Review on Cyclic Group and Affine Cryptosystem and its Application on Cryptography

A thesis that was submitted to the
Department of Mathematics and Natural Sciences, BRAC University,
For partial fulfillment of the requirements of the award of the degree of
Bachelor of Science in mathematics.

By
Samiha Tasnim
ID: 13216003
Department of Mathematics and Natural Sciences
BRAC University
27th September, 2018

DECLARATION

I do hereby declare that the thesis titled "Review on Cyclic Group and Affine Cryptosystem and its Application on Cryptography" is submitted to the Department of Mathematics and Natural Sciences of BRAC University in partial fulfillment of the Bachelor of Science in Mathematics. This is my review work and has not been submitted elsewhere for the award of any other degree or any other publication. Every work that has been used as reference for this work has been cited properly.

Date: 27th September, 2018

Candidate:

_____

Name: Samiha Tasnim

ID: 13216003

Certified:

_____

Mohammad Maruf Ahmed

Supervisor

Assistant Professor

Department of Mathematics and Natural Sciences

BRAC University, Dhaka.

DEDICATION

The completion of this thesis would not have been possible without the support of my beloved parents and my honorable faculty. Therefore I dedicate this thesis to them.

ACKNOWLEDGEMENT

First of all, I am grateful to the Almighty Allah for establishing me to complete this thesis successfully.

Now, I would like to express deep gratitude to my supervisor **Mohammad Maruf Ahmed** for his guidance, encouragement and gracious support throughout the course of my work, for his experience in the field that motivated me to work in this area and for his faith in me at every stage of this research.

My sincere thanks goes to **Professor A F M Yusuf Haider,** the Chairperson of the Department of Mathematics and Natural Sciences for giving his valuable time to go through my thesis on the day of my defense.

Also, I would like to thank all other honorable faculty members of the Department of Mathematics and Natural Sciences.

Then, I would like to thank my friends who have helped me, supported me and encouraged me throughout these four years.

Lastly, I would like to thank my parents for believing in me and supporting me spiritually throughout my life.

ABSTRACT

In this thesis, I have tried to briefly describe the concept of cryptography and group theory and the relation between them. In short, Cryptography is regarded as a medium which enables communications to take place under secure parameters. The process of encryption and decryption which uses algorithm and key to convert plain texts into encrypted ones and vice versa makes such secure communication possible even with the presence of malicious third parties. A major portion of the study of cryptography deals with Group Theory. Group theory, perhaps the primary algebraic structure to be studied abstractly, is one of the most fundamental structures. A group is a finite or infinite set of elements together with a binary operation that satisfies the four basic properties of closure, associability, the identity property and the inverse property. In this thesis, I have put the Diffie Hellman's protocol to demonstrate an application in which an outside client can privately communicate with members of a particular company without running the risk of important facts getting leaked elsewhere. Affine cryptosystem is used to encrypt and decrypt messages which uses the $\mathbb{Z}_{26}$ is also included in this thesis by which the client can send encrypted messages to any specific member he wants, and the receiver can also decrypt the message.

Table of Contents:

Chapter 1

Introduction

The need for cryptography emerged as a mean to protect information, which are required to remain hidden or disclosed, from falling within the reach of unintended users. Ancient forms of cryptography involved using pictures, symbols or distorted figures to represent proper alphabets in order to preserve the initial message. As time escalated and mathematical science developed, so did the field of cryptography and its implications.

The principle of cryptography puts much emphasis on the secret key which serves as the only tool for encrypting a plain text into an unreadable form and also to convert it back during decryption. Cryptography has three types: 1. Symmetric-Key Cryptography, 2. Public-Key Cryptography and 3. Hash Functions. All these different techniques approaches to make communication secure by applying the function of the key differently.

Like every other mathematical approach, cryptography comes with both advantages and drawbacks. Apart from making communication safe, cryptography also helps to preserve the authenticity of the message against forgeries. On the other hand, such complex measure of prevention and safety can be deemed to be troublesome in times that demand immediate response.

As mentioned previously, Group Theory has significant contribution to Cryptography. This thesis aims to explain the linking bridge that exists between the theoretical aspects of groups, its types, theorems and proofs with cryptography.

Most of the cryptographic schemes are based on group theory, which use finite groups. The Diffie Hellman key exchange which is the first public-key protocols has been applied in this thesis to show how two companies or parties can communicate privately by sharing keys. Theorems and algorithms were used and indicated with example to show how a sender from outside a company can securely communicate with an employee of a particular company by

sharing keys. After sharing keys the client and the employee will be able to communicate securely by using Affine cryptosystem, which encrypts and decrypts messages.

Literature review

Throughout history many secret organizations and firms have worked to keep their communication secret by encrypting it. Encryption is a technique which changes the information into an unreadable form. Thus, encryption has become an integral part of the computing world paving way for cryptography. Cryptography has an interesting history which has undergone many changes down through the centuries. Cryptography is the design and analysis of mathematical techniques that ensure secure communications in the presence of malicious adversaries. For many years, the concept of cryptography is used to ensure the safe transfer of messages for military purposes. This thesis has covered the theoretical aspects of groups and its types, some theorems and proofs. In addition, this thesis has covered the Diffie Hellman key exchange which was one of the first public-key protocols, originally conceptualized by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. In 1975 Diffie, Hellman and Merkle introduced public key cryptography. The basic idea is to use for encryption a so-called one way function, a function such that it is easy to compute $f(x)$ but difficult, in general, to compute $f^{-1}(y)$. In 1976 Diffie and Hellman presented the Discrete Logarithm Key agreement Protocol [5]. This protocol uses a finite cyclic group with generator g (the original implementation used the multiplicative group of integers modulo p as the basis group and a primitive root for$g$). A standard model for a cryptographic scheme is phrased as two parties, **A** and **B**, who wish to communicate securely over an insecure channel (such as a wireless link, or a conventional phone line). If **A** and **B** possess information in common that only they know (a shared secret key) they can use this, together with a symmetric key cipher such as AES (the Advanced Encryption Standard), to communicate. If **A** and **B** do not possess a secret key, they execute a protocol such as the Diffie–Hellman key agreement protocol to create one. 'Cryptosystem for Group Oriented Cryptography' by Tzonelih Hwang from National Cheng Kung University Institute of Information Engineering Tainan, Taiwan, R.O.C. Hwang proposed

a non-interactive scheme to simultaneously solve several open problems in group oriented cryptography, where the sender of the information is allowed to determine the destination of the information. The scheme which he proposed was based on the Diffie Hellman key distribution scheme and Shamir's secret sharing scheme to solve the open problems simultaneously.

Chapter 2

Basics of Cryptography

The field of Cryptography was discovered many thousands years ago with the motive of preventing valuable information and details from falling into the grasp of unintended audiences. Cryptography involves preserving and communicating information, by encrypting it into a different format, to the intended user(s). However, the main purpose of cryptography is not security only, it also works to conserve the authenticity of the data in question. Throughout the course of its existence, cryptography has evolved from being solely related to encryption to modern day mathematical theory and computer science practice. Modern cryptography uses sophisticated mathematical equations (algorithms) and secret keys to encrypt and decrypt data. [2] In addition, Public-Key Cryptography has drawn vast focus since its inception in 1976. Public-Key Cryptography, which is also known as Asymmetric-Key Cryptography, uses a lot of group theories. Different cryptosystems use different groups, such as the group of units in modular arithmetic and the group of rational points on elliptic curves over a finite field. Such use of group derivatives rose because of efficiency or for the difficulty of carrying out certain computations in groups, rather than from a "symmetrical" perspective. Other algebraic structures, such as lattices, are used to underline other Public-Key Cryptosystems. [1]

Principles of Cryptography

According to modern cryptographers, security should not only rely on the secrecy of the encryption method. The secrecy of the keys should also be a commanding factor. During comparison of cipher and plain texts, the secret keys should not be exposed and should remain beyond the knowledge of any individual. Modern algorithms are focused on mathematically difficult problems such as prime number factorization, discrete logarithms etc. There is no mathematical proof that these problems are in fact are hard, just empirical evidence.

It is beyond the capability of humans to master the art of executing modern cryptographic algorithms. Therefore, such algorithms can only be executed by computers or specialized hardware devices and can be mostly implemented in computer software.

The prime objective of the structure of the secure system using encryption techniques is to protect the secret keys. This can be done either by encrypting them under other keys or by protecting them physically. A successful method of encryption (a cipher) can be patented as an intellectual property and can be used by cryptographers to extract royalties from them when they are being used commercially. [9]

Types of Cryptography

Generally, three different types of cryptographic techniques can be used-

1. Symmetric-Key Cryptography

2. Public-Key Cryptography

3. Hash Functions.

Symmetric-Key Cryptography: A single key will be shared by the sender and the receiver. Using this key, the plain text will be encrypted by the sender and the cipher message will be delivered to the receiver. On receiving the text, the receiver will be using the same key to decode the message to retrieve the plain text.

Public-Key Cryptography: In this format of cryptography, two related keys, public and private, are used. Within this set of keys, the public one can be freely exposed however, the private one has to remain secret. The public key is used for encryption and the private key is used for decryption.

Hash Functions: This algorithm does not include any key. According to the plain text, a fixed-length hash is computed which makes it impossible to recover the contents of the plain text. Passwords of many operating systems are encrypted using hash functions**.**

Benefits and Drawbacks of Cryptography

The benefits of Cryptography:

Cryptography is an essential information security tool. It provides the four most basic services of information security –

1. Confidentiality: Unauthorized revelations and access of data can be prevented by using encryption techniques.
2. Authentication: The cryptographic techniques such as MAC and digital signatures can protect information against spoofing and forgeries.
3. Data Integrity: The cryptographic hash functions are playing vital role in assuring the users about the data integrity.
4. Non-repudiation**:** Non-repudiation services are provided by the digital signature to protect against disruptions that may result because of denial of passing message by the sender.

The drawbacks of Cryptography:

1. During times that require immediate response, it may prove to be rather difficult for an able user to access and use a strongly encrypted, authentic and digitally signed information. The network or the computer system can be attacked and rendered non-functional by an intruder.
2. Certain threats may rise due to poor design of systems, protocols and procedures. Unfortunately, cryptography fails to prevent such vulnerabilities. Such threats need to be addressed by using proper design and setting up of a defensive infrastructure.
3. Cryptography is both expensive and time consuming.

- Addition of cryptographic techniques in the information processing leads to delay.
- The use of public key cryptography requires setting up and maintenance of public key infrastructure requiring the handsome financial budget.[7]

Private Key cryptography

As mentioned earlier, one particular key is used for both encoding and decoding messages in a single or private key cryptosystem. In order to encrypt a plain text, some functions, such as f, will be applied to the message. This will result in the generation of an encrypted message. The original message can then be extracted from the encrypted form by applying the inverse transformation f^{-1}. The transformation f must be relatively easy to compute, as must f^{-1}; however, f must be extremely difficult to guess at if only examples of coded messages are available.[2]

Public Key Cryptography

If traditional cryptosystem is being used, an individual who is capable of encoding a message will also be capable of decoding an intercepted message. In the year 1976, public key cryptography, which depended on the observation that encryption and decryption did not require the same key, was proposed by W. Diffie and M. Hellman. This gave the hint that the encoding key no longer needed to be kept secret. Although computing the encoding function f was comparatively easier, calculating the function f^{-1} was much harder without additional information. Thus, someone who knew only the encoding key could not figure out the decoding key without prohibitive computation. It is interesting to note that to date, no system has been proposed that has been proven to be “one-way;" that is, for any existing public key cryptosystem, it has never been shown to be computationally prohibitive to decode messages with only knowledge of the encoding key. [2]

Some Definitions

1. Encryption: It's the technique by which plaintext or some other kind of information is changed over from an intelligible shape to an encoded rendition that must be decoded by another substance on the off chance that they approach a decoding key. Encryption is a standout amongst the most essential techniques for giving information security, particularly for end-to-end insurance of information transmitted crosswise over systems.[2]
2. Decryption: It is the way toward changing information that has been rendered garbled through encryption back to its decoded shape. In decoding, the framework concentrates and changes over the distorted information and changes it to writings and pictures that are effortlessly justifiable by the user as well as by the framework. Decoding might be acquired manually or automatically. It might likewise be performed with an arrangement of keys or passwords. [2]
3. Plaintext: Non-encoded content or message in its default form. Plain text is the contribution of an encryption procedure, and the yield of a decoding procedure. Likewise called clear text.[2]
4. Cipher-text: It's the encoded information achieved when any message is encrypted.
5. Cipher: The science (or calculation) in charge of transforming plaintext into cipher-text and returning cipher-text to plaintext.[2]

Chapter 3

Introduction to Group Theory

If we trace back the factors which instigated the development of abstract group theory in the mathematical literature of the nineteenth century, we will find the theory of algebraic equations, number theory and geometry. Group theoretic methods of reasoning are used by all the fields mentioned above. Groups are vital to modern algebra, their basic structure can be found in many mathematical phenomena. Groups can be found in geometry, representing phenomena such as symmetry and certain types of transformations. Group theory, perhaps the primary algebraic structure to be studied abstractly, is one of the most fundamental structures. A group can be defined as a set, both finite and infinite, of elements along with a binary operation which ensures the four basic properties of closure, associability, the identity property and the inverse property. The operation with respect to which a group is defined is often called the 'group operation' and a set is said to be a group 'under this operation'. In other words, a non-empty set G together with a binary operation $*$ is called a group if the following conditions are satisfied.

1. Closure law: $a * b \in G \; \forall \; a, b \in G$
2. Associative law: $(a * b) * c = a * (b * c) \; \forall \; a, b, c \in G$
3. Identity law: There exists an element $e \in G$ such that $a * e = e * a \; \forall \; a \in G$
4. Inverse law: For any $a \in G$, there exists a $b \in G$ such that $a * b = b * a = e$. Then a is the inverse of b and b is the inverse of a i.e. $b = a^{-1}$.

Out of the four functions, there is one property which is missed out that is commutative property. This is because in group theory commutative property is not assumed. Many of the most useful and interesting groups are not commutative. These groups in which the group operation is commutative is called Abelian group. [3]

Cyclic Group

A cyclic group G is a group that can be generated by a single element a, so that every element in G has the form a_i for some integer i. We denote the cyclic group of order n by $\mathbb{Z}_n$, since the additive group of $\mathbb{Z}_n$ is a cyclic group of order n.

Definition: Let G be a group under the binary operation multiplication. Then G is called the cyclic group if every element of G can be expressed as the power of a single element of G.

Or, let G be a group under the binary operation multiplication and$a \in G$. G is called cyclic if for any $x \in G$. That is, $x = a^n n \in \mathbb{Z}$.

The element a is called the generator of G. This group is generated by the element a. It is denoted by $G =$<a>. [2]

Example: $G = \{1, -1\}$

$$\Rightarrow (-1)^1 = -1$$

$$\Rightarrow (-1)^2 = 1$$

$$\Rightarrow (1)^1 = 1$$

$$\Rightarrow (1)^{-1} = 1$$

It is generated by -1.

$\therefore G = < -1 >$.

Application: Diffie-Hellman Key Exchange

In order to better understand the structure and proceedings of a standard model for a cryptographic scheme, let us consider the following scenario. The scheme includes two individual parties, **A** and **B** who intends to communicate securely but can do so only through an insecure channel. If both the involved parties possess a common information which is only confined within their knowledge, they can use this information together with a symmetric key cipher such as AES (Advanced Encryption Standard) to communicate. If **A** and **B** do not possess a secret key, they execute a protocol such as the Diffie- Hellman key agreement protocol to create one. The protocol enables unknown parties to construct together a shared secret key over an insecure communications channel. The key can then be used to encrypt relevant communications using a symmetrical key cipher. The scheme was first published publicly by Whitfield Diffie and Martin Hellman in 1976. Diffie-Hellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols. The Diffie-Hellman exchange by itself does not provide authentication of the communicating parties and is thus susceptible to a man-in-the-middle attack. It will be possible for an intruder to intercept the communication by establishing two different key exchanges with the involved parties by appearing as **A** to **B** and vice versa. This will enable the attacker to read and store and then re-encrypt the message. In order to prevent such occurrence, a method to validate the parties with each other is necessary. [4]

We give a brief description of this protocol as shown below:

Diffie–Hellman Key Agreement Protocol.

Let G be a cyclic group, and g a generator of G, where both g and its order d are publicly known. If **A** and **B** wish to create a shared key, they can proceed as follows:

1. **A** selects uniformly at random an integer $a \in [2, d-1]$, computes g^a, and sends it to **B**.

2. **B** selects uniformly at random an integer $b \in [2, d-1]$, computes g^b, and sends it to **A**.

3. **A** computes $k_a = (g^b)^a$, while **B** computes $k_b = (g^a)^b$

4. The shared key is thus $k = k_a = k_b \in G$. [5]

Diffie-Hellman's Features

The factors which makes the Diffie-Hellman algorithm attractive are:

There is no necessity to preserve secret keys for a long duration which may ultimately expose it to many threats. This algorithm enables such keys to be generated whenever required.

The interchange of messages can occur without the existence of any infrastructure other than an agreement on global parameters.

Unfortunately, the Diffie-Hellman's algorithm is not free from weaknesses. Such drawbacks are:

It does not provide the option to reveal the identities of the involved parties. Therefore, it gives opportunities to impersonators to intervene.

It is highly dependent on computations. As a result, clogging attacks may occur in which an opponent requests a high number of keys. Significant resources related to computing will be needed to perform unnecessary modular exponentiation.

Replay attacks cannot be avoided.

It is subject to a man-in-the-middle attack, in which a third party **C** impersonates while communicating with **A** and impersonates **A** while communicating with **B**. Both **A** and **B** end up negotiating a key with **C**, which can then listen to and pass on traffic.

Example:

Together **A** and **B** choose a large prime number p and a number g such that $1 < g < p$. (Usually g is chosen to be quite small, for ease of computation) These numbers do not need to be secret, so they can be communicated freely over a public channel. **A** secretly chooses an integer x, and **B** secretly chooses an integer y.

Now, **A** sends **B**: $g^x \ (mod\ p)$ and **B** sends **A**: $g^y (mod\ p)$.

Using her secret x, **A** computes $z \equiv (g^y)^x \equiv g^{yx} (mod\ p)$. Using his secret y, **B** computes $z \equiv (g^x)^y \equiv g^{xy} (mod\ p)$. Now **A** and **B** have a secret key z known only to them, which can be used to send messages via any number of secure private-key cryptosystems.

Let $p = 36721,\ g = 1797$. Suppose **A** picks $x = 79$ and **B** picks $y = 67$. Then **A** computes $1797^{79} \equiv 23881 (mod\ 36721)$ and **B** computes $1797^{67} \equiv 21209 (mod\ 36721)$. They send the result of these computations to each other. Upon receiving 23881 and 21209; **A** computes $21209^{79} \equiv 13308 (mod\ 36721)$, and **B** computes $23881^{67} \equiv 13308 (mod\ 36721)$ respectively. So they have a key that they agreed on without disclosing their secret numbers to each other. They can use this key to communicate securely via a cryptosystem of their choosing.

An eavesdropper (often called **E**) will see the results of any transmissions over public channels, so she will know $p, g, g^x (mod\ p),\ and\ g^y (mod\ p)$. Her goal is to compute $g^{yx}\ (mod\ p)$. If she can compute y and x then she has the same information as **A** and **B** and hence can read their messages. Computing the exponent x is like taking a "mod-p base-g logarithm of g^x, but this is expected to be a hard problem in general. It is known as the discrete log problem. There is no known algorithm for efficiently computing discrete logs $(mod\ p)$ in general.

It is also widely believed that there is no faster way for **E** to compute $g^{yx} (mod\ p)$ than by finding y and x so that the security of the Diffie-Hellman protocol is as strong as the difficulty of the discrete log problem.[8]

Algorithm:

1. **A** and **B** agrees to use a modulus p and a generator g.
2. **A** chooses a secret key . **B** chooses a secret key y.
3. **A** computes, $n = g^x mod\ p$. **B** computes $m = g^y mod\ p$.
4. **A** sends n to **B** and **B** sends m to **A**.
5. **A** then computes $z = n^x\ mod\ p$. **B** then computes $z = m^y\ mod\ p$.
6. **A** and **B** now share a secret key.

Code: The Python code for the DFH protocol is shown below:

```
class DHM:

    def __init__(self):
        """
        Initiate with any large prime and its primitive root
        """
        self.prime = 36721 #any prime
        self.root = 1797 #primitive root of prime

    def set_private_key(self):
        """
        Method that accepts and returns a private key.
        For user convenience only, class does not save private key.
        """
        try:
            k_pr = int(input('\nEnter your private key: '))
            return k_pr
        except ValueError:
            print('Enter integer')

```

```

    def get_public_key(self, k_pr):
        """
        Method that accepts a private key, and uses it to create a public key.
        """
        k_pb = (self.root**k_pr) % self.prime #(root^private_key) mod prime
        return k_pb

    def get_shared_key(self, k_pr, user_k_pb):
        """
        Accepts an user's private key and the public key of the other user to create a shared key.
        """
        key = (user_k_pb**k_pr) % self.prime #(B's public key^A's private key) mod prime
        return key

```

```
34
35 if __name__ == '__main__':
36     d = DHM()
37     #for user A
38     print "\nfor user A"
39     a_pr = d.set_private_key() #create a private key for user A
40     a_pb = d.get_public_key(a_pr) # calculate A's public key, to be shared with B
41
42     #for user B
43     print "\nfor user B"
44     b_pr = d.set_private_key() #create a private key for user B
45     b_pb = d.get_public_key(b_pr) #calculate B's public key, to be shared with A
46
47     a_k_sh = d.get_shared_key(a_pr, b_pb) #User A calls this, using his private key and B's
48                                           #public key
49
50     b_k_sh = d.get_shared_key(b_pr, a_pb) #User B uses his private key and A's public key to
51                                           #calculate shared key
52
53     #To show they are equal
54     print('\nuser A key: ' + str(a_k_sh) + '\n\nuser B key: ' + str(b_k_sh))
55
```

Output: The result is as follows,

```
In [1]: %run "C:/Users/sony vaio/Downloads/dhm_final.py"

for user A

Enter your private key: 79

for user B

Enter your private key: 67

user A key: 13308

user B key: 13308
```

Application of cryptography using Diffie Hellman and Affine Cryptosystem

Let us consider the following case as an example.

Company **A** operates to complete special and secret projects. A particular client wants to communicate with an employee of this company in order to appoint them for working on such a secret assignment. However, he intends to keep this communication as undisclosed as possible. Let us assume that the company has "a" number of employees with each employee possessing a secret key. Each member of the company is denoted by the symbol a_i. The client will now compute a secret random key, for example b by using Diffie- Hellman protocol:

$$Y=g^b(mod\ p)$$

After computing the value of Y, the client will want to convey it to a particular employee within the company. To do so, he will need to search for the specific receiver. The sender (client) will pick a random number m $[i = m\,;\ 0 < i < a]$. Using this random number i, the sender will determine the specific employee that the value of Y will be delivered to. The specific receiver of the company will compute a random secret key, for example s, and generate the value X by using Diffie Hellman protocol:

$$X = g^s(mod\ p)$$

Upon receiving the value of Y, the receiver will send X to the client. The client will use the value of X and his secret key b to compute K

$$K = X^b(mod\ p)$$

The employee will use the value of Y and his secret key s to compute K

$$K = Y^s(mod\ p)$$

Both the value of K will be equivalent and this signifies that the sender and the receiver has shared their mutual key. This will enable them to communicate privately without any threat or external intervention.

Algorithm:

Company_A= [$Employee_1, Employee_2, Employee_3 \ldots\ldots\ldots Employee_i$]

```
Function ConnectToEmployee(EmployeeId, SenderId)
        for i = 0 to N(Company_A)
                if (Employee_i == EmployeeId)
                        KeyUniqueToAllSender = GenerateKey(SenderId)
                        return KeyUniqueToAllSender
                Endif
        Endloop
Endfunction


Function DecryptMessage(message, key)
        plain_text_message = decryption(message, key)
        Return plain_text_message
EndFunction


Function EncryptMessage(message, key)
        encrypted_message = encryption(message, key)
        Return encrypted_message
```

```
EndFunction

#Firstly the sender need to connect to the employee and get the key

Key = ConnectToEmployee(EmployeeId, SenderId)

While Sender is communicating

        encrypted_message = EncryptMessage(message,Key)

        SendToEmployee(EmployeeId, SenderId)

EndLoop

#For all case to decrypt the conversation

        DecryptMessage(message, Key)
```

Example: Company A has 'a' employees and each employee is denoted by a_i, where $0 < i < a$. Lets say, $a = 127$, which means there are 127 employees in the company.

A client outside the company wants to communicate privately with any one of the employees by sharing secret keys. At first a generator $g = 117$ and a large prime number $p = 5371$ has been selected which is known to everyone. Now, the client will choose a random secret key, say $b = 97$ and by using the Diffie Hellman protocol he will generate the value of Y.

$$Y = g^b (mod\ p)$$

$$Y = 117^{97} (mod\ 5371)$$

$$Y = 1532$$

After computing Y, the client will determine whom he wants to deliver the value of Y to, by giving the value of i, he will be able to determine the specific employee, say $i = 92\ ; 0 < i < 127$. The 92^{nd} employee of the company now will choose a random secret key, say $s = 58$ and generate the value of X by using the Diffie Hellman protocol.

$$X = g^s (mod\ p)$$

$$X = 117^{58} (mod\ 5371)$$

$$X = 3313$$

The employee will now send his value of X to the client. Upon receiving the value of Y and X, both the employee and the client will compute the common key, which they will share to communicate securely.

The client will use the value of X and his secret key b, to get K

$$K = 3313^{97} (mod\ 5371)$$

$$K = 2872$$

The employee will use the value of Y and his secret key s to get K

$$K = 1532^{58} (mod\ 5371)$$

$$K = 2872$$

Finally, they are able to share a common key, which the client will use to transmit encrypted messages and the employee will be able to decrypt the message into plaintext.

To encrypt a message the client will use affine cryptosystem, where he uses the $\mathbb{Z}_{26}$ to make the message unreadable and send it to the employee by sharing the secret keys. Therefore, after receiving the encrypted message the employee will use the affine cryptosystem to decrypt the message and communicate securely.

Affine Groups

Affine Groups: The affine group over a field m is the group of all invertible affine transformations from the space into itself. [6]

Affine cryptosystems use the group $\mathbb{Z}_{26}$, which encodes messages using a mathematical function

$$f(x) = ax + b \; mod \; 26$$

where 26 is the number of letters in the alphabet and a and b are integers between $(0,1,2, \dots \dots 25)$ and x is the number equivalent to the corresponding letter. In order to decode the message, f^{-1} needs to be computed, so the function becomes,

$$f^{-1}(x) = a^{-1}x - a^{-1}b \; mod \; 26$$

Hence, we will simplify affine codes using matrices. We will use pairs of letters to encode a message as an alternative of encoding a single letter one at a time. For this procedure to work we choose an $a \in \mathbb{Z}_{26}$ that is invertible. This is only possible if $\gcd(a, 26) = 1$. So we will let $a = 5$ since $\gcd(5,26) = 1$ as a result $a^{-1} = 21$. [2]

Let's say we want to encode the message '**SEIZE**'. Therefore, the corresponding digits are

$$18 \; 4 \; 8 \; 25 \; 4$$

If a is a 2×2 matrix that is invertible with entries in $\mathbb{Z}_{26}$ and b is a fixed column vector say $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ and a is $\begin{bmatrix} 5 & 3 \\ 2 & 1 \end{bmatrix}$ and the matrix operations are performed in $\mathbb{Z}_{26}$. [5]

$$f(x) = \begin{bmatrix} 5 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 18 \\ 4 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} \; mod \; 26$$

$$f(x) = \begin{bmatrix} 102 \\ 40 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} \; mod \; 26$$

$$f(x) = \begin{bmatrix} 104 \\ 43 \end{bmatrix} \; mod \; 26$$

$$f(x) = \begin{bmatrix} 0 \\ 17 \end{bmatrix}$$

Hence, we can now say that for the digits $0\ 17$ the corresponding encrypted letters are AR. Similarly, we can now find the next pair,

$$f(x) = \begin{bmatrix} 5 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 25 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} \ mod\ 26$$

$$f(x) = \begin{bmatrix} 115 \\ 41 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} \ mod\ 26$$

$$f(x) = \begin{bmatrix} 117 \\ 44 \end{bmatrix} \ mod\ 26$$

$$f(x) = \begin{bmatrix} 13 \\ 18 \end{bmatrix}$$

Again, we can write corresponding encrypted letters for the digits $13\ 18$ as NS. Similarly, for the last letter E we have to add a dummy letter E to fill out the pair, so for the digit 4 we will add another digit 4 to make it a pair and hence a column vector. Then compute the encrypted pair of letters:

$$f(x) = \begin{bmatrix} 5 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} \ mod\ 26$$

$$f(x) = \begin{bmatrix} 32 \\ 12 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} \ mod\ 26$$

$$f(x) = \begin{bmatrix} 34 \\ 15 \end{bmatrix} \ mod\ 26$$

$$f(x) = \begin{bmatrix} 8 \\ 15 \end{bmatrix}$$

Finally, we can write the last corresponding encrypted letter for the digit $8\ 15$ as IP. Therefore, the final encrypted word is **ARNSIP**.

Now to decrypt this message first of all we need to find the inverse matrix of a.

$$a = \begin{bmatrix} 5 & 3 \\ 2 & 1 \end{bmatrix}$$

Therefore, $$a^{-1} = \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix}$$

Decryption: The encrypted message is **ARNSIP**, for which the corresponding digits are 0 17 13 18 8 15 .

$$f^{-1}(x) = a^{-1}x - a^{-1}b \; mod \; 26$$

Therefore, $$f^{-1}(x) = \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix} \begin{bmatrix} 0 \\ 17 \end{bmatrix} - \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 51 \\ 357 \end{bmatrix} - \begin{bmatrix} 59 \\ 67 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 18 \\ 290 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 18 \\ 4 \end{bmatrix}$$

So the decrypted letter for the corresponding digits are SE. Similarly, we can find the rest of the letters are as follows,

$$f^{-1}(x) = \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix} \begin{bmatrix} 13 \\ 18 \end{bmatrix} - \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 379 \\ 404 \end{bmatrix} - \begin{bmatrix} 59 \\ 67 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 320 \\ 337 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 8 \\ 25 \end{bmatrix}$$

The decrypted letters are IZ.

Again, $$f^{-1}(x) = \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix} \begin{bmatrix} 8 \\ 15 \end{bmatrix} - \begin{bmatrix} 25 & 3 \\ 2 & 21 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 245 \\ 331 \end{bmatrix} - \begin{bmatrix} 59 \\ 67 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 186 \\ 264 \end{bmatrix} \; mod \; 26$$

$$f^{-1}(x) = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

Therefore, the decrypted letters are EE. Hence, we can now get the decrypted message as **SEIZEE** . Since a dummy letter was added to complete the last pair during encryption, we can now remove the last letter to get the original plaintext. That is: **SEIZE .**

This is the procedure through which the client will encrypt the messages and send it to the employee using the shared key and the client will also be able to use this procedure to decrypt messages and transfer information privately. We can use a larger matrix to make decoding difficult for other members of the company.

Conclusion

This thesis gives a brief on cryptography and group theory and how group theory can be applied on cryptography to make communications secure and private. It also clearly shows the Diffie Hellman protocol, which uses a cyclic group on a finite set of elements, to securely communicate. In this thesis, I proposed that by using the Diffie Hellman protocol we can demonstrate an application where a sender can determine a specific member from a finite group of members to communicate without the risk of information getting leaked and that has been shown through an arbitrary case used as an example. Beside, sharing keys they can encrypt and decrypt messages using affine cryptosystem which uses $\mathbb{Z}_{26}$.

References

[1] Arora, P. (2016). USE OF GROUP THEORY IN CRYPTOGRAPHY. *International Journal Of Advance Research And Innovative Ideas In Education*, *2*(6), 1767–1772.

[2] Judson, T. W. (1993). *Abstract Algebra Theory and Applications*. PWS Pub. Co.

[3] Ruohonen, K. (2014). *MATHEMATICAL CRYPTOLOGY*.

[4] Li, N. (2010). Research on Diffie-Hellman key exchange protocol. *2010 2nd International Conference on Computer Engineering and Technology*, *4*, 634–647.

[5] Blackburn, S. R., Cid, C., & Mullan, C. (2010). *Group theory in Cryptography*.

[6] Milne, J. S. (2012). *Basic Theory of Affine Group Schemes*.

[7] https://www.tutorialspoint.com/cryptography/benefits_and_drawbacks.htm

[8] https://brilliant.org/wiki/diffie-hellman-protocol/

[9] http://www.laits.utexas.edu/~anorman/BUS.FOR/course.mat/SSim/history.html