & 165 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 15 & 6 & 1 & 21 \\ 11 & 18 & 10 & 17 \\ 6 & 18 & 17 & 4 \end{pmatrix}.$$

Now user A will send his public key to user B

$K_A = GBA = VA$

$$K_A = \begin{pmatrix} 15 & 6 & 1 & 21 \\ 11 & 18 & 10 & 17 \\ 6 & 18 & 17 & 4 \end{pmatrix} \begin{pmatrix} 10 & 1 & 3 & 3 \\ 3 & 10 & 1 & 3 \\ 3 & 3 & 10 & 1 \\ 1 & 3 & 3 & 10 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 192 & 141 & 124 & 274 \\ 211 & 272 & 202 & 267 \\ 169 & 249 & 218 & 129 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \bmod 23$$

$K_B = GAB = UB$

$$K_B = \begin{pmatrix} 10 & 0 & 15 & 14 \\ 2 & 22 & 9 & 21 \\ 3 & 18 & 3 & 2 \end{pmatrix} \begin{pmatrix} 3 & 15 & 6 & 3 \\ 3 & 3 & 15 & 6 \\ 6 & 3 & 3 & 15 \\ 15 & 6 & 3 & 3 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 330 & 279 & 147 & 297 \\ 441 & 249 & 432 & 336 \\ 261 & 180 & 333 & 1 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \bmod 23.$$

3.4 Encryption and Decryption by Rajarama1 et al. [15]

Suppose M be a message matrix whose elements belongs to set of integers and their range is 0 to $(p-1)$. The elements of M belong to $\mathbb{Z}_p$, and its size is $(n-1) \times (n-1)$. User A encrypts M and sends it to user B. Matrices A, B, G, U, V and scalar p are same as given in Example 3.3.1

We suppose that user B has sent V to user A that is already being recieved by user A. User A will do encryption by generating two crypto terms U and W are given as:

$$U = GA$$

$$W = MVA. \tag{3.1}$$

From Section 3.3.1 we have

$$W = MGBA = MGAB \tag{3.2}$$

Using Equation 3.1, Equation 3.2 becomes, Since, $K_B = GAB = GBA$

$$W = MK_B \tag{3.3}$$

K_B is not a square matrix because its size is $(n-1) \times n$, its direct inverse does not exist but 3.3 can be solved for M by using pseudoinverse of K_B as

$$M = WK_B^{\dagger} \tag{3.4}$$

Here $(K_B)^{\dagger}$ is a pseudo right modular inverse of K_B and is defined as

$$(K_B)^{\dagger} = K_B^T(K_BK_B^T)^{-1}. \tag{3.5}$$

Crypto-parameters G, A and B are chosen in such a way that K_B is a full rank matrix. The encrypter will send the pair (U, W) to intentional decryptor User B.

3.4.1 Decryption at User B

User B will recieve the pair (U, W) and then calculates K_B by using $K_B = UB$, now he find out $(K_B)^{\dagger}$ by using Equation 3.5 and hence recovers M out of Equation 3.4. All operations are being performed under modulo p.

We will illustrate this in the following example,

Example 3.4.1. Now we will illustrate the example to show decryption at user B. User A choose the message matrix M, A is the private key of user A, U and

V be the public key of user A and user B, where G be the generator matrix and p is prime number that is taken as 23, as given below:

$$M = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix}$$

Matrices A , B , G , U , V and p are same as in Example 3.3.1. M is taken as 3×3 matrix, A, B, being the private keys of user A and user B, where U and V are public keys of user A and user B, $p = 23$ and G be the generator matrix, now we will calculate W and $(K_B K_B^T)$ as follows

$$U = \begin{pmatrix} 10 & 0 & 15 & 14 \\ 2 & 22 & 9 & 8 \\ 3 & 18 & 3 & 12 \end{pmatrix}, V = \begin{pmatrix} 15 & 6 & 1 & 21 \\ 11 & 18 & 10 & 17 \\ 6 & 18 & 17 & 4 \end{pmatrix} \text{ and } K_{AB} = \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix}$$

As,

$$W = MK_{AB}.$$

$$W = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \text{ mod } 23$$

$$= \begin{pmatrix} 72+40+80 & 27+190+190 & 81+180+110 & 189+140+140 \\ 72+28+48 & 27+133+114 & 81+126+66 & 189+98+84 \\ 80+24+16 & 30+114+38 & 90+108+22 & 210+84+28 \end{pmatrix} \text{ mod } 23$$

$$= \begin{pmatrix} 192 & 407 & 371 & 469 \\ 148 & 274 & 273 & 371 \\ 120 & 182 & 236 & 322 \end{pmatrix} \text{ mod } 23$$

$$= \begin{pmatrix} 8 & 16 & 3 & 9 \\ 10 & 21 & 20 & 3 \\ 5 & 21 & 6 & 0 \end{pmatrix} \text{ mod } 23.$$

3.4.2 Decryption

$$M = W(K_B)^{\dagger} \tag{3.6}$$

Where $K_B^{\dagger}$ is pseudoinverse of K_B that is calculated as follows:

$$K_B^{\dagger} = K_B^T(K_BK_B^T)^{-1} \tag{3.7}$$

Where, K_B^T is the transpose of a matrix K_B.

As $K_B = \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix}$, $K_B^T = \begin{pmatrix} 8 & 4 & 8 \\ 3 & 19 & 19 \\ 9 & 18 & 11 \\ 21 & 14 & 14 \end{pmatrix}$

$$K_BK_B^T = \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \begin{pmatrix} 8 & 4 & 8 \\ 3 & 19 & 19 \\ 9 & 18 & 11 \\ 21 & 14 & 14 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 20 & 16 & 8 \\ 16 & 0 & 5 \\ 8 & 5 & 6 \end{pmatrix} \bmod 23$$

$$(K_BK_B^T)^{-1} = \begin{pmatrix} 7 & 12 & 19 \\ 12 & 11 & 17 \\ 19 & 17 & 22 \end{pmatrix} \bmod 23$$

Using K_B^T and $(K_BK_B^T)^{-1}$ in Equation 3.7, we get

$$K_B^{\dagger} = \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix} \bmod 23$$

Now putting the values of W and $K_B^\dagger$ in Equation 3.6

$$W(K_B)^\dagger = \begin{pmatrix} 8 & 16 & 3 & 9 \\ 10 & 21 & 20 & 3 \\ 5 & 21 & 6 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 24+192+15+54 & 256+0+0+30 & 40+256+18+18 \\ 30+252+100+18 & 336+0+0+200 & 50+336+120+6 \\ 15+252+30+0 & 0+336+60+0 & 25+336+36+0 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 285 & 286 & 332 \\ 400 & 536 & 512 \\ 297 & 396 & 397 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 21 & 5 & 6 \end{pmatrix} \mod 23.$$

3.5 Proxy Re-encryption by Rajarama et al. [15]

Proxy re-encryption described in [50] is the process of re-encoding a given cipher text so that now, it can be decoded by another receiver other than the original one. The process is so defined that the re-encrypter itself cannot recover the plain text or it can not get hold of a private keys of the concerned parties.

Consider the model shown in the FIGURE 3.1. Here, A, B, G, U, V, M, W, K_A and K_B are same as described in Section 3.3. C is a circulant matrix of size $n \times n$.

3.5.1 Common Secret Key between User B and User C.

Now the DH type common secret key between user B and user C is decided to have the size $n \times (n+1)$ that is large in size relative to K_A or K_B. The common key of bigger size is obtained as given below.

1. Choose another generator matrix H, and its size is chosen as $n \times (n+1)$.
2. The elements of H belongs to $\mathbb{Z}_p$.
3. User B and user C adopt additional private keys chosen to be circulant matrices B_2 and C_2.
4. The size of B_2 and C_2 is $(n+1) \times (n+1)$.
5. The corresponding public keys are HB_2 and HC_2 respectively, the private keys such that B_2 and C_2 cannot be evaluated by knowing H , HB_2 and HC_2 because H does not have left inverse.
6. User C calculates common secret key between user B and user C using HB_2 that is given below.

$$L_C = (HB_2)C_2 \tag{3.8}$$

Likewise user C will calculate common secret key between user B and user C given by:

$$L_B = (HC_2)B_2 \tag{3.9}$$

$$\text{Since,} \quad B_2C_2 = C_2B_2$$

L_B and L_C are given as

$$L = L_B = L_C \tag{3.10}$$

$$L = HC_2B_2 = HB_2C_2$$

The size of L is $n \times (n+1)$.

3.5.2 Encryption

Encryption done by user A is same as described in Equation 3.4. User A generates the two matrices U and W as given below.

$$U = GA. \tag{3.11}$$

$$W = MVA = MGBA.$$

$$= MGAB.$$

User A sends the encrypted data (U, W) to user B and the proxy server $B \to C$. Using Equation 3.5,

$$M = W(UB)^{\dagger}. \tag{3.12}$$

User B can decrypt ciphertext (U, W) using equation 3.8.

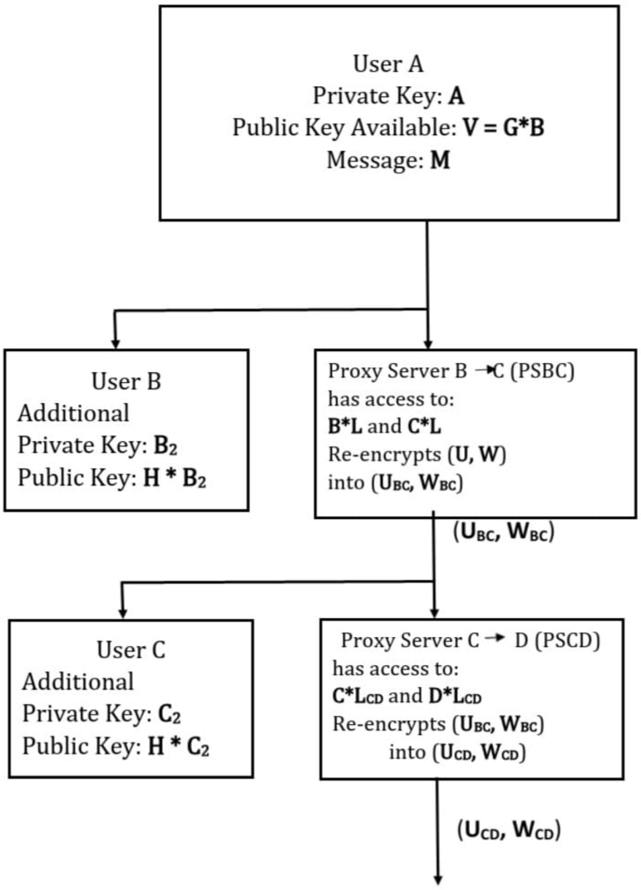


FIGURE 3.1: Proxy Re-encryption and decryption

3.5.3 Re-encryption at Proxy Server $B \to C$

User B or user C will request Proxy Server B→C (PSBC) to send the same data M to user C with proper re-encryption so that user C can decode it correctly.

PSBC has to translate the ciphertext intimated for user B to a new pattern such that translated ciphertext can be decoded by user C. The working of PSBC is given as follows:
PSBC accepts (U, W) as the input and re-encrypts using the same encryption scheme to generate (U_{BC}, W_{BC}) that is then sent to user C. W_{BC} is formulated in such a way that only user C can decrypt it. Moreover Proxy server itself is unable to retrieve M, B or C. During initiation user B and user C sends (BL) and (CL) to PSBC respectively. The size of (BL) and (CL) is defined to be $n \times (n+1)$.

3.5.4 Formulation of U_{BC} and W_{BC} at PSBC

For the purpose of re-encryption. PSBC formulates U_{BC} and W_{BC} from U and W as,

$$U_{BC} = U = GA \tag{3.13}$$

$$W_{BC} = W(CL)(BL)^{\dagger}. \tag{3.14}$$

Here $(BL)^{\dagger}$ is a right modular inverse of (BL). The size of (BL) is $n \times (n+1)$, By definition,

$$(BL)^{\dagger} = (BL)^T[(BL)(BL)^T]^{-1} \tag{3.15}$$

$$(BL)^{\dagger} = L^T B^T (BLL^T B^T)^{-1} \tag{3.16}$$

$$(BL)^{\dagger} = L^T B^T (B^T)^{-1} (LL^T)^{-1} B^{-1} \tag{3.17}$$

$$(BL)^{\dagger} = L^T (LL^T)^{-1} B^{-1}. \tag{3.18}$$

Using definition of pseudoinverse,

$$L^T (LL^T)^{-1} = L^{\dagger}. \tag{3.19}$$

Using Equations 3.15 and 3.16

$$(BL)^{\dagger} = L^{\dagger} B^{-1}. \tag{3.20}$$

Pluging Equation 3.17 in 3.10, we get,

$$W_{BC} = W(CL)L^{\dagger}B^{-1}. \tag{3.21}$$

As $L^{\dagger}$ is a pseudoinverse of L, above equation becomes

$$W_{BC} = WCB^{-1}. \tag{3.22}$$

C and B^{-1} are multiplicatively commutative because they are circulant matrices, so Equation (3.20) becomes

$$W_{BC} = WB^{-1}C. \tag{3.23}$$

From Section (3.5.2) substituting the value of W.

$$W_{BC} = (MGAB)B^{-1}C. \tag{3.24}$$

As B and B^{-1} cancel each other, above equation become

$$W_{BC} = MGAC. \tag{3.25}$$

Hence, we have eliminated the private key of User B that is B and plugged in the private key C in its place. The proxy server PSBC sends (U_{BC}, W_{BC}) to user C. Now user C decrypts W_{BC} given in Equation 3.23.

3.5.5 Decryption by User C

After recieving U_{BC} and W_{BC} by User C. Fron Equation 3.23. M is recovered. Using Equation 3.13 in Equation 3.25, we have

$$W_{BC} = M(U_{BC})C. \tag{3.26}$$

Therefore user C will take right modular pseudoinverse of shared secret key.

$$M = W_{BC}(U_{BC}C)^{\dagger}. \tag{3.27}$$

$(U_{BC}C)^{\dagger}$ is the right modular inverse of $(U_{BC}C)$, where $(U_{BC}C)$ is the shared secret key of the proxy server **PSBC**.

Example 3.5.1. The values of G, A, B, U, V,M, W and p are same as in Example 3.4.1. The values of C, H, B_2 and C_2 are taken as, C is to be taken as circulant matrix of order 4×4, order of H is 5×4, order of B_2 is 5×5 and order of C_2 is 5×5.

$$C = \begin{pmatrix} 9 & 15 & 9 & 6 \\ 6 & 9 & 15 & 9 \\ 9 & 6 & 9 & 15 \\ 15 & 9 & 6 & 9 \end{pmatrix}, H = \begin{pmatrix} 2 & 9 & 7 & 8 & 18 \\ 15 & 9 & 14 & 11 & 19 \\ 2 & 16 & 5 & 4 & 9 \\ 14 & 7 & 22 & 1 & 15 \end{pmatrix}, B_2 = \begin{pmatrix} 13 & 21 & 3 & 5 & 17 \\ 17 & 13 & 21 & 7 & 5 \\ 5 & 17 & 13 & 21 & 3 \\ 3 & 5 & 17 & 13 & 21 \\ 21 & 3 & 5 & 17 & 13 \end{pmatrix}$$

$$\text{and} \quad C_2 = \begin{pmatrix} 9 & 12 & 8 & 1 & 2 \\ 2 & 9 & 12 & 8 & 1 \\ 1 & 2 & 9 & 12 & 8 \\ 8 & 1 & 2 & 9 & 12 \\ 12 & 8 & 1 & 2 & 9 \end{pmatrix}$$

$$U_1 = HB_2.$$

$$V_1 = HC_2.$$

$$U_1 = \begin{pmatrix} 2 & 9 & 7 & 8 & 18 \\ 15 & 9 & 14 & 11 & 19 \\ 2 & 16 & 5 & 4 & 9 \\ 14 & 7 & 22 & 1 & 15 \end{pmatrix} \begin{pmatrix} 13 & 21 & 3 & 5 & 17 \\ 17 & 13 & 21 & 3 & 5 \\ 5 & 17 & 13 & 21 & 3 \\ 3 & 5 & 17 & 13 & 21 \\ 21 & 3 & 5 & 17 & 13 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 616 & 372 & 512 & 594 & 502 \\ 850 & 782 & 698 & 862 & 920 \\ 524 & 382 & 520 & 368 & 330 \\ 729 & 809 & 567 & 321 & 555 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 18 & 4 & 6 & 19 & 19 \\ 22 & 0 & 8 & 11 & 15 \\ 18 & 14 & 14 & 0 & 8 \\ 16 & 4 & 15 & 16 & 3 \end{pmatrix} \bmod 23.$$

$$V_1 = \begin{pmatrix} 2 & 9 & 7 & 8 & 18 \\ 15 & 9 & 14 & 11 & 19 \\ 2 & 16 & 5 & 4 & 9 \\ 14 & 7 & 22 & 1 & 15 \end{pmatrix} \begin{pmatrix} 9 & 12 & 8 & 1 & 2 \\ 2 & 9 & 12 & 8 & 1 \\ 1 & 2 & 9 & 12 & 8 \\ 8 & 1 & 2 & 9 & 12 \\ 12 & 8 & 1 & 2 & 9 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 323 & 271 & 221 & 266 & 327 \\ 483 & 452 & 396 & 392 & 454 \\ 195 & 254 & 270 & 244 & 189 \\ 350 & 396 & 411 & 373 & 358 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 1 & 18 & 14 & 13 & 5 \\ 0 & 15 & 4 & 1 & 17 \\ 11 & 1 & 17 & 14 & 5 \\ 5 & 5 & 20 & 5 & 13 \end{pmatrix} \bmod 23.$$

User B and user C will compute their private keys with public keys of other, shared secret key of both user is defined as:

$$K = K_{BC} = U_{BC}C.$$

L is calculated as follows,

$$L = HC_2B_2 = HB_2C_2$$

$$L = \begin{pmatrix} 4 & 21 & 4 & 9 & 4 \\ 14 & 20 & 9 & 17 & 7 \\ 1 & 20 & 9 & 15 & 4 \\ 9 & 22 & 1 & 10 & 7 \end{pmatrix} \bmod 23.$$

3.5.6 Encryption and Decryption

K_{BC} is found as $K_{BC} = (U_{BC}C)$

$$BL = \begin{pmatrix} 2 & 20 & 20 & 11 & 1 \\ 8 & 3 & 19 & 18 & 20 \\ 20 & 1 & 1 & 1 & 1 \\ 13 & 9 & 6 & 13 & 20 \end{pmatrix} \bmod 23$$

$BL^\dagger$ and CL is found to be

$$BL^\dagger = \begin{pmatrix} 14 & 11 & 8 & 8 \\ 18 & 19 & 11 & 13 \\ 11 & 16 & 20 & 12 \\ 7 & 20 & 6 & 6 \\ 6 & 1 & 11 & 16 \end{pmatrix} \bmod 23$$

$$CL \quad = \begin{pmatrix} 10 & 19 & 5 & 2 & 12 \\ 16 & 22 & 19 & 16 & 3 \\ 11 & 14 & 2 & 8 & 12 \\ 20 & 8 & 20 & 8 & 3 \end{pmatrix} \bmod 23.$$

W_{BC} and $(U_{BC}C)$ is calculated as $W_{BC} = MK_{BC}$,

$$W_{BC} = \begin{pmatrix} 7 & 18 & 9 & 18 \\ 14 & 10 & 3 & 5 \\ 7 & 22 & 13 & 22 \end{pmatrix} \bmod 23.$$

$$U_{BC}C = \begin{pmatrix} 21 & 21 & 10 & 20 \\ 17 & 11 & 3 & 5 \\ 20 & 11 & 5 & 11 \end{pmatrix} \bmod 23$$

$U_{BC}C^\dagger$ is calculated as,

$$U_{BC}C^\dagger = \begin{pmatrix} 2 & 9 & 18 \\ 8 & 13 & 9 \\ 10 & 9 & 14 \\ 11 & 0 & 21 \end{pmatrix} \bmod 23.$$

Finally M is calculated as,

$$M = W_{BC}(U_{BC}C)^{\dagger}$$

$$M = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \bmod 23.$$

Chapter 4

Cryptanalysis

In this chapter, we will show that encryption/ decryption scheme of C. Rajarama et al. [15] is vulnerable to Known plaintext attack. We will first recall the nature of attack.

4.1 Known Plaintext Attack

In this section, we will recall known plaintext attack as discussed in Subsection 2.8.4. In this category attacker has an apprehension of some of the ciphertext as well as its corresponding plaintext, On this basis, he attempts to recover the key or makes a logical algorithm to decode any further ciphertexts.

Note that the ciphertext W for plaintext M is as follows

$$W = MK \tag{4.1}$$

and decryption is done as

$$M = WK^{\dagger}. \tag{4.2}$$

Attacker has the knowledge of both M and W, so from the Equation 4.1

$$M^{-1}W = K. \tag{4.3}$$

Therefore, the given scheme is not secure for any consequent encryption. We will further illustrate the attack by applying it on the example given in section 3.4.1.

Example 4.1.1.

$$W = MK$$

$$\begin{pmatrix} 8 & 16 & 3 & 9 \\ 10 & 21 & 20 & 3 \\ 5 & 21 & 13 & 0 \end{pmatrix} = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix} \bmod 23.$$

After reducing this system of matrices into system of equations we get a linear system of 12 equations and 12 unknowns as follows:

$$\begin{aligned}
9k_{11} + 10k_{21} + 10k_{31} &= 8 \\
9k_{12} + 10k_{22} + 10k_{32} &= 16 \\
9k_{13} + 10k_{23} + 10k_{33} &= 3 \\
9k_{14} + 10k_{24} + 10k_{34} &= 9 \\
9k_{11} + 7k_{21} + 6k_{31} &= 10 \\
9k_{12} + 10k_{22} + 10k_{32} &= 21 \\
9k_{13} + 10k_{23} + 6k_{33} &= 20 \\
9k_{14} + 10k_{24} + 6k_{34} &= 3 \\
10k_{11} + 6k_{21} + 2k_{31} &= 5 \\
10k_{12} + 6k_{22} + 2k_{32} &= 21 \\
10k_{13} + 6k_{23} + 2k_{33} &= 13 \\
10k_{14} + 6k_{24} + 2k_{34} &= 0.
\end{aligned}$$

Now, we will find the value of K using the matrix inversion method and euclidean algorithm under modulo 23 as given below,

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix} = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 8 & 16 & 3 & 9 \\ 10 & 21 & 20 & 3 \\ 5 & 21 & 13 & 0 \end{pmatrix} \bmod 23 \tag{4.4}$$

Inverse of M would be calculated by using the formula given below, working under modulo we will find inverse of (det M) under modulo 23.

$$M^{-1} = \frac{AdjM}{detM} = AdjM(detM)^{-1} \tag{4.5}$$

$$AdjM = \begin{pmatrix} 1 & 7 & 13 \\ 19 & 10 & 13 \\ 7 & 0 & 19 \end{pmatrix} \bmod 23 \tag{4.6}$$

$det(M) = 62 \bmod 23 = 16$

$$det(M) = 16 \tag{4.7}$$

Now by using Extended eucledian algorithm we have
$(detM)^{-1} \bmod 23 = 16^{-1} \bmod 23 = 13 \bmod 23$

$$det(M) = 13 \tag{4.8}$$

Now, using Equation 4.7 and Equation 4.8 in Equation 4.5, we have

$$M^{-1} = \begin{pmatrix} 13 & 14 & 8 \\ 17 & 15 & 8 \\ 22 & 0 & 17 \end{pmatrix}$$

Using the value of M^{-1} in Equation 4.9

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix} = \begin{pmatrix} 13 & 14 & 8 \\ 17 & 15 & 8 \\ 22 & 0 & 17 \end{pmatrix} \begin{pmatrix} 8 & 16 & 3 & 9 \\ 10 & 21 & 20 & 3 \\ 5 & 21 & 13 & 0 \end{pmatrix} \bmod 23$$

$$K = \begin{pmatrix} 284 & 670 & 423 & 159 \\ 326 & 755 & 455 & 198 \\ 261 & 709 & 287 & 198 \end{pmatrix} \bmod 23$$

$$K = \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \bmod 23.$$

Hence, shared secret key between Alice and Bob is vulnerable against known plaintext attack.

4.2 Cryptanalysis Of Proxy Re-encryption

In Section 3.5, we have described the algorithm of proxy re-encryption. In this algorithm shared secret key of the proxy server BC is $K_{BC} = U_{BC}C$ used for encryption and its inverse is used for decryption, where $U_{BC} = U$ and C be the private key of user C. Re-encryption is given as,

$$W_{BC} = MK_{BC} \tag{4.9}$$

and decryption is done as

$$M = W_{BC}(K_{BC})^{\dagger}. \tag{4.10}$$

Attacker has the knowledge of both M and W_{BC}, so from the Equation 4.9

$$M^{-1}W_{BC} = K_{BC}. \tag{4.11}$$

Therefore the given scheme is again vulnerable for any consequent encryption. We will further illustrate the attack by applying it on the Example 3.5.1 given in.

Example 4.2.1.

$$W_{BC} = MK_{BC}$$

$$\begin{pmatrix} 7 & 18 & 9 & 18 \\ 14 & 10 & 30 & 5 \\ 7 & 222 & 13 & 22 \end{pmatrix} = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix} \bmod 23$$

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix} = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 7 & 18 & 9 & 18 \\ 14 & 10 & 30 & 5 \\ 7 & 222 & 13 & 22 \end{pmatrix} \bmod 23$$

As in Example 4.1.1

$$M^{-1} = \begin{pmatrix} 13 & 14 & 8 \\ 17 & 15 & 8 \\ 22 & 0 & 17 \end{pmatrix} \bmod 23$$

Putting this value of M^{-1} in above equation, we compute K_{BC} as follows

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix} = \begin{pmatrix} 13 & 14 & 8 \\ 17 & 15 & 8 \\ 22 & 0 & 17 \end{pmatrix} \begin{pmatrix} 7 & 18 & 9 & 18 \\ 14 & 10 & 30 & 5 \\ 7 & 22 & 13 & 22 \end{pmatrix} \bmod 23$$

$$K_{BC} = \begin{pmatrix} 343 & 550 & 641 & 480 \\ 385 & 632 & 707 & 557 \\ 273 & 770 & 419 & 770 \end{pmatrix} \bmod 23.$$

$$K_{BC} = \begin{pmatrix} 21 & 21 & 10 & 20 \\ 17 & 11 & 3 & 5 \\ 20 & 11 & 5 & 11 \end{pmatrix} \bmod 23.$$

Hence, shared secret key at proxyserver BC is vulnerable against known plaintext attack.

Now we will present another example to show the cryptanalysis of the scheme when value of n is set to be higher as $n = 8$. Illustrative example is given below

Example 4.2.2.

We suppose $n = 8$ and $p = 11$ then order of G is 7×8, The private keys of Alice and Bob are said to be A and B, G be the generator matrix of order 8×7. M be the plaintext of order 7 × 7, where A and B are chosen to be circulant matrices. The orders of A and B are taken to be 8 × 8. Matrices G, A and B are given

$$G = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 4 \\ 3 & 1 & 4 & 8 & 6 & 9 & 10 & 3 \\ 4 & 2 & 4 & 3 & 2 & 5 & 3 & 2 \\ 5 & 1 & 0 & 3 & 1 & 4 & 0 & 1 \\ 10 & 9 & 8 & 7 & 6 & 4 & 3 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 8 & 5 \\ 3 & 2 & 1 & 5 & 4 & 2 & 1 & 1 \end{pmatrix}, A = \begin{pmatrix} 10 & 0 & 4 & 3 & 4 & 1 & 2 & 3 \\ 3 & 10 & 0 & 4 & 3 & 4 & 1 & 2 \\ 2 & 3 & 10 & 0 & 4 & 3 & 4 & 1 \\ 1 & 2 & 3 & 10 & 0 & 4 & 3 & 4 \\ 4 & 1 & 2 & 3 & 10 & 0 & 4 & 3 \\ 3 & 4 & 1 & 2 & 3 & 10 & 0 & 4 \\ 4 & 3 & 4 & 1 & 2 & 3 & 10 & 0 \\ 0 & 4 & 3 & 4 & 1 & 2 & 3 & 10 \end{pmatrix},$$

$$B = \begin{pmatrix} 3 & 4 & 6 & 3 & 1 & 2 & 3 & 4 \\ 4 & 3 & 4 & 6 & 3 & 1 & 2 & 3 \\ 3 & 4 & 3 & 4 & 6 & 3 & 1 & 2 \\ 2 & 3 & 4 & 3 & 4 & 6 & 3 & 1 \\ 1 & 2 & 3 & 4 & 3 & 4 & 6 & 3 \\ 3 & 1 & 2 & 3 & 4 & 3 & 4 & 6 \\ 6 & 3 & 1 & 2 & 3 & 4 & 3 & 4 \\ 4 & 6 & 3 & 1 & 2 & 3 & 4 & 3 \end{pmatrix} \text{ and } M = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 5 & 2 & 4 & 7 & 6 & 1 \\ 8 & 3 & 2 & 1 & 2 & 5 & 7 \\ 9 & 2 & 1 & 3 & 5 & 9 & 2 \\ 2 & 1 & 2 & 3 & 5 & 6 & 3 \\ 1 & 9 & 3 & 7 & 8 & 7 & 6 \\ 2 & 4 & 6 & 9 & 10 & 2 & 8 \end{pmatrix}$$

Public key of Alice is calculated by computing G and A, while public key of Bob is found by computing G and B.

$$U = GA$$

$$U = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 4 \\ 3 & 1 & 4 & 8 & 6 & 9 & 10 & 3 \\ 4 & 2 & 4 & 3 & 2 & 5 & 3 & 2 \\ 5 & 1 & 0 & 3 & 1 & 4 & 0 & 1 \\ 10 & 9 & 8 & 7 & 6 & 4 & 3 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 8 & 5 \\ 3 & 2 & 1 & 5 & 4 & 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 10 & 0 & 4 & 3 & 4 & 1 & 2 & 3 \\ 3 & 10 & 0 & 4 & 3 & 4 & 1 & 2 \\ 2 & 3 & 10 & 0 & 4 & 3 & 4 & 1 \\ 1 & 2 & 3 & 10 & 0 & 4 & 3 & 4 \\ 4 & 1 & 2 & 3 & 10 & 0 & 4 & 3 \\ 3 & 4 & 1 & 2 & 3 & 10 & 0 & 4 \\ 4 & 3 & 4 & 1 & 2 & 3 & 10 & 0 \\ 0 & 4 & 3 & 4 & 1 & 2 & 3 & 10 \end{pmatrix} \bmod 11$$

$$U = \begin{pmatrix} 4 & 4 & 3 & 2 & 9 & 2 & 9 & 6 \\ 8 & 1 & 3 & 8 & 9 & 1 & 4 & 10 \\ 4 & 0 & 4 & 0 & 4 & 0 & 2 & 1 \\ 6 & 4 & 5 & 9 & 2 & 8 & 5 & 3 \\ 0 & 5 & 4 & 0 & 1 & 4 & 4 & 8 \\ 8 & 0 & 10 & 7 & 1 & 7 & 0 & 5 \\ 6 & 1 & 0 & 2 & 8 & 3 & 1 & 2 \end{pmatrix} \bmod 11$$

Now shared secret key of Alice and Bob is found as $K_A = GAB$ and $K_B = GBA$

$$K_A = \begin{pmatrix} 2 & 8 & 5 & 10 & 6 & 8 & 10 & 8 \\ 8 & 8 & 2 & 2 & 3 & 5 & 9 & 7 \\ 0 & 8 & 9 & 5 & 4 & 3 & 6 & 3 \\ 3 & 7 & 7 & 2 & 0 & 3 & 8 & 6 \\ 2 & 9 & 5 & 1 & 9 & 7 & 3 & 2 \\ 0 & 0 & 6 & 5 & 5 & 7 & 10 & 9 \\ 2 & 1 & 8 & 9 & 5 & 10 & 9 & 4 \end{pmatrix} \bmod 11$$

$$K_B = \begin{pmatrix} 134 & 129 & 115 & 120 & 116 & 129 & 142 & 129 \\ 129 & 162 & 156 & 123 & 124 & 159 & 163 & 128 \\ 44 & 52 & 53 & 49 & 48 & 47 & 50 & 47 \\ 135 & 128 & 139 & 134 & 143 & 146 & 127 & 138 \\ 101 & 97 & 71 & 78 & 86 & 73 & 80 & 90 \\ 110 & 132 & 138 & 115 & 137 & 128 & 109 & 119 \\ 157 & 67 & 85 & 75 & 60 & 76 & 97 & 81 \end{pmatrix} \bmod 11$$

$$K_B = \begin{pmatrix} 2 & 8 & 5 & 10 & 6 & 8 & 10 & 8 \\ 8 & 8 & 2 & 2 & 3 & 5 & 9 & 7 \\ 0 & 8 & 9 & 5 & 4 & 3 & 6 & 3 \\ 3 & 7 & 7 & 2 & 0 & 3 & 8 & 6 \\ 2 & 9 & 5 & 1 & 9 & 7 & 3 & 2 \\ 0 & 0 & 6 & 5 & 5 & 7 & 10 & 9 \\ 2 & 1 & 8 & 9 & 5 & 10 & 9 & 4 \end{pmatrix} \bmod 11.$$

Now Alice will send encrypted message to Bob using Encryption scheme that is $W = MK$.

$$W = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 5 & 2 & 4 & 7 & 6 & 1 \\ 8 & 3 & 2 & 1 & 2 & 5 & 7 \\ 9 & 2 & 1 & 3 & 5 & 9 & 2 \\ 2 & 1 & 2 & 3 & 5 & 6 & 3 \\ 1 & 9 & 3 & 7 & 8 & 7 & 6 \\ 2 & 4 & 6 & 9 & 10 & 2 & 8 \end{pmatrix} \begin{pmatrix} 2 & 8 & 5 & 10 & 6 & 8 & 10 & 8 \\ 8 & 8 & 2 & 2 & 3 & 5 & 9 & 7 \\ 0 & 8 & 9 & 5 & 4 & 3 & 6 & 3 \\ 3 & 7 & 7 & 2 & 0 & 3 & 8 & 6 \\ 2 & 9 & 5 & 1 & 9 & 7 & 3 & 2 \\ 0 & 0 & 6 & 5 & 5 & 7 & 10 & 9 \\ 2 & 1 & 8 & 9 & 5 & 10 & 9 & 4 \end{pmatrix}$$

$$W = \begin{pmatrix} 10 & 7 & 5 & 3 & 2 & 10 & 7 & 4 \\ 8 & 7 & 7 & 5 & 7 & 3 & 0 & 7 \\ 6 & 4 & 2 & 1 & 0 & 9 & 4 & 9 \\ 2 & 10 & 9 & 8 & 10 & 3 & 8 & 8 \\ 4 & 10 & 4 & 1 & 3 & 0 & 2 & 2 \\ 2 & 0 & 9 & 0 & 6 & 6 & 5 & 5 \\ 0 & 4 & 8 & 3 & 1 & 3 & 0 & 10 \end{pmatrix}.$$

Now Bob will decrypt the message using right modular pseudoinverse of K and get the plaintext M as $M = WK^{\dagger}$.

As,

$$K^{\dagger} = K^T(KK^T)^{-1}$$

so, $K^{\dagger}$ is found to be

$$K^{\dagger} = \begin{pmatrix} 4 & 10 & 2 & 4 & 10 & 7 & 3 \\ 4 & 8 & 5 & 5 & 9 & 3 & 6 \\ 8 & 9 & 3 & 4 & 4 & 4 & 3 \\ 4 & 2 & 2 & 5 & 8 & 10 & 9 \\ 8 & 7 & 4 & 2 & 4 & 9 & 3 \\ 8 & 6 & 9 & 0 & 1 & 1 & 9 \\ 8 & 0 & 5 & 5 & 1 & 1 & 2 \\ 1 & 10 & 9 & 5 & 1 & 7 & 9 \end{pmatrix} \mod 11$$

Decryption will be carried out as follows,

$$M = \begin{pmatrix} 10 & 7 & 5 & 3 & 2 & 10 & 7 & 4 \\ 8 & 7 & 7 & 5 & 7 & 3 & 0 & 7 \\ 6 & 4 & 2 & 1 & 0 & 9 & 4 & 9 \\ 2 & 10 & 9 & 8 & 10 & 3 & 8 & 8 \\ 4 & 10 & 4 & 1 & 3 & 0 & 2 & 2 \\ 2 & 0 & 9 & 0 & 6 & 6 & 5 & 5 \\ 0 & 4 & 8 & 3 & 1 & 3 & 0 & 10 \end{pmatrix} \begin{pmatrix} 4 & 10 & 2 & 4 & 10 & 7 & 3 \\ 4 & 8 & 5 & 5 & 9 & 3 & 6 \\ 8 & 9 & 3 & 4 & 4 & 4 & 3 \\ 4 & 2 & 2 & 5 & 8 & 10 & 9 \\ 8 & 7 & 4 & 2 & 4 & 9 & 3 \\ 8 & 6 & 9 & 0 & 1 & 1 & 9 \\ 8 & 0 & 5 & 5 & 1 & 1 & 2 \\ 1 & 10 & 9 & 5 & 1 & 7 & 9 \end{pmatrix} \mod 11.$$

4.2.1 Known Plaintext Attack

The encryption scheme is given by

$$W = MK$$

Adversary has the knowledge of plaintext and some of the ciphertext. In this case the key will be obtained by left modular multiplication of M that is M^{-1} with W as given below

$$M^{-1}W = K.$$

By using Extended euclidean algorithm, we have found the inverse of matrix M that is given below,

$$M^{-1} = \frac{adjM}{detM}$$

$$M^{-1} = \begin{pmatrix} 0 & 0 & 10 & 0 & 9 & 8 & 8 \\ 8 & 9 & 10 & 10 & 0 & 3 & \\ 9 & 9 & 9 & 1 & 5 & 2 & 7 \\ 9 & 5 & 3 & 3 & 2 & 3 & 3 \\ 9 & 10 & 9 & 8 & 2 & 9 & 1 \\ 1 & 10 & 1 & 9 & 1 & 7 & 7 \\ 2 & 8 & 8 & 2 & 6 & 9 & 7 \end{pmatrix} \mod 11.$$

$$K = \begin{pmatrix} 0 & 0 & 10 & 0 & 9 & 8 & 8 \\ 8 & 9 & 10 & 10 & 0 & 3 & \\ 9 & 9 & 9 & 1 & 5 & 2 & 7 \\ 9 & 5 & 3 & 3 & 2 & 3 & 3 \\ 9 & 10 & 9 & 8 & 2 & 9 & 1 \\ 1 & 10 & 1 & 9 & 1 & 7 & 7 \\ 2 & 8 & 8 & 2 & 6 & 9 & 7 \end{pmatrix} \begin{pmatrix} 10 & 7 & 5 & 3 & 2 & 10 & 7 & 4 \\ 8 & 7 & 7 & 5 & 7 & 3 & 0 & 7 \\ 6 & 4 & 2 & 1 & 0 & 9 & 4 & 9 \\ 2 & 10 & 9 & 8 & 10 & 3 & 8 & 8 \\ 4 & 10 & 4 & 1 & 3 & 0 & 2 & 2 \\ 2 & 0 & 9 & 0 & 6 & 6 & 5 & 5 \\ 0 & 4 & 8 & 3 & 1 & 3 & 0 & 10 \end{pmatrix} \bmod 11.$$

$$K = \begin{pmatrix} 2 & 8 & 5 & 10 & 6 & 8 & 10 & 8 \\ 8 & 8 & 2 & 2 & 3 & 5 & 9 & 7 \\ 0 & 8 & 9 & 5 & 4 & 3 & 6 & 3 \\ 3 & 7 & 7 & 2 & 0 & 3 & 8 & 6 \\ 2 & 9 & 5 & 1 & 9 & 7 & 3 & 2 \\ 0 & 0 & 6 & 5 & 5 & 7 & 10 & 9 \\ 2 & 1 & 8 & 9 & 5 & 10 & 9 & 4 \end{pmatrix} \bmod 11.$$

In the above example, we have applied known plaintext attack on the bigger system. Hence, we have found that given scheme is vulnerable against the known plaintext attack. We can conclude that this scheme is vulnerable for all $n \in \mathbb{Z}_p$.

Chapter 5

Conclusion and Future Work Suggestion

5.1 Conclusion

In this section we will discuss the strengths and weaknesses of encryption scheme based on circulant matrices using pseudoinverses presented by C. Rajarama et al. Private keys are to be taken from the subgroup of $GL(\mathbb{Z}_n, p)$ of circulant matrices. In this scheme mentioned above Diffie-Hellman type key exchange protocol using circulant matrices as the private keys is presented. The public keys U and V are obtained by the modular multiplication of generator matrix of order $(n-1) \times n$ with the private keys of user A and B. Encryption is presented as ElGamal type encryption and decryption using the pseudoinverse of common session key K.

The use of circulant matrices in the above mentioned scheme helped in key generation as well as in encryption/ decryption and served the purpose of less computational cost. Circulant matrices are also used as keys to provide multi-stage proxy re-encryption. In the given scheme modular multiplication is used to manipulate private and public keys that has made the scheme faster as compared to modular exponentiation, furthermore vulnerability of circulant matrices is observed. In a circulant matrix, first row is to be chosen independently but the succeeding rows can be obtained by the circular shift of preceeding rows as discussed in Section 2.6.1. Hence the author has used the effective key length of matrices. For example the key length of a matrix of size $n \times n$ is $n \times n \times m$, where m is the length of individual element in bits. The main advantage of these matrices is that they help in utilizing the less memory space. In Chapter 4, we have successfully done the cryptanalysis of the given scheme using right modular inverses. Hence it is proved that the given scheme is vulnerable against known plaintext attack.

In the next section, we will suggest the future work that will describe how we upgrade the level of hardness against plaintext attack and brute force attack in the given encryption scheme that is $W = MK$.

5.1.1 (Modified Work)

Modified work of this scheme is discussed below:

5.1.2 Key Generation Algorithm

1. Let G be the public generator matrix, rank of G to be chosen $(n-1)$.

2. A and B are circulant matrices of order $n \times n$ being the private keys of Alice and Bob chosen from $GL(n, p)$.

3. Alice will compute Public key as

$$U = GA.$$

4. Bob will compute Public key as

$$V = GB.$$

Here U, V, G, n and p are public while A, B are held to be private.

Now Alice will generate its common session key by computing its private key with public key of Bob and vice versa.

Alice :

$$K_{AB} = GAB.$$

Since, A and B are circulant matrices, they are being commutative such that, $AB = BA$

Bob:

$$= GBA = K_{BA}.$$

$$K = K_{AB} = K_{BA}.$$

Therefore, we will use K as a common session key in further calculations, such as for encrypting a plaintext, to decrypt a ciphertext.

5.1.3 Encryption and Decryption Algorithm

With the help of common session key K. Alice performs encryption in a way as follows:

1. Alice computes pseudoinverse of K of order $m \times n$ as $K^{\dagger}$.
2. Alice will compute ciphertext as:

$$W = K^{\dagger}MK. \tag{5.1}$$

3. Bob accepts ciphertext and decrypt it by taking pseudo inverse of common session key K. Decryption will be taken out as:

$$KW = KK^{\dagger}MK.$$

$$KWK^{\dagger} = IMKK^{\dagger}.$$

$$KWK^{\dagger} = IMI.$$

Hence,

$$M = (K^{\dagger})^{\dagger}WK^{\dagger}.$$

Since,

$$(K^{\dagger})^{\dagger} = I.$$

$$KWK^{\dagger} = M.$$

5.1.4 ALgorithm for Proxy Re-encryption

In this subsection we will discuss Proxy re-encryption carried out by proxy server $B \rightarrow C$ as described in section 3.5. A and B are private keys of Alice and Bob, U and V are public keys, $(U_B C)$ is a public key sent by Alice to PSBC.

1. Alice will compute its Public key as

$$U_{BC} = U = GA.$$

2. Alice will encrypt the text as

$$W = MVA.$$

3. Alice will send (U, W) to user B and to Proxy server $B \rightarrow C$

4. Bob will decrypt the encrypted data (U, W) as

$$M = W(UB)^{\dagger}.$$

5. Proxy serve $B \rightarrow C$ will re-encrypt the encrypted text W as W_{BC} such that it can be decoded by user C only.

$$W_{BC} = MGAC.$$

$$W_{BC} = M(U_{BC})C.$$

6. As per the suggested future work for the given scheme, proxy re-encryption by PSBC is carried out as given below.

$$W_{BC} = (U_{BC}C)^{\dagger}M(U_{BC}C). \tag{5.2}$$

7. Decryption by user C is carried out as

$$M = ((U_{BC}C)^{\dagger})^{\dagger}W_{BC}(U_{BC}C)^{\dagger}.$$

$$M = (U_{BC}C)W_{BC}(U_{BC}C)^{\dagger}. \tag{5.3}$$

Hence, we have obtained M by taking right modular pseudoinverse of $U_{BC}C$.

5.1.5 Correctness

The correctness of above mentioned asymmetric cipher can be recognized by the help of following demonstration

From Section 5.1.3 we have

$$W = K^{\dagger}MK$$
$$W_{n\times n} = K^{\dagger}_{n\times n-1}M_{n-1\times n-1}K_{n-1\times n}.$$

Multiplying $K_{n-1\times n}$ on both sides, we get

$$K_{n-1\times n}W_{n\times n} = K_{n-1\times n}K^{\dagger}_{n\times n-1}M_{n-1\times n-1}K_{n-1\times n}.$$

$$K_{n-1\times n}W_{n\times n} = I_{n-1\times n-1}M_{n-1\times n-1}K_{n-1\times n}.$$

Right multiplication of $K^{\dagger}_{n\times n-1}$ will give us,

$$K_{n\times m}W_{n\times n}K^{\dagger}_{n\times n-1} = I_{n-1\times n-1}M_{n-1\times n-1}K_{n-1\times n}K^{\dagger}_{n\times n-1}.$$

$$K_{n-1\times n}W_{n\times n}K^{\dagger}_{n\times n-1} = I_{n-1\times n-1}M_{n-1\times n-1}I_{n\times n}.$$

Hence,

$$M_{n\times n} = K_{n-1\times n}W_{n\times n}K^{\dagger}_{n\times n-1}.$$

5.1.6 Illustrative Examples

Let us consider some examples to demonstrate the proof of $W = K^{\dagger}MK$ and $W = (U^{\dagger}_{BC}C)M(U_{BC}C)$ using the parameters given in the scheme proposed by C.Rajarama et al.

Example 5.1.1. Let $n = 4$ The value of p is taken as 23. Matrices G, A and B are chosen as in Example 3.3.1

$$G = \begin{pmatrix} 10 & 4 & 11 & 3 \\ 7 & 9 & 11 & 10 \\ 3 & 6 & 8 & 0 \end{pmatrix}, A = \begin{pmatrix} 10 & 1 & 3 & 3 \\ 3 & 10 & 1 & 3 \\ 3 & 3 & 10 & 1 \\ 1 & 3 & 3 & 10 \end{pmatrix}, B = \begin{pmatrix} 3 & 15 & 6 & 3 \\ 3 & 3 & 15 & 6 \\ 6 & 3 & 3 & 15 \\ 15 & 6 & 3 & 3 \end{pmatrix}$$

$$K_B = \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix}, \text{and } M = \begin{pmatrix} 9 & 10 & 10 & \\ 9 & 7 & 6 & \\ 10 & 6 & 2 & \end{pmatrix}$$

Encryption:

From Equation 5.1 we have,

The encryption of our suggested future work will be conducted as follows:

$$W = K^{\dagger}MK. \tag{5.4}$$

Here $K^{\dagger}$ is computed as described in Equation 3.6

$$K^{\dagger} = K^T(KK^T)^{-1}.$$

From Equation 3.4.2

$$K^{\dagger} = \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix}$$

Putting values of $K^{\dagger}$, M and K in Equation 5.2

$$W = \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix} \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 77 & 60 & 40 \\ 412 & 328 & 248 \\ 195 & 156 & 122 \\ 74 & 72 & 64 \end{pmatrix} \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 8 & 14 & 17 \\ 21 & 6 & 18 \\ 11 & 18 & 7 \\ 5 & 3 & 18 \end{pmatrix} \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 226 & 613 & 511 & 602 \\ 336 & 519 & 495 & 777 \\ 216 & 508 & 500 & 581 \\ 196 & 414 & 297 & 399 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 3 & 15 & 5 & 4 \\ 14 & 13 & 12 & 18 \\ 9 & 2 & 17 & 6 \\ 12 & 0 & 21 & 8 \end{pmatrix} \mod 23.$$

Decryption:

From Equation 5.1 we can get M as

$$M = K_B W (K_B)^{\dagger} \tag{5.5}$$

Now putting values of W, (K_B) and $(K_B)^{\dagger}$ in Equation 5.3

$$M = \begin{pmatrix} 8 & 3 & 9 & 21 \\ 4 & 19 & 18 & 14 \\ 8 & 19 & 11 & 14 \end{pmatrix} \begin{pmatrix} 3 & 5 & 15 & 4 \\ 14 & 13 & 12 & 18 \\ 9 & 2 & 17 & 6 \\ 12 & 0 & 21 & 18 \end{pmatrix} \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 399 & 177 & 670 & 308 \\ 608 & 343 & 848 & 578 \\ 557 & 389 & 749 & 552 \end{pmatrix} \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 8 & 16 & 3 & 9 \\ 10 & 21 & 20 & 3 \\ 5 & 21 & 13 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 & 5 \\ 12 & 16 & 16 \\ 5 & 10 & 6 \\ 6 & 0 & 2 \end{pmatrix} \mod 23$$

$$= \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \mod 23.$$

Hence the obtained message is same as sended by Bob.

5.1.7 Proxy Re-encryption

The encryption and decryption of the scheme is defined in Equation 5.2 and 5.3, we will further prove the working of suggested future work for proxy re-encryption by the following illustrative example.

Example 5.1.2. The values of n, p, G, A and B will be taken same as in Equation 5.1.1 shared secret key between user B and user C is given in Section 3.5.5 as

$$U_{BC}C = \begin{pmatrix} 21 & 21 & 10 & 20 \\ 17 & 11 & 3 & 5 \\ 20 & 11 & 5 & 11 \end{pmatrix}$$

Pseudoinverse of $U_{BC}C$ is given as

$$(U_{BC}C)^{\dagger} = \begin{pmatrix} 2 & 9 & 18 \\ 8 & 13 & 9 \\ 10 & 9 & 14 \\ 11 & 0 & 21 \end{pmatrix}$$

Encryption:

From Equation 5.2

$$W_{BC} = (U_{BC}C)^{\dagger} M (U_{BC}C)$$

Putting the values of M, $(U_{BC}C)$ and $(U_{BC}C)^{\dagger}$ in above equation

$$W_{BC} = \begin{pmatrix} 2 & 9 & 18 \\ 8 & 13 & 9 \\ 10 & 9 & 14 \\ 11 & 0 & 21 \end{pmatrix} \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \begin{pmatrix} 21 & 21 & 10 & 20 \\ 17 & 11 & 3 & 5 \\ 20 & 11 & 5 & 11 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 279 & 191 & 110 \\ 279 & 225 & 176 \\ 311 & 247 & 182 \\ 309 & 236 & 152 \end{pmatrix} \begin{pmatrix} 21 & 21 & 10 & 20 \\ 17 & 11 & 3 & 5 \\ 20 & 11 & 5 & 11 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 3 & 7 & 18 \\ 3 & 18 & 15 \\ 12 & 17 & 21 \\ 10 & 6 & 14 \end{pmatrix} \begin{pmatrix} 21 & 21 & 10 & 20 \\ 17 & 11 & 3 & 5 \\ 20 & 11 & 5 & 11 \end{pmatrix} \bmod 23$$

$$= \begin{pmatrix} 542 & 338 & 141 & 293 \\ 669 & 426 & 159 & 315 \\ 961 & 670 & 276 & 556 \\ 592 & 430 & 188 & 384 \end{pmatrix} \bmod 23$$

$$W_{BC}=\begin{pmatrix}13 & 16 & 3 & 17\\ 2 & 12 & 21 & 16\\ 18 & 3 & 0 & 4\\ 17 & 16 & 4 & 16\end{pmatrix} \bmod 23.$$

Decryption:

From Equation5.2, we have

$$M = (U_{BC}C)W_{BC}(U_{BC}C)^{\dagger} \tag{5.6}$$

To find out the value of plaintext we will put the values of W_{BC}, shared secret key that is $(U_{BC}C)$ and right modular pseudoinverse of shared secret key that is $(U_{BC}C)^{\dagger}$ in above equation,

$$M = \begin{pmatrix}21 & 21 & 10 & 20\\ 17 & 11 & 3 & 5\\ 20 & 11 & 5 & 11\end{pmatrix}\begin{pmatrix}13 & 16 & 3 & 17\\ 2 & 12 & 21 & 16\\ 18 & 3 & 0 & 4\\ 17 & 16 & 4 & 16\end{pmatrix}\begin{pmatrix}2 & 9 & 18\\ 8 & 13 & 9\\ 10 & 9 & 14\\ 11 & 0 & 21\end{pmatrix} \bmod 23$$

$$= \begin{pmatrix}835 & 938 & 584 & 1053\\ 382 & 493 & 302 & 557\\ 559 & 643 & 335 & 712\end{pmatrix}\begin{pmatrix}2 & 9 & 18\\ 8 & 13 & 9\\ 10 & 9 & 14\\ 11 & 0 & 21\end{pmatrix} \bmod 23.$$

$$= \begin{pmatrix}7 & 18 & 9 & 18\\ 14 & 10 & 3 & 5\\ 7 & 22 & 13 & 22\end{pmatrix}\begin{pmatrix}2 & 9 & 18\\ 8 & 13 & 9\\ 10 & 9 & 14\\ 11 & 0 & 21\end{pmatrix} \bmod 23.$$

$$= \begin{pmatrix}446 & 378 & 792\\ 193 & 283 & 489\\ 562 & 466 & 968\end{pmatrix} \bmod 23.$$

$$M = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \bmod 23.$$

Finally Alice recieved the original message sent by Bob.

5.2 Security Analysis and Future Work

In the above section we have suggested the future work that will transform the linear system of equations into complex and non-linear system of equations. In this section we will show that plaintext attack is infeasible on suggested work.

First of all we will recall the public key cryptosystem given in the above mentioned research paper, that is $W = MK$. Here in this scheme we have applied known plaintext attack and retrieved the shared secret key K. The shared secret key of a proxy server (PSBC) is also found vulnerabale against known plaintext attack. The encryption of PSBC is given by $W_{BC} = M(U_{BC}C)$, by knowing M and W, we have calculated $K_{BC} = U_{BC}C$ by using matrix inversion method that is described in section 4.1. Our suggested future work is presented as $W = K^{\dagger}MK$, we suggested left modular multiplication of matrix $K^{\dagger}$ that is pseudoinverse of a rectangular matrix K with the original scheme that was given as $W = MK$. The security of proposed modified work against different attacks is discussed below.

5.2.1 Algebraic Attacks

Algebraic attacks is a kind of cryptanalysis in which attacker reduces the whole system of matrices into system of equations.

1. **Known Plaintext Attack:**

   In this attack, the attacker has knowledge about ciphertext and its corresponding plaintext. On the basis of this data he will try to recover the shared secret key K.

Example 5.2.1. Consider the plaintext, ciphertext pair (M, W) of Example 5.1.1 that is

$$M = \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \text{ and } W = \begin{pmatrix} 3 & 15 & 5 & 4 \\ 14 & 13 & 12 & 18 \\ 9 & 2 & 17 & 6 \\ 12 & 0 & 21 & 8 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 15 & 5 & 4 \\ 14 & 13 & 12 & 18 \\ 9 & 2 & 17 & 6 \\ 12 & 0 & 21 & 8 \end{pmatrix} = \begin{pmatrix} k_{11}^{'} & k_{12}^{'} & k_{13}^{'} \\ k_{21}^{'} & k_{22}^{'} & k_{23}^{'} \\ k_{31}^{'} & k_{32}^{'} & k_{33}^{'} \\ k_{41}^{'} & k_{42}^{'} & k_{43}^{'} \end{pmatrix} \begin{pmatrix} 9 & 10 & 10 \\ 9 & 7 & 6 \\ 10 & 6 & 2 \end{pmatrix} \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{pmatrix}.$$

Above system of matrices will be reduced into system of equations as follows,

$$\begin{aligned}
&(9k_{11}^{'} + 9k_{12}^{'} + 10k_{13}^{'})k_{11} + (10k_{11}^{'} + 7k_{12}^{'} + 6k_{13}^{'})k_{21} + (10k_{11}^{'} + 6k_{12}^{'} + 2k_{13}^{'})k_{31} = 3 \\
&(9k_{11}^{'} + 9k_{12}^{'} + 10k_{13}^{'})k_{12} + (10k_{11}^{'} + 7k_{12}^{'} + 6k_{13}^{'})k_{22} + (10k_{11}^{'} + 6k_{12}^{'} + 2k_{13}^{'})k_{32} = 15 \\
&(9k_{11}^{'} + 9k_{12}^{'} + 10k_{13}^{'})k_{13} + (10k_{11}^{'} + 7k_{12}^{'} + 6k_{13}^{'})k_{23} + (10k_{11}^{'} + 6k_{12}^{'} + 2k_{13}^{'})k_{33} = 5 \\
&(9k_{11}^{'} + 9k_{12}^{'} + 10k_{14}^{'})k_{14} + (10k_{11}^{'} + 7k_{12}^{'} + 6k_{13}^{'})k_{24} + (10k_{11}^{'} + 6k_{12}^{'} + 2k_{13}^{'})k_{34} = 4 \\
&(9k_{21}^{'} + 9k_{22}^{'} + 10k_{23}^{'})k_{11} + (10k_{21}^{'} + 7k_{22}^{'} + 6k_{23}^{'})k_{21} + (10k_{21}^{'} + 6k_{22}^{'} + 2k_{23}^{'})k_{31} = 14 \\
&(9k_{21}^{'} + 9k_{22}^{'} + 10k_{23}^{'})k_{12} + (10k_{21}^{'} + 7k_{22}^{'} + 6k_{23}^{'})k_{21} + (10k_{21}^{'} + 6k_{22}^{'} + 2k_{23}^{'})k_{32} = 13 \\
&(9k_{21}^{'} + 9k_{22}^{'} + 10k_{23}^{'})k_{13} + (10k_{21}^{'} + 7k_{22}^{'} + 6k_{23}^{'})k_{23} + (10k_{21}^{'} + 6k_{22}^{'} + 2k_{23}^{'})k_{33} = 12 \\
&(9k_{21}^{'} + 9k_{22}^{'} + 10k_{23}^{'})k_{14} + (10k_{21}^{'} + 7k_{22}^{'} + 6k_{23}^{'})k_{24} + (10k_{21}^{'} + 6k_{22}^{'} + 2k_{23}^{'})k_{34} = 18 \\
&(9k_{31}^{'} + 9k_{32}^{'} + 10k_{33}^{'})k_{11} + (10k_{31}^{'} + 7k_{32}^{'} + 6k_{33}^{'})k_{21} + (10k_{31}^{'} + 6k_{32}^{'} + 2k_{33}^{'})k_{31} = 9 \\
&(9k_{31}^{'} + 9k_{32}^{'} + 10k_{33}^{'})k_{12} + (10k_{31}^{'} + 7k_{32}^{'} + 6k_{33}^{'})k_{22} + (10k_{31}^{'} + 6k_{32}^{'} + 2k_{33}^{'})k_{32} = 2 \\
&(9k_{31}^{'} + 9k_{32}^{'} + 10k_{33}^{'})k_{13} + (10k_{31}^{'} + 7k_{32}^{'} + 6k_{33}^{'})k_{23} + (10k_{31}^{'} + 6k_{32}^{'} + 2k_{33}^{'})k_{33} = 17 \\
&(9k_{31}^{'} + 9k_{32}^{'} + 10k_{33}^{'})k_{14} + (10k_{31}^{'} + 7k_{32}^{'} + 6k_{33}^{'})k_{24} + (10k_{31}^{'} + 6k_{32}^{'} + 2k_{33}^{'})k_{34} = 6 \\
&(9k_{41}^{'} + 9k_{42}^{'} + 10k_{43}^{'})k_{11} + (10k_{41}^{'} + 7k_{42}^{'} + 6k_{43}^{'})k_{21} + (10k_{41}^{'} + 6k_{42}^{'} + 2k_{43}^{'})k_{31} = 12 \\
&(9k_{41}^{'} + 9k_{42}^{'} + 10k_{43}^{'})k_{12} + (10k_{41}^{'} + 7k_{42}^{'} + 6k_{43}^{'})k_{22} + (10k_{41}^{'} + 6k_{42}^{'} + 2k_{43}^{'})k_{32} = 0 \\
&(9k_{41}^{'} + 9k_{42}^{'} + 10k_{43}^{'})k_{13} + (10k_{41}^{'} + 7k_{42}^{'} + 6k_{43}^{'})k_{23} + (10k_{41}^{'} + 6k_{42}^{'} + 2k_{43}^{'})k_{33} = 21 \\
&(9k_{41}^{'} + 9k_{42}^{'} + 10k_{43}^{'})k_{14} + (10k_{41}^{'} + 7k_{42}^{'} + 6k_{43}^{'})k_{24} + (10k_{41}^{'} + 6k_{42}^{'} + 2k_{43}^{'})k_{34} = 8,
\end{aligned}$$

where, we can get values of $k_{11}^{'}, k_{12}^{'}, ..., k_{43}^{'}$ as $K^{\dagger} = K^T(KK^T)^{-1}$.

Above system of equations clearly depicting that on converting the system of matrices into system of equations, we get 12 unknowns and 16 equations that is number of unknowns are less than number of equations and in the result this system of equations will yield a non linear homogeneous system. Thus the suggested work would make it difficult to hack the key due to non linear complex system of equations with raised power of p.

Therefore our proposed future work is comparatively better than the previous one because that was easily breakable by using right modular pseudoinverses. After the application of new encryption scheme we get a complex system of non linear equations, that makes hard for adversary to solve system of non linear equations. Hence the key is relatively secured and in consequence of this the known plaintext attack would not be feasible. To prove our claim we will present a simplest possible case that is given below.

In general the scheme is presented as follows:

Let K be the unknown key of order 3×4

$$K = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{pmatrix} \bmod 23$$

With its pseudoinverse $K^{\dagger}$ given as

$$K^{\dagger} = \begin{pmatrix} k_{11}^{'} & k_{12}^{'} & k_{13}^{'} \\ k_{21}^{'} & k_{22}^{'} & k_{23}^{'} \\ k_{31}^{'} & k_{32}^{'} & k_{33}^{'} \\ k_{41}^{'} & k_{42}^{'} & k_{43}^{'} \end{pmatrix} \bmod 23$$

and $W = K^{\dagger}MK \bmod 23$

$$W_{n\times n} = K^{\dagger}_{n\times m}M_{m\times m}K_{m\times n},$$

where $m, n \in \mathbb{Z}_p$. In this system of non-linear equations, number of unknowns are $m \times n$, number of equations are n^2 and degree of the system is claimed to be $2n$.

Recall that

$$K^{\dagger} = K^T(KK^T)$$

So all $K_{ij}^{'}$ depends on K_{ij}. Hence the above system is clearly a non-linear system of equations of degree 6 in 12 unknowns. We tried to solve this system of equations by using ApCoCoA without any success.

As the working field is Z_{23}, we believe that there will be almost 23 possibilities for each $K_{ij}(i = 1, 2, 3, j = 1, 2, ., .4)$. Since the system is homogenous, the number of possible solutions for K will be almost $12(23) = 276$. The next step will be to find true K from all the possible solutions.

This is clearly hard to find K when the size of the field is very large such that of the order 2^{64} (for example).

2. **Brute Force Attack:**

   In this attack, attacker would try to guess every feasible guess for the key. As for the simplest possible case under modulo prime 11, attacker would have to guess 20 pairs of key out of which only 8 will satisfy that given homegenous equations. To make this attack infeasible we can take the large key length as $p \times q$. When we will increase the value of modular prime number he might have guess larger number of pairs that is infeasible for example for $p = 2^{128}$ would make the guess for pairs of keys with length $(n - 1) \times n$ and larger value of n infeasible. This will result the failure of Brute force attack.

Bibliography

[1] C. J. Monico, “Semirings and semigroup actions in public-key cryptography,” vol. 1, no. 32, pp. 265–274, 2002.

[2] T. Satoh and K. Araki, “On construction of signature scheme over a certain non-commutative ring,” *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 80, no. 1, pp. 40–45, 1997.

[3] B. Acharya, S. K. Panigrahy, S. K. Patra, and G. Panda, “Image encryption using advanced hill cipher algorithm,” *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, pp. 663–667, 2009.

[4] S. Noureen, “Student privacy preserving framework based on blockchain technology,” Ph.D. dissertation, 2019.

[5] W. Stallings, *Cryptography and network security.* Pearson Education India, 2006, vol. 4, no. 1.

[6] D. Coppersmith, “The data encryption standard (des) and its strength against attacks,” *IBM journal of research and development*, vol. 38, no. 3, pp. 243–250, 1994.

[7] R. C. Merkle and M. E. Hellman, “On the security of multiple encryption,” *Communications of the ACM*, vol. 24, no. 7, pp. 465–467, 1981.

[8] E. Barker and N. Mouha, “Recommendation for the triple data encryption algorithm (tdea) block cipher,” *National Institute of Standards and Technology*, vol. 24, no. 7, pp. 465–467, 2017.

[9] J. Daemen and V. Rijmen, “Reijndael: The advanced encryption standard.” *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, vol. 26, no. 3, pp. 137–139, 2001.

[10] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[11] D. Boneh *et al.*, “Twenty years of attacks on the rsa cryptosystem,” *Notices of the AMS*, vol. 46, no. 2, pp. 203–213, 1999.

[12] C.-H. Ling, S.-M. Chen, and M.-S. Hwang, “Cryptanalysis of tseng-wu group key exchange protocol.” *IJ Network Security*, vol. 18, no. 3, pp. 590–593, 2016.

[13] P. Gaudry, “An algorithm for solving the discrete log problem on hyperelliptic curves,” vol. 1807, no. 2, pp. 203–213, 2000.

[14] C. Meshram, S. A. Meshram, and M. Zhang, “An id-based cryptographic mechanisms based on gdlp and ifp,” *Information Processing Letters*, vol. 112, no. 19, pp. 753–758, 2012.

[15] C. Rajarama, J. N. Sugatoor, and T. Y. Swamy, “Diffie-hellman type key exchange, elgamal like encryption/decryption and proxy re-encryption using circulant matrices.” *IJ Network Security*, vol. 20, no. 4, pp. 617–624, 2018.

[16] X. Fan and F.-H. Liu, “Proxy re-encryption and re-signatures from lattices,” *International Conference on Applied Cryptography and Network Security*, vol. 11464, no. 19, pp. 363–382, 2019.

[17] S. S. Chow, J. Weng, Y. Yang, and R. H. Deng, “Efficient unidirectional proxy re-encryption,” *International Conference on Cryptology in Africa*, vol. 6055, no. 10, pp. 316–332, 2010.

[18] M. S. Iqbal, S. Singh, and A. Jaiswal, “Symmetric key cryptography: Technological developments in the field,” *International Journal of Computer Applications*, vol. 117, no. 15, pp. 325–335, 2015.

[19] H. Delfs and H. Knebl, “Symmetric-key encryption,” vol. 36, no. 10, pp. 11–31, 2007.

[20] M. A. Musa, E. F. Schaefer, and S. Wedig, “A simplified advance encryption standard algorithm and its linear and differential cryptanalyses,” *Cryptologia*, vol. 27, no. 2, pp. 148–177, 2003.

[21] W. G. Barker, “Introduction to the analysis of the data encryption standard (des),” *Aegean Park Press*, vol. 34, no. 3, pp. 160–190, 1991.

[22] C.-W. Shiu, Y.-C. Chen, and W. Hong, “Encrypted image-based reversible data hiding with public key cryptography from difference expansion,” *Signal Processing: Image Communication*, vol. 39, no. 12, pp. 226–233, 2015.

[23] S. Inam and R. Ali, “A new elgamal-like cryptosystem based on matrices over groupring,” *Neural Computing and Applications*, vol. 29, no. 11, pp. 1279–1283, 2018.

[24] S. William, *Cryptography and Network Security: for VTU.* Pearson Education India, 2006.

[25] C. Reutenauer and H. Straubing, “Inversion of matrices over a commutative semiring,” *Journal of Algebra*, vol. 88, no. 2, pp. 350–360, 1984.

[26] J. J. Rotman, *Journey into mathematics: An introduction to proofs*, 1st ed., 2013.

[27] J. Kwapisz, “Rigidity and mapping class group for abstract tiling spaces,” *Ergodic Theory and Dynamical Systems*, vol. 31, no. 6, pp. 17–25, 2011.

[28] C. J. Benvenuto, “Galois field in cryptography,” *University of Washington*, vol. 1, no. 1, pp. 1–11, 2012.

[29] K. Ali, “Noncommutative cryptography using extra special group and galois field,” MPhil Thesis, Capital University of Science and Technology, Islamabad, 2019.

[30] A. J. Menezes, J. Katz, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*, 1st ed. CRC press, 1996.

[31] K. S. McCurley, "The discrete logarithm problem, cryptography and computational number theory," *Proceedings of Symposia in Applied Mathematics*, vol. 42, no. 11, pp. 49–61, 2000.

[32] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," *international conference on the theory and applications of cryptographic techniques*, vol. 37, pp. 127–144, 1998.

[33] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. S. Kang, and C. Park, "New public-key cryptosystem using braid groups," *annual international cryptology conference*, vol. 18, no. 2, pp. 110–123, 2000.

[34] S. Noor, "Cryptographic schemes based on enhanced matrix power function," MPhil Thesis, Capital University of Science and Technology, Islamabad, 2019.

[35] I. Kra and S. R. Simanca, "On circulant matrices," *Notices of the AMS*, vol. 59, no. 3, pp. 368–377, 2012.

[36] C. Kimberling, "A visual euclidean algorithm," *International Journal of Scientific Engineering and Applied Science*, vol. 76, no. 2, pp. 108–119, 1983.

[37] J. Kerl, "Computation in finite fields," *Arizona State University and Lockheed Martin Corporation*, vol. 1, no. 1, pp. 1–12, 2004.

[38] P. Jovanovic and M. Kreuzer, "Algebraic attacks using sat-solvers," *Groups–Complexity–Cryptology*, vol. 2, no. 2, pp. 247–259, 2010.

[39] M. R. Albrecht, C. Cid, L. Grassi, and D. Khovratovich, "Algebraic cryptanalysis of stark-friendly designs: application to marvellous and mimc," *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 13, no. 9, pp. 371–382, 2019.

[40] B. Acharya, G. S. Rath, S. K. Patra, and S. K. Panigrahy, "Novel methods of generating self-invertible matrix for hill cipher algorithm," *International Journal of Security*, vol. 1, no. 3, pp. 20–33, 2007.

[41] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[42] A. Ben-Israel and T. Greville, “Generalized inverses for non invertible matrices,” *Journal of Cryptology*, vol. 31, no. 8, pp. 73–85, 2003.

[43] C. A. Klein and C.-H. Huang, “Review of pseudoinverse control for use with kinematically redundant manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics*, no. 2, pp. 245–250, 1983.

[44] E. H. Moore, “On the reciprocal of the general algebraic matrix,” *Bull. Am. Math. Soc.*, vol. 26, no. 12, pp. 394–395, 1920.

[45] J. Koliha, “The drazin and moore-penrose inverse,” *Mathematical Proceedings of the Royal Irish Academy*, vol. 6, no. 10, pp. 17–27, 1999.

[46] G. H. Golub, “Cf vanloan, matrix computations,” *The Johns Hopkins*, vol. 113, no. 10, pp. 23–36, 1996.

[47] S. Ayinde and R. Ogunrinde, “On error analysis comparison of some numerical experimental results,” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 41, no. 2, pp. 351–364, 2007.

[48] G. Matsaglia and G. PH Styan, “Equalities and inequalities for ranks of matrices,” *Linear and multilinear Algebra*, vol. 2, no. 3, pp. 269–292, 1974.

[49] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications*, 2nd ed. Springer, 2003.

[50] P. Ananth and Chandran, “Achieving privacy in verifiable computation with multiple servers–without fhe and without pre-processing,” *International journal of Cryptography*, vol. 83, no. 3, pp. 149–166, 2014.