CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Asymmetric Cryptographic Schemes Based on Noncommutative Structures

by

Shamsa Kanwal

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

in the

Faculty of Computing

Department of Mathematics

2019

Asymmetric Cryptographic Schemes Based on Noncommutative Structures

By

Shamsa Kanwal

(PA-131001)

Dr. Ayesha Khalid

Queen's University Belfast, UK

(Foreign Evaluator 1)

Dr. Leo Zhang

Deakin University, Australia

(Foreign Evaluator 2)

Dr. Rashid Ali

(Thesis Supervisor)

Dr. Muhammad Sagheer

(Head, Department of Mathematics)

Dr. Muhammad Abdul Qadir

(Dean, Faculty of Computing)

DEPARTMENT OF MATHEMATICS

CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

ISLAMABAD

2019

This dissertation is dedicated to

My Father

For earning a good and honest living for us and teaching me that so much could be done with little

My Mother

A strong and gentle soul who taught me to trust in Almighty Allah, believed in hard work and encouraged me to believe in myself

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD
Expressway, Kahuta Road, Zone-V, Islamabad
Phone:+92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: https://www.cust.edu.pk

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled "**Asymmetric Cryptographic Schemes Based on Noncommutative Structures**" was conducted under the supervision of **Dr. Rashid Ali**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Mathematics, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Mathematics**. The open defence of the thesis was conducted on **January 14, 2019**.

Student Name : Ms. Shamsa Kanwal (PA131001) ____________

The Examining Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a) External Examiner 1: Dr. Tariq Shah
Professor
QAU, Islamabad
Tariq Sh

(b) External Examiner 2: Dr. Muhammad Ashiq
Associate Professor
MCS, NUST, Islamabad
M. Ashiq

(c) Internal Examiner : Dr. Dur-e-Shahwar Sagheer
Assistant Professor
Capital University of Science & Technology, Islamabad
Dur-e-Sh

Supervisor Name : Dr. Rashid Ali
Assistant Professor
Capital University of Science & Technology, Islamabad

Name of HoD : Dr. Muhammad Sagheer
Professor
Capital University of Science & Technology, Islamabad

Name of Dean : Dr. Muhammad Abdul Qadir
Professor
Capital University of Science & Technology, Islamabad

AUTHOR'S DECLARATION

I, **Ms. Shamsa Kanwal (Registration No. PA131001)**, hereby state that my PhD thesis titled, '**Asymmetric Cryptographic Schemes Based on Noncommutative Structures**' 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/ world.

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

(Ms. Shamsa Kanwal)

Dated: 14, January, 2019

Registration No : PA131001

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Asymmetric Cryptographic Schemes Based on Noncommutative Structures**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly 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 PhD Degree, the University reserves the right to withdraw/ revoke my PhD 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 thesis.

(Ms. Shamsa Kanwal)
Registration No. : PA131001

Dated: 14, January, 2019

List of Publications

It is certified that following publication has been made out of the research work that has been carried out for this thesis:-

1. **S. Kanwal** and R. Ali, “A cryptosystem with noncommutative platform groups”, *Neural Computing and Application*, volume 29, issue 11, 1273-1278, June 2018.

Shamsa Kanwal
(PA-131001)

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful.

All praise to Almighty Allah, the Creator of knowledge, with Whose grace I have been able to accomplish the task assigned to me.

There are a lot of people without whom this research and thesis might not have been completed. First of all, I pay my gratitude to all my teachers for bringing me this stage of academic zenith.

Foremost, special appreciation goes to my research supervisor Dr. Rashid Ali, Department of Mathematics, Capital University of Science and Technology, Islamabad, Pakistan, for his supervision and constant support. His invaluable help of constructive comments and suggestions throughout the research phase and thesis write up, have contributed towards the success of this research.

I would like to express my gratitude to honorable vice chancellor Prof. Dr. Muhammad Mansoor Ahmed, Capital University of Science and Technology, Islamabad, Pakistan, for providing me with the financial help in the form of university scholarship. My sincere acknowledgment is owed to Dr. Muhammad Sagheer, Head Department of Mathematics, Capital University of Science and Technology, Islamabad, Pakistan, for helping towards my graduate affairs and for providing ideal atmosphere of study and research in the department.

At this stage, I think of my parents whose selfless sacrificial life and their great efforts with unceasing prayers have enabled me to reach the present position in life. I would never be able to pay back the love and affection showered upon by my parents. I am especially indebted to my M.Phil supervisor Dr. Naseer Ahmed (late), Quaid-i-Azam University, Islamabad, who gave me professional guidance and taught me a great deal about both scientific research and life in general. I express my appreciation and deep sense of gratitude from the core of my heart to my husband.

My acknowledgement cannot be completed without appreciating my children for abiding my ignorance and the patience they showed during the period of my Ph.D. Words would never say how grateful I am to them. I consider myself the luckiest in the world to have such a lovely and caring family, standing beside me with their love and unconditional support. A special note of gratitude to all my relatives for the love, concern and encouragement.

Collective and individual acknowledgments are also owed to all my friends and colleagues specially, Dr. Sadia Hina, Head Department of Mathematical Sciences, Dr. Munazza Naz, Dr. Afshan Batool, Ms. Saba Inam and Ms. Bushra Kanwal, Lecturers Department of Mathematical Sciences, Fatima Jinnah Women University, Rawalpindi, Pakistan. They were always beside me during the happy and hard moments to push me and motivate me.

Finally, I thank all those who have helped me directly or indirectly in the successful completion of my thesis. Anyone missed in this acknowledgement are also thanked. If I did not mention someones name here, it does not mean that I do not acknowledge your support and help. Again, I would like to thank everyone who supported and helped me during my Ph.D study, in any way.

Shamsa Kanwal

Abstract

Asymmetric cryptography based on groups is mainly concerned with the role of noncommutative groups. The origin of group based cryptography goes back in the 1980s. Since then, numerous cryptographic proposals based on noncommutative groups have been evolved. In this thesis, we consider several noncommutative settings in different cryptographic context. This study adds to the cryptographic literature by demonstrating new asymmetric cryptographic schemes. On the one hand, two new public key exchange protocols and two asymmetric cryptosystems are constructed. On the other hand, different primitives like suggestion of platform, size of different parameters, security and efficiency aspects regarding these proposals, are also elaborated.

Contents

Author's Declaration v

Plagiarism Undertaking vi

List of Publications vii

Acknowledgements viii

Abstract x

List of Figures xiv

List of Tables xv

Abbreviations xvi

Symbols xviii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Motivation . . . 3
- 1.3 Contribution of the Thesis . . . 3
- 1.4 Organization of the Thesis . . . 4

2 Mathematical Background 6
- 2.1 Basic Concepts in Number Theory . . . 6
  - 2.1.1 Obtaining the Greatest Common Divisor . . . 8
  - 2.1.2 Modular Arithmetic . . . 9
  - 2.1.3 Properties of Modular Arithmetic . . . 10
- 2.2 Some Algebraic Structures . . . 14
  - 2.2.1 Finite Fields . . . 21
- 2.3 Polynomial Arithmetic . . . 22
  - 2.3.1 Ordinary Polynomial Arithmetic . . . 23
  - 2.3.2 Polynomial Arithmetic with Coefficients in $\mathbb{Z}_p$ . . . 24
- 2.4 Finite Fields of the Form $GF(2^n)$ . . . 26

2.4.1 Modular Polynomial Arithmetic 29
2.4.2 Multiplicative Inverse of a Polynomial 30

3 Cryptographic Background 32
3.1 Basic Cryptographic Primitives 32
3.1.1 Characterization of Cryptosystems 33
3.1.2 Cryptanalysis and Techniques of Attacks 35
3.1.3 Cryptographic Applications 36
3.2 Public Key Cryptography 37
3.2.1 The General Model for Public Key Cryptosystem 38
3.3 Some Public Key Protocols 40
3.3.1 Diffie–Hellman Protocol 40
3.3.2 RSA Cryptosystem 41
3.3.3 El-Gamal Cryptosystem 43
3.3.4 Public Key Cryptography Based on Groups 45
3.3.5 Ko–Lee Protocol 47
3.3.6 Anshel–Anshel–Goldfeld Protocol 48

4 A Cryptosystem with Noncommutative Platform Groups 51
4.1 Motivating Literature 52
4.1.1 Shamir's Secret Sharing Scheme 52
4.1.2 The BCFRX Scheme 53
4.1.3 Stickel's Key Exchange Protocol 55
4.2 Proposed Cryptosystem 57
4.2.1 Suggested Platform and Parameters for the Proposed System 60
4.3 Security Analysis 69
4.3.1 Ciphertext Only Attack 70
4.3.2 Known Plaintext Attack 70
4.4 Efficiency Analysis 71
4.4.1 Computational Complexity of Some Matrix Operations . . . 71
4.4.2 Bit-Complexity of Proposed Cryptosystem 73

5 New Variants of Stickel's Key Exchange Protocol 74
5.1 Some Cryptographic Hard Problems and Noncommutativity 74
5.2 Motivating Literature 76
5.2.1 Diffie-Hellman-Like Protocol For Noncommutative Rings . . 77
5.3 Proposed Key Exchange Protocols 78
5.4 Security Aspects of the Proposed Protocols 90

6 A Novel Algebraic Public Key Cryptosystem 93
6.1 Proposed Cryptosystem 93
6.2 Underlying Work Structure and Parameters 96
6.3 Security of Proposed Cryptosystem 104

7 Conclusion and Future Work 106

Bibliography **108**

List of Figures

2.1 Relation Between Different Algebraic Structures 21

3.1 Symmetric Cryptosystem 33
3.2 Asymmetric Cryptosystem 34
3.3 Diffie Hellman Protocol 41
3.4 Key Generation of ElGamal Cryptosystem 44
3.5 Encryption and Decryption of ElGamal Cryptosystem 45

5.1 Stickel's Variant Protocol 80

List of Tables

2.1 The Properties of Modular Arithmetic. 11
2.2 The Multiplicative Inverse Using Extended Eucleadian Algorithm. . 13
2.3 Addition in $\mathbb{Z}_8$. 27
2.4 Multiplication in $\mathbb{Z}_8$. 28
2.5 Addition in $GF(2^3)$. 28
2.6 Multiplication in $GF(2^3)$. 28
2.7 Polynomial Addition Modulo $(x^3 + x + 1)$. 30
2.8 Polynomial Multiplication Modulo $(x^3 + x + 1)$. 30

3.1 Cryptanalytic Attacks. 36

4.1 Bit-Complexity of Basic Modular Operations 71

5.1 The size of set of polynomials of different degrees α and primes p with order of matrices $n = 2$ 85
5.2 The size of set of polynomials of different degrees α and β and $p = 29$ with order of matrices $n = 2$. 90

Abbreviations

AES	Advance Encryption Standard
ApCoCoA	Applied Computations in Commutative Algebra
BCFRX Scheme	Baumslag, Camps, Fine, Rosenberger and Xu Scheme
CSP	Conjugacy Search Problem
DES	Data Encryption Standard
DH	Diffie Hellman
DLP	Discrete Logarithm Problem
DP	Decomposition Problem
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
GDP	Generalized Decomposition Problem
GF	Galois Field
GSDP	Generalized Symmetric Decomposition Problem
IFP	Integer Factorization Problem
KMOV	Komaya, Maurer, Okamoto and Vanston
LUC	Lucas Function
PGDP	Polynomial Generalized Decomposition Problem
PKC	Public Key Cryptography
PSDP	Polynomial Symmetric Decomposition Problem
RC4	Rivest Cipher 4
RSA	Rivest-Shamir-Adleman

SDP	Symmetric Decomposition Problem
WP	Word Problem

Symbols

$\gcd(a, b)$	Greatest Common Divisor of a and b
$\mathbb{Z}$	set of integers
$\mathbb{Z}_{>0}$	set of positive integers
$\mathbb{Z}_n$	set of integers modulo n
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of Real numbers
$\mathbb{C}$	set of Complex numbers
$\mathbb{F}$	Field
G	Group
$Z\left(G\right)$	Center of group G
R	Ring
$R[x]$	Polynomial ring in one indeterminate x over the ring R
$GL(n, R)$	General linear group of matrices of order n over the ring R
$M(n, R)$	set of matrices of order n over the ring R

Chapter 1

Introduction

1.1 Background

The two strategies of keeping our information secret are the *Steganography* and the *Cryptography.* In *Steganography,* the existence of original message is concealed within a covered message. *Cryptography* is the art and science of keeping information unintelligible from unauthorized audiences. Conversely, the art and science of breaking unintelligible form of data is known as *Cryptanalysis.* Both branches Cryptography and Cryptanalysis together are called *Cryptology.*

Different cryptological aspects are indeed based on mathematical grounds. The advent of public key cryptography is a tremendous demonstration of role of mathematics, that brings cryptography to enter a new and exciting phase. Usually, public key cryptography is based on different primitives of commutative algebraic structures. The most pioneer examples are the Diffie-Hellman key exchange protocol [15] and the Rivest-Shamir-Adleman (RSA) algorithm [46]. These two have gained the wider acceptance and their security relies on the discrete logarithm problem (DLP) and integer factorization problem (IFP), respectively, over the commutative structures. A key exchange protocol proposed by Habeeb et. al. [22] similar to the Diffie Hellman protocol, has its own features and some important advantages. The finite cyclic semigroup is used for the protocol. The authors suggested the semigroup of matrices over the groupring $\mathbb{Z}_7[A_5]$. Different modifications of RSA like LUC's scheme (based on Lucas function) [53], Cao's schemes [8, 9], Rabin-Williams [45, 58, 59] schemes and elliptic curve variant of RSA like KMOV (Komaya, Maurer, Okamoto and Vanston) [32] are based on the commutative structures. The underlying platform of the extended multi-dimension RSA cryptosystem [9] is also the commutative structure $\mathbb{Z}_n[x]$. The family of ElGamal like public key cryptosystems is another good example in this regard, including the basic ElGamal cryptosystem [17], elliptic curve cryptosystem, McCurley scheme [38] and variant of ElGamal scheme [24].

Due to Shor [51], Kitaev [30] and Proos-Zalka [44] algorithms, IFP, DLP and DLP over elliptic curves (ECDLP) may efficiently be solved on quantum computer. So, the algorithms based on these problems are believed to be insecure, in future. In order to enrich cryptographic protocols, there are different kinds of group theoretical results [34] which can replace IFP and DLP. Especially, the cryptographic techniques based on noncommutative structures attract more attention. The use of noncommutative groups in public key cryptography was initially proposed by Wagner and Magyarik [36] in 1985. A brief account of group based cryptographic methods is given in the book, titled, 'Group-based Cryptography' by Myasnikov et. al. [43]. For devising new cryptographic and cryptanalytic techniques, the algebraic properties of different noncommutative algebraic structures, play a very important role.

The work of Charalambos [11], based on the structure of grouprings is an important addition to the field of noncommutative cryptography. Various cryptosystems using noncommutative algebraic systems are based on the conjugacy search problem over certain noncommutative structures see, for instance [21, 41]. Although, the conjugacy search problem is meaningful in noncommutative structures, it is inconvenient to design public key cryptosystems over such structures. That is why utilizing noncommutativity is really a challenging problem for developing a public key cryptosystem over these kind of algebraic structures.

1.2 Motivation

Research has been going on using noncommutative groups for public key encryption algorithms, for example see [2, 31]. They use braid groups which have various computationally hard problems. Also, these groups are ideal for achieving implementation efficiency. Side by side attacks on braid group based cryptography have also been published in the literature [23, 35, 42].

E. Stickel [56] proposed a public key exchange scheme using matrices in a certain subgroup of $GL(n, \mathbb{F}_q)$. The methods presented by Stickel, is not a generalization of the classical Diffie-Hellman protocol rather, it is a reminiscent of latter to noncommutative groups. This approach can be used for the purpose of key exchange, as well as, authentication protocols. Stickel suggested the use of general linear group of matrices for his proposal. Different matrices based structures have a great potential to be used as a noncommutative platform group in variety of ways in cryptography. See, for instance [1, 25, 37, 40] and the references therein.

The present study is concerned with the development of cryptographic techniques based on noncommutative algebraic structures. Specifically, we examine the role of some noncommutative groups and rings of matrices as the platform group for devising several new settings in cryptography. Indeed, the focus of our study is on the development of new key exchange protocols and public key cryptosystems. But, we also demonstrate several discussions regarding the security and suggest values of involved parameters of the proposals.

1.3 Contribution of the Thesis

The main contribution of this thesis is as follows:

- Based on Stickel's protocol, we have developed a cryptosystem. We have proved the correctness of the proposed cryptosystem and discussed different related issues like the choice of platform and parameters involved. A brief note on the security analysis of the said system is also given. Our article based on the exposition of this work is published in journal of Neural Computing and Applications [26].

- Keeping in mind the spirit of Stickel's protocol, we have presented two variant key exchange protocols. The underlying work structure for these protocols is the set of polynomials over noncommutative ring. Different aspects including its security and parameter values are also elaborated. This work is submitted for a possible publication.

- Another cryptosystem which uses the polynomials over noncommutative groups as underlying work structure. The useful feature of this cryptosystem is that it provides high security because of the use of inner automorphisms of a noncommutative group. This work is also submitted for a possible publication.

1.4 Organization of the Thesis

The conventional way to read this thesis is of course to read it sequentially. A reader can also adopt the way of reading the Chapter 1 and then continue to read the Chapters 4-6, containing the actual contents. However, the general considerations of each chapter specifies organization of that chapter and the dependency structure of its sections, is also given at the start of each chapter.

Overview of chapters

To read individual chapters, the following information might be helpful:

- Chapter 2 and 3 contain definitions, basic discussions and notations used in the rest of the thesis. These specific definitions and discussions can be read out separately, when needed. All the necessary prerequisites of number theory, algebra and cryptography may be found in many basic books of these subjects. For example, we refer lecture notes [16] and the book [19] for fundamental algebraic foundations. For number theoretic concepts, one can see the reference [7]. The references [28], [39] and Stinson [57] can be seen for cryptographic concepts.

- In Chapter 4, a new cryptosystem with noncommutative groups is presented. The discussion related to different primitives of this cryptosystem is also given.
- Chapter 5 is based on two new variants of Stickel's key exchange protocol. The platform for these protocols is ring of polynomials over noncommutative rings.
- Chapter 6 is related to the study of a new cryptosystem based on the protocol presented in Chapter 5.
- Chapter 7 describes the concluding remarks related to all proposals presented in the thesis. There is also a brief discussion of some new directions for possible future work.

Chapter 2

Mathematical Background

Modular arithmetic and finite fields have become increasingly important in cryptography. Various cryptographic techniques rely deeply upon different properties of modular arithmetic and finite fields. This chapter provides the sufficient background of modular arithmetic and finite fields which enables to comprehend the rest of the cryptographic techniques of this dissertation. We first introduce the basic concepts from modular arithmetic. Then comes a concise overview of some algebraic structures. Next, some background of finite fields, is given. This chapter concludes with a discussion of polynomial arithmetic.

2.1 Basic Concepts in Number Theory

This section provides some background of number theory. The concepts of divisibility and greatest common divisor are elaborated. The division algorithm and the Euclidian algorithm are also discussed in detail.

Definition 2.1.1. (**Divisibility**)
An integer $b \neq 0$ divides an integer a, if

$$a = bc, \text{ for some integer } c.$$

We use the notation $b \mid a$ to say that b divides a. The integer b is called a divisor (factor) of a and a is said to be a multiple of b.

Theorem 2.1.2. (**The Division Algorithm/The Euclid Theorem**)
For any integer $b > 0$ and any integer $a \geqslant 0$, there exist unique integers q and r such that

$$a = bq + r, \qquad 0 \leq r < b. \tag{2.1}$$

In other words, on dividing a by b, we obtain two integers q and r for which the equation (2.1) is satisfied. The integers q and r are known as quotient and remainder, respectively. This is also referred as the division algorithm. The remainder r is referred as a residue under mod b.
From number theory, recall that the integer $c \neq 0$, is said to be a common divisor of a and b if $c \mid a$ and $c \mid b$.

Definition 2.1.3. (**Greatest Common Divisor**)
The integer $d > 0$ is known as the greatest common divisor of a and b, if

1. $d \mid a$ and $d \mid b$.

2. Any other divisor c of a and b, divides d.

We denote the greatest common divisor of a and b by $\gcd(a, b)$.
If $\gcd(a, b) = 1$, then a and b are said to be relatively prime or co-prime. That is equivalently written as $\gcd(a, b) = 1$.

Definition 2.1.4. (**Prime Number**)
An integer $p > 1$ is called a prime number if its only positive divisors are 1 and p.

Example 2.1.5.

1. The greatest common divisor of 24 and 60 or $\gcd(24, 60) = 12$.

2. The integers 25 and 128 are relatively prime because $\gcd(25, 128) = 1$.

2.1.1 Obtaining the Greatest Common Divisor

Euclid developed an algorithm which is used to find out the gcd of two integers. For finding the gcd(a, b), where $a > b$ and $b > 0$, we proceed as follows:
By Euclid's Theorem 2.1.2, there exist unique integers q_1 and r_1 that satisfy the equation

$$a = bq_1 + r_1, \qquad 0 \leq r_1 < b.$$

Here, we have two cases according to two values of remainder r_1.

1. If $r_1 = 0$ then $b = \gcd(a, b)$.

2. If $r_1 \neq 0$ then again by applying Euclid's theorem, we can obtain unique integers q_2 and r_2 that satisfy the relation

$$b = r_1q_2 + r_2, \qquad 0 \leq r_2 < r_1.$$

1. If $r_2 = 0$ then $r_1 = \gcd(a, b)$.

2. If $r_2 \neq 0$ then unique integers q_3 and r_3 exist, such that

$$r_1 = r_2q_3 + r_3, \qquad 0 \leq r_3 < r_2.$$

We repeat this procedure until we obtain a zero remainder r_{m+1} say at $(m + 1)th$ stage. The following set of equations is obtained:

$$\begin{array}{rcll}
a & = & bq_1 + r_1, & 0 \leq r_1 < b, \\
b & = & r_1q_2 + r_2, & 0 \leq r_2 < r_1, \\
r_1 & = & r_2q_3 + r_3, & 0 \leq r_3 < r_2, \\
 & & \vdots & \vdots \\
r_{m-2} & = & r_{m-1}q_m + r_m, & 0 \leq r_m < r_{m-1}, \\
r_{m-1} & = & r_mq_{m+1} + 0, & r_{m+1} = 0.
\end{array}$$

Here, we note the following:

1. $r_m > 0$

2. $r_m \mid a$ and $r_m \mid b$

3. For every iterative step, $d = \gcd(r_j, r_{j+1})$ and finally

$$d = \gcd(r_m, 0) = r_m.$$

By applying the division algorithm, repetitively, the gcd of two integers can be obtained. This procedure is given as the following algorithm:

Algorithm 2.1.6. (**The Euclidean Algorithm**)

Input : Two positive integers a and b.

Output: $\gcd(a, b)$

```
1. $M \longleftarrow a; \quad N \longleftarrow b$
2. if $N = 0$, return $M = \gcd(a, b)$
3. $R \longleftarrow M \mod N$
4. $M \longleftarrow N$
5. $N \longleftarrow R$
6. goto step 2
```

Example 2.1.7.

For finding $\gcd(143, 110)$, we have

$$\begin{aligned} 143 &= 1 \times 110 + 33, \\ 110 &= 3 \times 33 + 11, \\ 33 &= 3 \times 11 + 0. \end{aligned}$$

Therefore, $\gcd(143, 110) = \gcd(110, 33) = \gcd(33, 11) = \gcd(11, 0) = 11$.

2.1.2 Modular Arithmetic

We now give a brief account of modulus and related modular arithmetic concepts.

Definition 2.1.8. (**The Modulus**)

The remainder obtained by dividing an integer a by a positive integer n, is defined as $a \mod n$. We call the integer n as the modulus.

By Theorem 2.1.2, on dividing a by n, we have

$$a = nq + r, \qquad 0 \leq r < n,$$

or

$$r = a \mod n.$$

Two integers a and b are congruent to each other modulo n whenever

$$a \mod n = b \mod n.$$

We write it as

$$a \equiv b \mod n.$$

Remark 2.1.9.

Following are some properties of congruence:

1. $a \equiv b \mod n$ implies $b \equiv a \mod n$.

2. $a \equiv b \mod n$ if $n \mid (a - b)$.

3. $a \equiv b \mod n$ and $b \equiv c \mod n$ imply $a \equiv c \mod n$.

2.1.3 Properties of Modular Arithmetic

Observe that the operator $\mod n$ is a mapping which maps all integers into the set $\{0, 1, \cdots, (n-1)\}$. We can confine ourselves to perform arithmetic operations within this set. These techniques are known as modular arithmetic. Denote the set $\{0, 1, \cdots, (n-1)\}$ of nonnegative integers less than n as $\mathbb{Z}_n$. This set is called the set of residues, or residue classes $\mod n$. Every integer r in $\mathbb{Z}_n$ is a representative of a complete residue class. A residue class corresponding to r is labeled by

$$[r] = \{a \mid a \text{ is an integer with the property } a \equiv r \mod n\}.$$

That is all the residue classes $\mod n$ are $[0], [1], [2], \cdots, [n-1]$. The residue classes $\mod 5$ are

$$\begin{aligned}
[0] &= \{\cdots, -20, -15, -10, -5, 0, 5, 10, 15, 20 \cdots\}, \\
[1] &= \{\cdots, -19, -14, -9, -4, 1, 6, 11, 16, 21 \cdots\}, \\
[2] &= \{\cdots, -18, -13, -8, -3, 2, 7, 12, 17, 22 \cdots\}, \\
[3] &= \{\cdots, -17, -12, -7, -2, 3, 8, 13, 18, 23 \cdots\}, \\
[4] &= \{\cdots, -16, -11, -6, -1, 4, 9, 14, 19, 24 \cdots\}.
\end{aligned}$$

A residue class is represented by the smallest nonnegative integer present in that residue class. For the integers $x_1, x_2, x_3 \in \mathbb{Z}_n$, the basic modular arithmetic operations are given as:

1. $[(x_1 \mod n) + (x_2 \mod n)] \mod n = (x_1 + x_2) \mod n,$
2. $[(x_1 \mod n) - (x_2 \mod n)] \mod n = (x_1 - x_2) \mod n,$
3. $[(x_1 \mod n) \times (x_2 \mod n)] \mod n = (x_1 \times x_2) \mod n.$

The properties of modular arithmetic within $\mathbb{Z}_n$ are described in TABLE 2.1.

Property	Expression
Commutative Laws	$(x_1 + x_2) \mod n = (x_2 + x_1) \mod n$
	$(x_1 \times x_2) \mod n = (x_2 \times x_1) \mod n$
Associative Laws	$[(x_1 + x_2) + x_3] \mod n = [x_1 + (x_2 + x_3)] \mod n$
	$[(x_1 \times x_2) \times x_3] \mod n = [x_1 \times (x_2 \times x_3)] \mod n$
Distributive Law	$[x_1 \times (x_2 + x_3)] \mod n = [(x_1 \times x_2) + (x_1 \times x_3)] \mod n$
Additive Identity	$(0 + x_1) \mod n = x_1 \mod n$
Multiplicative Identity	$(1 \times x_1) \mod n = x_1 \mod n$
Additive Inverse	For each $x \in \mathbb{Z}_n$, there exists a $x' \in \mathbb{Z}_n$ such that $x + x' = 0 \mod n$.

TABLE 2.1: The Properties of Modular Arithmetic.

The modular arithmetic is somehow different from ordinary arithmetic. Note that

$$\text{if } (x_1 + x_2) \equiv (x_2 + x_3) \mod n, \text{ then } x_1 \equiv x_3 \mod n$$

because of existence of the additive inverse. But, the statement for muliplication is true only with an additional condition:

$$\left.\begin{array}{c}\text{if } (x_1 \times x_2) \equiv (x_2 \times x_3) \mod n, \text{ then } x_1 \equiv x_3 \mod n, \\ \text{provided that } x_2 \text{ is relatively prime to } n.\end{array}\right\} \tag{2.2}$$

The Euclidean algorithm in $\mathbb{Z}_n$ relies on the following theorem:

For any integer $a \geqslant 0$ and any integer $b > 0$,

$$\gcd(a, b) = \gcd(b, a \mod b). \tag{2.3}$$

The equation (2.3) can be used repetitively, to find the $\gcd(a, b)$.
For a positive integer $a < n$, if $\gcd(a, n) = 1$ then there exists $a^{-1} \in \mathbb{Z}_n$ such that

$$a^{-1}a = aa^{-1} = 1 \mod n.$$

In order to determine a^{-1}, the Euclidean algorithm can be extended. If $\gcd(a, n) = 1$, the algorithm returns the multiplicative inverse a^{-1} of a modulo n. This algorithm is known as the Extended Euclidean algorithm.

Algorithm 2.1.10. (**The Extended Euclidean Algorithm**)
Input: Two positive integers a and n with $a < n$.
Output: Multiplicative inverse a^{-1} of a modulo n.

```
1. $(U_1, U_2, U_3) \longleftarrow (1, 0, n)$;        $(V_1, V_2, V_3) \longleftarrow (0, 1, a)$
2. If $V_3 = 0$ return $U_3 = \gcd(a, n)$; no inverse
3. If $V_3 = 1$ return $V_3 = \gcd(a, n)$; $V_2 = a^{-1} \mod n$
4. $W = \left\lfloor \frac{U_3}{V_3} \right\rfloor$
5. $(T_1, T_2, T_3) \longleftarrow (U_1 - WV_1, U_2 - WV_2, U_3 - WV_3)$
6. $(U_1, U_2, U_3) \longleftarrow (V_1, V_2, V_3)$
7. $(V_1, V_2, V_3) \longleftarrow (T_1, T_2, T_3)$;
```

8. goto 2.

Example 2.1.11.

To understand the execution of the Algorithm 2.1.10, consider that $\gcd(1759, 550) = 1$. Therefor, the inverse $(550)^{-1} \mod 1759$ exists which can be obtained as shown in TABLE 2.2.

W	U_1	U_2	U_3	V_1	V_2	V_3
	1	0	1759	0	1	550
3	0	1	550	1	3	109
5	1	3	109	5	16	5
21	5	16	5	106	339	4
1	106	339	4	111	355	1

TABLE 2.2: The Multiplicative Inverse Using Extended Eucleadian Algorithm.

Definition 2.1.12. (Euler's Totient Function)

For any integer $n > 0$, the Euler totient function, denoted by $\phi(n)$, is defined as the number of integers a such that $\gcd(a, n) = 1$, where $1 \leq a \leq n$. That is, $\phi(n)$ is the number of integers that are relatively or co-prime with n.

example $\phi(5) = 4$, and $\phi(24) = 8$.

Theorem 2.1.13.

The Euler totient function satisfies the following properties:

1. If $\gcd(a, b) = 1$, then $\phi(ab) = \phi(a)\phi(b)$.

2. $\phi(p^n) = p^n - p^{n-1}$, where p is a prime integer.

3. $\phi(n^k) = n^{k-1}\phi(n)$.

Theorem 2.1.14. (Fermat's Little Theorem)

For any prime p, and any integer $a \not\equiv 0 \mod p$, we have $a^{p-1} \equiv 1 \mod p$. Further, we can also say that for any integer a, we have $a^p \equiv a \mod p$.

Theorem 2.1.15. (Euler's Theorem)

For an integer a which is relatively prime to a positive integer n that is

$$\gcd(a, n) = 1,$$

we have

$$a^{\phi(n)} = 1 \mod n.$$

The Fermat little theorem is the special case of the Euler theorem with n is prime.

2.2 Some Algebraic Structures

Groups, rings, and fields are the elementary structures of abstract algebra or modern algebra. While dealing with abstract algebra, the operations within these structures, are not limited to ordinary arithmetical operations. Perhaps, to combine two elements of the set, different algebraic operations can be defined depending on the nature of the elements of the set. This becomes clear as we proceed further in this section.

Definition 2.2.1. (**Groups**)
A set G with a binary operation $*$, is called a group such that $*$ associates to each order pair $(g_1, g_2) \in G \times G$, an element $(g_1 * g_2)$ of G, provided that the following axioms are satisfied:

$(G1)$ Closure: For all $g_1, g_2 \in G$,

$$(g_1 * g_2) \in G$$

$(G2)$ Associative: For all $g_1, g_2, g_3 \in G$,

$$g_1 * (g_2 * g_3) = (g_1 * g_2) * g_3$$

$(G3)$ Identity Element: For all $g \in G$, there exists an element $e \in G$, such that

$$g * e = e * g = g$$

$(G4)$ Inverse Element: For all $g \in G,$ there exists an element $g' \in G$, called the inverse of g, such that

$$g * g' = g' * g = e.$$

The subsets of a group which are also groups are called subgroups of that group. That is, a subset H of a group $(G, *)$ which contains the identity element $e \in G$ is called a subgroup of G, if it is also a group with the same binary operation $*$. Symbolically, we write it as $H \leq G$.

Definition 2.2.2. (**Order of a Group**)
A group G is referred as finite group if it has a finite number of elements in it. The order $|G|$ of a finite group G is the number of elements it contains.

Definition 2.2.3. (**Commutative Group**)
A group is called commutative or abelian with the following additional condition:
$(G5)$ For all $g_1, g_2 \in G$,

$$g_1 * g_2 = g_2 * g_1.$$

The exponentiation of elements of a group is defined as the repetitive application of the group operator $*$. For example

$$g^3 = g * g * g.$$

Definition 2.2.4. (**Order of an Element of a Group**)
For an element $g \in G$, the smallest positive integer n is called the order of g, denoted by $|g| = n$, such that $g^n = e$ and then g is said to have finite order.

Definition 2.2.5. (**Cyclic Group**)
A group G is said to be cyclic if each element of G can be expressed as a power of a specific fixed element, say $g \in G.$ The element g is known as the generator of the group G. In that case the group G can also be denoted as

$$G = \langle g \rangle \, .$$

Example 2.2.6.

1. The set of complex numbers $\mathbb{C}$, the set of real numbers $\mathbb{R}$, the set of rational numbers $\mathbb{Q}$ and the set of integers $\mathbb{Z}$ form groups with respect to addition. For all these sets 0 is the identity element. Moreover, $\mathbb{Z}$ is subgroup of $\mathbb{Q}$, that is, $\mathbb{Z} \leq \mathbb{Q}$. Similarly, $\mathbb{Q} \leq \mathbb{R} \leq \mathbb{C}$.

2. The set of complex numbers $\mathbb{C}$, the set of real numbers $\mathbb{R}$ and the set of rational numbers $\mathbb{Q}$ (excluding 0) with respect to multiplication are groups. The element 1 is the identity for these groups. All these are infinite groups.

3. The set $\mathbb{Z}_n$ of all remainders modulo n forms a cyclic group under addition modulo n. It is a finite group of order n.

Definition 2.2.7. (**Center of a Group**)
The center $Z(G)$ of a group G is defined as the set

$$Z(G) = \{x \in G \mid gx = xg \ \forall g \in G\}.$$

That is, $Z(G)$ is the set of all those elements that commute with every element of G.

Definition 2.2.8. (**Conjugate Elements**)
Two elements $g, h \in G$, are said to be conjugate if $g^t = t^{-1}gt = h$, for some $t \in G$.

Definition 2.2.9. (**Normal Subgroup**)
A subset N which commutes with every element of group G is called invariant or self-conjugate. Particularly, if N is also a subgroup of G, then N is said to be a normal or invariant or self-conjugate subgroup of G.
Equivalently, we can also say that a subgroup N is called normal if $g^{-1}Ng = N$, for all $g \in G$. We write $N \rhd G$ to express that N is a normal subgroup of G.

For example the center $Z(G)$ of a group G is a normal subgroup of G.

Definition 2.2.10. (**Homomorphism**)
A mapping $\psi : G \longrightarrow H$ from a group G to another group H is said to be a

(group) homomorphism, if the group operation is preserved in the sense that

$$\psi(g_1 *_G g_2) = \psi(g_1) *_H \psi(g_2), \text{ for all } g_1, g_2 \in G,$$

where, $*_G$ and $*_H$ are the binary operations defined in G and H, respectively.

The kernel of a homomorphism ψ is

$$\ker \psi = \{g \in G \mid \psi(g) = e_H\},$$

where e_H is the identity of group H.

The image of ψ is the set

$$\text{img } \psi = \{h \in H \mid \exists\ g \in G \text{ such that } \psi(g) = h\}.$$

Note that, usually we do not use the binary operation symbol, $*$, but concatenation of two elements is used to show the specific binary operation of the corresponding group.

Example 2.2.11.

1. Consider a group G and an element $a \in G$. The exponential function $\psi : \mathbb{Z} \longrightarrow G$ defined by
$$\psi(n) = a^n,$$
for all $n \in \mathbb{Z}$, is a group homomorphism from $\mathbb{Z}$ to G.

2. The map $\varphi : \mathbb{Z} \to \mathbb{Z}_n$, $\varphi(k) = k \mod n$, is a homomorphism from $(\mathbb{Z}, +)$ to $(\mathbb{Z}_n, +)$.

3. The set $(\mathbb{R} \setminus \{0\}, \cdot)$ is a group. The map $det : GL(n, \mathbb{R}) \to \mathbb{R} \setminus \{0\}$, defined as
$$det(M) = |M|,$$
is a homomorphism.

Definition 2.2.12. (**Isomorphism**)

If a group homomorphism $\psi : G \longrightarrow H$ from a group G to another group H, possesses an inverse homomorphism, then ψ is said to be an isomorphism and the corresponding groups are said to be isomorphic to each other. For the isomorphic groups G and H, we write $G \cong H$.

Example 2.2.13.

1. The map $n \longmapsto (-1)^n$ is an isomorphism from $(\mathbb{Z}_2, +)$ to $(\{1, -1\}, \cdot)$.
2. For $x \in \mathbb{R} \setminus \{0\}$, the map $f_x : \mathbb{R} \longrightarrow \mathbb{R}$, defined as $f_x(y) = xy$, is an isomorphism from $(\mathbb{R}, +)$ to $(\mathbb{R}, +)$.

Definition 2.2.14. (**Endomorphism and Automorphism**)

An endomorphism is just a homomorphism $\psi : G \longrightarrow G$, where the domain and codomain are the same group G, and an automorphism is just an isomorphism $\psi : G \longrightarrow G$, from group G to the same group G.

The set of all automorphisms forms a group which is denoted by $Aut(G)$, with the binary operation as the composition of mappings.

Example 2.2.15. (**Inner Automorphism**)

The conjugate of any element $g \in G$ by an element $h \in G$ is $h^{-1}gh$ and we denote it by g^h. The automorphism defined as the conjugation by an element of the group is known as inner automorphism. The collection of all inner automorphisms of a group, denoted by $Inn(G)$, is a subgroup of the group $Aut(G)$.

Definition 2.2.16. (**Finitely Presented Groups**)

A group G on finitely many generators defined by finitely many relations between these generators, is said to be finitely presented or finitely presentable group. A set S of generators and a set R of relations among those generators can be specified. Each element of such a group is a product of powers of some of these generators. We can express G as

$$G = \langle S | R \rangle .$$

Example 2.2.17.

The group of symmetries of a regular polygon which includes rotations and reflections known as dihedral group, is the simplest example of finitely presented groups. The dihedral group D_n of order $2n$ with a rotation r and a reflection f has presentation

$$D_n = \left\langle r, f \mid r^n = f^2 = (rf)^2 = 1 \right\rangle .$$

Definition 2.2.18. (Word)
A word in a subset X of a group G, is any expression of the form

$$x_1^{\varepsilon_1} x_2^{\varepsilon_2} \ldots x_n^{\varepsilon_n}$$

where $x_1, x_2, ..., x_n$ are elements of X and each ε_i is either 1 or -1. The number n is called the length of the word.

Every word in X represents an element of G. For the elements x, y and $z \in G$, the products of the form xy, $z^{-1}xzz$ and $y^{-1}zxx^{-1}yz^{-1}$ are examples of words in the set $\{x, y, z\}$.

Definition 2.2.19. (Ring)
A set R in which two binary operations addition "+"and multiplication "·"are defined, is said to be a ring such that $(R, +)$ is an abelian group and satisfies the following conditions with respect to the multiplication operation "·":

$(M1)$ With respect to the operation of multiplication, R is closed that is, for all $r_1, r_2 \in R$

$$r_1 r_2 \in R,$$

where by $r_1 r_2$, we mean $r_1 \cdot r_2$.

$(M2)$ With respect to the operation of multiplication R is associative that is, for all $r_1\ , r_2,\ r_3 \in R$

$$r_1 (r_2 r_3) = (r_1 r_2) r_3,$$

$(M3)$ Distributivity holds in R that is, for all $r_1\ , r_2,\ r_3 \in R$

$$\begin{aligned} r_1 (r_2 +\ r_3) &= r_1 r_2 + r_1\ r_3, \\ (r_1 + r_2) r_3 &= r_1\ r_3 + r_2\ r_3. \end{aligned}$$

If a ring R also satisfies the following condition, then it is called a commutative ring.

$(M4)$ With respect to multiplication, a ring R is commutative, if for all r_1 , $r_2 \in R$

$$r_1 r_2 = r_2 r_1.$$

Definition 2.2.20. (Integral Domain)

A set R is called an integral domain, if it is a commutative ring and satisfies the following additional properties:

$(M5)$ Multiplicative Identity: For every $r \in R$, There exists an element $1 \in R$, such that

$$1r = r1 = r.$$

$(M6)$ No Zero Divisors: For $r_1, r_2 \in R$, if $r_1 r_2 = 0$, then either $r_1 = 0$, or $r_2 = 0$.

Definition 2.2.21. (Field)

An integral domain $\mathbb{F}$ is called field, if it obeys the following additional axiom:

$(M7)$ Multiplicative Inverse: For each nonzero $a \in \mathbb{F}$, there exists an element $a' \in \mathbb{F}$, called the inverse of a, such that

$$aa' = a'a = 1.$$

The relation between different algebraic structures is show in FIGURE 2.1

Example 2.2.22.

1. The set of complex numbers $\mathbb{C}$, the set of real numbers $\mathbb{R}$, the set of rational numbers $\mathbb{Q}$ and the set of integers $\mathbb{Z}$, are rings.

2. The set $M(n, R)$ of $n \times n$ matrices with entries from a ring R forms a non-commutative ring, under the usual matrix addition and multiplication.

3. Consider the set $GL(n, R) = \{A \in R^{n \times n} \mid det(A) \neq 0\}$ of all $n \times \ n$ matrices over R, where R is any ring having an identity element. We call it the general

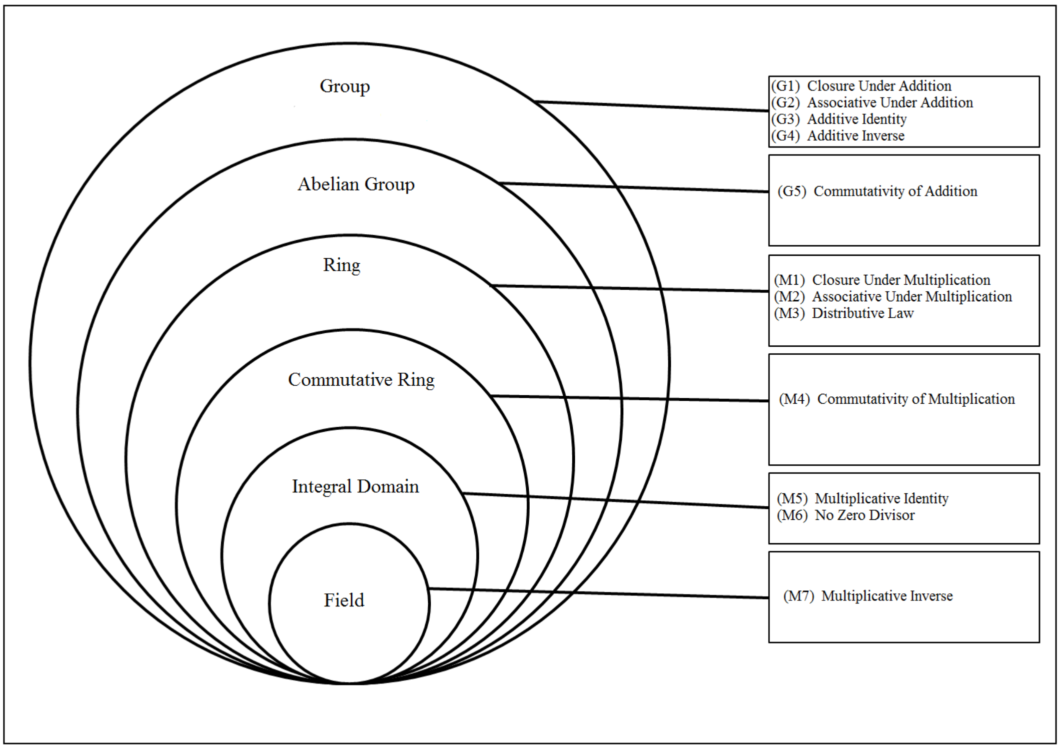

Group
Abelian Group
Ring
Commutative Ring
Integral Domain
Field
(G1) Closure Under Addition
(G2) Associative Under Addition
(G3) Additive Identity
(G4) Additive Inverse
(G5) Commutativity of Addition
(M1) Closure Under Multiplication
(M2) Associative Under Multiplication
(M3) Distributive Law
(M4) Commutativity of Multiplication
(M5) Multiplicative Identity
(M6) No Zero Divisor
(M7) Multiplicative Inverse

FIGURE 2.1: Relation Between Different Algebraic Structures

linear group of all $n\times\ n$ matrices over R. This set with matrix multiplication forms a noncommutative group.

4. $\mathbb{Z}_n$ is a commutative ring.

5. The set of integers $\mathbb{Z}$ with ordinary addition and multiplication forms an integral domain.

6. The set of complex numbers $\mathbb{C}$, the set of real numbers $\mathbb{R}$ and the set of rational numbers $\mathbb{Q}$ are all fields.

7. The set $\mathbb{Z}_p$, for a prime p is a finite field.

2.2.1 Finite Fields

In the context of cryptography, infinite fields are not of particular interest. However, there are many cryptographic algorithms which are based on finite fields. The order of a finite field can be shown to be a positive integer power of a prime number p. The finite field having order p^n, where n is a positive integer, is generally written as $GF(p^n)$. Here $GF(p^n)$ stands for Galois field, due to a famous mathematician Galois. For cryptographic techniques, we are interested in two special cases. These two cases are $n = 1$ and $n > 1$. The finite field $GF(p)$ (with $n = 1$) have different structure than that of $GF(p^n)$. First, let us look at the structure of finite fields having form $GF(p)$.

The finite field having form $GF(p)$ is defined as the set $\mathbb{Z}_p$ of integers less than p. That is

$$\mathbb{Z}_p = GF(p) = \{0, 1, \cdots, (p-1)\}.$$

The order of set $\mathbb{Z}_p$ is p. All the arithmetic operations are under modulo p. We have already mentioned in a previous section that any integer $a \in \mathbb{Z}_n$ possesses a multiplicative inverse if a and n are relatively prime. Observe that for a prime p, each nonzero integers in set $\mathbb{Z}_p$ is relatively prime to p. Therefore, each nonzero integers in set $\mathbb{Z}_p$ has a multiplicative inverse in $\mathbb{Z}_p$. So, the set $\mathbb{Z}_p$ is a finite field of order p. Hence, in this particular case, the equation (2.2) can be rewritten without the condition, as

$$\text{if } (x_1 \times x_2) \equiv (x_2 \times x_3) \mod p, \text{ then } x_1 \equiv x_3 \mod p.$$

2.3 Polynomial Arithmetic

One of our concerning algebraic structure is the set of polynomials in one variable. First, we give the definition of a polynomial.

Definition 2.3.1. (**Polynomial**)
For an integer $n \geq 0$, an expression having the form

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i, \ a_n \neq 0$$

is called a polynomial of degree n. Here a_i*'s* $\in A$, A is any designated set called the coefficient set. This set can be a ring or a field. Such polynomials are said to be defined over the coefficient set A. This form of polynomial with variable x, is referred to as the polynomial in one indeterminate.

To continue further discussion, we now revisit the following three kinds of polynomial arithmetic:

A) Ordinary polynomial arithmetic, employing the basic algebraic rules.

B) Polynomial arithmetic with the coefficients of polynomials are from $GF(p)$ and the arithmetic operations are performed on the coefficients under modulo p.

C) Polynomial arithmetic in which the coefficients of polynomials are from $GF(p)$ and the arithmetic operations are performed under modulo a specific polynomial $M(x)$.

First two classes are examined in this section and the third class is discussed in next section.

2.3.1 Ordinary Polynomial Arithmetic

All the operations are defined in a usual way. The addition or subtraction of any two polynomials can be carried out by adding or subtracting corresponding coefficients of the polynomials. Therefore, for any two polynomials

$$P(x) = \sum_{i=0}^{n} a_i x^i, \qquad Q(x) = \sum_{i=0}^{m} b_i x^i, \qquad n \geq m,$$

addition and multiplication are defined as

$$\begin{aligned} P(x) + Q(x) &= \sum_{i=0}^{m} (a_i + b_i)\, x^i + \sum_{i=m+1}^{n} a_i x^i, \\ P(x) \times Q(x) &= \sum_{i=0}^{m+n} c_i x^i, \qquad \text{where} \\ c_i &= a_0 b_i + a_1 b_{i-1} + \cdots a_{i-1} b_1 + a_0 b_i. \end{aligned}$$

In the formula of c_i, a_i*'s* are treated as zero for $i > n$ and b_i's are zero for $i > m$. The resulting polynomial obtained by the product of two polynomials $P(x)$ and $Q(x)$, has the degree equal to $n + m$.

Definition 2.3.2. (**Polynomial Ring**)
A set of all polynomials in one indeterminate with coefficients from a ring R, is denoted by $R[x]$. It is not hard to show that the set $R[x]$, with defined addition and multiplication, forms a ring, called a polynomial ring.

2.3.2 Polynomial Arithmetic with Coefficients in $\mathbb{Z}_p$

Now consider the polynomial ring $\mathbb{F}[x]$, where $\mathbb{F}$ is any field. In such a case, division can be performed on polynomials. Here, the exact division may not be possible. We explain this difference as follows:

In a field, we can say that b completely divides a, for two elements a and b of a field. That is the quotient $(a/b) \in \mathbb{F}$. But in a ring R, division, generally, produces a quotient and a remainder. That is in a ring R, an element may or may not completely divides another element. For example, consider the set of rational numbers and let 5 and 3 be the elements of this set. The division of 5 by 3 produces a rational number, as set of rational numbers is a field. Similarly, consider 5 and 3 being the elements of set $\mathbb{Z}_7$. We note that $5/3 = (5 \times 3^{-1}) \mod 7 = (5 \times 5) \mod 7 = 4 \mod 7$, as $\mathbb{Z}_7$ is a field. However, if we consider 5 and 3 as elements of set of integers which is a ring, the division of 5 by 3 produces a quotient and a remainder which are 1 and 2, respectively. Thus, 3 does not completely divide 5 over the set of integers.

The division has different aspects when performed over polynomials with coefficient set A. When set A is a ring the division may not be defined, always. For example, $(5x^2) \diagup (3x)$ has no solution, if coefficient set is the set of integers. Over the polynomial with coefficient set as a field, the division is always defined. For example, the same division over the polynomials with coefficients set $\mathbb{Z}_7$, gives $(5x^2) \diagup (3x) = 4x$, and $4x \in \mathbb{Z}_7[x]$. However the complete division is not always possible. Generally, a quotient and a remainder is produced as a result of such division. Consider the division $(5x^2 + 6) \diagup (3x)$ as an example over coefficients set $\mathbb{Z}_7$. This division will give a quotient $4x$ and a remainder equal to 6 which both are the valid polynomial over $\mathbb{Z}_7$.

Now, let us restate the division algorithm defined in Theorem 2.1.2 for ring of polynomials $\mathbb{F}[x]$ as follows:

Theorem 2.3.3. Given two polynomials $P(x)$ and $Q(x)$ of degrees n and m $(n \geq m)$, respectively. The division of $P(x)$ by $Q(x)$, will give a qoutient $Q^{'}(x)$ and a remainder $R(x)$ that satisfy the following relation:

$$P(x) = Q^{'}(x)Q(x) + R(x),$$

where $\deg(Q^{'}(x)) = n - m$ and $\deg(R(x)) \leq m - 1$.

Analogous to integer arithmetic, if remainder $R(x) = 0$, then it can be said that $Q(x) \mid P(x)$ or $Q(x)$ is a factor of $P(x)$. Also, the remainder $R(x)$ can be written as

$$R(x) = P(x) \mod Q(x),$$

Definition 2.3.4. (**Irreducible Polynomial**)

A polynomial $P(x)$ in $\mathbb{F}[x]$ is said to be an irreducible or a prime polynomial if it cannot be written as a product of two polynomials $Q(x)$ and $H(x) \in \mathbb{F}[x]$ and both $Q(x)$ and $H(x)$ having degree smaller than that of $P(x)$.

For example, the polynomial $P(x) = x^3 + 1$ over $GF(2)$ can be expressed as

$$x^3 + 1 = (x + 1)\left(x^2 + x + 1\right).$$

So, it is reducible. An example of irreducible polynomial over $GF(2)$ is $x^3 + x + 1$.

Definition 2.3.5. (**Greatest Common Divisor of Two Polynomials**)

The polynomial $D(x)$ is called the greatest common divisor of two polynomials $P(x)$ and $Q(x)$ if $D(x)$ divides both $P(x)$ and $Q(x)$ and any other divisor of $P(x)$ and $Q(x)$ is also a divisor of $D(x)$.

The Euclidean Algorithm 2.1.6 can be employed to determine the gcd of two polynomials. The equation (2.3) can be rewritten for polynomials as

$$\gcd[P(x), Q(x)] = \gcd[Q(x), P(x) \mod Q(x)].$$

2.4 Finite Fields of the Form $GF(2^n)$

It is mentioned earlier that a finite field must be of order of the form p^n, for any prime p and a positive integer n. We also talked about the special kind of finite fields having order p. Note that, for polynomials with coefficients from p^n, with $n > 1$ (operations modulo p^n), is not a field. Now, we discuss which structure becomes a field with p^n elements. Specially, we concentrate on $GF(2^n)$ which is important for cryptographic perspective. Because, for implementation convenience and efficiency, we have to deal with integers that fit exactly into a given number of bits. That is why, the integers in the range 0 to $2^n - 1$, which can be considered and fitted to an n-bit word are preferred to work with.

For example, assume that we wish to use and perform division in 8 bits data for an encryption algorithm. The integers in the range 0 to 255 can be represented with 8 bits. If we perform arithmetics in $\mathbb{Z}_{256}$, this set is not a field because 256 is not a prime number. The prime number less than and closest to 256 is 251 and the set $\mathbb{Z}_{251}$is a field under modulo 251. But, now we can not use the 8 bit patterns to represent the integers from 251 to 255. This results in excessive and inefficient use of storage. Keeping in mind this discussion, if all the integers ranging from 0 to $2^n - 1$ are desired to be represented in n bits, then the set of integers with arithmetic modulo 2^n will not generate a field. Moreover, there are also some other restrictions on the use of set $\mathbb{Z}_{2^n}$ for the encryption algorithm, as illustrated in the following example:

Consider a particular encryption algorithm in which 3 bit blocks are used and only the operations of addition and multiplication are involved. The operations of addition and multiplication in the set $\mathbb{Z}_8$ modulo 8 are as given in TABLE 2.3

and TABLE 2.4, respectively. Observe that in the TABLE 2.4, the occurrence of non-zero integers is not in an equal number of times. For instance, the integer 3 is appearing four times and 4 is appearing twelve times. But for the finite field $GF(2^3)$ of order $2^3 = 8$, the nonzero integers are occurring in a uniform manner for multiplication, as shown in TABLE 2.6. Let us just ignore the question of how TABLE 2.5 and TABLE 2.6 are constructed, observe the following:

1. The two tables verify the commutative law with respect to their corresponding operations.
2. All the nonzero elements of $GF(2^3)$ are invertible with respect to multiplication, unlike the case with TABLE 2.4.
3. The set of elements of $GF(2^3)$ with multiplication defined as in TABLE 2.6, becomes a finite field.

+	0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7	0
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

TABLE 2.3: Addition in $\mathbb{Z}_8$.

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

Table 2.4: Multiplication in $\mathbb{Z}_8$.

		000	001	010	011	100	101	110	111
	+	0	1	2	3	4	5	6	7
000	0	0	1	2	3	4	5	6	7
001	1	1	0	3	2	5	4	7	6
010	2	2	3	0	1	6	7	4	5
011	3	3	2	1	0	7	6	5	4
100	4	4	5	6	7	1	0	2	3
101	5	5	4	7	6	1	0	3	2
110	6	6	7	4	5	2	3	0	1
111	7	7	6	5	4	3	2	1	0

Table 2.5: Addition in $GF(2^3)$.

		000	001	010	011	100	101	110	111
	×	0	1	2	3	4	5	6	7
000	0	0	0	0	0	0	0	0	0
001	1	0	1	2	3	4	5	6	7
010	2	0	2	4	6	3	1	7	5
011	3	0	3	6	5	7	4	1	2
100	4	0	4	3	7	6	2	5	1
101	5	0	5	1	4	2	7	3	6
110	6	0	6	7	1	5	3	2	4
111	7	0	7	5	2	1	6	4	3

Table 2.6: Multiplication in $GF(2^3)$.

Generally, an attractive cryptographic algorithm is one that maps the integers uniformly onto themselves unlike an algorithm that maps themselves unevenly. Therefore, in cryptographic context, the use of the finite fields $GF(2^n)$ is preferred.

To summarize, we seek a set of order 2^n in which addition and multiplication are defined and that set becomes a field. So that the integer ranging from 0 to $2^n - 1$ can be associated uniquely to the element of that set.
Now, we show how polynomial can be used to provide a mean for constructing the required field.

2.4.1 Modular Polynomial Arithmetic

Let us consider the collection A_n of all the polynomials over the field $\mathbb{Z}_p$, such that every polynomial of this set has degree at most $n-1$. So, every polynomial is of the following form:

$$P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n-1} a_i x^i,$$

where $a_i \in \mathbb{Z}_p$. The set A_n has a total of p^n different polynomials.

Considering each such set A_n with the arithmetic operations that obey the following rules, makes it a finite field:

1. The ordinary rules of algebra are obeyed in polynomial arithmetic but with the following additional conditions:

2. The coefficients arithmetic can be done modulo p. The rules followed by the coefficients are those which are followed in finite field $\mathbb{Z}_p$.

3. In case of multiplication of polynomials, if we get a polynomial with degree greater than $(n-1)$, then that resulting polynomial should be reduced modulo some irreducible polynomial $M(x)$ of degree n. This means that the resulting polynomial is divided by $M(x)$ and replaced with the remainder.

In modular polynomial arithmetic, there is also the idea of a set of residues, just as in ordinary modular arithmetic. There are p^n elements in the set of residues modulo $M(x)$, where $M(x)$ is an nth-degree irreducible polynomial. Each element of this set can be expressed as one of the p^n polynomials with degree less than n. Further, a fact can be established that this set of polynomials forms a finite field. The modular polynomial arithmetic is explained in the following example:

The finite field $GF(2^3)$ can be constructed by choosing a polynomial having degree 3 which is irreducible. Note that there are only two irreducible polynomials having degree 3 which are $(x^3 + x^2 + 1)$ and $(x^3 + x + 1)$. Let us choose and use the irreducible polynomial $(x^3 + x + 1)$ to construct the results of addition and multiplication for $GF(2^3)$ as given in TABLE 2.7 and TABLE 2.8.

		000	001	010	011	100	101	110	111
	$+$	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
000	0	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
001	1	1	0	$x+1$	x	x^2+1	x^2	x^2+x+1	x^2+x
010	x	x	$x+1$	0	1	x^2+x	x^2+x+1	x^2	x^2+1
011	$x+1$	$x+1$	x	1	0	x^2+x+1	x^2+x	x^2+1	x^2
100	x^2	x^2	x^2+1	x^2+x	x^2+x+1	0	1	x	$x+1$
101	x^2+1	x^2+1	x^2	x^2+x+1	x^2+x	1	0	$x+1$	x
110	x^2+x	x^2+x	x^2+x+1	x^2	x^2+1	x	$x+1$	0	1
111	x^2+x+1	x^2+x+1	x^2+x	x^2+1	x^2	$x+1$	x	1	0

TABLE 2.7: Polynomial Addition Modulo $(x^3 + x + 1)$.

		000	001	010	011	100	101	110	111
	$\times$	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
000	0	0	0	0	0	0	0	0	0
001	1	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
010	x	0	x	x^2	x^2+x	$x+1$	1	x^2+x+1	x^2+1
011	$x+1$	0	$x+1$	x^2+x	x^2+1	x^2+x+1	x^2	1	x
100	x^2	0	x^2	x^2+1	x^2+x+1	x^2+x	x	x^2+1	1
101	x^2+1	0	x^2+1	1	x^2	x	x^2+x+1	$x+1$	x^2+x
110	x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2
111	x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$

TABLE 2.8: Polynomial Multiplication Modulo $(x^3 + x + 1)$.

2.4.2 Multiplicative Inverse of a Polynomial

The extended Euclidean algorithm can be used to determine the multiplicative inverse of a polynomial. For finding the multiplicative inverse of $P(x) \mod M(x)$, the polynomial $P(x)$ is required to have degree less than that of the polynomial $M(x)$ and the two polynomials should be relatively prime that is $\gcd[M(x), P(x)] = 1$. For an irreducible polynomial $M(x)$, the $\gcd[M(x), P(x)] = 1$, always holds. The algorithm is as follows:

Algorithm 2.4.1.

Input: Two Polynomials $P(x)$ and $M(x)$

Output: Inverse of $P(x) \mod M(x)$

```
1. $[P_1(x), P_2(x), P_3(x)] \longleftarrow [1, 0, M(x)]; [G_1(x), G_2(x), G_3(x)] \longleftarrow [0, 1, P(x)]$
2. if $G_3(x) = 0$ return $P_3(x) = \gcd[M(x), P(x)]$; There is no inverse
3. if $G_3(x) = 1$ return $G_3(x) = \gcd[M(x), P(x)]$; $G_2(x) = (P(x))^{-1} \mod M(x)$
4. $Q(x) = quotient\ of\ P_3(x) \diagup G_3(x)$
5. $[C_1(x), C_2(x), C_3(x)] \longleftarrow [P_1(x) - Q(x)G_1(x), P_2(x) - Q(x)G_2(x), P_3(x) - QG_3(x)]$
6. $[P_1(x), P_2(x), P_3(x)] \longleftarrow [G_1(x), G_2(x), G_3(x)]$
7. $[G_1(x), G_2(x), G_3(x)] \longleftarrow [C_1(x), C_2(x), C_3(x)]$
8. return to step 2.
```

Chapter 3

Cryptographic Background

Every field of study has its own language and terminologies including the specific terms that support an understanding of the object being investigated. We discuss different preliminary concepts and material related to the field of cryptography, in this chapter. The definitions and discussions contained therein will be used to understand rest of the study. The structure of this chapter is as follows:
First section is about a few basic primitives of cryptography, including its characterizations and applications. In the second section, a detailed discussion of public key cryptography with some of public key protocols and systems, is given.

3.1 Basic Cryptographic Primitives

Before proceeding further, we first discuss the main components of a cryptosystem. In a cryptosystem, any two communicating parties are usually referred as *Alice* and *Bob*. The original unprocessed message is called the *plaintext* and the processed message is known as the *ciphertext*. The procedure of transforming the plaintext in to the ciphertext is called *enciphering or encryption algorithm*, while reverting the procedure to get the plaintext back from the ciphertext is known as *deciphering or decryption algorithm*. A *key* is a secret essential piece of information that is used to get the original message from the cipher. Without the key, no plaintext would be produced.

3.1.1 Characterization of Cryptosystems

Cryptographic systems are characterized according to two approaches: the number of keys used and the way of dealing the plaintext. First approach may further be divided in two types: *secret key cryptography and public key cryptography.* The second approach of characterization of a cryptosystems is based on the manner of processing the plaintext for encryption. This characterization has also two types: *block cipher encryption and stream cipher encryption.*

Secret key cryptography makes use of a single key. This single key is shared by both sender and receiver. This type of system is also known as single-key, symmetric, or conventional cryptosystem. These cryptosystems are fast, highly secure and not very computationally intensive. The main disadvantage of this type of cryptography is that a key has to be sent through a highly secure channel before starting the communication itself. Usually, the single-key used in these cryptosystems is a lengthy one. The most famous examples of symmetric cryptographic algorithms are Data Encryption Standard (DES) [12] and Advanced Encryption Standard (AES) [13]. The model of secret key cryptosystem is shown in FIGURE 3.1, where the plaintext X is encrypted using an encryption algorithm E and the key K to compute ciphertext $C = E(K, X)$. The ciphertext is then decrypted by receiver using corresponding decryption algorithm and symmetric key K to get $X = D(K, C)$. While in *public key cryptography,* two keys are involved: a key

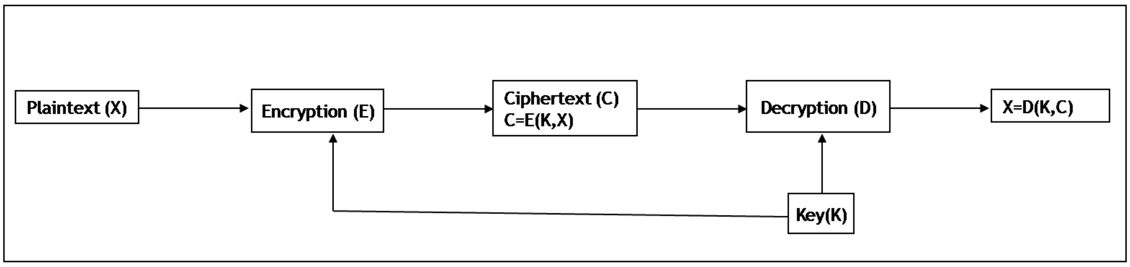


FIGURE 3.1: Symmetric Cryptosystem

K', known only to receiver called the private key and a key K, publicly available to anyone who wants to communicate with the receiver which is called the public key. The two keys are mathematically connected. This type of cryptosystem is also known as asymmetric cryptosystem. If there is no mean to exchange a key before starting communication, the public keys are used to establish secure communication. These cryptosystems are computationally intensive as compared to secret key cryptosystems. In such cryptosystems, the encryption algorithm E is also publicly known to every body. So, they can never exhibit unconditional security because an adversary looking at a ciphertext C, can try to encrypt each possible plaintext X by using the encryption algorithm until he gets a plaintext whose ciphertext is C. The model of public key cryptosystem is shown in FIGURE 3.2.

The two most famous and commonly used public key cryptosystems are RSA (Rivest-Shamir-Adleman) [46] and elliptic curve cryptosystem [18].

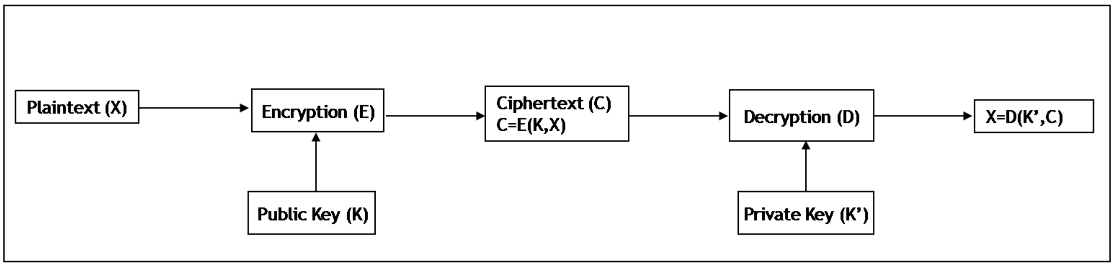


FIGURE 3.2: Asymmetric Cryptosystem

An other characterization of a cryptosystem is based on the way of dealing the plaintext. In a *block cipher encryption*, one block of a message is processed at a time and corresponding to each input block, an output block is produced. While in *stream cipher encryption*, the input message is processed continuously, as it goes along and one output element is produced at a time. Hill cipher, playfair cipher, DES, IDEA (International Data Encryption Algorithm) [33], RC5 [60], AES [13] and Blowfish [49] ciphers are the examples of block ciphers. Caesar cipher, shift register [20] and RC4 [47] are the examples of stream ciphers.

3.1.2 Cryptanalysis and Techniques of Attacks

The main goal of cryptography is to keep the original message secret from the adversaries. The adversaries are always trying to break the cryptosystem by getting some secret information about the plaintext or to change the message. Therefore, the cracking science of secrets is known as Cryptanalysis. Through different techniques of cryptanalysis, an adversary may target cryptographic protocols and authentication schemes. It is assumed that adversaries have a complete access to the communication system (encryption algorithm, decryption algorithm and the public key). This is the fundamental principle due to Kerckhoff [29] which states that adversaries know everything about cryptographic algorithms except the keys (private keys). The main aim of an attack is to observe and understand the working of different cryptographic algorithms. So that the weaknesses and flaws involved in these algorithms can be used to victimize or defeat them. Two different approaches are there to break a cryptographic algorithm: *Cryptanalytic attacks and brute-force attack*. Some cryptanalytic attacks are as follows:

In **ciphertext-only attack**, the adversary only knows some encrypted messages but has no knowledge of the plaintext and any thing about the key being used.
In a **known-plaintext attack**, the cryptanalyst has some of the plaintext and corresponding ciphertext. The main objective of this attack is to discover the secret key used to encrypt the messages.
In a **chosen-plaintext attack**, the attacker has some access to the computer or device which is used for the encryption. The attacker encrypts some plaintext of his choice with the encryption algorithm and tries to find out the key.
In **chosen-ciphertext attack**, the cryptanalyst has access to the decryption algorithm. He tries to obtain the key by choosing different ciphertexts to be decrypted. These attacks are generally mounted against the public key cryptosystems.
In **chosen-text attack**, the attacker may have some access to encryption and decryption algorithms. The attacker observes what will be encrypted and decrypted and then determines the key by using observed results.

The TABLE 3.1 shows a summary of the information available to the cryptanalyst or attacker for mounting different cryptanalytic attacks.

In the *brute-force attack*, the strategy of the attack is to try every key from the key space, on a ciphertext as far as an understandable plaintext is obtained. To find out the actual key, an attacker has to check half of the key space, averagely.

Attack Type	**Attacker Knowledge**
Ciphertext-only attack	1) Ciphertext
Known-plaintext attack	1) Ciphertext 2) Some plaintext ciphertext pairs created by using the private key
Chosen-plaintext attack	1) Ciphertext 2) Plaintext selected by attacker plus its respective ciphertext formed by using the private key.
Chosen-ciphertext attack	1) Ciphertext 2) Purported ciphertext selected by attacker, plus its respective plaintext formed by using the private key.
Chosen-text attack	1) Ciphertext 2) Plaintext selected by attacker plus its respective ciphertext formed by using the private key 3) Purported ciphertext selected by attacker, plus its respective plaintext formed by using the private key

Table 3.1: Cryptanalytic Attacks.

3.1.3 Cryptographic Applications

Today, open communication channels are used to send secure information. For instance, while using internet to purchase items, for financial transactions, in banking and alike situations. Unauthorized parties are there to get secret information from the data transmitted over these open channels. This is one of the main reasons behind the development of modern cryptography. As far as the applications of modern cryptography are concerned, secrecy of the message is not the only purpose of the cryptography. Perhaps, cryptography is also used to address the solution of the following problems.

1. *Data Integrity:* Data integrity simply means that the recipient of the message should be able to verify whether the message was altered during transmission, either accidently or deliberately. The unauthorized parties should be unable to modify even a part of a message.
2. *Authentication:* In the authentication scenario, the recipient of the message can verify the origin of the received message. In other words, at the time of initiating the communication sender and receiver should be able to identify each other. The unauthorized parties should be unable of pretending to be Alice and send a message to Bob.
3. *Non-repudiation:* One of the purpose of modern cryptography is the non-repudiation which means that the sender should be unable to deny the sending of a message, later on.

3.2 Public Key Cryptography

While using the symmetric key cryptography, the sender and receiver only know the encryption and decryption schemes, respectively. Moreover, once the encrypting scheme is exposed, the decryption scheme can be judged. Whereas in public key cryptography, the encryption and decryption methods as well as a key are publicly available to everyone but the private key used in decryption, is known only to the receiver. Generally, when the encrypting algorithm involved in a secret key (symmetric) cryptosystem is known, the order and the magnitude of time to implement the decryption algorithm, is approximately same. But the implementation of the decryption method involved in a public key cryptosystem, is much more difficult.

Definition 3.2.1. (Public key cryptosystem)

This type of cryptosystem consists of the following three algorithms:

1. The algorithm KG is used for the purpose of key generation . A pair (P, S) of two mathematically connected public and secret keys P and S, respectively, are produced through this algorithm.

2. The encryption algorithm E_P is used to produce a ciphertext C by taking a plaintext message M and a public key P, as an input. That is $C = E_P(M)$.

3. The decryption algorithm D_S which takes a ciphertext C and the secret key S and gives the plaintext M as an output. That is $M = D_S(C)$.

3.2.1 The General Model for Public Key Cryptosystem

The construction of a public key cryptosystem is based on the idea of having a one-way function or trapdoor function. It is easy to implement a one-way function but any computation of its inverse remains infeasible without a specific secret information. That is why encryption of a message is simple but decryption is too hard, without knowing the inverse even if an adversary can have enough computational resources and time (even those capable of using thousands of supercomputers for tens of years). It is a formidable task to find an appropriate one way function.

For example

1. Modular exponentiation: $f(x) = a^x \mod n$.

2. Prime number multiplication $f(p, q) = pq$.

3. Modular squaring $f(y) = y^2 \mod n,$ where n is a Blum integer. A natural number n is a Blum integer if $n = p \times q$ is a semi-prime and p and q are distinct prime numbers congruent to $3 \mod 4$. A natural number is known as semi-prime if it can be written as the product of two prime numbers which need not to be distinct.

Generally, the following model is used for public key systems:

Suppose (P_A, S_A) and (P_B, S_B) are the public and secret key pairs of Alice and Bob, respectively. The encryption algorithm (map) E_{P_A} of Alice as well as the encryption algorithm E_{P_B} of Bob are known publicly. But only Alice and Bob know their respective secret information used in decryption algorithms (inverse maps) D_{S_A} and D_{S_B}, respectively.

1. Suppose Alice wishes to communicate a message M to Bob, she sends ciphertext C by computing
$$C = E_{P_B}(M).$$
2. For decryption, Bob simply applies D_{S_B} for which the secret information is known to him, only. This gives him
$$M = D_{S_B}(C),$$
to get the message M.

Let us now consider a little twist in this communication model as follows:

1. If Alice wishes to communicate a message M to Bob, she sends
$$E_{P_B}(D_{S_A}(M)).$$
2. For decryption, Bob applies first D_{S_B}. This gives him
$$D_{S_B}E_{P_B}(D_{S_A}(M)).$$
3. Now, he looks up publicly available E_{P_A} and applies this
$$E_{P_A}(D_{S_A}(M)) = M$$
to get the sent message.

The question arises here, why didn't Alice just send $(E_{P_B}(M))$ to Bob? As Bob is the only person who can decrypt this. The answer lies in the idea of *authentication.*

That is, in the later method, Bob can confirm that the message actually sent from Alice.

3.3 Some Public Key Protocols

In the subsequent subsections, we recall some well known number theoretic and group based protocols and some cryptosystems.

3.3.1 Diffie–Hellman Protocol

Diffie and Hellman [15] formulated the innovative public key exchange idea which is based on the discrete log problem. The discrete log problem states that in modular arithmetic finding a power of an element is easy but given an element, determining whether it is a power of another element, is difficult. Specially, let the cyclic multiplicative group with prime order p and $y = x^k \mod p$, for some integer k, then the integer k is the *discrete log* of y to the base x. Given x and $y \in \mathbb{Z}_p$, the problem of finding such integer k, is called the discrete log problem (DLP).

The standard form of the protocol, is the following:

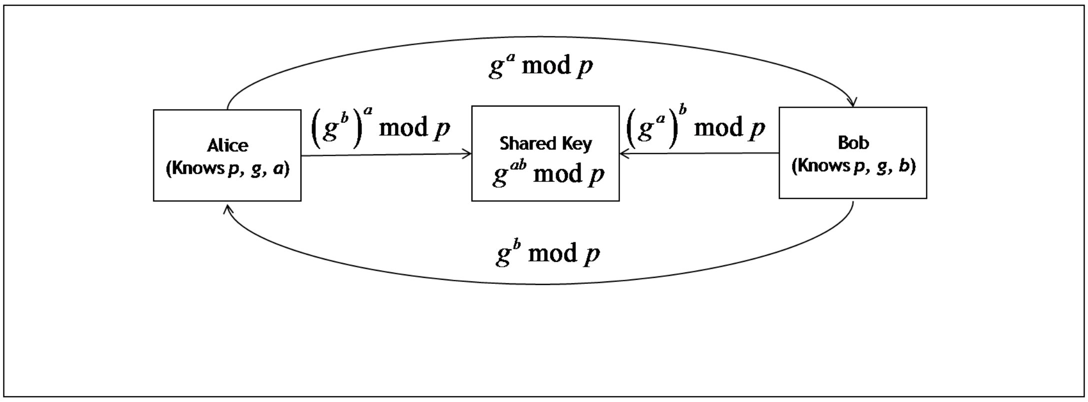
Suppose Bob and Alice want to share a secret key k with $2 < k < p-1$, where p is a prime. Let g be a generator of $\mathbb{Z}_p^*$ which is the multiplicative group of $\mathbb{Z}_p$. The generator g is announced publicly. Note that with prime p, this group is cyclic.

1. Alice selects an $a \in \mathbb{Z}_p$ with $2 < a < p - 1$. She calculates $g^a \mod p$ and made it public.
2. Bob chooses an element $b \in \mathbb{Z}_p$ and anounces $g^b \mod p$, publicly .
3. Alice can discover the secret key $K = g^{ab} \mod p$ as $(g^b)^a \mod p$.
4. Similarly Bob computes $K = g^{ab} \mod p = (g^a)^b \mod p$.

However, g^a and g^b and g are only known to an attacker but not the secret exponents a and b. This protocol is based on the solution of the DLP. This algorithm is shown diagrammatically, in FIGURE 3.3.

The problem of getting the secret key from the known information p, g, g^a, g^b, is called the discrete log problem (DLP). The discrete log problem becomes very hard if the prime p and the generator g are chosen to be very large. Because for large p, the order of g remains very large.

The *man in the middle attack* is one of the attacks on the Diffie–Hellman key exchange protocol. For this attack, attacker receives information from Alice, by pretending to be Bob and then pretending as Alice for getting information from Bob. So that, the attacker can find the shared secret key. The enhanced Diffie-Hellman based protocol [4] easily defeats the man in the middle attack by using authentication. This protocol is preferably used for communicating data but it is not recommended for storing data or archived over long periods of time.



$g^a \bmod p$
Alice (Knows p, g, a)
$(g^b)^a \bmod p$
Shared Key $g^{ab} \bmod p$
$(g^a)^b \bmod p$
Bob (Knows p, g, b)
$g^b \bmod p$

FIGURE 3.3: Diffie Hellman Protocol

3.3.2 RSA Cryptosystem

Currently, the RSA algorithm and its different variations are extensively used public key cryptosystems. This algorithm is developed by Ron Rivest, Adi Shamir and Leonard Adleman in 1977 [46]. There are two different keys, one public and one private, are used in RSA algorithm. These keys are mathematically connected to each other. The public key is announced and known to everyone, while the private key is kept secure. Any of the keys involved in RSA cryptography, can be used for encrypting a message; the opposite key from the one used for encryption, is used for decryption of the message. Due to this attribute, the RSA cryptosystem is one of the most widely used public key cryptosystems. It can also provide confidentiality, integrity, authenticity and non-reputability of communication.

This algorithm is based on the problem of factoring two large prime integers. It is easy to multiply these two integers, but it remains infeasible to determine the original prime numbers from the product. This is known as integer factorization problem (IFP). The working of RSA algorithm is as follows:

1. Two large primes p_A, q_A are chosen randomly, by Alice. After that, Alice also chooses a integer P_A, relatively prime to

   $$\phi(p_A q_A) = (p_A - 1)(q_A - 1),$$

   where ϕ is the Euler totient function. The primes should be chosen quite large. Initially, the primes involved in RSA were of approximately 100 decimal digits, however, with the increase in computing ability, the utilization of larger primes has to be needed.

2. Now, Alice takes

   $$N_A = p_A q_A$$

   and she computes the multiplicative inverse of P_A modulo $\phi(N_A)$ which is S_A. This means S_A satisfies the equation

   $$S_A P_A \equiv 1 \mod \phi(N_A).$$

3. Alice announces her public key

   $$K_A = (N_A, P_A).$$

4. The ciphertext C_A is obtained by using encryption algorithm E_{P_A} (which is public knowledge of everyone) as follows:

$$C_A = E_{P_A}(M) = M^{P_A} \mod N_A,$$

where $M \in \mathbb{Z}_{N_A}$, is a plaintext.

5. The decryption algorithm is

$$M = D_{S_A}(C_A) = C_A^{S_A} \mod N_A.$$

Bob also follows the above mentioned algorithm to choose his parameters p_B, q_B, P_B. He lets

$$N_B = p_B q_B.$$

His public key would be

$$K_B = (N_B, P_B).$$

For sending an authenticated message to Alice, Bob sends

$$E_{P_A}(D_{S_B}(M)),$$

as described for the process of authentication in Section 3.2.1.

3.3.3 El-Gamal Cryptosystem

In 1985 Taher ElGamal [17] presented a cryptosystem that utilizes the Diffie–Hellman key exchange method for encrypting the message.

Consider the communication between Bob and Alice through some open channel.

1. The parameters p and g are chosen by Alice, as discussed in the Diffie–Hellman key exchange protocol.

2. Alice then chooses an integer a with $1 < a < p-1$ and computes

$$A = g^a \mod p.$$

3. She announces her public key

$$(p, g, A),$$

as described in FIGURE 3.4.

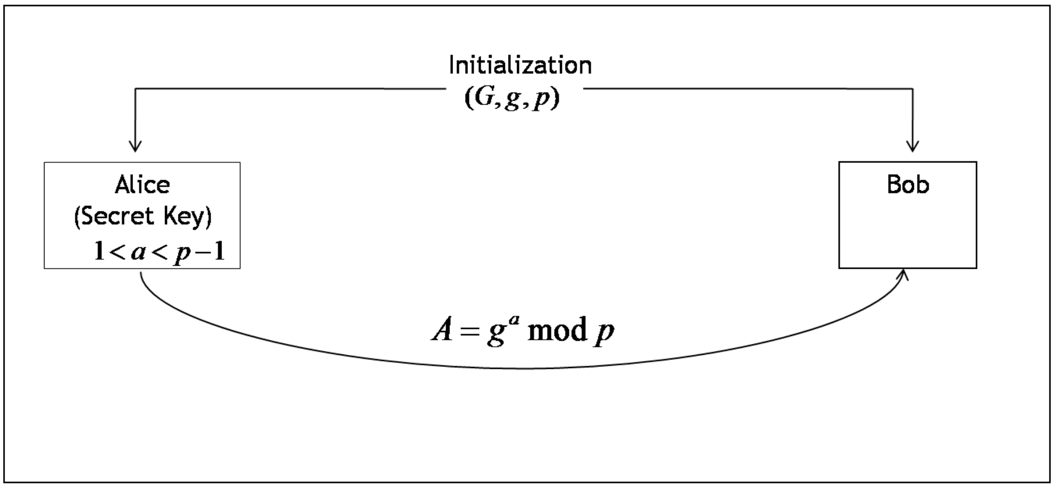


FIGURE 3.4: Key Generation of ElGamal Cryptosystem

4. Bob, first converts the message M to an integer $m \mod p$.

5. Now, Bob chooses an integer b with $1 < b < p-2$, randomly and computes

$$B = g^b \mod p.$$

6. After that, he sends

$$c = A^b m \mod p,$$

which means Bob formulates the encrypted message by multiplying the original message m with the shared key.

7. The ciphertext is in the form of the pair (B, c).

Now, Alice decrypts as follows:

1. The shared key can be computed as B^a.
2. Then c can be divided by this key g^{ab} to get m.
3. After obtaining the message m, only she can find the plaintext message M.

The encryption and decryption are described in FIGURE 3.5.

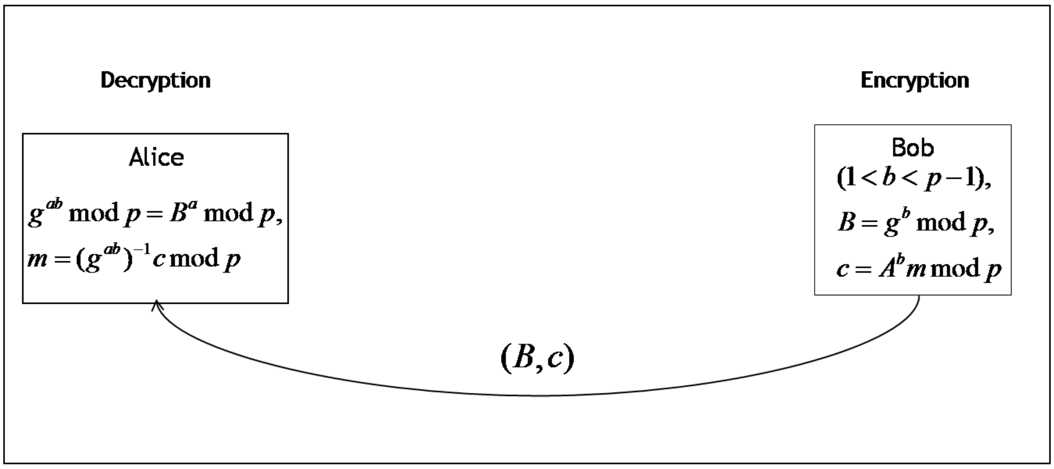


FIGURE 3.5: Encryption and Decryption of ElGamal Cryptosystem

The RSA cryptosystem, Diffie–Hellman protocol and El-Gamal cryptosystem are based on the number theoretic methods. So, they depend on the commutative structures. Now, the recent research deals with the development of cryptographic techniques using noncommutative platforms. Among these the primitive schemes are of Anshel, Anshel and Goldfeld and Ko and Lee. In these schemes the authors, proposed noncommutative structures and combinatorial group theory for developing public key exchange protocols.

3.3.4 Public Key Cryptography Based on Groups

In public key cryptography, using noncommutative groups is first time proposed by Wagner and Magyarik [36]. The difficulty of their cryptosystem depends on the *Word Problem* in finitely presented groups. Although, that scheme is theoretical one which has various unresolved issues. Wagner and Magyarik's scheme is important because it interplays between cryptography and combinatorial group theory.

The main reason to use combinatorial group theory for developing cryptographic algorithms, is that the elements of groups can be written as words in some alphabet. Max Dehn [14] proposed the following problems of combinatorial group theory:

1. *Word Problem:* Let G be a finitely presented group. For an arbitrary word w, can we say that there exists an algorithm for determining w to be one of the generators of G that defines the identity of G?

2. *Conjugacy Problem:* Let G be a finitely presented group. Is there an algorithm to decide whether a pair of words w_1, w_2 in the generators of G to be conjugate elements?

3. *Isomorphism Problem:* Given two arbitrary finite presentations, can we determine through some algorithm, whether the groups they present are isomorphic or not?

The problem to determine the conjugator $x \in G$, given elements g, h in a group G, where it is known that

$$g^x = x^{-1}gx = h,$$

is the called the conjugator search problem.

Remark 3.3.1. If a group G has a finite presentation, then the elements $g, x, h \in G$ can be expressed as words of the generators of the group G, otherwise $g, x, h \in G$ are just the elements of the group G.

In the present study, our different proposals depend on the conjugator search problem, somehow.

Now onwards, for an integer k, the notation g^k refers to exponentiation of an element $g \in G$. This means we multiply g by itself k-times. While g^x is to be used for the conjugacy of g by an element $x \in G$, that is

$$g^x = x^{-1}gx. \tag{3.1}$$

Two very famous group-based key exchange protocols proposed by Anshel, Anshel and Goldfeld [2] and Ko et al [31] are based on the cojugator search problem. Their key agreement schemes are considered as a generalization of the Diffie–Hellman protocol to noncommutative groups.

3.3.5 Ko–Lee Protocol

For a nonabelian group, Ko and Lee [31] proposed a public key agreement protocol that generalizes the protocol of Diffie and Hellman, given in Section 3.3.1. This protocol relies on the hard problem of conjugacy of elements in a nonabelian group. We take a noncommutative finitely presented platform group G. Following the notation of (3.1), the protocol is as follows:

1. Two secret commuting subgroups A and B of the group G are chosen by Alice and Bob, respectively. Let $g \in G$ be a public element.
2. A random secret element $a \in A$ is chosen by Alice. She then computes g^a and makes it public, where $g^a = a^{-1}ga$.
3. Similarly, a random choice of secret element $b \in B$, Bob computes g^b and makes it public, where $g^b = b^{-1}gb$.
4. The shared secret is $K = g^{ab}$ can now be computed by both Alice and Bob as follows:
$$(g^a)^b = g^{ab} = g^{ba} = (g^a)^b.$$

This follows by the commuting property of elements of the chosen subgroups.

Here the underlying hard problem is due to the difficulty of the CSP.

Generally, the CSP is undecidable but there are groups where it is a bit hard. As with the CSP, it is known that the structures of such groups can be considered as a suitable platform for the Ko–Lee protocol. Use of braid groups is suggested as such a platform group. Some weaknesses of this protocol are highlighted in [35]. These weaknesses further lead to develop new algebraic protocols.
As with the case of standard Diffie–Hellman protocol, an El-Gamal like cryptosystem can be constructed through the Ko–Lee protocol.

3.3.6 Anshel–Anshel–Goldfeld Protocol

In this section, a brief discussion of the Anshel–Anshel–Goldfeld [2] public key agreement protocol is given. Consider a goup G having a finite presentation. For sharing a secret key, the following steps are followed:

1. Alice and Bob select two subgroups A and B of G, respectively. A generating set for each subgroups
$$A = \{a_1, ..., a_n\}$$
and
$$B = \{b_1, ..., b_m\}$$
is made public.

2. A secret group word
$$a = a(a_1, ..., a_n)$$
is chosen by Alice from her subgroup.

3. Similarly, Bob also makes choice of a secret group word
$$b = b(b_1, ..., b_m)$$
from his selected subgroup.

4 Alice computes the conjugates

$$b_i^a, \qquad i = 1, ..., m,$$

from the known generators b_i of Bob's subgroup and her secret word a. She sends them to Bob. Bob also sends the conjugates

$$a_j^b, \qquad j = 1, ..., n,$$

to Alice.

5 Alice computes

$$a(a_1^b, a_2^b, ..., a_n^b) = a^b = b^{-1}ab.$$

Also, she multiplies $b^{-1}ab$ by a^{-1}, from the left to get $a^{-1}b^{-1}ab$.
Similarly, Bob computes

$$b(b_1^a, b_2^a, ..., b_m^a) = b^a = a^{-1}ba.$$

Now, he multiplies $a^{-1}ba$ by b^{-1}, from the left and takes the inverse of whole thing as

$$(b^{-1}a^{-1}ba)^{-1} = a^{-1}b^{-1}ab.$$

The shared secret key will be the commutator

$$[a, b] = a^{-1}b^{-1}ab.$$

An adversary has to find the corresponding conjugator to attack on this system. This means that he must know the element that conjugates each of the generators. There are some groups in which the CSP is solvable but, generally, the complexity of solving the CSP is considered to be "hard". The groups for which the CSP is hard, seem to be the ideal candidates for the Anshel–Anshel–Goldfeld protocol. This protocol is successfully cryptanalyzed in [42].

Various public key exchange protocols based on nonabelian groups are discussed by Myasnikov, Shpilrain and Ushakov, in their book [43]. These authors themselves have proposed various protocols based on different "hard" group theoretic problems.

Chapter 4

A Cryptosystem with Noncommutative Platform Groups

This chapter presents a new public key cryptosystem that uses noncommutative groups as platform group. The underlying hard problem of the proposed cryptosystem is a combination of discrete log problem and conjugacy search problem. Due to use of noncommutative platform groups, it is expected that the presented cryptosystem provides higher levels of security against known attacks. Some important issues regarding the choice of platform and parameters of this cryptosystem are addressed. Further, a brief analysis of security aspects is also presented.

The material of this chapter is based on the exposition of our paper [26] and is organized as follows:

In the first section, some motivating literature behind our main work, is discussed. Second section presents the proposed cryptosystem. We also discuss implementation aspects of our proposed scheme by giving an example. The security analysis is given in Section 4.3.

4.1 Motivating Literature

Here, we discuss some key exchange protocols which serve as motivation for the work presented in this chapter. We begin our discussion by describing Shamir's secret sharing scheme [50] which relies on the hardness of DLP. This is also referred as Shamir's 'no-key' key transport protocol. After that, the matrix based key exchange protocol by Baumslag, Camps, Fine, Rosenberger and Xu, [6] is discussed. The scheme is referred as the BCFRX scheme which is described in the subsequent subsection. At the end, Stickel's scheme [56] is given.

4.1.1 Shamir's Secret Sharing Scheme

Let $G = \langle g \rangle$ be a cyclic group, generated by g. Both g and its order d is announced publicly. The following procedure is executed to share a secret key:

1. Alice selects at random a key $k = g^r \in G$ and an integer $2 \leq a \leq d-1$. She sends
$$k^a \in G$$
to Bob.

2. Bob chooses, randomly an integer $2 \leq b \leq d-1$ and sends
$$(k^a)^b = k^{ab} \in G$$
to Alice.

3. Then Alice computes
$$\left(k^{ab}\right)^{a^{-1}} = k^b$$
and sends it to Bob.

4. Finally, Bob computes
$$\left(k^b\right)^{b^{-1}} = k.$$

Thus at the end of step 4, both Alice and Bob have a common shared secret k. As discussed earlier that the security of Diffie–Hellman Problem relies on the discrete logarithm problem (DLP). If the discrete logarithm problem can be efficiently solved then one can efficiently break the Diffie–Hellman Protocol. Likewise, Shamir's 'no-key' protocol depends on the discrete logarithm problem. Thus, for implementing these types of protocols, finding difficult instances of the DLP is a minimum requirement. Also, the difficulty of the discrete logarithm problem strongly depends on the representation of the group G. So, the appropriate choice of group G makes the DLP computationally infeasible.

4.1.2 The BCFRX Scheme

Before executing the BCFRX scheme [6], Alice and Bob have to share a common secret which is their long term secret key. To establish a session key for subsequent communication, is the main purpose of this protocol. For this goal, Bob sends a session key to Alice, in three passes, as follows:

Consider two commuting subgroups A and B (so that $AB = BA$ that is, for all $x \in A$ and $y \in B$, $xy = yx$) of a finitely presented group G. Here the group G is publicly announced but the subgroups A and B are the long term secret key of Alice and Bob, respectively. Although, the subgroups A and B may be infinite,but have finite presentation. So, they are kept secret. Now,

1. Bob selects a session key $k \in G$ and two elements b, $b^{'} \in B$. He forms

$$k_1 = bkb^{'}$$

   and sends it to Alice.

2. Alice chooses two elements a, $a^{'} \in A$ and sends

$$k_2 = ak_1a^{'} = abkb^{'}a^{'}$$

to Bob.

3. Bob then sends

$$k_3 = b^{-1}k_2b^{'-1} = aka^{'},\ \text{(Since } abkb^{'}a^{'} = baka^{'}b^{'}\text{)}$$

to Alice.

4. Alice finds

$$k = a^{-1}k_3a^{-1}.$$

This protocol is similar to that of Shamir's 'no-key' protocol, in the sense that the exponentiation operation is replaced by the multiplication on the left and right by an element of the group. Baumslag et al. [6] suggested various platform groups for their protocol, but we are concerned here, only with their matrix based proposal. They take $G = SL(4, \mathbb{Z})$, that is the group of matrices of order 4 over the integers and having determinant equal to 1. The commutative subgroups are constructed as follows:

Define two subgroups U and V of G as

$$U = \begin{bmatrix} SL(2,\mathbb{Z}) & 0 \\ 0 & I_2 \end{bmatrix} \qquad \text{and} \qquad V = \begin{bmatrix} I_2 & 0 \\ 0 & SL(2,\mathbb{Z}) \end{bmatrix},$$

where

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Let $M \in SL(4, \mathbb{Z})$ known to both Alice and Bob and this matrix can be considered as the long-term secret key. Now define two subgroups as

$$A = M^{-1}UM \quad \text{and} \quad B = M^{-1}VM.$$

In BCFRX protocol, the authors have not fully specified about the choice of long term secret key M. It is also not mentioned how to choose the elements from A and B by Alice and Bob, respectively.

4.1.3 Stickel's Key Exchange Protocol

In 2005, Stickel [56] proposed a key agreement scheme for noncommutative groups. This protocol can be easily implemented and it is expected to provide higher levels of security in noncommutative setting. This method, is not a generalization of the classical Diffie-Hellman protocol rather, it is a reminiscent of latter to noncommutative groups. This approach can be used for the purpose of key exchange, as well as, authentication protocols. Stickel suggested the use of general linear group of matrices for his proposal. Different matrices based structures have a great potential to be used in various ways for cryptographic algorithms. See for example [11, 25]. The main protocol as given by Stickel [56], is as follows:

Protocol 4.1.1.

Let $a, b \in G$, where G is a noncommutative finite group and $ab \neq ba$. Let $n_1 = |a|$ and $n_2 = |b|$. To share a common secret key, Alice and Bob will do the following steps:

1. Two secret natural numbers r and s with $0 < r < n_1$ and $0 < s < n_2$ are chosen, randomly by Bob. He then forms

   $$c = a^r b^s$$

   and transmit it to Alice.

2. Two secret natural numbers v and w with $0 < v < n_1$ and $0 < w < n_2$ are chosen, randomly by Alice. She forms

   $$d = a^v b^w$$

   and transmit it to Bob.

3. Alice computes the secret key K as

   $$K = a^v c\, b^w. \tag{4.1}$$

4. Bob similarly computes K as

$$K = a^r d\, b^s. \tag{4.2}$$

In this way, Protocol 4.1.1 enables Alice and Bob to have common secret key K. This follows immediately by putting value of c in expression (4.1) and value of d in expression (4.2).

A variant of key-exchange Protocol 4.1.1 proposed by Stickel [56], can also be implemented by slightly modifying Step 2 as follows:

Protocol 4.1.2.

Let $a, b \in G$, where G is a noncommutative finite group and $ab \neq ba$. Let $n_1 = |a|$ and $n_2 = |b|$. To share a common secret key, Alice and Bob will do the following steps:

1. Bob computes $c = a^r b^s$ as in Protocol 4.1.1 and submits it to Alice.

2. Two secret natural numbers v and w with $0 < v < n_1$ and $0 < w < n_2$ are chosen, randomly by Alice. She forms

   $$K = a^v b^w$$

   and

   $$d = a^v c\, b^w.$$

   Here, K is the secret key and d is submitted to Bob.

3. The secret key K can be computed by Bob as

   $$K = a^{-r} d\, b^{-s}.$$

These protocols are for noncommutative groups and may be considered as a reminiscent of Diffie-Hellman protocol. The main idea is the use of noncommutative behavior of product of elements of the form a^v and b^w of a group G.

4.2 Proposed Cryptosystem

We are going to propose a cryptosystem based on the Stickel approach. Many key exchange protocols can be used to design a cryptosystem. The most prominent example is the ElGamal [17] cryptosystem which employs the Diffie-Hellman key establishment [15] for encryption and decryption purpose. In the same fashion, we have used the variant of Stickel key exchange Protocol 4.1.2 to create a public key cryptosystem. The important feature of the proposed cryptosystem is that it can be implemented in a noncommutative structure. The main focus is on the development of a cryptographic scheme based on noncommutative groups. Particularly, the role of general linear group of matrices over finite field will be examined for the implementation of the scheme. The scheme relies on both, the DLP and the CSP in a noncommutative setting. Therefore, our scheme is believed to be more secure in post-quantum world.
Now, we present the general scheme for a public key cryptosystem as follows:

Cryptosystem 4.2.1.
Initial setup: Let G be a noncommutative group having the elements with large orders. Let $a, b \in G$ such that $ab \neq ba$. The order of the elements a and b be n_1 and n_2, respectively. Suppose that in G the DLP and CSP are intractable.

For developing a communication between Alice and Bob, the following steps will be executed:
Key Generation (KG)
By KG, Alice randomly chooses two natural numbers r and s with $0 < r < n_1$ and $0 < s < n_2$. The pair (r, s) is the secret key S_A of Alice. Then, she forms

$$K_A = a^r b^s \tag{4.3}$$

and announces her public key $P_A = (a, b, K_A)$.
Encryption Algorithm (E_{P_A})
Input: Plaintext message $M \in G$ and public key $P_A = (a, b, K_A)$.
Output: Ciphertext $C_A = (C', C)$.

To send a message (plaintext) $M \in G$ to Alice, Bob chooses randomly v and w with $0 < v < n_1, 0 < w < n_2$. Then, he executes Steps E-1 to E-5 as given below:

E-1) using Alice's public key $P_A = (a, b, K_A)$ Bob, computes

$$x = a^v b^w,$$

E-2) using x and K_A, he computes

$$C_1 = (K_A)^x,$$

E-3) then finds $x^{-1} \in G$ to computes

$$C_2 = (K_A)^{x^{-1}},$$

E-4) given the plaintext message M, Bob computes C as

$$C = C_1 M C_2,$$

E-5) and finally he computes

$$C^{'} = a^v K_A b^w. \tag{4.4}$$

The resulting transmitted ciphertext is the pair $(C^{'}, C)$.

Decryption Algorithm (D_{S_A})

Input: Ciphertext message C_A and secret key $S_A = (r, s)$.

Output: Plaintext M.

When Alice receives the ciphertext C_A, she executes the following steps:

D-1) By using her secret key $S_A = (r, s)$, Alice computes

$$\tilde{D} = (a^r)^{-1} C^{'} (b^s)^{-1} \tag{4.5}$$

D-2) Now with the help of $\tilde{D}$, she computes

$$\tilde{D}_1 = \left(K_A^{-1}\right)^{\tilde{D}}, \tag{4.6}$$

and

D-3)

$$\tilde{D}_2 = (K_A^{-1})^{\tilde{D}^{-1}}, \tag{4.7}$$

D-4) Finally, she gets the plaintext as

$$M = \tilde{D}_1 C \tilde{D}_2.$$

Theorem 4.2.2.

In view of specified notation of Cryptosystem 4.2.1, the correctness of its decryption is guaranteed.

Proof. **Correctness:** The correctness of the scheme is guaranteed as follows: First note the expression (4.5) in step **D-1** which is

$$\tilde{D} = (a^r)^{-1} C' (b^s)^{-1}.$$

In view of expressions (4.4) and (4.3), respectively, we have

$$\begin{aligned} \tilde{D} &= (a^r)^{-1} (a^v K_A b^w) (b^s)^{-1}, \\ &= (a^r)^{-1} (a^v a^r b^s b^w) (b^s)^{-1}, \\ &= (a^r)^{-1} \left(a^{v+r} b^{w+s}\right) (b^s)^{-1}, \\ &= a^v b^w, \\ &= x. \end{aligned}$$

Therefore, we can write expressions (4.6) and (4.7), respectively as

$$\tilde{D}_1 = \left(K_A^{-1}\right)^{\tilde{D}} = (x)^{-1} K_A^{-1} x,$$

$$\tilde{D}_2 = \left(K_A^{-1}\right)^{\tilde{D}^{-1}} = x K_A^{-1} (x)^{-1}.$$

Now consider that

$$\begin{aligned}
\tilde{D}_1 C \tilde{D}_2 &= \left[(x)^{-1})K_A^{-1}x\right] C \left[xK_A^{-1}\left(x\right)^{-1}\right], \\
&= \left[(x)^{-1} K_A^{-1}x\right] C_1 M C_2 \left[xK_A^{-1}\left(x\right)^{-1}\right], \\
&= \left[(x)^{-1} K_A^{-1}x\right] K_A^x M K_A^{(x)^{-1}} \left[xK_A^{-1}\left(x\right)^{-1}\right], \\
&= \left[(x)^{-1} K_A^{-1}x\right] \left[(x)^{-1} K_A x\right] M \left[xK_A\left(x\right)^{-1}\right] \times \\
&\quad \left[xK_A^{-1}\left(x\right)^{-1}\right], \\
&= M.
\end{aligned}$$

□

4.2.1 Suggested Platform and Parameters for the Proposed System

The proposed cryptosystem can be implemented to any noncommutative group having elements of large order. We suggest to use $GL\left(n, \mathbb{F}_p\right)$ for implementing proposed scheme, where $GL\left(n, \mathbb{F}_p\right)$ is the general linear group of matrices of order n over the field $\mathbb{F}_p$, for some prime p. The size of $GL(n, \mathbb{F}_p)$ can be calculated by the following formula [48]

$$N = |GL(n, \mathbb{F}_p)| = \prod_{i=0}^{n-1} \left(p^n - p^i\right).$$

That is we have a space of size N from which we can choose the public matrices A and B in $GL(n, \mathbb{F}_p)$.

Since the encryption involves the conjugation of matrices which can be considered as we are dealing with inner automorphisms of $GL(n, \mathbb{F}_p)$. In this case the inner automorphism is working linearly on the n^2-dimensional algebra of matrices of degree n over $\mathbb{F}_p$. So, the discrete logarithm problem reduced to the discrete logarithm problem in $GL(n^2, \mathbb{F}_p)$. Now the natural question arises, is there any algorithm to solve this problem?

A quite good algorithm for solving the DLP in $GL(n, \mathbb{F}_p)$, is proposed by Menezes and Wu [40]. The authors have proved that for any two matrices $A, B \in GL(n, \mathbb{F}_p)$, where $A^k = B$, $k \in \mathbb{N}$; the solution of DLP can be obtained if the characteristic polynomial of A have small degree irreducible polynomial as its factors. If the characteristic polynomial is irreducible then DLP in $\langle A \rangle$ reduces to the DLP in $\mathbb{F}_{p^n}$. While working in $GL(n^2, \mathbb{F}_p)$, the characteristic polynomials has degree n^2. In this case it can be easily seen that if the characteristic polynomial is irreducible then the extension of the minimum degree in which the characteristic polynomial will split is $\mathbb{F}_{p^{n^2}}$. So, DLP of this cryptosystem depends on solution of DLP on $\mathbb{F}_{p^{n^2}}$ which becomes harder as the field gets huge.

Today's recommendation for key management used for public key cryptography is 112-bit security [5]. So the best known attack to the cryptosystem should take at least 2^{112} steps. Hence, we have to make sure that to find exponents from public information A, B, K_A, one needs at least 2^{112} steps. We suppose that for a breaking algorithm, the computations in $\mathbb{F}_p$ and in $GL(n, \mathbb{F}_p)$ takes the same time. The size of the field should be 2^{112}. In the view of this discussion, we can make the choices of p to be a 112-bit prime.

Example 4.2.3. To comprehend the implementation details of the proposed cryptosystem, we take $GL\left(2, \mathbb{Z}_{101}\right)$, as the platform group. All the computations are performed in ApCoCoA [3].

Initial setup: Let

$$A = \begin{bmatrix} 21 & 47 \\ 27 & 95 \end{bmatrix} \in GL\left(2, \mathbb{Z}_{101}\right)$$

and

$$B = \begin{bmatrix} 89 & 17 \\ 19 & 67 \end{bmatrix} \in GL\left(2, \mathbb{Z}_{101}\right).$$

The order of these matrices are $n_1 = |A| = 1275$ and $n_2 = |B| = 1700$, respectively.

Key generation: Alice randomly chooses two natural numbers $r = 15 < 1275$ and $s = 21 < 1700$. The secret key of Alice is $S_A = (15, 21)$. She then forms

$$\begin{aligned} K_A &= A^r B^s = A^{15} B^{21} \mod 101, \\ &= \begin{bmatrix} 57 & 34 \\ 21 & 78 \end{bmatrix} \mod 101 \end{aligned}$$

and announces her public key $P_A = (A, B, K_A)$.

Encryption (E_{P_A}): To send a message (plaintext) $M = \begin{bmatrix} 25 & 20 \\ 60 & 69 \end{bmatrix} \in GL\left(2, \mathbb{Z}_{101}\right)$ to Alice, Bob chooses $\nu = 37 < 1275$ and $w = 51 < 1700$. He then forms

E-1)

$$\begin{aligned} X = A^v B^w = A^{37} B^{51} \mod 101, \\ = \begin{bmatrix} 88 & 60 \\ 71 & 17 \end{bmatrix} \mod 101, \end{aligned}$$

E-2)

$$\begin{aligned} C_1 = K_A^X = (X)^{-1} K_A X \mod 101, \\ = \begin{bmatrix} 14 & 80 \\ 73 & 20 \end{bmatrix} \mod 101, \end{aligned}$$

E-3)

$$\begin{aligned} C_2 = K_A^{(X)^{-1}} = X K_A (X)^{-1} \mod 101, \\ = \begin{bmatrix} 5 & 95 \\ 76 & 29 \end{bmatrix} \mod 101, \end{aligned} \tag{4.8}$$

E-4)

$$\begin{aligned} C = C_1 M C_2 \mod 101, \\ = \begin{bmatrix} 31 & 41 \\ 79 & 75 \end{bmatrix} \mod 101, \end{aligned} \tag{4.9}$$

E-5)

$$\begin{aligned} C^{'} = A^{v}K_{A}B^{w} = A^{37}K_{A}B^{51} &\mod 101, \\ = \begin{bmatrix} 14 & 97 \\ 61 & 3 \end{bmatrix} &\mod 101. \end{aligned} \tag{4.10}$$

The ciphertext transmitted to Alice is

$$C_A = (C^{'}, C).$$

Decryption (D_{S_A}): Upon receiving the ciphertext, Alice computes

D-1)

$$\begin{aligned} \tilde{D} &= (A^{r})^{-1}C^{'}(B^{s})^{-1} = A^{-15}C^{'}B^{-21} \mod 101, \\ &= \begin{bmatrix} 88 & 60 \\ 71 & 14 \end{bmatrix} \mod 101, \end{aligned}$$

D-2)

$$\begin{aligned} \tilde{D}_1 &= \left(K_A^{-1}\right)^{\tilde{D}} = \tilde{D}^{-1}K_A^{-1}\tilde{D} \mod 101, \\ &= \begin{bmatrix} 97 & 16 \\ 55 & 78 \end{bmatrix} \mod 101, \end{aligned}$$

D-3)

$$\begin{aligned} \tilde{D}_2 &= (K_A^{-1})^{\tilde{D}^{-1}} = \tilde{D}K_A^{-1}\tilde{D}^{-1}, \\ &= \begin{bmatrix} 75 & 19 \\ 96 & 100 \end{bmatrix} \mod 101. \end{aligned}$$

D-4)

$$M = \tilde{D}_1 C \tilde{D}_2 = \begin{bmatrix} 25 & 20 \\ 60 & 69 \end{bmatrix} \mod 101.$$

Example 4.2.4. In this example, we show the computation of the proposed cryptosystem by taking account the set $GL(3, \mathbb{Z}_{1009})$, the general linear group of matrices of order 3 over $\mathbb{Z}_{1009}$, as the platform group. All the computations are performed in ApCoCoA [3].

Initial setup: Let

$$A = \begin{bmatrix} 28 & 470 & 12 \\ 6 & 27 & 95 \\ 10 & 2 & 90 \end{bmatrix} \in GL(3, \mathbb{Z}_{1009})$$

and

$$B = \begin{bmatrix} 8 & 70 & 120 \\ 60 & 207 & 950 \\ 1 & 22 & 9 \end{bmatrix} \in GL(3, \mathbb{Z}_{1009}).$$

The order of these matrices are $n_1 = |A| = 1018080$ and $n_2 = |B| = 50904$, respectively.

Key generation: Alice randomly chooses two natural numbers $r = 1007 < 1018080$ and $s = 500 < 50904$. The secret key of Alice is $S_A = (1007, 500)$. She then forms

$$\begin{aligned} K_A &= A^r B^s = A^{1007} B^{500} \mod 1009, \\ &= \begin{bmatrix} 919 & 240 & 970 \\ 162 & 594 & 314 \\ 301 & 931 & 241 \end{bmatrix} \begin{bmatrix} 848 & 828 & 328 \\ 754 & 170 & 278 \\ 885 & 887 & 416 \end{bmatrix} \mod 1009 \\ &= \begin{bmatrix} 504 & 299 & 796 \\ 447 & 53 & 787 \\ 67 & 730 & 725 \end{bmatrix} \mod 1009 \end{aligned}$$

and announces her public key $P_A = (A, B, K_A)$.

Encryption (E_{P_A}): To send a message (plaintext)

$$M = \begin{bmatrix} 453 & 717 & 135 \\ 504 & 93 & 477 \\ 816 & 533 & 732 \end{bmatrix} \in GL\left(3, \mathbb{Z}_{1009}\right)$$

to Alice, Bob chooses $\nu = 100 < 1018080$ and $w = 51 < 50904$. He then forms

E-1)

$$\begin{aligned} X &= A^{v}B^{w} \\ &= A^{100}B^{51} \quad \text{mod } 1009, \\ &= \begin{bmatrix} 736 & 813 & 604 \\ 796 & 426 & 456 \\ 814 & 487 & 589 \end{bmatrix} \begin{bmatrix} 474 & 804 & 9 \\ 157 & 387 & 394 \\ 218 & 74 & 715 \end{bmatrix} \quad \text{mod } 1009 \\ &= \begin{bmatrix} 759 & 593 & 38 \\ 752 & 111 & 584 \\ 526 & 513 & 74 \end{bmatrix} \quad \text{mod } 1009, \end{aligned}$$

E-2)

$$\begin{aligned} C_1 &= K_A^X \\ &= (X)^{-1} K_A X \quad \text{mod } 1009, \\ &= \begin{bmatrix} 926 & 978 & 260 \\ 587 & 267 & 373 \\ 543 & 701 & 802 \end{bmatrix} \begin{bmatrix} 504 & 299 & 796 \\ 447 & 53 & 787 \\ 67 & 730 & 725 \end{bmatrix} \times \\ & \quad \begin{bmatrix} 759 & 593 & 38 \\ 752 & 111 & 584 \\ 526 & 513 & 74 \end{bmatrix} \quad \text{mod } 1009 \\ &= \begin{bmatrix} 424 & 289 & 336 \\ 448 & 166 & 748 \\ 390 & 235 & 692 \end{bmatrix} \quad \text{mod } 1009, \end{aligned}$$

E-3)

$$\begin{aligned}
C_2 &= K_A^{X^{-1}} \\
&= XK_AX^{-1} \mod 1009, \\
&= \begin{bmatrix} 759 & 593 & 38 \\ 752 & 111 & 584 \\ 526 & 513 & 74 \end{bmatrix} \begin{bmatrix} 504 & 299 & 796 \\ 447 & 53 & 787 \\ 67 & 730 & 725 \end{bmatrix} \times \\
&\quad \begin{bmatrix} 926 & 978 & 260 \\ 587 & 267 & 373 \\ 543 & 701 & 802 \end{bmatrix} \mod 1009 \\
&= \begin{bmatrix} 57 & 900 & 362 \\ 979 & 1006 & 18 \\ 212 & 631 & 219 \end{bmatrix} \mod 1009,
\end{aligned}$$

E-4)

$$\begin{aligned}
C &= C_1MC_2 \\
&= \begin{bmatrix} 424 & 289 & 336 \\ 448 & 166 & 748 \\ 390 & 235 & 692 \end{bmatrix} \begin{bmatrix} 453 & 717 & 135 \\ 504 & 93 & 477 \\ 816 & 533 & 732 \end{bmatrix} \times \\
&\quad \begin{bmatrix} 57 & 900 & 362 \\ 979 & 1006 & 18 \\ 212 & 631 & 219 \end{bmatrix} \mod 1009 \\
&= \begin{bmatrix} 230 & 158 & 395 \\ 722 & 519 & 29 \\ 56 & 779 & 37 \end{bmatrix} \mod 1009.
\end{aligned}$$

E-5)

$$\begin{aligned}
C' &= A^vK_AB^w \\
&= A^{100}K_AB^{51} \mod 101,
\end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} 736 & 813 & 604 \\ 796 & 426 & 456 \\ 814 & 487 & 589 \end{bmatrix} \begin{bmatrix} 504 & 299 & 796 \\ 447 & 53 & 787 \\ 67 & 730 & 725 \end{bmatrix} \times \\
&\quad \begin{bmatrix} 474 & 804 & 9 \\ 157 & 387 & 394 \\ 218 & 74 & 715 \end{bmatrix} \mod 1009 \\
&= \begin{bmatrix} 799 & 295 & 610 \\ 264 & 96 & 77 \\ 163 & 823 & 297 \end{bmatrix} \mod 1009.
\end{aligned}
$$

The ciphertext transmitted to Alice is

$$C_A = (C^{'}, C).$$

Decryption (D_{S_A}): Upon receiving the ciphertext, Alice computes

D-1)

$$
\begin{aligned}
\tilde{D} &= (A^r)^{-1}C^{'}(B^s)^{-1} = A^{-1007}C^{'}B^{-500} \mod 1009, \\
&= \begin{bmatrix} 149 & 119 & 899 \\ 316 & 196 & 227 \\ 540 & 233 & 797 \end{bmatrix} \begin{bmatrix} 799 & 295 & 610 \\ 264 & 96 & 77 \\ 163 & 823 & 297 \end{bmatrix} \times \\
&\quad \begin{bmatrix} 1004 & 588 & 620 \\ 319 & 167 & 1000 \\ 29 & 753 & 204 \end{bmatrix} \mod 1009 \\
&= \begin{bmatrix} 759 & 593 & 38 \\ 752 & 111 & 584 \\ 526 & 513 & 74 \end{bmatrix} \mod 1009.
\end{aligned}
$$

D-2)

$$\tilde{D_1} = \left(K_A^{-1}\right)^{\tilde{D}} = \tilde{D}^{-1}K_A^{-1}\tilde{D} \mod 1009,$$

$$= \begin{bmatrix} 926 & 978 & 260 \\ 587 & 267 & 373 \\ 543 & 701 & 802 \end{bmatrix} \begin{bmatrix} 288 & 809 & 568 \\ 597 & 993 & 691 \\ 288 & 807 & 386 \end{bmatrix} \times$$

$$\begin{bmatrix} 759 & 593 & 38 \\ 752 & 111 & 584 \\ 526 & 513 & 74 \end{bmatrix} \mod 1009$$

$$= \begin{bmatrix} 17 & 535 & 125 \\ 911 & 1 & 64 \\ 222 & 312 & 580 \end{bmatrix} \mod 1009,$$

D-3)

$$\tilde{D_2} = (K_A^{-1})^{\tilde{D}^{-1}} = \tilde{D} K_A^{-1} \tilde{D}^{-1} \mod 1009,$$

$$= \begin{bmatrix} 759 & 593 & 38 \\ 752 & 111 & 584 \\ 526 & 513 & 74 \end{bmatrix} \begin{bmatrix} 288 & 809 & 568 \\ 597 & 993 & 691 \\ 288 & 807 & 386 \end{bmatrix} \times$$

$$\begin{bmatrix} 926 & 978 & 260 \\ 587 & 267 & 373 \\ 543 & 701 & 802 \end{bmatrix} \mod 1009$$

$$= \begin{bmatrix} 821 & 423 & 534 \\ 96 & 893 & 72 \\ 39 & 966 & 902 \end{bmatrix} \mod 1009,$$

D-4)

$$M = \tilde{D_1} C \tilde{D_2} \mod 1009$$

$$= \begin{bmatrix} 17 & 535 & 125 \\ 911 & 1 & 64 \\ 222 & 312 & 580 \end{bmatrix} \begin{bmatrix} 230 & 158 & 395 \\ 722 & 519 & 29 \\ 56 & 779 & 37 \end{bmatrix} \times$$

$$\begin{bmatrix} 821 & 423 & 534 \\ 96 & 893 & 72 \\ 39 & 966 & 902 \end{bmatrix} \mod 1009$$

$$= \begin{bmatrix} 453 & 717 & 135 \\ 504 & 93 & 477 \\ 816 & 533 & 732 \end{bmatrix} \mod 1009,$$

4.3 Security Analysis

We will now discuss some good aspects that can be expected from the proposed cryptosystem. Since the complexity depends both on matrix CSP and DLP, our approach can cover following features.

First, note from equation (4.3) that r and s are exponents of two different group elements a and b, respectively. Because of the involvement of DLP of two elements a and b, it is hard to extract the exponents r and s from publicly known K_A.

Shpilrain [52] and Sramka [54] discussed the cryptanalysis of the Stickel main protocol. Sramka's attack reveals the private exponents r, s, ν, w, whereas, Shpilrain more efficiently worked to get the shared key K of main protocol without knowing any of private exponents. To prevent these attacks, we have used Stickel's variant key exchange protocol for sufficient security.

The attack presented in [42] does not seem to be applicable to our proposed cryptosystem, as our proposal is based on a mixture of CSP and DLP.

If we take ℓ, as well as $(2^\ell - 1)$ to be the prime numbers. The number ℓ is called the Mersenne exponent and the numbers having form $(2^\ell - 1)$ are called Mersenne primes [56]. For the matrices of order $(2^\ell - 1)$, the total number of different elements of the form of product $A^r B^s$ is equal to $(2^\ell - 1)^2$. For brute force attacks, an adversary has to check the space of this size. To enlarge the size of this space, the choice of order n of matrices to be a Mersenne prime $n > 31$, is recommended. Further, the matrices A and B which have irreducible characteristic polynomials should be selected, so that the eigenvalue and eigenvector attacks [56] remain infeasible.

The matrices C_1 and C_2 involved in encryption, are the matrix conjugates of the matrix K_A by the private matrix X. The relevant problem of the matrices C_1 and C_2, is the conjugacy search problem. For conducting matrix conjugation attacks, an adversary has to find the unknown matrices C_1, C_2, first. That is why these types of attacks can also be avoided.

4.3.1 Ciphertext Only Attack

Suppose an adversary knows only the ciphertext $(C^{'}, C)$, as given in expressions (4.10) and (4.9), respectively. In (4.9) the unknown matrices M, C_1 and C_2 are involved. To know C_1 and C_2 an adversary has to find the solution of

$$XC_1 = K_A X, \qquad\qquad C_2 X = X K_A, \tag{4.11}$$

where the matrix X is also unknown. For the underlying structure $GL(n, \mathbb{F}_p)$, system (4.11) will lead to a large system of equations for the choice of sufficiently large n. For the discussion on choice of n, see Section 4.2.1.

Also the DLP of matrices A and B is involved in the expression (4.10) of $C^{'}$. As discussed earlier in Section 4.2.1, the DLP of this cryptosystem depends on solution of DLP on $\mathbb{F}_{p^{n^2}}$ which becomes harder as the field gets huge. Due to all these reasons, this type of attack becomes infeasible.

4.3.2 Known Plaintext Attack

Let $(C_i^{'}, C_i)$ be the ciphertext corresponding to the plaintext M_i known to the adversary. From this information he needs to find the next plaintext M_{i+1} from the corresponding ciphertext $(C_{i+1}^{'}, C_{i+1})$. In our situation this type of attack can be made infeasible by using different exponents v and w in the encryption of every new message. So the knowledge of previous plaintext-ciphertext pairs provide not enough information to find the next plaintext.

4.4 Efficiency Analysis

This section deals with some aspects related to the efficiency of the proposed cryptosystem. The details regarding the bit-complexity analysis are elaborated. Then bit-complexity of our proposed cryptosystem is estimated and compared with the classical RSA cryptosystem. There are more efficient modifications of RSA cryptosystem, but we are comparing to the classical one, the state of the art (SoTA) rival algorithm.

4.4.1 Computational Complexity of Some Matrix Operations

The bit-complexity of basic operations in the residue ring $\mathbb{Z}_n$ are given in TABLE (4.1) [39].

$Operations\ (\forall z_1, z_2, z \in \mathbb{Z}_n)$		$Bit - Complexity$
Modular addition	$(z_1 + z_2) \mod n$	$O(\lg n)$
Modular subtraction	$(z_1 + z_2) \mod n$	$O(\lg n)$
Modular inversion	$z^{-1} \mod n$	$O\left((\lg n)^2\right)$
Modular multiplication	$(z_1 z_2) \mod n$	$O\left((\lg n)^2\right)$
Modular exponentiation	$z^k \mod n,\ k < n$	$O\left((\lg n)^3\right)$

Table 4.1: Bit-Complexity of Basic Modular Operations

Now, we give the bit-complexity of modular arithmetic in matrices. The subsequent discussion is for matrices of order 2.

1. **Bit-Complexity of Matrix Multiplication**

   In the matrices of order 2, the multiplication of any two matrices consists of 8-modular multiplication and 4-modular additions. By considering only the multiplication, the bit-complexity of matrix multiplication is estimated as $8(\lg n)^2$-bit operations, where $\lg n$ are the number of bits in binary representation of n.

   For a 112-bit prime, the estimate of bit-complexity of matrix multiplication is

$$8(112)^2 = 2^3(16 \times 7)^2 = 2^3(2^8 \times 7^2) = 2^{10} \times 98 \approx 9 \cdot 8 \times 10^4 - \text{bit operations.}$$

2. **Bit-Complexity of Scalar Multiplication**

   This computation involves 4-modular multiplications. The estimated bit-complexity of scalar multiplication is $4(\lg n)^2$-bit operations.

   For a 112-bit prime, the estimate of bit-complexity of scalar multiplication is

   $$4(112)^2 = 2^{10} \times 49 \approx 4 \cdot 9 \times 10^4 - \text{bit operations.}$$

3. **Bit-Complexity of Matrix Inversion**

   The calculation of determinant of a matrix involves $2-$modular multiplication and $1-$modular subtraction. The process of finding inverse of a matrix consists of modular inversion of value of determinant and its scalar multiplication with the matrix. So, the estimated bit-complexity of matrix inversion is $2(\lg n)^2 + (\lg n)^2 + 4(\lg n)^2 = 7(\lg n)^2$-bit operations.

   For a 112-bit prime, the estimate of bit-complexity of matrix inversion is

   $$7(112)^2 = 7^3 \times 2^8 = 3 \cdot 43 \times 10^2 \times 2 \cdot 56 \times 10^2 \approx 8 \cdot 8 \times 10^4 - \text{bit operations.}$$

4. **Bit-Complexity of Matrix Exponentiation**

   Assume that the exponent has size in bits proportional to that of n. That is the size of exponent is $O(\lg n)$. Thus, the operation of matrix exponentiation involves $\lg n$ modular multiplication of matrices. So, the estimate of bit-complexity of matrix exponentiation is $(\lg n) \cdot 8(\lg n)^2 = 8(\lg n)^3$-bit operations.

   For a 112-bit prime, the estimate of bit-complexity of matrix exponentiation is

   $$8(112)^2 = 7^3 \times 2^{15} = 3 \cdot 2768 \times 10^4 \times 3 \cdot 43 \times 10^2 \approx 1 \cdot 1 \times 10^7 - \text{bit operations.}$$

4.4.2 Bit-Complexity of Proposed Cryptosystem

There are 2−exponentiation, 1−inversion and 9−multiplications of matrices are involved in the encryption of our proposed Cryptosystem 4.2.1. Neglecting other operations as compared to the matrix exponentiation, the bit-complexity of encryption is

$$2 \times (1 \cdot 1 \times 10^7) = 2 \cdot 2 \times 10^7 - \text{bit operations.} \tag{4.12}$$

In decryption, there are 4−inversion and 8−multiplications of matrices are involved. Therefore the estimated bit-complexity is

$$4 \times (8 \cdot 8 \times 10^4) + 8 \times (4 \cdot 9 \times 10^4) = 7 \cdot 44 \times 10^5 - \text{bit operations.} \tag{4.13}$$

As mentioned earlier, we are considering the RSA cryptosystem as point of reference for comparing the efficiency of our cryptosystem. The encryption and decryption of RSA cryptosystem involve only modular exponentiation. The bit-complexity of RSA encryption and decryption is $(\lg n)^3$−bit operations. For 1024−bit n, we obtain the estimate of bit-complexity as

$$(1024)^3 = (2^{10})^3 \approx (10^3)^3 = 10^9 - \text{bit operations.} \tag{4.14}$$

In view of expressions (4.12), (4.13) and (4.14), we can say that our proposed cryptosystem is more efficient than classical RSA cryptosystem.

Chapter 5

New Variants of Stickel's Key Exchange Protocol

In this chapter, we present two variants of Stickel's protocol based on noncommutative structures. Particularly, the use of polynomials over noncommutative rings is examined. The underlying idea of the protocols is to generalize the decomposition problem over noncommutative rings. Due to use of noncommutative structure and polynomials over noncommutative rings, it is expected that the proposed protocols provide high security levels. Some issues regarding the choice of parameters involved in the protocols, are also addressed. Further, a brief note on the security of the protocols, is also presented.

5.1 Some Cryptographic Hard Problems and Noncommutativity

As discussed earlier, the trap door one way function can be used behind the public key cryptography. Therefore, there must exist computationally hard problems for defining the protocols based on public keys. The following are the problems involved in the security of different noncommutative group based cryptographic proposals. We also defined a variant of these problems and use it in our proposal.

Definition 5.1.1. (**Decomposition Problem** [10])

Let G be a noncommutative group and S be a subset of G. Given two elements $g, h \in G$, the problem of finding two elements $k_1, k_2 \in S$, where

$$h = k_1 g k_2$$

is known as the decomposition problem (DP).

Generally, for a noncommutative group G, the decomposition problem is considered difficult enough regarding the cryptographic assumptions. More specifically, the DP is intractable which means that there is no probabilistic polynomial time algorithm that is used to solve DP with non negligible accuracy.

Definition 5.1.2. (**Symmetric Decomposition Problem** [10])

Let G be a noncommutative group, two elements $g, h \in G$ and $m, n \in \mathbb{Z}$. Finding the element $k \in G$, where

$$h = k^m g k^n,$$

is known as the symmetric decomposition problem (SDP).

Definition 5.1.3. (**Generalized Symmetric Decomposition Problem** [10])

Let G be a noncommutative group, a subset S of G, two elements $g, h \in G$ and $m, n \in \mathbb{Z}$. Finding the element $k \in S$, where

$$h = k^m g k^n,$$

is known as the generalized symmetric decomposition problem (GSDP).

In the view of the these problems, we now define the following cryptographic problem over a noncommutative group G.

Definition 5.1.4. (**Generalized Decomposition Problem**)

Let G be a noncommutative group, two subsets $S_1, S_2 \subseteq G$ and two elements $g, h \in G$ and $m, n \in \mathbb{Z}$. Finding two elements $k_1 \in S_1$, and $k_2 \in S_2$, where

$$h = k_1^m g k_2^n$$

is known as the generalized decomposition problem (GDP).

Note that the GDP can be considered as a special form of DP. If the size of sets S_1 and S_2 are taken to be more than 200 integers and also extracting k_1 and k_2 from $k_1^m g k_2^n$ is hard from the membership information of sets S_1 and S_2, then the GDP is at least as hard as DP. It follows that the generalized decomposition assumption states that the GDP is intractable, which means there does not exist any probabilistic polynomial time algorithm that can solve GDP with non negligible accuracy.

5.2 Motivating Literature

In [10], Cao at. el. proposed a scheme for devising a public key cryptosystem based on noncommutative rings. It is proposed that for a noncommutative ring, the set of polynomials can be considered as the underlying work structure. The authors construct Diffie-Helman like key exchange protocol and ElGamal like cryptosystems, using polynomials over noncommutative ring. They also extend their technique to noncommutative groups. The difficulty of their Diffie-Hellman-Like key agreement proposal is based on the following hard problem:

Definition 5.2.1. (**Polynomial Symmetric Decomposition Problem [10]**) Let R be a noncommutative ring. For any element $a \in R$, consider the set $S_a \subseteq R$ defined as

$$S_a = \{P(a) \mid P(x) \in \mathbb{Z}_{>0}[x]\}$$

and $m, n \in \mathbb{Z}$. Given two elements $g, h \in R$, finding the element $k \in S_a$, where

$$h = k^m g k^n \ ,$$

is known as the polynomial symmetric decomposition problem (PSDP).

5.2.1 Diffie-Hellman-Like Protocol For Noncommutative Rings

A common secret key can be shared between two communicating parties Alice and Bob, as follows:

1. To launch the protocol, two elements $a, b \in R$ and two small (say, less than 10) positive integers $m, n \in \mathbb{Z}_{>0}$ are chosen by Alice and transmitted to Bob.

2. Alice selects a random polynomial $P(x) \in \mathbb{Z}_{>0}[x]$, such that $P(a) \neq 0$ and keeps $P(a)$ as her private key.

3. A random polynomial $Q(x) \in \mathbb{Z}_{>0}[x]$, such that $Q(a) \neq 0$ is selected by Bob and $Q(a)$ is his private key.

4. Alice computes
$$r_A = P(a)^m b P(a)^n$$
and transmits it to Bob.

5. Bob computes
$$r_B = Q(a)^m b Q(a)^n$$
and transmits r_B to Alice.

6. The common secret key computed by Alice is
$$K_A = P(a)^m r_B P(a)^n.$$

7. Bob computes
$$K_B = Q(a)^m r_A Q(a)^n$$
as the common secret key.

We now define a new variant of GDP over a noncommutative ring R and name it as polynomial generalized decomposition problem (PGDP).

Definition 5.2.2. (**Polynomial Generalized Decomposition Problem**)
Let R be a noncommutative ring. Let $Z(R)$ be the centre of R and $Z(R)[X]$ be the polynomial ring over $Z(R)$. For any random elements a_1, $a_2 \in R$, consider the sets $S_{a_1} \subseteq R$ and $S_{a_2} \subseteq R$ defined as

$$\begin{aligned} S_{a_1} &= \{P(a_1) : P(X) \in Z(R)[X]\}, \\ S_{a_2} &= \{P(a_2) : P(X) \in Z(R)[X]\} \end{aligned}$$

and $m, n \in Z$. Given two elements $g, h \in R$, finding two elements $k_1 \in S_{a_1}$ and $k_2 \in S_{a_2}$, where

$$h = k_1^m g k_2^n \ ,$$

is known as the polynomial generalized decomposition problem (PGD).

So, the PGD cryptographic assumption states that PGDP over R is intractable which means there does not exist any probabilistic polynomial time algorithm that can solve PGDP with nonnegligible accuracy. We are going to use PGDP in our proposed key exchange protocols as described in the following section.

5.3 Proposed Key Exchange Protocols

In this section, a novel key exchange protocol is proposed to start a communication session between two parties, namely, Alice and Bob. Our scheme is based on a noncommutative ring.

Protocol 5.3.1.
Consider a noncommutative ring R. Let $Z(R)$ be the centre of R and $Z(R)[x]$ be the polynomial ring over $Z(R)$. The following steps would be executed for sharing a secret key.

Global Agreement/Parameters: The elements $c \in R \setminus Z(R)$ and $a_1, a_2 \in R$.

1. Alice chooses a polynomial $P(x) \in Z(R)[x]$, randomly, as her private key such that $P(a_1) \neq 0$ and $P(a_2) \neq 0$. She also chooses $r, s \in \mathbb{Z}_{>0}$. She

computes

$$K_A = (P(a_1))^r\, c\, (P(a_2))^s \tag{5.1}$$

and sends it to Bob.

2. Now, a random polynomial $Q(x) \in Z(R)[x]$, is chosen by Bob as his private key such that $Q(a_1) \neq 0$ and $Q(a_2) \neq 0$. He also chooses $u, v \in \mathbb{N}$. He then computes

$$K_B = (Q(a_1))^u\, c\, (Q(a_2))^v \tag{5.2}$$

and sends it to Alice.

3. The shared key computed by Alice, is

$$W_A = (P(a_1))^r\, K_B\, (P(a_2))^s\,. \tag{5.3}$$

4. Bob finds the shared secret key as

$$W_B = (Q(a_1))^u\, K_A\, (Q(a_2))^v\,. \tag{5.4}$$

The correctness of the proposed Protocol 5.3.1 is shown as follows:

Theorem 5.3.2.

In view of specified notation of Protocol 5.3.1, it follows that the shared secret keys obtained by both entities Alice and Bob are same that is $W_A = W_B$.

Proof. First consider the expression (5.3)

$$W_A = (P(a_1))^r\, K_B\, (P(a_2))^s\,,$$

by using (5.2), above expression becomes

$$W_A = (P(a_1))^r\, (Q(a_1))^u\, c\, (Q(a_2))^v\, (P(a_2))^s\,. \tag{5.5}$$

Expression (5.4) with (5.1) gives

$$W_B = \left(Q\left(a_1\right)\right)^u \left(P\left(a_1\right)\right)^r c\left(P\left(a_2\right)\right)^s \left(Q\left(a_2\right)\right)^v . \tag{5.6}$$

Although the ring R is noncommutative, but we have

$$\left(P\left(g\right)\right)^{\ell} \left(Q\left(g\right)\right)^m = \left(Q\left(g\right)\right)^m \left(P\left(g\right)\right)^{\ell} , \qquad \begin{array}{c} \forall\ g \in R, \quad \forall\ \ell, m \in \mathbb{Z}_{>0} \\ \forall\ P(x), Q(x) \in Z(R)[x]. \end{array} \tag{5.7}$$

In view of property (5.7), expressions (5.5) and (5.6) are same. □

Obviously, for the proposed public key exchange protocol, the passive attack can be resisted with the PGD assumption over the noncommutative ring.

Practically, the steps 1 and 2 of Protocol 5.3.1 can be done simultaneously. Alice communicates to Bob through one pass and similarly, in the step 2, there is a requirement of only one pass from Bob to Alice. Finally, the steps 3 and 4 can be executed by both entities, without any further communications. This communication is described in FIGURE 5.1.

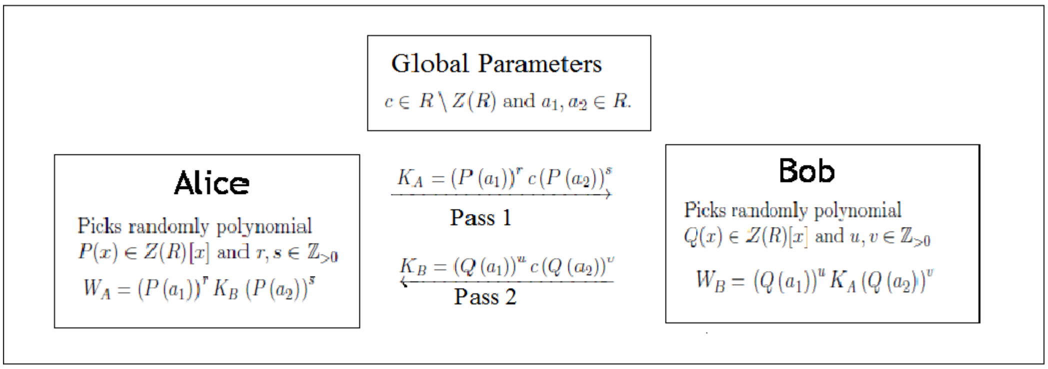

Global Parameters
$c \in R \setminus Z(R)$ and $a_1, a_2 \in R$.
Alice
Picks randomly polynomial
$P(x) \in Z(R)[x]$ and $r, s \in \mathbb{Z}_{>0}$
$W_A = \left(P\left(a_1\right)\right)^r K_B \left(P\left(a_2\right)\right)^s$
$K_A = \left(P\left(a_1\right)\right)^r c\left(P\left(a_2\right)\right)^s$
Pass 1
$K_B = \left(Q\left(a_1\right)\right)^u c\left(Q\left(a_2\right)\right)^v$
Pass 2
Bob
Picks randomly polynomial
$Q(x) \in Z(R)[x]$ and $u, v \in \mathbb{Z}_{>0}$
$W_B = \left(Q\left(a_1\right)\right)^u K_A \left(Q\left(a_2\right)\right)^v$

FIGURE 5.1: Stickel's Variant Protocol

We naturally employ the square and multiply procedure to speed up the calculations for conducting exponentiation. This procedure helps to compute large powers of random elements very quickly [55].

Example 5.3.3.

To understand the implementation details of the Protocol 5.3.1, we take $M(2, \mathbb{Z}_p)$, with a large prime p. The care must be taken in the choice of a large value of prime, approximately of the order of 60 decimal digits. But for simplifying the calculations, we take $\mathbb{Z}_{29}$ in this example.

In the initial set up of the Protocol 5.3.1, we take $M(2, \mathbb{Z}_{29})$ as the noncommutative ring. All the computations are performed in ApCoCoA [3]. The center of $M(2, \mathbb{Z}_{29})$ is the set

$$\begin{aligned} Z(M(2,\mathbb{Z}_{29})) &= \{gI \mid g \in \mathbb{Z}_{29}\}, \qquad \text{where} \\ I &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

is the identity matrix. Now the following steps of Protocol 5.3.1 would be executed.

Global Parameters: Assume that the following matrices are the global parameters:

$$\left.\begin{aligned} A_1 = \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}, \\ A_2 = \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix} \end{aligned}\right\} \in M(2, \mathbb{Z}_{29})$$

and

$$C = \begin{bmatrix} 11 & 3 \\ 2 & 0 \end{bmatrix} \in M(2, \mathbb{Z}_{29}) \setminus Z(M(2, \mathbb{Z}_{29}))$$

1. Alice chooses a random polynomial from $Z(M(2, \mathbb{Z}_{29}))[x]$

$$P(x) = \begin{bmatrix} 21 & 0 \\ 0 & 21 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} x^2 + \begin{bmatrix} 13 & 0 \\ 0 & 13 \end{bmatrix} x^4$$

and $(r, s) = (2, 8)$. Then

$$P(A_1) = \begin{bmatrix} 21 & 0 \\ 0 & 21 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}^2 + \begin{bmatrix} 13 & 0 \\ 0 & 13 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}^4$$

$$= \begin{bmatrix} 13 & 0 \\ 12 & 9 \end{bmatrix}$$

and

$$\begin{aligned} P\left(A_2\right) &= \begin{bmatrix} 21 & 0 \\ 0 & 21 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix}^2 + \begin{bmatrix} 13 & 0 \\ 0 & 13 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix}^4 \\ &= \begin{bmatrix} 16 & 20 \\ 3 & 4 \end{bmatrix}. \end{aligned}$$

Now she computes

$$\begin{aligned} K_A = \left(P\left(A_1\right)\right)^2 C \left(P\left(A_2\right)\right)^8 &= \begin{bmatrix} 13 & 0 \\ 12 & 9 \end{bmatrix}^2 \begin{bmatrix} 11 & 3 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 16 & 20 \\ 3 & 4 \end{bmatrix}^8 \\ &= \begin{bmatrix} 23 & 9 \\ 7 & 0 \end{bmatrix} \end{aligned}$$

and sends it to Bob.

2. Now, a random polynomial is chosen by Bob

$$Q(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix} x^3 \in Z(M(2, \mathbb{Z}_{29}))[x]$$

and $(u, v) = (3, 4)$. Then

$$\begin{aligned} Q(A_1) &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}^3 \\ &= \begin{bmatrix} 8 & 0 \\ 2 & 17 \end{bmatrix} \end{aligned}$$

$$Q(A_2) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix} + \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix}^3$$

$$= \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}.$$

Then he computes

$$\begin{aligned} K_B = (Q(A_1))^3 C (Q(A_2))^4 &= \begin{bmatrix} 8 & 0 \\ 2 & 17 \end{bmatrix}^3 \begin{bmatrix} 11 & 3 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}^4 \\ &= \begin{bmatrix} 13 & 22 \\ 16 & 6 \end{bmatrix} \end{aligned}$$

and sends it to Alice.

3. The shared key computed by Alice is

$$\begin{aligned} W_A = (P(A_1))^2 K_B (P(A_2))^8 &= \begin{bmatrix} 13 & 0 \\ 12 & 9 \end{bmatrix}^2 \begin{bmatrix} 13 & 22 \\ 16 & 6 \end{bmatrix} \begin{bmatrix} 16 & 20 \\ 3 & 4 \end{bmatrix}^8 \\ &= \begin{bmatrix} 14 & 8 \\ 25 & 18 \end{bmatrix}. \end{aligned}$$

4. Bob finds the shared key as

$$\begin{aligned} W_B = (Q(A_1))^3 K_A (Q(A_2))^4 & \\ &= \begin{bmatrix} 8 & 0 \\ 2 & 17 \end{bmatrix}^3 \begin{bmatrix} 23 & 9 \\ 7 & 0 \end{bmatrix} \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}^4 \\ &= \begin{bmatrix} 14 & 8 \\ 25 & 18 \end{bmatrix}. \end{aligned}$$

Note that although the elements A_1 and A_2 are publicly known, but for an adversary the polynomials $P(x)$ and $Q(x) \in Z(M(2, \mathbb{Z}_{29}))[x]$ are unknown. Hence, the following elements

$$(P(A_1))^r = \begin{bmatrix} 24 & 0 \\ 3 & 23 \end{bmatrix}, \qquad (P(A_2))^s = \begin{bmatrix} 6 & 1 \\ 19 & 17 \end{bmatrix},$$

$$(Q(A_1))^u = \begin{bmatrix} 19 & 0 \\ 21 & 12 \end{bmatrix}, \qquad (Q(A_2))^v = \begin{bmatrix} 7 & 0 \\ 0 & 7 \end{bmatrix},$$

are also unknown.

Let us suppose that an adversary intercepts K_A and K_B. For obtaining the shared secret key, he only knows the following expressions

$$\begin{aligned} K_A &= (P(A_1))^r C (P(A_2))^s = \begin{bmatrix} 23 & 9 \\ 7 & 0 \end{bmatrix}, \\ K_B &= (Q(A_1))^u C (Q(A_2))^v = \begin{bmatrix} 13 & 22 \\ 16 & 6 \end{bmatrix}. \end{aligned}$$

That is equivalent to find the solution of generalized decomposition problem. For the brute force attack, one has to check the set of polynomials whose coefficients come from the center of the ring. The feasibility of brute force attack could be avoided because the number of polynomials having degree α and coefficients from $Z(M(2, \mathbb{Z}_p))$, is $(p-1)p^\alpha$. The values of $(p-1)p^\alpha$ for different values of α and p are shown in TABLE 5.1.

$\alpha \diagdown p$	2	3	5	7	11	13	17	19	$\cdots$
2	4	18	100	294	1210	2028	4624	6498	$\cdots$
3	8	54	500	2058	13310	26364	78608	123462	$\cdots$
4	16	162	2500	14406	146410	342732	1336336	2345778	$\cdots$
5	32	486	12500	100842	1610510	4455516	22717712	44569782	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
12	4096	1062882	976562500	83047723206	31384283767210	279577021469772	9321955795676180	39839668543190900	$\cdots$
13	8192	3188646	4882812500	581334062442	345227121439310	3634501279107040	158473248526495000	756953702320627000	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
20	1048576	6973568802	381469726562500	478753597785672000	6727499949325600000000	2280595652985700000000	650277025063612000000000	676619522235827000000000	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

TABLE 5.1: The size of set of polynomials of different degrees α and primes p with order of matrices $n = 2$

.

We note that Protocol 5.3.1 exhibits some kind of symmetry in the sense that computation of public keys K_A and K_B involves the same polynomial which is multiplied with element c, from both sides. This symmetry can be avoided by introducing two different polynomials for each user. Therefore, Protocol 5.3.1 can be modified as follows:

Protocol 5.3.4.

Global Agreement/Parameters: The elements $c \in R \setminus Z(R)$ and $a_1, a_2 \in R$.

1. Alice chooses two random polynomials $P_1(x)$ and $P_2(x) \in Z(R)[x]$ such that $P_1(a_1) \neq 0$ and $P_2(a_2) \neq 0$. She also chooses $r, s \in \mathbb{Z}_{>0}$. She computes

$$K_A = \left(P_1\left(a_1\right)\right)^r c \left(P_2\left(a_2\right)\right)^s$$

   and sends it to Bob.

2. Bob also selects two polynomials $Q_1(x)$ and $Q_2(x) \in Z(R)[x]$ such that $Q_1(a_1) \neq 0$ and $Q_2(a_2) \neq 0$. He also chooses $u, v \in \mathbb{Z}_{>0}$. He then computes

$$K_B = \left(Q_1\left(a_1\right)\right)^u c \left(Q_2\left(a_2\right)\right)^v$$

   and sends it to Alice.

3. The shared key computed by Alice, is

$$W_A = \left(P_1\left(a_1\right)\right)^r K_B \left(P_2\left(a_2\right)\right)^s. \tag{5.8}$$

4. Bob finds the key as

$$W_B = \left(Q_1\left(a_1\right)\right)^u K_A \left(Q_2\left(a_2\right)\right)^v. \tag{5.9}$$

The shared secret keys obtained in both cases (5.8) and (5.9) are same, following a similar argument as in Protocol 5.3.1.

The next example shows the procedure of the Protocol 5.3.4.

Example 5.3.5. The platform group and global parameters are selected as in Example 5.3.3.

1. Alice chooses two random polynomial

$$P_1(x) = \begin{bmatrix} 21 & 0 \\ 0 & 21 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} x^2 + \begin{bmatrix} 13 & 0 \\ 0 & 13 \end{bmatrix} x^4 \in Z(M(2, \mathbb{Z}_{29}))[x],$$

$$P_2(x) = \begin{bmatrix} 11 & 0 \\ 0 & 11 \end{bmatrix} + \begin{bmatrix} 24 & 0 \\ 0 & 24 \end{bmatrix} x^5 + \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} x^9 \in Z(M(2, \mathbb{Z}_{29}))[x]$$

and $(r, s) = (2, 6)$. Then

$$\begin{aligned} P_1(A_1) &= \begin{bmatrix} 21 & 0 \\ 0 & 21 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}^2 + \begin{bmatrix} 13 & 0 \\ 0 & 13 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}^4 \\ &= \begin{bmatrix} 13 & 0 \\ 12 & 9 \end{bmatrix} \qquad \text{and} \\ P_2(A_2) &= \begin{bmatrix} 11 & 0 \\ 0 & 11 \end{bmatrix} + \begin{bmatrix} 24 & 0 \\ 0 & 24 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix}^5 + \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix}^9 \\ &= \begin{bmatrix} 9 & 16 \\ 14 & 11 \end{bmatrix}. \end{aligned}$$

She computes

$$\begin{aligned} K_A &= (P_1(A_1))^2 C (P_2(A_2))^6 \\ &= \begin{bmatrix} 13 & 0 \\ 12 & 9 \end{bmatrix}^2 \begin{bmatrix} 11 & 3 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 9 & 16 \\ 14 & 11 \end{bmatrix}^6 \\ &= \begin{bmatrix} 2 & 1 \\ 27 & 9 \end{bmatrix} \end{aligned}$$

and sends it to Bob.

2. Bob selects two polynomials, randomly

$$\begin{aligned} Q_1(x) &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix} x^3 \in Z(M(2,\mathbb{Z}_{29}))[x], \\ Q_2(x) &= \begin{bmatrix} 16 & 0 \\ 0 & 16 \end{bmatrix} + \begin{bmatrix} 19 & 0 \\ 0 & 19 \end{bmatrix} x + \begin{bmatrix} 27 & 0 \\ 0 & 27 \end{bmatrix} x^{10} \in Z(M(2,\mathbb{Z}_{29}))[x] \end{aligned}$$

and $(u, v) = (3, 4)$. Then

$$\begin{aligned} Q_1(A_1) &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}^3 \\ &= \begin{bmatrix} 8 & 0 \\ 2 & 17 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} Q_2(A_2) &= \begin{bmatrix} 16 & 0 \\ 0 & 16 \end{bmatrix} + \begin{bmatrix} 19 & 0 \\ 0 & 19 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix} + \begin{bmatrix} 27 & 0 \\ 0 & 27 \end{bmatrix} \begin{bmatrix} 11 & 2 \\ 9 & 4 \end{bmatrix}^{10} \\ &= \begin{bmatrix} 12 & 15 \\ 24 & 3 \end{bmatrix}. \end{aligned}$$

Then he computes

$$\begin{aligned} K_B &= \left(Q_1\left(A_1\right)\right)^3 C \left(Q_2\left(A_2\right)\right)^4 \\ &= \begin{bmatrix} 8 & 0 \\ 2 & 17 \end{bmatrix}^3 \begin{bmatrix} 11 & 3 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 12 & 15 \\ 24 & 3 \end{bmatrix}^4 \\ &= \begin{bmatrix} 12 & 10 \\ 16 & 26 \end{bmatrix} \end{aligned}$$

and sends it to Alice.

3. The shared key computed by Alice is

$$W_A = \left(P_1\left(A_1\right)\right)^2 K_B \left(P_2\left(A_2\right)\right)^6$$

$$
\begin{aligned}
&= \begin{bmatrix} 13 & 0 \\ 12 & 9 \end{bmatrix}^2 \begin{bmatrix} 12 & 10 \\ 16 & 26 \end{bmatrix} \begin{bmatrix} 9 & 16 \\ 14 & 11 \end{bmatrix}^6 \\
&= \begin{bmatrix} 2 & 11 \\ 3 & 0 \end{bmatrix}.
\end{aligned}
$$

4. Bob finds the shared key as

$$
\begin{aligned}
W_B &= (Q_1(A_1))^3 K_A (Q_2(A_2))^4 \\
&= \begin{bmatrix} 8 & 0 \\ 2 & 7 \end{bmatrix}^3 \begin{bmatrix} 2 & 1 \\ 27 & 9 \end{bmatrix} \begin{bmatrix} 12 & 15 \\ 24 & 3 \end{bmatrix}^4 \\
&= \begin{bmatrix} 2 & 11 \\ 3 & 0 \end{bmatrix}.
\end{aligned}
$$

Here, an adversary knows A_1 and A_2 since they are public, but the polynomials $P_1(x)$, $P_2(x)$, $Q_1(x)$ and $Q_2(x) \in Z(M(2, \mathbb{Z}_{29}))[x]$ remain unknown. So the following elements are also unknown

$$
\begin{aligned}
(P_1(A_1))^r &= \begin{bmatrix} 24 & 0 \\ 3 & 23 \end{bmatrix}, \qquad (P_2(A_2))^s = \begin{bmatrix} 2 & 5 \\ 8 & 28 \end{bmatrix}, \\
(Q_1(A_1))^u &= \begin{bmatrix} 19 & 0 \\ 21 & 12 \end{bmatrix}, \qquad (Q_2(A_2))^v = \begin{bmatrix} 8 & 8 \\ 7 & 9 \end{bmatrix}.
\end{aligned}
$$

Let us suppose that an adversary intercepts K_A and K_B. Similar argument as given for Protocol 5.3.1, for obtaining the shared secret key, he only knows the following expressions

$$
\begin{aligned}
K_A &= (P_1(A_1))^r C (P_2(A_2))^s = \begin{bmatrix} 2 & 1 \\ 27 & 9 \end{bmatrix}, \\
K_B &= (Q_1(A_1))^u C (Q_2(A_2))^v = \begin{bmatrix} 12 & 10 \\ 16 & 26 \end{bmatrix},
\end{aligned}
$$

where the polynomials $P_1(x)$, $P_2(x)$, $Q_1(x)$ and $Q_2(x) \in Z(M(2, \mathbb{Z}_{29}))[x]$ and the integers r, s, u and v have to be found, first. This problem leads to solve generalized decomposition problem.

There are two polynomials involved in the public key of each user. If the degrees of two polynomials are α and β, respectively, then the total number of possible polynomials for one user is $(p-1)^2p^{(\alpha+\beta)}$. The values of $(p-1)^2p^{(\alpha+\beta)}$ for different values of α, β and $p = 29$ are shown in TABLE 5.2.

$\alpha \diagdown \beta$	3	7	11	$\cdots$
2	16080740816	11373602445081300	8044332910959540000000	$\cdots$
3	466341483664	329834470907358000	233285654417827000000000	$\cdots$
4	13523903026256	9565199656313370000	6765283978116980000000000	$\cdots$
5	392193187761424	277390790033088000000	196193235365392000000000000	$\cdots$
6	11373602445081300	8044332910959540000000	5689603825596380000000000000	$\cdots$
7	329834470907358000	233285654417827000000000	164998510942295000000000000000	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

TABLE 5.2: The size of set of polynomials of different degrees α and β and $p = 29$ with order of matrices $n = 2$.

5.4 Security Aspects of the Proposed Protocols

This section presents discussion on the security analysis of the protocols proposed in Section 5.3.4.

As mentioned earlier, the security of the protocols depends on the solution of the generalized decomposition problem. For solving such problem in a noncommutative ring, no polynomial time algorithm is known, to the best of our knowledge. In case of both protocols, an adversary has to find the solution of the decomposition problems which is expressed as the following system of equations

$$M_AM_B = M_BM_A, \tag{5.10}$$

$$N_AN_B = N_BN_A, \tag{5.11}$$

$$M_ACN_A = K_A, \tag{5.12}$$

$$M_BCN_B = K_B. \tag{5.13}$$

The adversary also knows the elements a_1,$a_2 \in R$ and $c \in R \setminus Z(R)$. To break the Protocol 5.3.1, the adversary has to find the elements M_A, M_B, N_A and N_B. For this, the adversary tries to find out two polynomials $H_1(x), H_2(x) \in Z(R)[x]$ and numbers $\ell_1, \ell_2, m_1, m_2 \in \mathbb{N}$ such that

$$\begin{aligned} (H_1(a_1))^{\ell_1} &= M_A, \\ (H_1(a_2))^{\ell_2} &= N_A, \\ (H_2(a_1))^{m_1} &= M_B \qquad \text{and} \\ (H_2(a_2))^{m_2} &= N_B. \end{aligned}$$

Then, one can guarantee the conditions (5.10) and (5.11).

We note that the size of space of polynomials over $Z(R)$ is actually a set of all possible random choices. Also, the adversary has to verify the conditions (5.12) and (5.13). By taking the space of polynomials over $Z(R)$ to be large enough, the possibility of brute force attack would be infeasible.

In the same fashion, for Protocol 5.3.4, an adversary has to find the elements M_A, M_B, N_A, and N_B. Also, the polynomials $H_1(x), G_1(x), H_2(x), G_2(x) \in Z(R)[x]$ and natural numbers $\ell_1, \ell_2, m_1, m_2 \in \mathbb{N}$ have to be determined, such that

$$\begin{aligned} (H_1(a_1))^{\ell_1} &= M_A, \\ (G_1(a_2))^{\ell_2} &= N_A, \\ (H_2(a_1))^{m_1} &= M_B \qquad \text{and} \\ (G_2(a_2))^{m_2} &= N_B. \end{aligned}$$

Notice that conditions (5.10) and (5.11) are again guaranteed. In this case, there is an increase in the number of polynomials which would add an extra difficulty.

The solution of intractable decomposition problem is involved in a brute force attack. The only possibility to get solution of generalized decomposition problem is to search out the space of polynomials $Z(R)[x]$. This search becomes infeasible when the order of the space of the polynomials would be taken large enough. To make brute force attack infeasible, it is suggested to make the choice of a prime p of order of 60 decimal digits and polynomials of degree 20.

The order of the matrices n can be chosen so that $2^n - 1$ is a Mersenne prime. The choice of a Mersenne prime $n > 31$ is recommended.

Chapter 6

A Novel Algebraic Public Key Cryptosystem

Different public key exchange protocols can be employed to design a cryptosystem. The most famous example is the ElGamal cryptosystem [17] which uses the Diffie-Hellman key establishment [15] to do encryption and decryption. In the same fashion, we propose a public key cryptosystem in spirit of a variant of Stickel key exchange protocol presented in Chapter 5. This variant uses the polynomials over noncommutative groups as underlying work structure. The useful feature of the presented cryptosystem is that it provides favorable security level because of use of inner automorphisms of a noncommutative group. The issues regarding the choice of parameters and platform are discussed. A brief note on security of the proposed cryptosystem is also presented.

6.1 Proposed Cryptosystem

Now, we present the general scheme for a public key cryptosystem as follows:

Cryptosystem 6.1.1.

Initial Setup: Let G be a noncommutative group. Let $Z(G)$ be the centre of G and $Z(G)[y]$ be the polynomial ring over $Z(G)$.

For developing a communication, Bob and Alice will do the following steps:

Key Generation (KG)

To compute secret and public key pair (S_A, P_A) of Alice, KG works as follows: The elements $g_1, g_2 \in G, c \in G \setminus Z(G)$, a polynomial $P(y) \in Z(G)[y]$, such that $P(g_1) \neq 0$ and $P(g_2) \neq 0$ and $r, s \in \mathbb{N}$ are picked randomly. The secret and public keys are $S_A = (r, s, P(y))$ and $P_A = (g_1, g_2, c, K_A)$, respectively, where

$$K_A = \left(P\left(g_1\right)\right)^r c \left(P\left(g_2\right)\right)^s . \tag{6.1}$$

Encryption Algorithm (E_{P_A})

Input: Plaintext message $M \in G$, Public key of Alice $P_A = (g_1, g_2, c, K_A)$

Output: Ciphertext C_A

To send a message (plaintext) $M \in G$ to Alice, Bob executes following steps:

E-1) He chooses random polynomial $Q(y) \in Z(G)[y]$ such that $Q(g_1) \neq 0$ and $Q(g_2) \neq 0$ and elements $u, \nu \in \mathbb{N}$. Then, he computes

$$B = \left(Q\left(g_1\right)\right)^u c \left(Q\left(g_2\right)\right)^v .$$

E-2) Now he uses the inner automorphism

$$\phi : g \longmapsto B^{-1} g B, \qquad \text{for } g \in G$$

of the group G to compute

E-3)

$$C_1 = \phi(K_A).$$

E-4) Finally, he computes

$$C = C_1 M.$$

E-5) The ciphertext transmitted to Alice is $C_A = (C', C)$, where

$$C' = (Q(g_1))^u K_A (Q(g_2))^v . \tag{6.2}$$

Decryption Algorithm (D_{S_A})

Input: Ciphertext message C_A, secret key of Alice $S_A = (r, s, P(y))$

Output: Plaintext M

When Alice receives the ciphertext $C_A = (C', C)$,, she executes following steps:

D-1) She computes

$$\tilde{D} = (P(g_1))^{-r} C'(P(g_2))^{-s}. \tag{6.3}$$

D-2) She defines an inner automorphism

$$\rho : g \longmapsto \tilde{D}^{-1} g \tilde{D}, \qquad \text{for } g \in G$$

and computes

D-3)

$$\tilde{D_1} = \rho\left(K_A^{-1}\right).$$

D-4) Then plaintext message M can be obtained as

$$M = \tilde{D_1} C.$$

Theorem 6.1.2.

In view of specified notation of Cryptosystem 6.1.1, the correctness of its decryption is guaranteed.

Proof. **Correctness** : The correctness of the scheme is guaranteed as follows:

Consider expression (6.3)

$$\tilde{D} = (P(g_1))^{-r} C'(P(g_2))^{-s},$$

using expression (6.2) and then (6.1) above expression reduces to

$$\tilde{D} = \left(P\left(g_1\right)\right)^{-r}\left(\left(Q\left(g_1\right)\right)^{u}\left(\left(P\left(g_1\right)\right)^{r} c\left(P\left(g_2\right)\right)^{s}\right)\left(Q\left(g_2\right)\right)^{v}\right)\left(P\left(g_2\right)\right)^{-s}. \tag{6.4}$$

Although the group G is noncommutative, but we have

$$\left.\left(P\left(g\right)\right)^{\ell}\left(Q\left(g\right)\right)^{m} = \left(Q\left(g\right)\right)^{m}\left(P\left(g\right)\right)^{\ell}, \quad \begin{array}{c} \forall\ g \in G, \quad \forall\ \ell, m \in \mathbb{N} \\ \forall\ P(y), Q(y) \in Z(G)[y]. \end{array}\right\} \tag{6.5}$$

In view of property (6.5) and associativity of elements of a group, expression (6.4) becomes

$$\begin{aligned} \tilde{D} &= \left(Q\left(g_1\right)\right)^{u}\left(\left(P\left(g_1\right)\right)^{-r}\left(\left(P\left(g_1\right)\right)^{r} c\left(P\left(g_2\right)\right)^{s}\right)\left(P\left(g_2\right)\right)^{-s}\right)\left(Q\left(g_2\right)\right)^{v}, \\ &= \left(Q\left(g_1\right)\right)^{u} c\left(Q\left(g_2\right)\right)^{v} \\ &= B. \end{aligned}$$

Also

$$\begin{aligned} \tilde{D}_1 C &= \rho\left(K_A^{-1}\right)\phi(K_A)M, \\ &= \left(\tilde{D}^{-1} K_A^{-1}\tilde{D}\right)(B^{-1}K_A B)M \\ &= M. \end{aligned}$$

□

6.2 Underlying Work Structure and Parameters

We note that, generally our proposed cryptosystem can be used with any noncommutative group G. But we suggest the use the general linear group of matrices of order n over a Galois field of characteristic 2 because it reduces the key size. The computations become efficient due to use of the known algorithms of fast calculations in a field of characteristic 2. Particularly, the group of matrices over

$GF(2^{127})$ is recommended as underlying work structure. The Galois field $GF(2^{127})$ is the factor algebra $\mathbb{Z}_2[x]/\langle q(x)\rangle$, where $\langle q(x)\rangle$ is the ideal generated by the irreducible polynomial $q(x) = x^{127} + x^{63} + 1$. So the elements of $GF(2^{127})$ are the polynomials of degree at most 126 with coefficients in $\mathbb{Z}_2$. Each entry of a matrix over the Galois field $GF(2^{127})$ is an element of $GF(2^{127})$ which is a string of 127 bits. The complexity of a matrix of order n over $GF(2^{127})$ is $127{\cdot}n^2$ bits.

The set of polynomials whose coefficients come from $Z(GL(2, GF(2^{127})))$ is the space for brute force attack. The brute force attack can be made infeasible because the number of polynomials with degree α, is $(2^{127} - 1)(2^{127})^{\alpha}$.

Example 6.2.1. To comprehend the proposed cryptosystem, we use $GL\left(2, GF(2^3)\right)$ as the noncommutative group in the following example. The irreducible polynomial $q(x) = x^3 + x^2 + 1$ is used to do computations in $GL\left(2, GF(2^3)\right)$. The inverses of the elements of $GF(2^3)$, are computed by using Algorithm 2.4.1. We use ApCoCoA [3] for calculations.

Initial Setup : The center of $GL\left(2, GF(2^3)\right)$ is the set

$$\begin{aligned} Z(GL\left(2, GF(2^3)\right)) &= \left\{gI \mid g \in GF(2^3)\right\}, \qquad \text{where} \\ I &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

For developing a communication, Bob and Alice will do the following steps:

Key Generation (KG) : Alice chooses a random polynomial as

$$P(y) = \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} + \begin{bmatrix} x+1 & 0 \\ 0 & x+1 \end{bmatrix} y^2 + \begin{bmatrix} x^2 & 0 \\ 0 & x^2 \end{bmatrix} y^4 \in Z(GL\left(2, GF(2^3)\right))[y]$$

as her private key. She also chooses elements

$$\left.\begin{array}{r} N_1 = \begin{bmatrix} x & x^2+1 \\ x+1 & x^2 \end{bmatrix}, \\ N_2 = \begin{bmatrix} x^2+1 & x \\ x+1 & 0 \end{bmatrix}, \end{array}\right\} \in GL\left(2, GF(2^3)\right)$$

$$N = \begin{bmatrix} 1 & x+1 \\ 0 & 1 \end{bmatrix} \in GL\left(2, GF(2^3)\right) \setminus Z(GL\left(2, GF(2^3)\right))$$

and $(r,s) = (2,8)$. So

$$\begin{aligned} P(N_1) &= \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} + \begin{bmatrix} x+1 & 0 \\ 0 & x+1 \end{bmatrix} \begin{bmatrix} x & x^2+1 \\ x+1 & x^2 \end{bmatrix}^2 + \begin{bmatrix} x^2 & 0 \\ 0 & x^2 \end{bmatrix} \times \\ & \quad \begin{bmatrix} x & x^2+1 \\ x+1 & x^2 \end{bmatrix}^4 \mod (x^3+x^2+1) \\ &= \begin{bmatrix} x^2+x & x^2+1 \\ x+1 & 0 \end{bmatrix} \mod (x^3+x^2+1) \end{aligned}$$

and

$$\begin{aligned} P(N_2) &= \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} + \begin{bmatrix} x+1 & 0 \\ 0 & x+1 \end{bmatrix} \begin{bmatrix} x^2+1 & x \\ x+1 & 0 \end{bmatrix}^2 + \begin{bmatrix} x^2 & 0 \\ 0 & x^2 \end{bmatrix} \times \\ & \quad \begin{bmatrix} x^2+1 & x \\ x+1 & 0 \end{bmatrix}^4 \mod (x^3+x^2+1) \\ &= \begin{bmatrix} x+1 & x^2+x+1 \\ x & x^2+1 \end{bmatrix} \mod (x^3+x^2+1). \end{aligned}$$

Now she computes

$$\begin{aligned} K_A &= \left(P\left(N_1\right)\right)^2 N \left(P\left(N_2\right)\right)^8 \\ &= \begin{bmatrix} x^2+x & x^2+1 \\ x+1 & 0 \end{bmatrix}^2 \begin{bmatrix} 1 & x+1 \\ 0 & 1 \end{bmatrix} \times \end{aligned}$$

$$\begin{aligned} & \begin{bmatrix} x+1 & x^2+x+1 \\ x & x^2+1 \end{bmatrix}^8 \quad \text{mod } (x^3+x^2+1) \\ = & \begin{bmatrix} x^2+x & x+1 \\ 0 & x^2+1 \end{bmatrix} \quad \text{mod } (x^3+x^2+1) \end{aligned}$$

and announces her public key (N_1, N_2, N, K_A).

Encryption (E_{P_A}) : To send a message (plaintext)

$$M = \begin{bmatrix} x^2+x & x+1 \\ 0 & x^2+1 \end{bmatrix} \in GL\left(2, GF(2^3)\right)$$

to Alice, Bob chooses random polynomial

$$Q(y) = \begin{bmatrix} x^2+1 & 0 \\ 0 & x^2+1 \end{bmatrix} y + \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} y^3 \in Z(GL\left(2, GF(2^3)\right))[y]$$

and $(u, v) = (3, 4)$. Then

$$\begin{aligned} Q(N_1) \quad = & \begin{bmatrix} x^2+1 & 0 \\ 0 & x^2+1 \end{bmatrix} \begin{bmatrix} x & x^2+1 \\ x+1 & x^2 \end{bmatrix} + \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} \times \\ & \begin{bmatrix} x & x^2+1 \\ x+1 & x^2 \end{bmatrix}^3 \quad \text{mod } (x^3+x^2+1) \\ = & \begin{bmatrix} x^2+x+1 & 0 \\ 0 & x^2+x+1 \end{bmatrix} \quad \text{mod } (x^3+x^2+1) \end{aligned}$$

and

$$\begin{aligned} Q(N_2) \quad = & \begin{bmatrix} x^2+1 & 0 \\ 0 & x^2+1 \end{bmatrix} \begin{bmatrix} x^2+1 & x \\ x+1 & 0 \end{bmatrix} + \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} \times \\ & \begin{bmatrix} x^2+1 & x \\ x+1 & 0 \end{bmatrix}^3 \quad \text{mod } (x^3+x^2+1) \end{aligned}$$

$$= \begin{bmatrix} x+1 & x^2+x+1 \\ x & x^2+1 \end{bmatrix} \mod (x^3+x^2+1).$$

Then he computes

$$\begin{aligned} B &= \left(Q\left(N_1\right)\right)^3 N \left(Q\left(N_2\right)\right)^4 \\ &= \begin{bmatrix} x^2+x+1 & 0 \\ 0 & x^2+x+1 \end{bmatrix}^3 \begin{bmatrix} 1 & x+1 \\ 0 & 1 \end{bmatrix} \times \\ & \quad \begin{bmatrix} x+1 & x^2+x+1 \\ x & x^2+1 \end{bmatrix}^4 \mod (x^3+x^2+1) \\ &= \begin{bmatrix} x^2+1 & x^2 \\ x^2+1 & x^2+1 \end{bmatrix} \mod (x^3+x^2+1). \end{aligned}$$

Also

$$|B| = (x^2+1) \mod (x^3+x^2+1).$$

Using Algorithm 2.4.1

$$|B|^{-1} = (x^2+x+1) \mod (x^3+x^2+1).$$

The matrix B^{-1} can be computed as follows:

$$\begin{aligned} B^{-1} &= \frac{1}{|B|} Adj(B) \\ &= |B|^{-1} Adj(B) \\ &= (x^2+x+1) \begin{bmatrix} x^2+1 & x^2 \\ x^2+1 & x^2+1 \end{bmatrix} \mod (x^3+x^2+1) \\ B^{-1} &= \begin{bmatrix} 1 & x^2+x \\ 1 & 1 \end{bmatrix} \mod (x^3+x^2+1). \end{aligned}$$

Now he uses the inner automorphism

$$\phi : g \longmapsto B^{-1}gB, \qquad \text{for } g \in GL\left(2, GF(2^3)\right)$$

to compute

$$\begin{aligned}
C_1 &= \phi(K_A) \\
&= B^{-1}K_AB \\
&= \begin{bmatrix} 1 & x^2+x \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x^2+x & x+1 \\ 0 & x^2+1 \end{bmatrix} \times \\
&\quad \begin{bmatrix} x^2+1 & x^2 \\ x^2+1 & x^2+1 \end{bmatrix} \quad \text{mod } (x^3+x^2+1) \\
&= \begin{bmatrix} x^2+1 & x+1 \\ 0 & x^2+x \end{bmatrix} \quad \text{mod } (x^3+x^2+1).
\end{aligned}$$

Encryption is

$$\begin{aligned}
C &= C_1M \\
&= \begin{bmatrix} x^2+1 & x+1 \\ 0 & x^2+x \end{bmatrix} \begin{bmatrix} x^2+x & x+1 \\ 0 & x^2+1 \end{bmatrix} \quad \text{mod } (x^3+x^2+1) \\
&= \begin{bmatrix} x^2+x & x^2+x \\ x^2+x & x \end{bmatrix} \quad \text{mod } (x^3+x^2+1).
\end{aligned}$$

The ciphertext transmitted to Alice is $C_A = (C', C)$, where

$$\begin{aligned}
C' &= \left(Q\left(N_1\right)\right)^3 K_A \left(Q\left(N_2\right)\right)^4 \\
&= \begin{bmatrix} x^2+x+1 & 0 \\ 0 & x^2+x+1 \end{bmatrix}^3 \begin{bmatrix} x^2+x & x+1 \\ 0 & x^2+1 \end{bmatrix} \times \\
&\quad \begin{bmatrix} x+1 & x^2+x+1 \\ x & x^2+1 \end{bmatrix}^4 \quad \text{mod } (x^3+x^2+1) \\
&= \begin{bmatrix} x^2+x+1 & 1 \\ x^2+x & x^2+x \end{bmatrix} \quad \text{mod } (x^3+x^2+1).
\end{aligned}$$

Decryption (D_{S_A}) : When Alice receives the ciphertext, she computes

$$\begin{aligned}
\tilde{D} &= (P(N_1))^{-2}\, C'(P(N_2))^{-8} \\
&= \begin{bmatrix} x^2+x & x^2+1 \\ x+1 & 0 \end{bmatrix}^{-2} \begin{bmatrix} x^2+x+1 & 1 \\ x^2+x & x^2+x \end{bmatrix} \times \\
&\quad \begin{bmatrix} x+1 & x^2+x+1 \\ x & x^2+1 \end{bmatrix}^{-8} \quad \bmod (x^3+x^2+1) \\
&= \begin{bmatrix} x^2+1 & x^2 \\ x^2+1 & x^2+1 \end{bmatrix} \quad \bmod (x^3+x^2+1) \text{ and} \\
\tilde{D}^{-1} &= \begin{bmatrix} 1 & x^2+x \\ 1 & 1 \end{bmatrix} \quad \bmod (x^3+x^2+1).
\end{aligned}$$

Also, she computes the inverse of matrix K_A as follows:

$$|K_A| = (x^2) \quad \bmod (x^3+x^2+1).$$

The Algorithm 2.4.1 yields

$$|K_A|^{-1} = (x+1) \quad \bmod (x^3+x^2+1).$$

So,

$$\begin{aligned}
K_A^{-1} &= \frac{1}{|K_A|} Adj(K_A) \\
&= |K_A|^{-1} Adj(K_A) \\
&= (x+1) \begin{bmatrix} x^2+1 & x+1 \\ 0 & x^2+x \end{bmatrix} \quad \bmod (x^3+x^2+1) \\
K_A^{-1} &= \begin{bmatrix} x & x^2+1 \\ 0 & x^2+x+1 \end{bmatrix} \quad \bmod (x^3+x^2+1).
\end{aligned}$$

She defines the inner automorphism

$$\rho : g \longmapsto \tilde{D}^{-1} g \tilde{D}, \qquad \text{for } g \in GL\left(2, GF(2^3)\right).$$

She computes

$$\begin{aligned}
\tilde{D_1} &= \rho\left(K_A^{-1}\right) \\
&= \tilde{D}^{-1} K_A^{-1} \tilde{D} \\
&= \begin{bmatrix} 1 & x^2+x \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x & x^+1 \\ 0 & x^2+x+1 \end{bmatrix} \times \\
&\quad \begin{bmatrix} x^2+1 & x^2 \\ x^2+1 & x^2+1 \end{bmatrix} \quad \text{mod } (x^3+x^2+1) \\
&= \begin{bmatrix} x^2+x+1 & x^2+1 \\ 0 & x \end{bmatrix} \quad \text{mod } (x^3+x^2+1).
\end{aligned}$$

Then

$$\begin{aligned}
M &= \tilde{D_1} C \\
&= \begin{bmatrix} x^2+x+1 & x^2+1 \\ 0 & x \end{bmatrix} \begin{bmatrix} x^2+x & x^2+x \\ x^2+x & x \end{bmatrix} \\
&= \begin{bmatrix} 1 & x \\ 1 & x^2 \end{bmatrix} \quad \text{mod } (x^3+x^2+1).
\end{aligned}$$

Notice that although the elements N_1 and N_2 are public knowledge, but for an adversary the polynomials $P(y)$ and $Q(y) \in Z(GL\left(2, GF(2^3)\right))[y]$ are unknown. Hence, the elements $\left(P\left(N_1\right)\right)^r, \left(P\left(N_2\right)\right)^s, \left(Q\left(N_1\right)\right)^u$ and $\left(Q\left(N_2\right)\right)^v$ are also unknown. He only knows the following expressions

$$\begin{aligned}
K_A &= \left(P\left(N_1\right)\right)^r N \left(P\left(N_2\right)\right)^s = \begin{bmatrix} x^2+x & x+1 \\ 0 & x^2+1 \end{bmatrix} \quad \text{mod } (x^3+x^2+1), \\
C' &= \left(Q\left(N_1\right)\right)^u K_A \left(Q\left(N_2\right)\right)^v = \begin{bmatrix} x^2+x+1 & 1 \\ x^2+x & x^2+x \end{bmatrix} \quad \text{mod } (x^3+x^2+1).
\end{aligned}$$

That is equivalent to find the solution of generalized decomposition problem because the integers r, s and u, v are unknown to the adversary. Also $P(y)$ and $Q(y)$ have to be determined, first.

6.3 Security of Proposed Cryptosystem

We now address some computational hardness related to the security and performance considerations of the proposed scheme.

The generalized decomposition problem is involved in the security of the proposed cryptosystem. In a noncommutative group, no polynomial time algorithm is known to obtain the solution of such problem. In our case the decomposition problem gives the following equations:

$$U_A U_B = U_B U_A, \tag{6.6}$$

$$V_A V_B = V_B V_A, \tag{6.7}$$

$$U_A W V_A = K_A, \tag{6.8}$$

$$U_B K_A V_B = C'. \tag{6.9}$$

The elements g_1,$g_2 \in G$ and $c \in G \setminus Z(G)$ are known to an adversary. For breaking the cryptosystem, the adversary has to find the elements U_A, U_B, V_A and V_B, for which, the adversary has to find out two polynomials $H_1(y), H_2(y) \in Z(G)[y]$ and numbers $\ell_1, \ell_2, m_1, m_2 \in \mathbb{N}$ such that

$$\begin{aligned} (H_1(g_1))^{\ell_1} &= U_A, \\ (H_1(g_2))^{\ell_2} &= V_A, \\ (H_2(g_1))^{m_1} &= U_B \qquad \text{and} \\ (H_2(g_2))^{m_2} &= V_B. \end{aligned}$$

Then, the conditions (6.6) and (6.7) may be guaranteed. Actually, the size of the space of polynomials over $Z(G)$ is the set of all possible random choices and an adversary has to verify the Conditions (6.8) and (6.9). By taking the space of polynomials over $Z(G)$ to be large enough, the possibility of brute force attack would be infeasible.

The only possibility to get solution of generalized decomposition problem is to search out the space of polynomials $Z(G)[y]$. This search becomes infeasible when the order of the space of the polynomials would be taken large enough. For the infeasibilty of brute force, it is suggested to make the choice of a prime p of order of 60 decimal digits and polynomials of degree 20.

These matrix equations produce a large set of equations involving unknown matrices U_A, U_B, V_A, and V_B. However an adversary has to face a problem with these types of equations. That is, no matter how he rearranges these equations, the problem of having a product of two unknown matrices can not be avoided which leads to a large system of nonlinear equations in their entries.

Chapter 7

Conclusion and Future Work

Public key cryptography is an essential need in every field of today's life. Current cryptosystems depend on problems which are hard to break with today's knowledge and computing ability. Because of the continuous growing computational ability, the key length should be enlarged permanently, for achieving a high level of security. Therefore, researchers always looking for different approaches in algebraic structures. In this regard noncommutative groups are considered to be ideal.

Keeping in mind all these things, we have presented a new public key cryptosystem in spirit of Stickel's key exchange protocol, in chapter 4. This new cryptosystem can be implemented by employing CSP and DLP in any noncommutative setting. The correctness of the algorithm is proved. The proposed algorithm is simple and efficient to implement with suggested platform and parameter values. A toy example is also given to explain the algorithms. Finally, some security and efficiency aspects are also discussed.

As mentioned in Section 4.3 that the proposed cryptosystem is secure against the attack mentioned in [42]. However, as a future work, the possibility of an extension of such kind of attack can be explored.

In chapter 5, we have shown that the noncommutative rings can be used to develop public key exchange protocols. These protocols are the variants of Stickel's key exchange protocol. Specifically, we proposed two protocols based on the ring

$M(n, \mathbb{Z}_p)$ and use polynomials over the center of the ring. These polynomials are kept secret as the private keys by each user. It was discussed that for obtaining shared secret key, an adversary, first has to find the polynomials. After that, solution of the generalized decomposition problem would be acquire. For solving such problem in a noncommutative ring, no polynomial time algorithm is known.

The basic idea used in the presented protocol can be used to develop a signature verification scheme. The possibility of exploring this idea for signature verification, is open for future work.

We have presented new cryptosystem based on inner automorphism of a group, in chapter 6. It is based on the variant of Stickel's key exchange protocol for polynomials over noncommutative groups. The security and suggestions related to different parameters are discussed in detail. The combined use of polynomials and automorphisms of a noncommutative platform enables this cryptosystem a convenient and sensible choice for future security.

The development of another cryptosystem based on automorphisms with different encryption and decryption schemes, is also under consideration.

For the three proposals presented in this study, more platforms can be explored for achieving better security and efficiency. Further, the implementation of all the new techniques proposed in this thesis is open for future study.

Bibliography

[1] R. Álvarez, F. Martı´nez, J. Vicent and A. Zamora, "Cryptographic applications of 3 ×3 block upper triangular matrices", in: Proceedings of Hybrid Artificial Intelligent Systems – 7th International Conference, HAIS 2012, Part II, LNCS, vol. 7249, Springer, 97–104, 2012.

[2] I. Anshel, M. Anshel and D. Goldfeld, "An algebraic method for public key cryptography", Mathematical Research Letters 6, 287–291, 1999.

[3] ApCoCoA team. ApCoCoA, "Applied Computations in Commutative Algebra". Available at http://www.apcocoa.org.

[4] Aryan, C. Kumar and D. R. Vincent, "Enhanced Diffie-Hellman Algorithm for Reliable Key Exchange", IOP Conf. Series: Materials Science and Engineering 263, 2017, doi:10.1088/1757-899X/263/4/042015.

[5] E. Barker and A. Roginsky, "Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths", 2015.

[6] G. Baumslag, T. Camps, B. Fine, G. Rosenberger, X. Xu, "Designing key transport protocols using combinatorial group theory", Algebraic methods in cryptography:AMS/DMV Joint International Meeting, 16-19, 2005.

[7] D. M. Burton, "Elementary Number Theory", McGraw-Hill, 6 edition, 2007.

[8] Z. Cao, "Conic analog of RSA cryptosystem and some improved RSA cryptosystems", Journal of Natrual Science of Heilongjiang University, vol. 16, 1999.

[9] Z. Cao, “The multi-dimension RSA and its low exponent security”, Science in China (E Series), vol. 43, 349-354, 2000.

[10] Z. Cao, X. Dong and L. Wang, “New Public Key Cryptosystems using polynomials over Non-commutative rings”, Cryptology e-print Archive, 2007.

[11] M. Charalambos and C. Koupparis, “Non-commutative cryptography: Diffie-Hellman and CCA secure cryptosystems using matrices over group rings and digital signatures”, ProQuest LLC, Ann Arbor, Thesis (Ph.D.), City University of New York, 2012.

[12] D. Coppersmith, “The Data Encryption Standard (DES) and Its Strength Against Attacks”, IBM Journal of Research and Development, May 1994.

[13] J. Daemen and V. Rijmen, “AES Proposal: Rijndael, AES algorithm submission”, September 3, 1999.

[14] M. Dehn, Uber unendliche diskontinuerlich gruppen, Math. Ann. 69, 116–144, 1911.

[15] W. Diffie and M. Hellman, “New directions in cryptography”, IEEE Transactions on Information Theory, vol. 22, 644–654, 1976.

[16] M. Drmota, “Lecture Notes on Linear Algebra 1”, Technische Universitat Wien, 2005.

[17] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, vol. 31, 469-472, 1985.

[18] A. Fernandes, “Elliptic Curve Cryptography”, Dr. Dobb’s Journal, December 1999.

[19] J. B. Fraleigh, “A first course in abstract algebra”, Pearson Education India, 1971.

[20] S.W. Golomb, “Shift Register Sequence”, Holden-Day, San Francisco, 1967. Reprinted by Aegean Park Press, 1982.

[21] L. Gu, L. Wang, K. Ota, M. Dong, Z. Cao, Y. Yang, "New public key cryptosystems based on non-Abelian factorization problems", Security and Communication Networks, vol. 6, 912-922, 2013.

[22] M. Habeeb, D. Kahrobaei, C. Koupparis, V. Shpilrain, "Public key exchange using semidirect product of (semi)groups", in: ACNS 2013, Lecture Notes in Computer Science 7954, 475–486, 2013.

[23] D. Hofheinz, R. Steinwandt, "A Practical Attack on Some Braid Group Based Cryptographic Primitives", Public Key Cryptography PKC, Springer, 187-198, 2003.

[24] S. Inam and R. Ali, "A new ElGamal-like cryptosystem based on matrices over groupring", Neural Computing and Applications, vol. 29, 2018.

[25] D. Kahrobaei, C. Koupparis and V. Shpilrain, "Public key exchange using matrices over group rings", Groups Complex. Cryptol., vol. 5, 97-115, 2013.

[26] S. Kanwal and R. Ali, "A cryptosystem with noncommutative platform groups", Neural Computing and Applications, vol. 29, 1273-1278, 2018.

[27] S. Kanwal and R. Ali, "Two New Variants of Stickel's Key Exchange Protocol Based on Polynomials over Noncommutative Rings", Submitted for Possible Publication.

[28] J. Katz and Y. Lindell, "Introduction to Modern Cryptography", Chapman & Hall/CRC Press, 2008.

[29] A. Kerckhoffs, "La cryptographie militaire, Journal des Sciences Militaires ", 9th Series, 161-191, 1883.

[30] A. Kitaev, "Quantum measurements and the Abelian Stabilizer Problem", Preprint arXir: cs.CR/quant-ph/9511026, 1995.

[31] K.H. Ko, J. S. Lee, J. H. Cheon, J. H. Han, J. S. Kang and C. Park, "New public-key cryptosystems using Braid groups", Advances in Cryptography, Proceedings of Crypto 2000, Lecture Notes in Computer Science 1880, 166–183, 2000.

[32] K. Komaya, U. Maurer, T. Okamoto and S. Vanston, "Newpublic-key schemes bases on elliptic curves over the ring Z_n", In J. Feigenbaum (Ed.): Crypto'91, LNCS 576, Springer-Verlag, 252-266, 1992.

[33] X. Lai, J.L. Massey and S. Murphy, "Markov ciphers and differential cryptanalysis", Advances in CryptologyEUROCRYPT 91 (LNCS 547), 1738, 1991.

[34] E. Lee, "Braig groups in cryptography", IEICE Trans. Fundamentals, vol. E87-A, no. 5, 986-992, 2004.

[35] S. J. Lee and E. Lee, "Potential Weaknesses of the Commutator Key Agreement Protocol Based on Braid Groups", Knudsen, L. (Ed.), Advances in Cryptology EUROCRYPT, Springer, 14-28, 2002.

[36] R. Magyarik and N. R. Wagner, "A Public Key Cryptosystem Based on the Word Problem", Advances in Cryptology – CRYPTO 1984, Lecture Notes in Computer Science 196, 19–36. Springer, Berlin, 1985.

[37] A. Mahalanobis, "The discrete logarithm problem in the group of non-singular circulant matrices", Groups Complexity Cryptology 2, 83-89, 2010.

[38] K. McCurley, "A key distribution system equivalent to factoring", Journal of Cryptology 1, 95-100, 1988.

[39] A. J. Menezes, S. A. Vanstone and P. C. Van Oorschot, "Handbook of Applied Cryptography. Discrete Mathematics and Its Applications", CRC Press, Inc., 1996.

[40] A. Menezes and Y. Wu, "The discrete logarithm problem in GL(n, q)", Ars Combinatorica 47, 23–32, 1997.

[41] C. Mullan, "Some Results in Group-Based Cryptography", Thesis submitted to the University of London for the Degree of Doctor of Philosophy, (2012).

[42] A. D. Myasnikov, A. Ushakov, "Cryptanalysis of the Anshel-Anshel-Goldfeld-Lemieux key agreement protocol", Groups, Complexity, and Cryptology, 1, 63-75, 2009.

[43] A. G. Myasnikov, V. Shpilrain and A. Ushakov, “Group-Based Cryptography, Advanced Courses in Mathematics”, CRM Barcelona, 2007.

[44] J. Proos and C. Zalka, “Shor’s discrete logarithm quantum algorithm for elliptic curves”, Quantum Information and Computation, vol. 3, 317-344, 2003.

[45] M.O. Rabin, “Digitized signatures and public-key functions as intractible as factorization”, MIT Laboratory for Computer Science Technical Report, LCS/TR-212, 1979.

[46] R. Rivest, A. Shamir, L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, Communications of the ACM, vol. 21, 120–126, 1978.

[47] M. Robshaw, “Stream Ciphers. RSA Laboratories Technical Report TR-701”, 1995. http://www.rsasecurity.com/rsalabs.

[48] J. Rotman, “The Theory of Groups”, Boston, Allyn and Bacon, 1965.

[49] B. Schneier, “Description of a new variable-length key, 64-bit block cipher (Blowfish)”, R. Anderson, editor, Fast Software Encryption, Cambridge Security Workshop (LNCS 809), Springer-Verlag, 191-204, 1994.

[50] A. Shamir, “How to share a secret”, Communications of the ACM, vol. 22, 612–613, 1979.

[51] P. Shor, “Polynomail-time algorithms for prime factorization and discrete logarithms on a quantum computer”, SIAM Journal of Computing, vol. 5, 1484-1509, 1997.

[52] V. Shpilrain, “Cryptanalysis of Stickel’s key exchange scheme”, Proceedings of Computer Science in Russia 5010, 283–288, 2008 .

[53] P. Smith and M. Lennon, LUC, “A newpublic key system”, Proceedings of the IFIP TC11 Ninth International Conference on Information Security, IFIP/Sec 93, 103-117, North-Holland, 1993.

[54] M. Sramka, “On the security of Stickels key exchange scheme”, available at http://crises-deim.urv.cat/msramka/pubs/ sramka-stickelkesecurity.pdf.

[55] W. Stalling, “Cryptography and Network Security: Principles and Practices”, Sixth Edition, Prentice Hall, 2013.

[56] E. Stickel, “A New Method for Exchanging Secret Keys”, Proceedings of the Thirteenth International Conference on Information Technology and Applications, vol. 2, 426–430, 2005.

[57] D. R. Stinson, “Cryptography: Theory and Practice”, Third Edition, Chapman & Hall, Boca Raton, 2005.

[58] H.C. Williams, “A Modification of the RSA Public-Key Encryption Procedure”, IEEE Transactions on Information Theory, vol. 6, 726-729, 1980.

[59] H.C. Williams, “Some public-key crypto-funtions as intractible as factorization”, In G.R. Blakley and D.Chaum (Eds), CRYPTO’84, LNCS 196, Springer-Verlag, 66-70, 1985.

[60] B. Preneel, “The RC5 encryption algorithm ”, Fast Software Encryption, Second International Workshop (LNCS 1008), Springer-Verlag, 86-96 1995.