Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Education for Pure Sciences
Department of Mathematics

Asymmetric Encryption Schemes Based on The Numerical Optimization Problems

A Thesis
Submitted to the council of the College of Education for Pure Sciences
University of Babylon
as partial Fulfillment of the Requirements for the
Degree of Master in Education / Mathematics

By

Huda Kadhim Mahmood Kadhim

Supervised by

Asst. Prof.

Ruma Kareem K. Ajeena

2022 A.D 1444 A.H

بسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيم

﴿فَأَمَّا الزَّبَدُ فَيَذْهَبُ جُفَاءً ۖ وَأَمَّا مَا يَنفَعُ النَّاسَ فَيَمْكُثُ فِي الْأَرْضِ (١٧)﴾

صدَقَ اللهُ العليُّ العظـم

﴿سورة الرعد، آية١٧﴾

DEDICATION

To our greatest and honored prophet Mohamed (God prays on him).

to my mother.

to my father.

to my husband.

to my brothers.

to my sister

to my children.

to my supervisor Dr. Ruma Kareem K. Ajeena.

to all my friends and relatives.

I dedicate this work.

ACKNOWLEDGEMENT

First, my thanks and gratitude to Allah, Almighty Who awarded me the opportunity and health to complete this work.

I would like to thank my supervisor Dr. Ruma Kareem K. Ajeena for her help, encouragement, guidance, and support to complete my work in this thesis. My prayer for her and I will remember her help and kindness with me. My gratitude and thankfulness for her.

I would like to thank other professors who helped during the master study. Finally, I'm very thankful to all my family, especially my father, mother, and my husband.

Huda Kadhim

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table No. **Title of Table**

MATHEMATICAL SYMBOLS

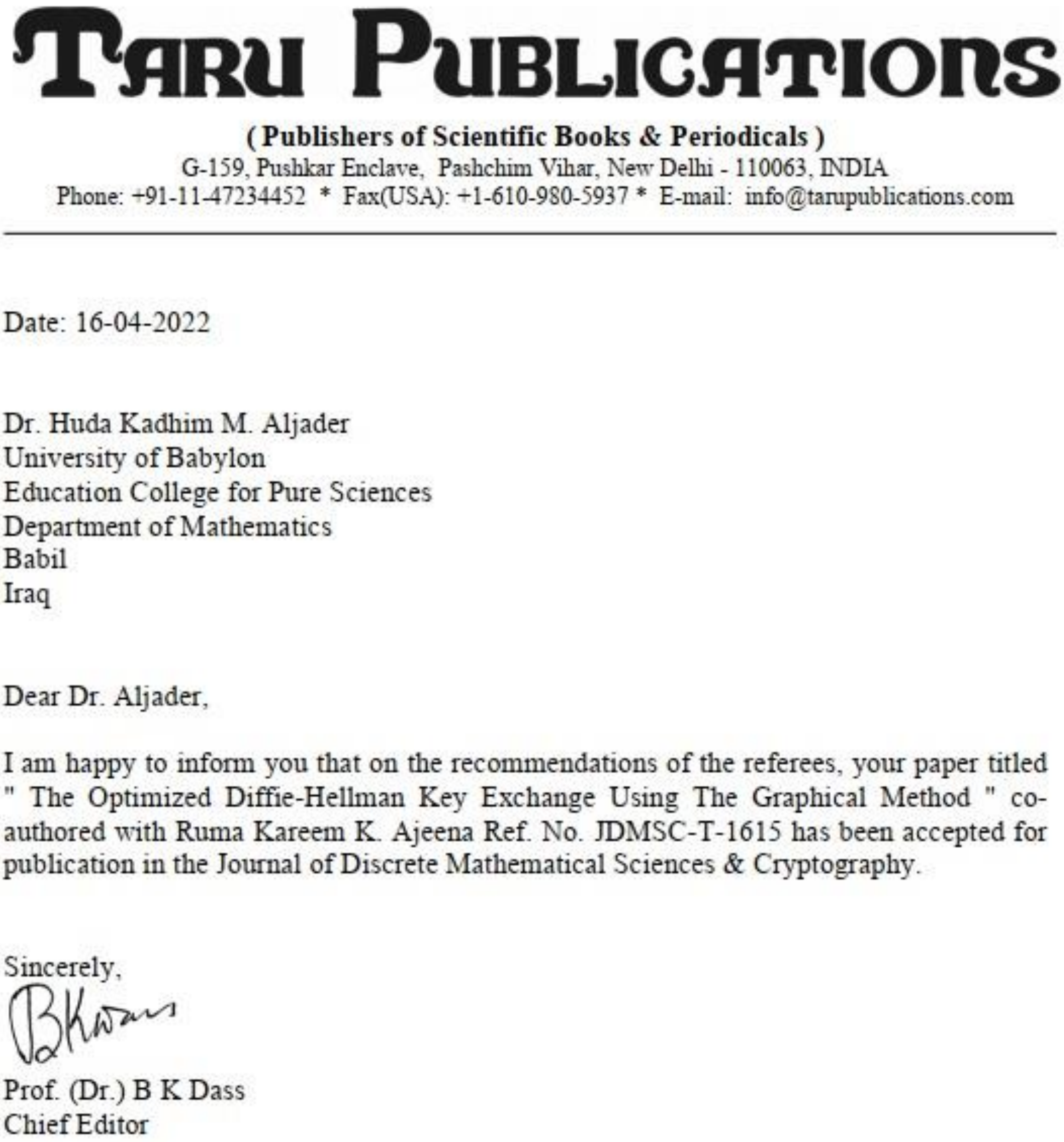
F	Field.
p	Prime number.
F_p	Prime field.
g	Primitive root.
h	Nonzero element.
a, b	Privat keys.
k	Ephemeral key.
Max	Maximization.
Min	Minimization.
m	Plaintext message.
(C_1,C_2)	Ciphertext
$\equiv$	Congruence.
M	a very large value.
S	slack variable.
R	Artificial variable.
r	A new objective functions.
mod	Modulo.
P_A	Public key.

ABBREVIATIONS

EEA	Extended Euclidean Algorithm.
AES	Advanced Encryption Standard.
AGDH	Algebraically Generalized Diffie-Hillman.
AKE	Authenticated key Exchange.
DHKE	Diffie-Hellman key Exchange.
DLFs	Discrete Logarithm Functions.
DLP	Discrete Logarithm Problem.
EC	Elliptic Curve.
ECC	Elliptic Curve Cryptosystem.
EPKC	ElGamal Public Key Cryptosystem.
gcd	The Greatest Common Divisor.
IFP	Integer factorization Problem.
LP	Linear Programming.
LPP	Linear Programming Problem.
MEC	Modified ElGamal Cryptosystem.
MECA	Modified El Gamal Cryptosystem Algorithm.
MOR	Model Order Reduction.
ODHKE	Optimized Diffie-Hellman key Exchange.
ODLPs	Optimized Discrete Logarithm Problem.
OEPKC	Optimized El-Gamal Public Key Cryptosystem.
RFID	Radio Frequency Identification.

RSA	Rivest–Shamir–Adleman.
SK	Secret Key.
SSK	Share Secret Key.
TMIS	Telecare Medicine Information System.

Publications



TARU PUBLICATIONS

(Publishers of Scientific Books & Periodicals)
G-159, Pushkar Enclave, Pashchim Vihar, New Delhi - 110063, INDIA
Phone: +91-11-47234452 * Fax(USA): +1-610-980-5937 * E-mail: info@tarupublications.com

Date: 16-04-2022

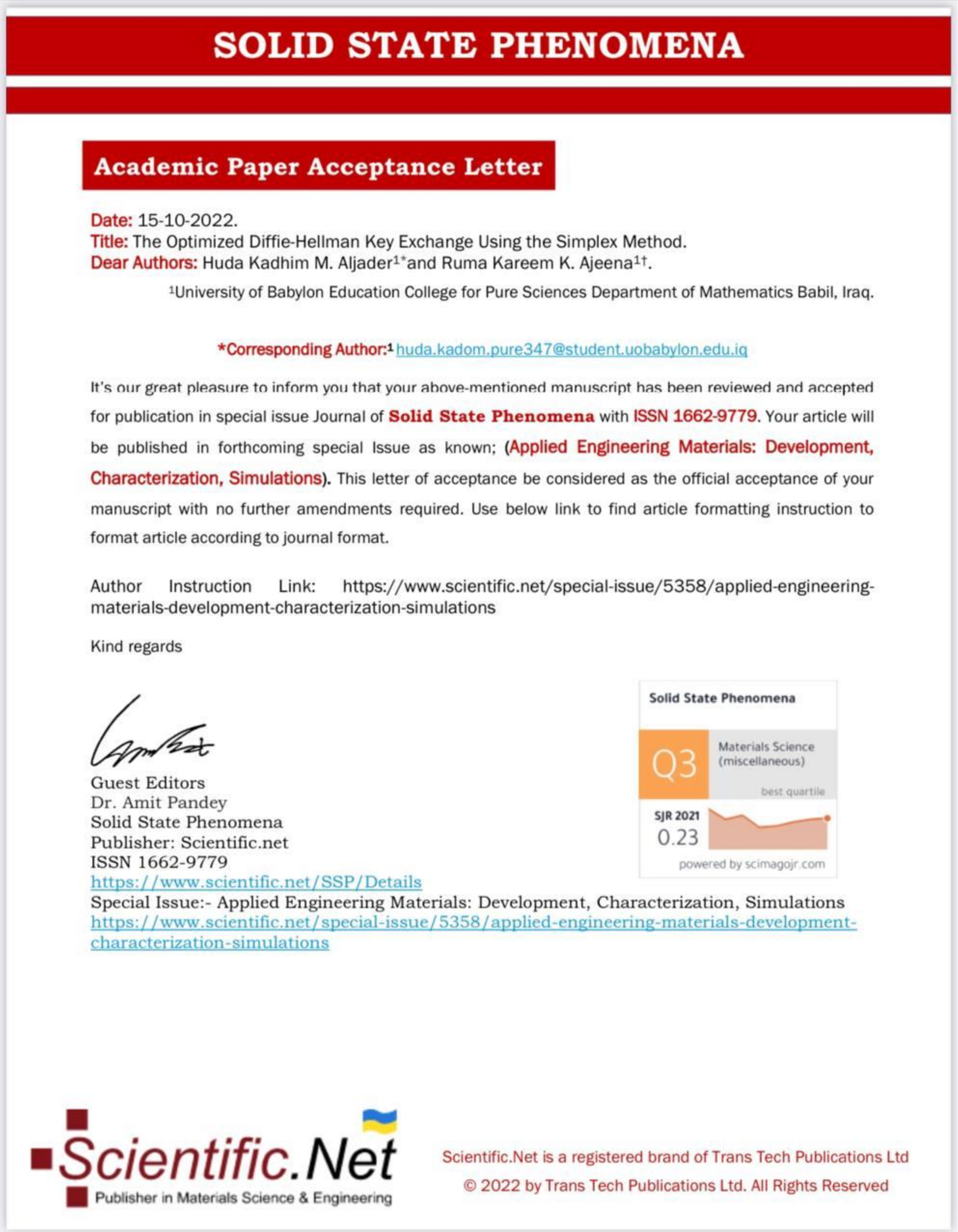
Dr. Huda Kadhim M. Aljader
University of Babylon
Education College for Pure Sciences
Department of Mathematics
Babil
Iraq

Dear Dr. Aljader,

I am happy to inform you that on the recommendations of the referees, your paper titled " The Optimized Diffie-Hellman Key Exchange Using The Graphical Method " co-authored with Ruma Kareem K. Ajeena Ref. No. JDMSC-T-1615 has been accepted for publication in the Journal of Discrete Mathematical Sciences & Cryptography.

Sincerely,

Prof. (Dr.) B K Dass
Chief Editor



SOLID STATE PHENOMENA

Academic Paper Acceptance Letter

Date: 15-10-2022.
Title: The Optimized Diffie-Hellman Key Exchange Using the Simplex Method.
Dear Authors: Huda Kadhim M. Aljader[1*]and Ruma Kareem K. Ajeena[1†].

[1]University of Babylon Education College for Pure Sciences Department of Mathematics Babil, Iraq.

***Corresponding Author:**[1] huda.kadom.pure347@student.uobabylon.edu.iq

It's our great pleasure to inform you that your above-mentioned manuscript has been reviewed and accepted for publication in special issue Journal of **Solid State Phenomena** with ISSN 1662-9779. Your article will be published in forthcoming special Issue as known; (**Applied Engineering Materials: Development, Characterization, Simulations**). This letter of acceptance be considered as the official acceptance of your manuscript with no further amendments required. Use below link to find article formatting instruction to format article according to journal format.

Author Instruction Link: https://www.scientific.net/special-issue/5358/applied-engineering-materials-development-characterization-simulations

Kind regards

Guest Editors
Dr. Amit Pandey
Solid State Phenomena
Publisher: Scientific.net
ISSN 1662-9779
https://www.scientific.net/SSP/Details
Special Issue:- Applied Engineering Materials: Development, Characterization, Simulations
https://www.scientific.net/special-issue/5358/applied-engineering-materials-development-characterization-simulations

Solid State Phenomena
Q3 Materials Science (miscellaneous)
best quartile
SJR 2021
0.23
powered by scimagojr.com

Scientific.Net
Publisher in Materials Science & Engineering

Scientific.Net is a registered brand of Trans Tech Publications Ltd

Abstract

The Diffie Hellman key exchange (DHKE) method and the EL-Gamal public key encryption (EPKC) method are asymmetric key encryption algorithms, which propose new schemes with the purpose of increasing the security of the two systems to protect confidential information from breach through the difficulty of accessing the secret key and the prime number by studying the problems of numerical optimization. The researcher found a link between these two methods by proposing asymmetric encryption schemes that depend on methods for solving numerical optimization problems that aim to raise the level of security for these two systems. The proposed version was based on the improved discrete logarithm functions (ODLFs) that were introduced as a new definition in this work. The graphical method, the simplex method, the Big-M method, and the two-phase method are essential tools for calculating a shared secret key (SSK) in the optimized Diffie-Hellman Enhanced Key Exchange ODHKE and the optimized EL-Gamal public key cryptography system OEPKC. OEPKC and ODHKE are more secure protocols for computing SSK compared to the original DHKE and EPKC. The security considerations of the proposed new OEPKC and ODHKE are determined based on the difficult computation of ODLFs. Compared to the original DHKE and EPKC.

Chapter One

Introduction

General Introduction

1.1 Introduction

Cryptography is all around, from ATMs, mobile phones and the internet, to the security systems that protect secrets in the workplace, to the civilian and military codes that protect skies.

The idea behind encryption schemes is to change confidential information in such a way that its meaning becomes incomprehensible to any unauthorized person. The two most common uses of encryption are to securely store data in a computer file or to transmit it over an insecure channel such as the internet. In either case, the fact that the document is encrypted does not prevent unauthorized people from accessing it, but it does ensure that they cannot understand what they see [1].

The information encrypted is often called 'plaintext', while the process of changing it is called 'encryption'. The encrypted text is called the ciphertext or the cipher statement, and the set of rules used to encrypt the plaintext information is called the encryption algorithm. Typically, this algorithm relies on an "encryption key"; It is an entry to it in addition to the message. For the recipient to retrieve the message through the ciphertext, there must be a 'decryption algorithm' which, when used with the appropriate 'decryption key', retrieves the plaintext from the ciphertext [1].

An important fact is that knowing the encryption key is not necessary to get the message through the ciphertext. The simple observation is the basis of Diffie-Hellman's highly influential work. It had a great influence on modern cryptography and resulted in a natural division between two types of cryptographic systems. They are the symmetric cryptosystem and the asymmetric cryptosystem. This work focuses on asymmetric cryptosystem, which depends on the discrete logarithm functions (DLFs) in its algorithm [2].

Among these cryptosystems are Diffie-Hellman key exchange (DHKE) and El-Gamal public key cryptosystem (EPKC).

Numerous mathematical specializations, in particular graph theory and number theory, have been linked to the creation of several encryption techniques. One of the key agreement procedures proposed by Diffie and Hellman in 1976 is the Diffie-Hellman key exchange (DHKE). Its goal is to make it possible for two users to communicate a shared secret key that can be used to encrypt plaintext [3]. Numerous academics have published a variety of research papers to improve this approach. The Diffie-Hellman protocol was recently updated in 2017 by Aryan et al [4] to obtain a stronger secret key. The DHKE key was altered in 2018 by W. Jirakitpuwapat1 and P. Kumam [5], utilizing the maximal abelian subgroup of group automorphism. In this work, a revised version of the DHKE is proposed based on the proposed hard mathematical problem which is optimized discrete logarithm function (ODLF) based on different optimization methods.

On the other hand, El-Gamal public key cryptosystem is an encryption algorithm invented by Taher El-Gamal in 1985 that is used to encrypt the plaintexts. It depended on the DLP that is used in the Diffie-Hellman key exchange. Many researchers introduced several research works to give different versions to improve this protocol. In 2021 [6], Hussein, H. I., & Abduallah, W. M. introduced a modified E-lGamal cryptosystem to reduce file size growth after encryption. The modified El-Gamal cryptosystem and classic Elliptic Curve (EC) are tested utilizing different-sized text data in the same programming environment. In 2011 [7], Ahmed, J.M., and Ali, Z.M. presented the RSA and El-Gamal-based paradigm. RSA employs the Integer Factorization Problem (IFP), while El-Gamal uses DLP. This paradigm combines IFP and DLP to allow the asymmetric cryptosystems to quickly compute two challenging issues. Their proposed is speedier up than El-Gamal and RSA. In 2002 [8], Hwang, M. S., Chang, C. C., & Hwang, K. F proposed the symmetric cryptosystem to encrypt a large private message whereas the asymmetric cryptosystem sends the secret key SK. The work in this thesis also proposes a new version of EPKC based on the proposed hard mathematical problem which is ODLF using different optimization methods.

1.2 Previous Studies

In 2001 [9], Bresson, E., Chevassut, O., & Pointcheval, D. propose group dynamic Diffie-Hellman protocols for authenticated key exchange (AKE) work in a scenario where group membership is unknown in advance and parties can join and leave the multicast group at any moment. A security model for AKE is determined. The security of a modified Diffie-Hellman protocol in this scenario is established.

Also, in 2002, Bresson, E., Chevassut, O., & Pointcheval, D., [10] presented two principals are agreed with public/private keys and a shared secret via authenticated Diffie-Hellman key exchange. The security models are determined. An authenticated dynamic group Diffie-Hellman protocol is defined and proved.

In 2003, Yang, C. C. et al [11], generalized the group-oriented cryptosystem, the sender can transmit a conditional message to a group of users so that only certain users can decrypt it. Also, they employed an ElGamal and an elliptic curve ElGamal cryptosystem to generalize and group-orient respectively.

In 2007, Bresson, E., Chevassut, O., & Pointcheval, D. [12], summarized provably a secure group DH key exchange.

In 2010, Nan Li [13], proposed another version of the Diffie-Hellman that allowed for two users to securely exchange a secret key for message encryption. The protocol exchanges keys.

In 2011, Sharma, P., Sharma, S., & Dhakar, R. S. [14], their study compared between the Modified El-Gamal Cryptosystem Algorithm (MECA) and El-Gamal's security and complexity.

In 2012, Mahalanobis, A. [15], presented a study of the Model Order Reduction (MOR) cryptosystem using the special linear group over finite fields. The automorphism group of the special linear group is analyzed for this purpose. The MOR cryptosystem has better security in compare with the ElGamal cryptosystem over finite fields.

In 2013, Rewagad, P., & Pawar, Y. [16], they propose the three-way mechanism that permitted the authentication, data security, and verification. Digital signature, Diffie Hellman key exchange, and AES are recommended in their study to secure cloud data confidentiality. Diffie Hellman key exchange renders hacked keys useless without the user's private key. The three-way approach protects cloud data from hackers.

In 2017, Ara, A., et al [17], proposed a new signature scheme together with an implementation of the Diffie-Hellman key distribution scheme that achieves a public key cryptosystem. The security of both systems relies on the difficulty of computing the discrete logarithms over finite fields.

In 2017, Juha Partala [18], proposed an algebraically generalized Diffie–Hellman scheme (AGDH) that can be applied of any algebra for key exchange. He showed that the proposed scheme is secure if the problem of computing the images under an unknown homomorphism is infeasible. Also, he presented a survey on the algebraic properties of existing key exchange schemes and the family of algebraic structures for each scheme.

In 2019, Arboleda, E. R. [19], proposed combining two encryption techniques. This combining depended on the security and fast chaos with ElGamal cryptography.

In 2020, Salem, F. M., & Amin, R. [20], they provided a privacy-preserving radio frequency identification (RFID), authentication approach for a telecare medicine information system (TMIS), based on El-Gamal. The proposed protocol can withstand attacks.

1.3 Statement of the Problem

It is clear from the previous studies that the Diffie-Hellman Key Exchange method and the AL-Gamal Public Key method were linked to the rest of the disciplines, and the results of these studies aimed either to raise the security of these systems or to use these systems to address a problem. We need to continuously develop data encryption methods to protect confidential information from being violated by increasing the difficulty of accessing the secret key and the prime number. Studies did not address the relationship between encryption and numerical improvement problems. By studying numerical optimization problems, which aim to reach the best solutions from the researcher found a link between these two methods by proposing asymmetric encryption schemes that depend on methods for solving numerical optimization problems aimed at raising the security of these two systems. We will rely on these methods to find the shared secret keys and the public key by proposing an improved discrete logarithmic function that is more difficult to calculate than the original discrete logarithmic function, and then we will make a comparison between the proposed systems and the two original systems.

1.4 Objectives of the study

The objectives of this thesis are:

i. Introducing new versions of the Diffie-Hellman key exchange DHKE, which is depending on the linear optimization form for computing the shared secret key SSK.

ii. Introducing new versions of the El-Gamal public key cryptosystem EPKC which depended on the linear optimization problems to calculate the keys, encryption and decryption.

iii. Providing the ODHKE and OEPKC protocols which are more secure in compared to the original DHKE and EPKC protocols.

iv. Comparison of the original DHKE and EPKC with the proposed versions ODHKE and OEPKC.

1.5 Thesis Structure

The outline of this study is as follows: in addition to chapter 1, it contains

- Chapter 2 It includes the basic facts of finite fields. It also explains some of the basic tools in number theory, such as the Euclidean theorem, the division algorithm, the greatest common divisor, the extended Euclidean algorithm, the discrete logarithm problem (DLP). Also, it included an introduction to cryptography. Finally, this chapter presents some important fact of the optimization problem, especially the linear programming and some methods for solving the linear programming, including the graphical method to solve the problem of linear programming and algebraic methods to solve the problem of linear programming, which is the simplified method, the Big-M method, and the two-phase method.

- Chapter 3 In this chapter, DL-optimized public key ciphers are presented, which include enhanced Diffie-Hellman key exchange using the graphical method. The El-Gamal Enhanced Public Key Cryptography System using the graphical method is also proposed. An improved Diffie-Hellman method using the simplex method is explained with two examples in case (maximum) and case (minimum). An optimized public key cipher using the simplex method has also been proposed.

- Chapter 4 Other improved public key ciphers, which include Diffie-Helman optimized using the Big-M method. Two examples are discussed in case (maximum) and case (minimum). Jamal's Enhanced Public Key Cryptography System Using the Big-M Method. Enhanced Diffie-Helman key exchange using the two-stage method. Optimized public key cipher using the two-stage method. Next, the security considerations for these proposed schemes are outlined. Comparison of the proposed enhanced cipher schemes with the original cipher systems.

- Chapter 5 draws the conclusions and future works.

Chapter Two

Mathematical Background to Cryptography and Optimization Problem

Mathematical Background to Cryptography and Optimization Problems

2.1 Introduction

In this chapter, will present two DLP-based encryption schemes. One of them is the Diffie-Hellman key exchange [21] which is a method of securely exchanging cryptographic keys over a public channel and was one of the first major public protocols conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellmann. [21] DH is one of the earliest practical examples of public key exchange applied in the field of cryptography. Published in 1976 by Diffie and Hellman, this is the earliest work known to the public that proposed the idea of the private key and the corresponding public key. The other is El-Gamal encryption is an encryption algorithm invented by Taher El-Gamal in 1985 that is used to encrypt public keys. It is based on the Diffie-Hellman principle for exchanging cryptographic keys. On the other hand, some methods for solving linear optimization problems will be presented, which are the graphical method and algebraic methods, including the simplex method, the Big-M method, and the two-stage method. These methods are the basis for extracting the shared secret key (SSK) and creating a new schemas ODHKE and OEPKC.

2.2 Finite Fields

In this section, the mathematical concepts related to fields, especially finite fields are discussed as follows.

Definition 2.2.1. (Field). A Field is a nonempty set F of elements with two operations "$+$" (called addition) and "$\cdot$" (called multiplication) satisfying the following axioms: for all $a,b,c \in F$,

i. F is closed under $+$ and $\cdot$, i.e., $a + b$ and $a \cdot b$ are in F;

ii. Commutative laws: $a+b = b+a$, $a \cdot b = b \cdot a$;

iii. Associative laws: $(a+b)+c=a+(b+c), a\cdot(b\cdot c)=(a\cdot b)\cdot c$;

iv. Distributive law: $a\cdot(b+c)=a\cdot b+a\cdot c$.

Furthermore, two distinct identity elements 0 and 1 (called the additive and multiplicative identities, respectively) must exist in F satisfying:

v. $a+0=a$ for all $a\in F$;

vi. $a\cdot 1=a$ and $a\cdot 0=0$ for all $a\in F$;

vii. $\forall a$ in F, there exists an additive inverse element $(-a)$ in F such that $a+(-a)=0$;

viii. $\forall a\neq 0$ in F, there exists a multiplicative inverse element a^{-1} in F such that $a\cdot a^{-1}=1$ [22].

Definition 2.2.2. (Finite Field or Galois field). A field with finitely many elements is called a finite field. The finite field with q elements is denoted by F_q. Finite fields are also called Galois fields [22].

If a finite field has p elements, and p is prime, then the field is called a prime field, which is defined as follows.

Definition 2.2.3. For a prime p, let F_p be a set $\{0,1,...,p-1\}$ of integers and let $\varphi: Z/(p)\to F_p$ be the mapping defined by $\varphi([a])=a$ for $a=0,1,...,p-1$. Then F_p, endowed with the field structure induced by φ, is a finite field, called the Galois field of order p [22].

Example 2.2.1 If $p=7$, a set with five elements is

$F_7=\frac{\mathbb{Z}}{7\mathbb{Z}}=\{\{...,-14,-7,0,7,14,...\}\ \ \{...,-13,-6,1,8,15,...\}\ \ \{...,-12,-5,2,9,16,...\}$ $\{...,-11,-4,3,10,17,...\}\ \ \{...,-10,-3,4,11,18,...\}\ \ \{\ldots,-9,\ -2,5,12,19,\ldots\}\ \ \{\ldots,-8,$ $-1,6,13,20,\ldots\}\}$ [23].

Definition 2.2.4. (The Characteristic of a Field). Let F be a field. The characteristic of F is the least positive integer p such that, $p \cdot 1 = 0$, where 1 is the multiplicative identity of F. If no such p exists, we define the characteristic to be 0 [24].

Example 2.2.2 [24]. (i) The characteristics of $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ are 0.

(ii) The characteristics of the field Z_p is p for any prime p.

Example 2.2.3 The ring $F_{11} = Z/11Z$ is a prime field of characteristic 11. The F_{11} has exactly 11 elements. So, it is a finite. It is easy to check F_{11} as a field. Every element in F_{11} is equal to zero under multiplication by 11, thus, the characteristic is 11.

Lemma 2.2.1: Let F be a field

1. If the characteristic of F is positive, then $char(F)$ is prime.

2. The finite fields have $char(F) > 0$. By the first part of this lemma, a finite field has a prime characteristic [22].

Theorem 2.2.1 (Primitive Root Theorem). Let p be a prime number. Then there exists an element $g \in F_p{}^*$ whose powers give every element of $F_p{}^*$, i.e.

$$F_p{}^* = \{1, g, g^2, g^3, \dots, g^{p-2}\}.$$

Elements with this property are called primitive roots of F_p or generators of $F_p{}^*$. They are the elements of $F_p{}^*$ having order $p - 1$ [25].

Example 2.2.4 Suppose $p = 17$, and $F_{17}{}^* = \{1, 2, \ldots, 16\}$. An element $a = 3$ is a primitive root of $F_{17}{}^*$, since $a = 3$ generates all elements in $F_{17}{}^*$.

In other words,

$3^0 \equiv 1, 3^1 \equiv 3, 3^2 \equiv 9, 3^3 \equiv 10, 3^4 \equiv 13, 3^5 \equiv 5, 3^6 \equiv 15, 3^7 \equiv 11, 3^8 \equiv 16, 3^9 \equiv 14, 3^{10} \equiv 8,$
$3^{11} \equiv 7, 3^{12} \equiv 4, 3^{13} \equiv 12, 3^{14} \equiv 2, 3^{15} \equiv 6, 3^{16} \equiv 1.$

On the other hand, 3 is not a primitive root for F_{13}, since in F_{13},

$2^0 \equiv 1, 2^1 \equiv 2, 2^2 \equiv 4, 2^3 \equiv 8, 2^4 \equiv 16, 2^5 \equiv 15, 2^6 \equiv 13, 2^7 \equiv 9, 2^8 \equiv 1,$

Corollary 2.2.1. Every finite field contains at least one primitive element [26].

2.3 Arithmetic operation on a Prime Field

The arithmetical operations, addition, subtraction, and multiplication can be computed over a prime field F_p [27]. A simple example with a small value $p = 5$ can be given to explain the arithmetic on F_5. The addition operation can be done by $3 + 4 \pmod 5 \equiv 7 \pmod 5 \equiv 2 \pmod 5$.

And the subtraction computes by $3 - 4 \pmod 5 \equiv -1 \pmod 5 \equiv 4 \pmod 5$. Whereas, the multiplication can be calculated by $3 \cdot 4 \pmod 5 \equiv 12 \pmod 5 \equiv$ $2 \pmod 5$.

Now, what about computing the division operation over F_p? and how to find the value of $a/b \pmod p$? The answer is: it is possible to compute the division operation over F_p by computing the inverse element modulo p which is given in the following relation

$$\frac{a}{b}\ (mod\ \ p) \equiv a \cdot (\frac{1}{b})\ (mod\ \ p). \tag{2.1}$$

There are some methods to compute the inverse element mod p, one of them is the extended Euclidean algorithm (EEA).

2.4 Discrete Logarithm Problem

The discrete logarithm problem is a mathematical problem that arises in many settings, including the mod p version version that will be studied later. The first published public key construction, due to Diffie and Hellman, is based on the discrete logarithm problem in a finite field F_{p}, where recall that F_{p} is a field with a s prime number of elements [28].

Definition 2.4.1 Let g be a generator of F_p and let h be a nonzero element of $\mathrm{F_p}$. The Discrete Logarithm Problem (DLP) is the problem of finding an exponent x such that

$$g^x \equiv h(\mathrm{mod} p).$$

The number x is called the discrete logarithm of h to the base g and is denoted by $\log_g (h)$ [29].

Remark 2.4.1. The discrete logarithm problem is a well-defined problem, namely, to find an integer exponent x such that $g^x = h$. However, if there is one solution, then there are infinitely many, because Fermat's little theorem tells us that $g^{p-1} \equiv 1(\mathrm{mod} p)$. Hence if x is a solution to $g^x = h$, then $x + k(p-1)$ is also a solution for every value of k, because

$$g^{x+k(p-1)} = g^x \cdot (g^{p-1})^k \equiv h \cdot 1^k \equiv h(\mathrm{mod} p).$$

Thus, $\log_g (h)$ is defined only up to adding or subtracting multiples of $p-1$ In other words, $\log_e (h)$ is really defined modulo $p-1$. It is not hard to verify that $\log_g$ gives a well-defined function

$$log_g \colon F_{p^*} \longrightarrow \frac{z}{(p-1)Z^*}$$

Sometimes, for concreteness, we refer to the discrete logarithm as the integer x lying between 0 and $p-2$ satisfying the congruence $g^x \equiv h$ (mod p) [28].
Remark 2.4.2. It is not hard to prove that

$$\log_g (ab) = \log_g (a) + \log_g (b)$$

for all $a, b \in F_{p^*}$ [28].

Example 2.4.1 The number $p = 56509$ is prime, and one can check that $g = 2$ is a generator modulo p. How would we go about calculating the discrete logarithm of $h = 38679$?. The only method that is immediately obvious is to compute

$$2^2, 2^3, 2^4, 2^5, 2^6, 2^7, \ldots (\text{mod } 56509)$$

until we find some power that equals 38679. It would be difficult to do this by hand, but using a computer, we find that $\log_p (h) = 11235$. You can verify this by calculating 2^{11235} mod 56509 and checking that it is equal to 38679 [28].

Definition 2.4.2. Let G be a group whose group law we denoted by the symbol $*$. The Discrete Logarithm Problem for G is to determine, for any two given elements g and h in G, an integer x satisfying [28].

$$\underbrace{g * g * \ldots * g}_{x\text{-times}} = \underline{h}$$

2.5 Extended Euclidean Algorithm (EEA).

The Euclidean algorithm can be extended to compute the integers t and s such that $at + bs = d$, where d is a greatest common divisor of a and b. The extended Euclidean algorithm (EEA) can be used also for computing the inverse element modulo p, where p is a prime number. Before explaining of this algorithm, some mathematical concepts which are needed to present the EEA should be illustrated [29].

Theorem 2.5.1 (Division algorithm). Given integer a and b, with $b > 0$, there exist unique integers q and r satisfying

$$a = qb + r, \text{ with } 0 \le r < b. \qquad (2.2)$$

The integers q and r are called, respectively, the quotient and remainder in the division of a by b [29].

Definition 2.5.1 An integer b is said to be divisible by an integer $a \neq 0$, in symbols $a \mid b$, if there exists some integer c such that $b = ac$. Whereas, one can write $a \nmid b$ to denote b is not divisible by a [29].

Definition 2.5.2 Let a and b be given integers, with at least one of them different from zero. The greatest common divisor of a and b, denoted by $\gcd(a,b)$, is the positive integer d satisfying the following [29]:

(a) $d \mid a$ and $d \mid b$.

(b) If $c \mid a$ and $c \mid b$ with $c \leq d$, then $c \mid d$.

Theorem 2.5.2 [35]. Given integer a and b, not both of which are zero, there exist integers t and s such that

$$\gcd(a,b) = at + bs. \tag{2.3}$$

Definition 2.5.3 Two integers a and b, not both of which are zero, are said to be relatively prime whenever $\gcd(a,b) = 1$ [29].

Theorem 2.5.3 Let a and b be integers, not both zero. Then a and b are relatively prime if and only if there exist integers t and s such that $1 = at + bs$ [29].

The Euclidean Algorithm can be described as follows.

Theorem 2.5.4 (The Euclidean Algorithm). Let a and b be two integers such that $a \geq b > 0$. The first step is to apply the division algorithm, as given in Theorem (2.5.1), to a and b to get

$$a = q_1 b + r_1, \text{ with } 0 \leq r_1 < b.$$

If $r_1 = 0$, then $b \mid a$ and $\gcd(a,b) = b$. Whereas, if $r_1 \neq 0$, divide b by r_1 to produce integers q_2 and r_2 which satisfy

$$b = q_2 r_1 + r_2, \text{ with } 0 \le r_2 < r_1.$$

If $r_2 = 0$, then it should stop. Otherwise, proceed as before to get

$$r_1 = q_3 r_2 + r_3, \text{ where } 0 \le r_3 < r_2.$$

The division processing continues until getting a zero remainder. For instance, on the $(n+1)$the stage, r_{n-1} is divided by r_n. A zero remainder appears, since the decreasing sequence $b > r_1 > r_2 > \ldots \ge 0$ cannot contain more than b integers.

This process can be expressed by the system of the following equations:

$$\begin{array}{ll}
a = q_1 b + r_1 & 0 < r_1 < b \\
b = q_2 r_1 + r_2 & 0 < r_2 < r_1 \\
r_1 = q_3 r_2 + r_3 & 0 < r_3 < r_2 \\
\quad\vdots & \\
r_{n-2} = q_n r_{n-1} + r_n & 0 < r_n < r_{n-1} \\
r_{n-1} = q_{n+1} r_n + 0. &
\end{array}$$

The last nonzero remainder r_n is the $\gcd(a,b)$ [29].

Lemma 2.5.1 If $a = qb + r$, then $gcd(a,b) = gcd(b,r)$ [29].

Example 2.5.1 Suppose $a = 93$ and $b = 57$. The computation of $\gcd(93,57)$ by using the Euclidean algorithm can be done based on Theorem (2.5.4). The greatest common divisor of 93 and 57 is equal to 3.

In other words,

$$3 = gcd(93{,}57).$$

For representing the positive integer 3 as a linear combination of the integers 93 and 57, the back substitution way can be used on the remainders 0,3,6,15,21 and 36.

Thus,

$$3 = gcd(93,57) = 93t + 57s$$

where $t = 8$ and $s = -13$. There are other possibilities to write the positive integer 3 as a linear combination of 93 and 57. In other words, the values of t and s to express the positive integer 3 are not unique.

The back substitution way of the Euclidean algorithm to compute the values of t and s in the following relation

$$d = gcd(a, b) = at + bs$$

is known by the extended Euclidean algorithm (EEA). Algorithm (2.5.1) computes

$$a\,t_1 + b\,s_1 = y,\ \ a\,t_2 + b\,s_2 = x, \text{with } y \leq x,$$

where $y = 0$, the EEA terminates. In this case $x = gcd(a, b)$ and $t = t_2,\ s = s_2$ that satisfy

$$a\,t + b\,s = d.$$

Algorithm 2.5.1 Extended Euclidean algorithm for integers [30],

Input: Positive integers a and b with $a \geq b$.

Output: $d = gcd(a, b)$ and integerssatisfying t,s $a\,t + b\,s = d$.

```
1. $\boldsymbol{x \leftarrow a, y \leftarrow b}$.
2. $\boldsymbol{t_1 \leftarrow 1, s_1 \leftarrow 0, t_2 \leftarrow 0, s_2 \leftarrow 1}$.
3. **While** $y \neq 0$ **do**
    3.1 $q \leftarrow \left\lfloor \frac{x}{y} \right\rfloor, r \leftarrow x - qy, t \leftarrow t_2 - qt_1, s \leftarrow s_2 - qs_1$.
    3.2 $\boldsymbol{x \leftarrow y, y \leftarrow r, t_2 \leftarrow t_1, t_1 \leftarrow t, s_2 \leftarrow s_1, s_1 \leftarrow s}$.
4. **End while**
5. $d \leftarrow y, t \leftarrow t_2, s \leftarrow s_2$.
6. **Return** $(d, r, t, s)$.
```

Example 2.5.2 Suppose $a = 107$ and $b = 23$. The $gcd(107,23) = 1$, since 107 and 23 are relatively prime (see Definition (2.5.3)). The aim here is to find the values of t, s and the remainders r using Algorithm (2.5.1).

Suppose $a = x = 107, b = y = 23$ and the initial values is $t_1 = 1, s_1 = 0, t_2 = 0$, $s_2 = 1$.

Now, if $y = 23 \neq 0$ then

$$q = \left\lfloor \frac{107}{23} \right\rfloor = 4, r = 107 - 4 \cdot 23 = 15, t = 0 - 4 \cdot 1 = -4, s = 1 - 4 \cdot 0 = 1.$$

So, $x = 23, y = 15, t_2 = 1, t_1 = -4, s_2 = 0, s_1 = 1$.

When $y = 15 \neq 0$ then

$$q = \left\lfloor \frac{23}{15} \right\rfloor = 1, r = 23 - 1 \cdot 15 = 8, t = 1 - 1 \cdot (-4) = 5, s = 0 - 1 \cdot 1 = -1.$$

Therefore, $x = 15, y = 8, t_2 = -4, t_1 = 5, s_2 = 1, s_1 = -1$.

If $y = 8 \neq 0$ then

$$q = \left\lfloor \frac{15}{8} \right\rfloor = 1, r = 15 - 8 = 7, t = -4 - 1 \cdot (5) = -9, s = 1 - 1 \cdot (-1) = 2.$$

Thus, $x = 8, y = 7, t_2 = 5, t_1 = -9, s_2 = -1, s_1 = 2$.

When $y = 7 \neq 0$ then

$$q = \left\lfloor \frac{8}{7} \right\rfloor = 1, r = 8 - 7 = 1, t = 5 - 1 \cdot (-9) = 14, s = -1 - 1 \cdot (2) = -3.$$

Hence, $x = 7, y = 1, t_2 = -9, t_1 = 14, s_2 = 2, s_1 = -3$.

With $y = 1 \neq 0$ then

$q = \left\lfloor \frac{7}{1} \right\rfloor = 7, r = 7 - 7 = 0, t = -9 - (7) \cdot (14) = -107, s = 2 - 7 \cdot (-3) = 23$. Hence, $x = 1, y = 0, t_2 = 14, t_1 = -107, s_2 = 2, s_1 = 23$.

Thus, $1 = gcd(23,107)$. Also, $r = \{15,8,7,1,0\}, t = \{-4,5,-9,14,-107\}$ and $s = \{1,-1,2,-3,23\}$.

The EEA (2.5.1) can be executed with the input (a,p). On the step (3.1), the last nonzero remainder r is equal to 1. So, the integers y, t_1, s_1 on the step (3.2) satisfy $a\, t_1 + b\, s_1 = y$ with $y = 1$. Therefore $a\, t_1 \equiv 1 \pmod{p}$.

In other words, $a^{-1} \equiv t_1 \pmod{p}$. So, the inverse element mod p, $a^{-1} \bmod p$, can be computed using the following algorithm

Algorithm 2.5.2. Computing the inversion on F_p using the EEA [30]

Input: Prime p and $a \in [1, p-1]$.

Output: $a^{-1} \bmod p$.

```
1. $x \leftarrow p, y \leftarrow a$.
2. $t_1 \leftarrow 1, t_2 \leftarrow 0$.
3. While $y \neq 1$ do
     3.1  $q \leftarrow \left\lfloor \frac{x}{y} \right\rfloor, r \leftarrow x - qy, t \leftarrow t_2 - qt_1$.
     3.2  $x \leftarrow y, y \leftarrow r, t_2 \leftarrow t_1, t_1 \leftarrow t$.
4. End while
5. Return $(t_1 \bmod p)$.
```

Example 2.5.3 Let $a = 84$ and $p = 97$. Computing $84^{-1} \bmod 97$ can be done by using algorithm (2.5.1) as follows.

Suppose $a = y = 84, p = x = 97$ and the initial values are $t_1 \leftarrow 1, t_2 \leftarrow 0$.

When $y = 84 \neq 0$ then $q = \left\lfloor \frac{97}{84} \right\rfloor = 1, r = 97 - 1 \cdot 84 = 13, t = 0 - 1 \cdot 1 = -1$. So, $x = 84, y = 13, t_2 = 1, t_1 = -1$.

Now, $y = 13 \neq 0$ then $q = \left\lfloor \frac{84}{13} \right\rfloor = 6, r = 84 - 6 \cdot 13 = 6, t = 1 - 6 \cdot (-1) = 7$. Therefore, $x = 13, y = 6, t_2 = -1, t_1 = 7$.

When $y = 6 \neq 0$ then $q = \left\lfloor \frac{13}{6} \right\rfloor = 2, r = 13 - 2 \cdot 6 = 1, t = -1 - 2 \cdot 7 = -15$. Hence, $x = 6, y = 1, t_2 = 7\ t_1 = -15$.

Now, $y = 1 \neq 0$ then $q = \left\lfloor \frac{6}{1} \right\rfloor = 6, r = 6 - 1 \cdot 6 = 0, t = 7 - 6 \cdot (-15) = 97$. Hence, $y = 0, t_2 = -15, t_1 = 97$.

Therefore, $t_1 \equiv -15 \pmod{97} \equiv 82 \pmod{97}$.

2.6 Cryptography

Some basic concepts related to cryptography are discussed as follows.

In this section, some important definitions are presented as follows

Definition 2.6.1 Cryptography is the design and analysis of mathematical techniques that enable secure communications in the presence of adversaries [31].

Definition 2.6.2 Cryptosystem. A cryptographic system is specifically a set of methods (algorithms) for computing (implementing) the encryption and decryption [32].

Definition 2.6.3 Cryptanalysis is the study of analyzing cryptosystem in order to study the hidden aspects of the systems [33].

Definition 2.6.4 Plaintext. The information which we want to protect from other people (attackers) [32].

Definition 2.6.5 Security. It means that the difficulty to know the information which transferred over the channel easily [32].

2.7 Basic Communications Model

In Figure (2.1), entities **A** (Alice) and **B** (Bob) are communicating over an unsecured channel. We assume that all communications take place in the presence of an adversary **E** (Eve) whose objective is to defeat any security services being provided to **A** and **B.**

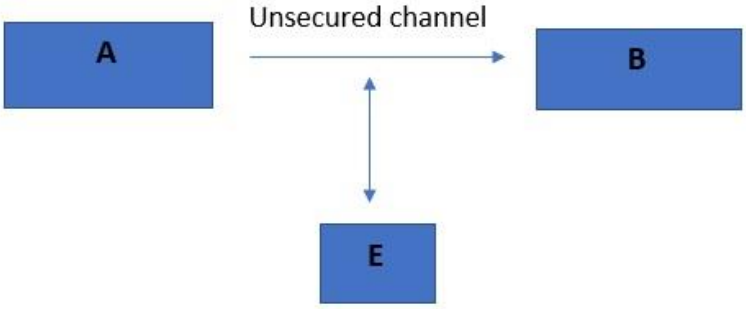


Figure 2.1 Basic communications model [31].

For example, A and B could be two people communicating over a cellular telephone network, and E is attempting to eavesdrop on their conversation [31].

Some Important Kinds of Cryptosystems

2.7.1 Symmetric-Key Cryptosystems.

The cryptosystems which use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information. Basically, symmetric encryption uses a single key for both encryption and description. And it is fast in execution. It is algorithm DES, 3DES, AES, and RC4. The purpose of the symmetric encryption is used for bulk data transmission [34].

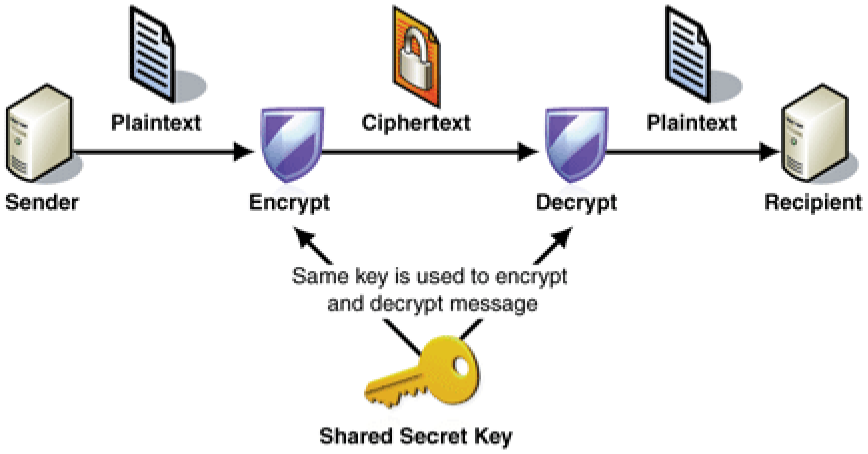


Figure 2.2 Symmetric-Key Cryptosystems [34].

2.7.2 Asymmetric-Key Cryptosystems (Public-Key Cryptosystems).

They use public and private keys to encrypt and decrypt data. The keys are not identical (asymmetric). One key in the pair can be shared with everyone; it is called the public key and another one can stay as a secret, which is called a private key. Basically, asymmetric encryption uses a different key for encryption and decryption. And it is slow in execution due to the high computation burden. It is algorithm Diffie-Hellman, RSA. The purpose of the asymmetric encryption is often used for securely exchanging secret key [34].

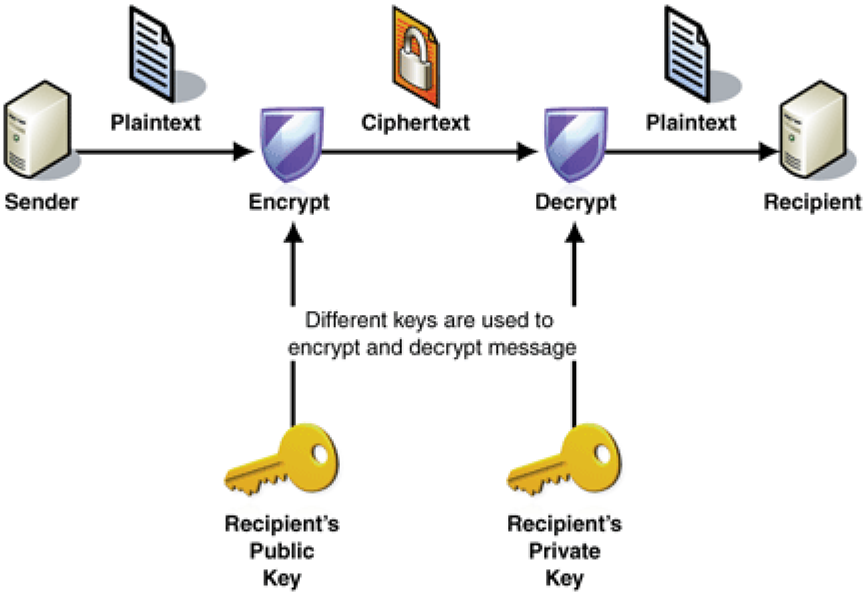


Figure 2.3 Asymmetric-Key Cryptosystems (Public-Key cryptosystems) [34].

2.8 DL-Public key cryptosystem

In this section, the original discrete logarithm public key cryptosystems are discussed as follows.

2.8.1 Diffie- Hellman Key Exchange

The Diffie-Hellman Key Exchange (DHKE) is explained as follows

Two entities agreed to choose the public parameters which are a large prime *p*, and a generator element g of a prime field F_p. Alice picks a secret integer a and Bob picks an integer b that he keeps secret. At the same time, they use the secret integers to compute.

$$A \equiv g^a (mod\, p) \text{ and } B \equiv g^b (mod\, p). \qquad (3.1)$$

These computations are exchanged between Alice and Bob, namely Alice sends *A* to Bob and Bob sends *B* to Alice. Finally, Bob and Alice again use their secret integers to compute

$$A' = B^a \ (\text{mod } p) \text{ and} B' = A^b \ (\text{mod } p). \qquad (3.2)$$

The values that are computed, A' and B' respectively, are actually the same, since

$$A' \equiv B^a \ \equiv \left(g^b\right)^a \equiv g^{a\cdot b} \equiv g^{b\cdot a} \equiv (g^a)^b \equiv A^b \equiv B' \,(\text{mod}\, p). \qquad (3.3)$$

The value $A' \equiv B'(\text{mod}\, p)$ is a shared secret key for Alice and Bob [38].

Example 2.8.1 Let p=47 be a prime number. The generator element $g = 3$ in F_{47}^*. Alice picks a secret a=10, while at the same time Bob picks an integer b = 6 that he keeps it as a secret. They use the secret integers to compute [38]

$$A \equiv g^a (mod\, p) \equiv 3^{10} (mod\, 4\,7) \equiv 17 (mod\, 4\,7),$$

and

$$B \equiv g^{a} \ \ (mod\, p) \equiv 3^{6} \ \ (mod\, 4\, 7) \equiv 24 \ \ \ (mod\, 4\, 7).$$

These computations are exchanged between Alice and Bob, namely Alice sends A = 17 to Bob and Bob sends B = 24 to Alice. Finally, Alice and Bob again use their secret integers to compute

$$A' \equiv B^{a}(mod\ p) \equiv 24^{10}\ (mod\ 47) \equiv 14\ (mod\ 47)$$

and

$$B' \equiv A^{a}(mod\, p) \equiv 17^{6}\ (mod\, 4\, 7) \equiv 14\ (mod\, 4\, 7)$$

The value $A' \equiv B' \equiv 14(mod\, 4\, 7)$is a shared secret key for Alice and Bob.

2.8.2 El-Gamal Public Key Cryptosystem

In 1985, Tahir El-Gamal proposed a public key cryptosystem (EPKC). The EPKC depended on the DLP that is given in Definition (2.4.1). On the EPKC, two entities, Alice, and Bob, agreed about selecting the public domain parameters. The public domain parameters are a prime number p and a generator element g over F_p^*, where $F_p^* = F_p \setminus \{0\}$.

For generating the keys, Alice selects randomly an integer a from the range [2, p - 1] as her private key. She computes her public key P_A using the following relation

$$P_A \equiv g^{a} \ (\text{mod } p). \qquad (3.4)$$

Now, Bob wants to encrypt a plaintext M and send it to Alice. So, he randomly selects M from the range [2, p -1]. Also, he selects an integer k from the range [2, p-1] as a random ephemeral key. He uses Alice's public key P_A to compute a pair of the ciphertext (C_1, C_2) as follows.

$$C_1 \equiv g^k \ (mod \ \ p) \text{ and } C_2 \equiv Mp_A^k (\text{mod}\, p) \tag{3.5}$$

Eventually, he will send pair of the ciphertext (C_1, C_2) to Alice.

Upon Alice receiving the ciphertext (C_1, C_2), some steps have been calculated by Alice to recover the original plaintext *M*. She first computes the value x through the following relation

$$x \equiv C_1^a \ (\text{mod } p) \tag{3.6}$$

Also, she computes the inverse value $x^{-1}(\text{mod}\, p)$ *of* x. One can use extended Euclidean algorithm (EEA) for computing the inverse element modulo *p*.

Finally, she computes the relation

$$x^{-1} * C_2 \equiv m \ (\text{mod } p) \tag{3.7}$$

to recover a plaintext *M* [39].

Example 2.8.2 Let p = 29 be a prime number. The generator element $g = 5$ in F_{29}^*. Alice selects her private key a = 9. She computes her public key P_A as follows [39].

$$\begin{aligned} P_A &\equiv g^a \ (mod \ \ p) \\ &\equiv 5^9 \ (mod \ \ 29) \\ &\equiv 4 \ (mod \ \ 29). \end{aligned}$$

Now, Bob selects the plaintext M = $20 \in [2, 28]$. He selects an integer $k = 15 \in [1, 28]$. He computes

$$\begin{aligned} C_1 &\equiv g^k \ (mod \ \ p) \\ &\equiv 5^{15} \ (mod \ \ 29) \\ &\equiv 5 \ (mod \ \ 29) \end{aligned}$$

and

$$\begin{aligned} C_2 &\equiv Mp_A^k \,(\mathrm{mod}\, p) \\ &\equiv 20.4^{15} (\mathrm{mod}\, 29) \\ &\equiv 22 \ \ (\mathrm{mod}\, 29). \end{aligned}$$

So, a pair of the ciphertext is $\left(C_1, C_2\right) = \left(5, 22\right)$ that will be send to Alice.

Now, Alice receives the ciphertext (5,22), so she wants to decrypt it and recover the plaintext M through the following calculation

$$\begin{aligned} (C_1^a)^{-1} \cdot C_2 \ (mod\, p) &\equiv (5^9)^{-1}.22 \ (mod \ \ 29) \\ &\equiv 22.22 (mod\, 2\, 9) \\ &\equiv 20 \ (mod\, 2\, 9) \ = \ M. \end{aligned}$$

2.9 Optimization problem

Optimization has been one of the most fundamental and successful tools in our daily lives. Optimization is an essential mathematical tool that aims to find the best value of variables that provide the minimum value or the maximum for a mathematical function (the objective function). Optimization algorithms are a fundamental and successful tool in mathematical programming to reach a solution, generally with the assistance of a computer. Optimization algorithms start with an initial estimated value of the variables and by an iterative technique generates a sequence of improved estimates, or iterates, until an optimal solution is reached. A great algorithm should be efficient, fast, accurate, and robust. It should generate a good approximation of an optimal solution [35].

The problem described:

- Maximize or minimize: Objective function
- Subject to: Constraints

This format is sufficiently general to include all optimization problems (most of life's problems too for that matter). Since the mathematical methods for solving such problems are more interested, it is necessary that the statement be reduced to symbolic form. For example [35]:

- Maximize: $F(x_1;\ x_2;\ x_3;\ )$
- Subject to: $g(x_1;\ x_2;\ x_3) \ = \ 0.$

Then statement reads as follows: Maximize some function F of $x_1;\ x_2;\ x_3$ by setting x_1, x_2 and x_3 subject to the requirement that another function g of x_1; x_2 and x_3, takes on the value zero [35].

The general optimization problem form

$$\begin{cases} \text{minimize} & f(x) & \text{objective function} \\ \text{subject to} & g(x) = 0 & \text{Equality Constraints,} \\ & h(x) \geq 0 & \text{Inequaliy} \end{cases}$$

2.10 Linear programming (LP)

Linear programming [35] is subfield of optimization. Linear programming is a mathematical technique for finding optimal solutions to the problems. Linear Programming deals with the problem of optimizing a linear objective function subject to linear equality and inequality constraints on the decision variables. Linear programming is not a programming language like C++, Java, or Visual Basic, its mathematical model. A feasible solution is a solution that satisfies all of the constraints. The feasible set or feasible region is the set of all feasible solutions. Finally, an optimal solution is the feasible solution that produces the best objective function value possible as shown in Figures (2.4) and (2.5).

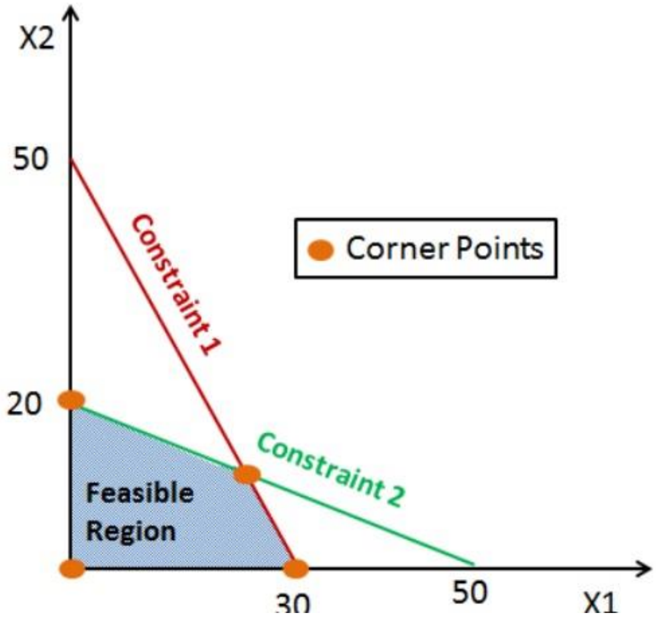


Figure 2.4 The feasible region [35].

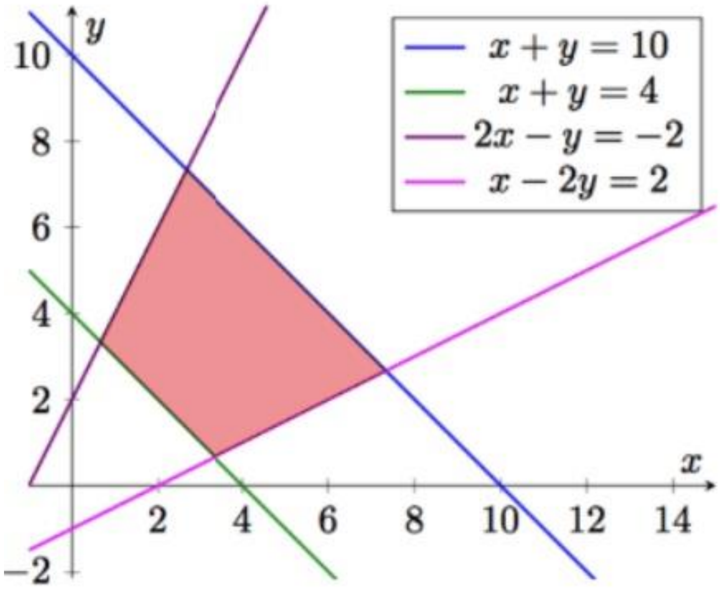


Figure 2.5 The feasible region of four constraints [35].

Definition 2.10.1 (Feasible region) is the set of points defined by the constraints [35].

Definition 2.10.2 (Optimal solution). An optimal solution x^*, for all $x \in F$, then $C^T x \leq C^T x^*$ [35].

Example 2.10.1 Consider the following simple linear programming problem [35]:

$$\begin{cases} maximize & 3x+5y \\ subject\ to & x+y \leq 4, \\ & x+3y \leq 6 \\ & x \geq 0, y \geq 0. \end{cases}$$

The feasible points are (substitute in the objective point)

- $(0,0) \Rightarrow 3(0) + 5(0) = 0$
- $(0,2) \Rightarrow 3(0) + 5(2) = 10$
- $(4,0) \Rightarrow 3(4) + 5(0) = 12$
- $(3,1) \Rightarrow 3(3) + 5(1) = 14$

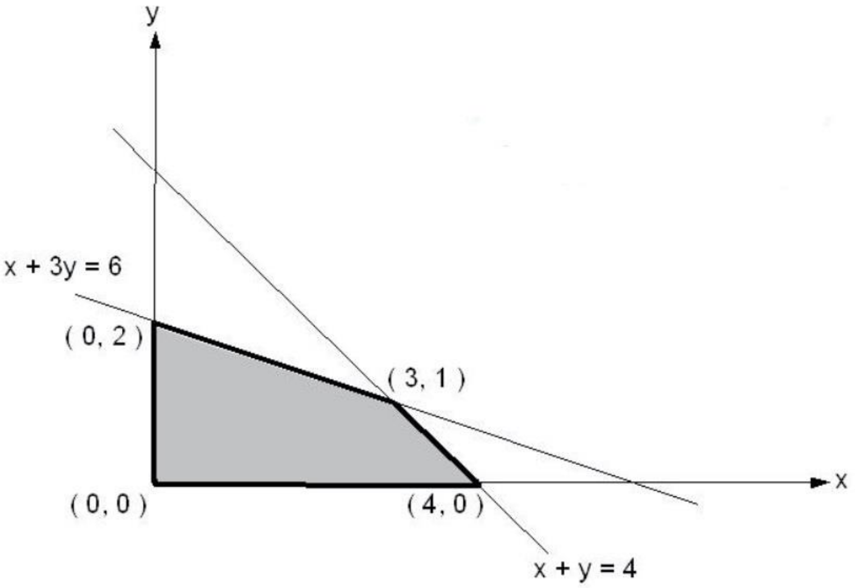


Figure 2.6 The feasible region for Example 2.10.1 [35].

- Therefore, the maximize objective value is 14 at the point (3,1).

2.11 Graphical Methods for Solving Linear Mathematical Programming Problems

In the previous section, some models called linear programming models are discussed. In each case, the model had a function called an objective function, which was to be maximized or minimized while satisfying several conditions or constraints.

If there are only two variables, one can use a graphical method of solution. Let us begin with the set of constraints and consider them as a system of inequalities. The solution of this system of inequalities is a set of points, S. Each point of the set S is called a feasible solution. The objective function can be evaluated for different feasible solutions and the maximum or minimum values obtained [36].

Definition 2.11.1. Graph (Linear). A linear graph consists of a number of nodes or junction points, each joined to some or all the others by arcs or lines [36].

2.11.1 Methods for Solving Graphical Problem

There are three methods of solving graphical problem. They are

i. General Graphical Method

ii. Corner Point Solution, and

iii. Computer Solution Method

Steps for Solving General Graphical Problem

The various steps for solving graphical problems are as follows [36].

1. Formulate to problem with mathematical form by
   - Specifying the decision variables.
   - Identifying the objective function.
   - Writing the constraint equations.

2. Plot the constraint equation on a graph.
3. Identify the area of feasible solution.
4. Locate the corner points of the feasible region.
5. Plot the objective function.
6. Choose the points where objective functions have optimal values.

Corner Point Solution

The search procedure adopted can be simplified by taking advantage of the first characteristics of feasible solution and the objective function

The following statements are of fundamental importance in linear programming

1. The solution set for a group of linear inequalities is a convex set. Therefore, the area of feasible solution for a linear programming problem is a convex set.
2. Given a linear objective function linear programming problem, the optimal solution will always include a corner point in the area of feasible solution.

Thus, the corner point method for solving linear programming problem has following steps

3. Step 1: Graphically identify the area of feasible solution.
4. Step 2: Determine the coordinates of each corner point on the area of feasible solution.
5. Step 3: Substitute the coordinates of the corner point in the objective function to determine the corresponding value of Z
6. Step 4: An optimal solution occurs in a maximization problem at the corner point yielding the highest value of Z and in a minimization problem at the corner point yielding the lowest value of Z.

Computer Solution Method

In actual application, LP problems are solved by computer methods as today much efficient computer software are readily available. The person who knows in detail about the LP problem can use this software effectively. Yet, following guidelines will be helpful for the person who wants to use computer software

1. One should fully understand the LP model
2. One may be able to make assumptions.
3. Have necessary skill to formulate the problem.
4. Ability to arrange solutions using a computer.
5. Capable of interpreting the output from such packages [36].

Example 2.11.1

Maximize: $Z = 4x + 5y$

Subject to:

$$2x + 5y \leq 25$$
$$6x + 5y \leq 45$$
$$x \geq 0, y \geq 0$$

Solution:

To solve the above linear programming model using the graphical method, we shall turn each constraints inequality to equation and set each variable equal to zero (0) to obtain two (2) coordinate points for each equation (i.e., using double intercept form). Having obtained all the coordinate points, we shall determine the range of our variables which enables us to know the appropriate scale to use for our graph. Thereafter, we shall draw the graph and join all the coordinate points with required straight line.

$2x + 5y = 25$ [Constraint 1]

When $x = 0, y = 5$ and when$y = 0, x = 12.5$.

$6x + 5y = 45$ [Constraint 2]

When $x = 0, y = 9$ and

when $y = 0, x = 7.5$.

Minimum value of x is $x = 0$.

Maximum value of x is $x = 12.5$.

Range of x is $0 \le x \le 12.5$.

Minimum value of y is $y = 0$.

Maximum value of y is $y = 9$.

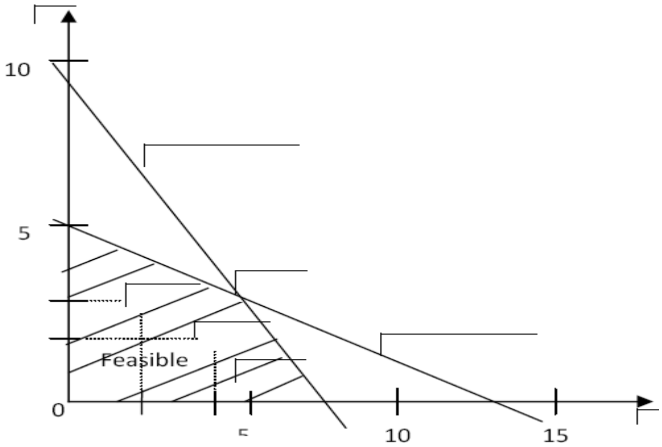


Figure 2.7 the feasible region for Example 2.11.1

The constraints give a set of feasible solutions as graphed above. To solve the linear programming problem, we must now find the feasible solution that makes the objective function as large as possible. Some possible solutions are listed below:

Table 2.1: A point of Feasible solutions and the value of Objective function at every point for Example 2.11.1

Feasible solutions (A point in the solution set of the system)	Objective function Z= 4x + 5y
(2,3)	4(2) +5(3) = 8 + 15 = 23
(4,2)	4(4) +5(2) = 16 + 10 = 26
(5, 1)	4(5) +5(1) = 20 + 5 = 25
(7, 0)	4(7) +5(0) = 28 + 0 = 28
(0, 5)	4(0) +5(5) = 0 + 25 = 25

In this list, the point that makes the objective function the largest is (7,0). But is this the largest for all feasible solutions? How about (6,1)? or (5,3)? It turns out that (5,3) provide the maximum value: $4(5)+5(3)=20+15=35$.

Hence, maximum profit at point (5,3) and it is the objective functions which have optimal values [36]

Example 2.11.2

The following linear programming problem:

Minimize: $z = 60x + 30y$

Subject to:

$$2x + 3y \geq 120$$
$$2x + y \geq 80$$
$$x \geq 0, y \geq 0.$$

Solution:

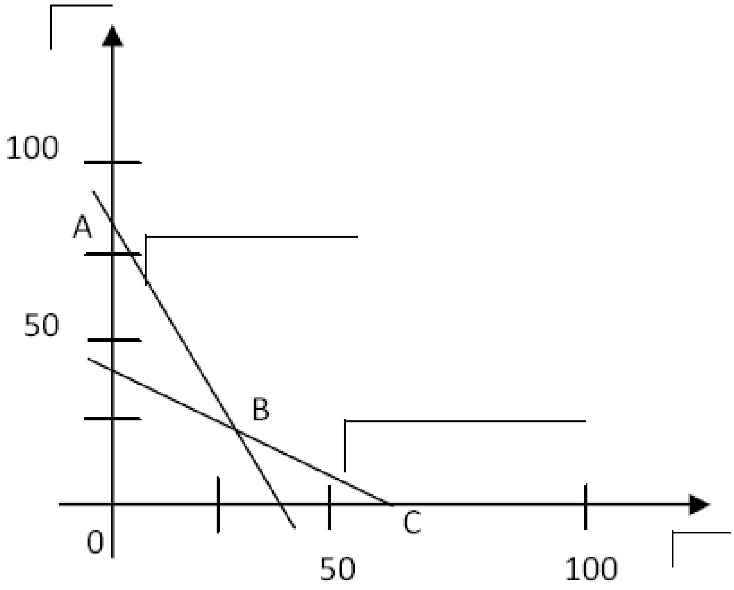


Figure 2.8 The feasible region and intersection point of for Example 2.11.2

Corner points *A* = (0,80) and *C* = (60,0) are found by inspection point B:

System: $2x + 3y = 120 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$

$$2x + y = 80 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$(1) - (2) = 2y = 40$$

$$y = 20$$

Substitute for y = 20 in (2):

$$2x + 20 = 80.$$

$$2x = 6.$$

$$x = 30.$$

$$Point\ B: (30,20).$$

Table 2.2: Extreme Values for Example 2.11.2

Corner point	Objective function Z = 60x + 30 y
(0, 80)	60(0) + 30(80) = 2400
(30, 20)	60(30) + 30(20) = 2400
(60, 0)	60(60) + 30(0) = 3600

From the table above, there are two minimum values for the objective function: $A = (0,80)$ and $B = (30,20)$. In this situation, the objective function will have the same minimum value (2,400) at all points along the boundary line segment A and B [36].

2.12 Algebraic Methods for Solving Linear Mathematical Programming Problems

2.12.1 Simplex method

Simplex method is the method to solve (LPP) models which contain two or more decision variables.

Basic variables: the variables which coefficients one in the equations and zero in the other equations.

Non-Basic variables: the variables which coefficients are taking any of the values, whether positive or negative or zero.

Slack, surplus & artificial variables:

a) If the inequality be (less than or equal, then we add a slack variable + S to change $\leq$ to =.

b) If the inequality be (greater than or equal, then we subtract a surplus variable - S to change $\geq$ to =.

c) If we have = we use artificial variables.

The steps of the simplex method:

Step 1: Determine a starting basic feasible solution.

Step 2: Select an entering variable using the optimality condition. Stop if there is no entering variable.

Step 3: Select a leaving variable using the feasibility condition [37].

2.12.1.1 Optimality condition:

The entering variable in a maximization (minimization) problem is the non-basic variable having the most negative (positive) coefficient in the Z-row.

The optimum is reached at the iteration where all the Z-row coefficient of the non-basic variables are non-negative (non-positive) [37].

2.12.1.2 Feasibility condition:

For both maximization and minimization problems the leaving variable is the basic associated with the smallest non-negative ratio (with strictly positive denominator) [37].

2.12.1.3 Pivot row:

a) Replace the leaving variable in the basic column with the entering variable.

b) new pivot row equal to current pivot row divided by pivot element.

c) All other rows:

New row=current row - (pivot column coefficient) * new pivot row [37].

Example 2.12.1 Extract the optimal solution to the problem of linear programming.

$$\text{Max } Z = 10x_1 + 8x_2$$

s.t

$$\begin{gathered} 2x_1 + 4x_2 \leq 36 \\ 4x_1 + 2x_2 \leq 48 \\ x_1, x_2 \geq 0 \end{gathered}$$

$$\text{Max } Z = 10x_1 + 8x_2 + 0S_1 + 0S_2$$

s.t

$$\begin{gathered} 2x_1 + 4x_2 + S_1 = 36 \\ 4x_1 + 2x_2 + S_2 = 48 \\ x_1, x_2, S_1, S_2 \geq 0 \end{gathered}$$

$$\text{Max } Z - 10x_1 - 8x_2 - 0S_1 - 0S_2 = 0$$

s.t

$$\begin{gathered} 2x_1 + 4x_2 + S_1 = 36 \\ 4x_1 + 2x_2 + S_2 = 48 \\ x_1, x_2, S_1, S_2 \geq 0 \end{gathered}$$

Table 2.3: An initial value of Objective function and constraints for Example 2.12.1

	↓ x_1	x_2	S_1	S_2	R.H.S
S_1	2	4	1	0	36
← S_2	$\boxed{4}$	2	0	1	48
Z	-10	-8	0	0	0

Then we extract the input variable x_1 that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then we extract the output variable S_2, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then we find the commonality between the intrinsic and extrinsic variable is 4.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1)) (New interior variable elements)

And we continue until all the values of the table change and we get a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max and negative or zeros if they are min, then this means that we have reached the optimal solution, but if not, we return to the first step and so on until we reach to the optimal solution.

Table 2.4: New values of Objective function and constraints for Example 2.12.1

	x_1	$\downarrow x_2$	S_1	S_2	R.H.S
$\leftarrow S_1$	0	$\boxed{3}$	1	-0.5	12
x_1	1	0.5	0	0.25	12
Z	0	-3	0	2.5	120

Table 2.5: New values of Objective function and constraints and the optimal solution for Example 2.12.1

	x_1	x_2	S_1	S_2	R.H. S
x_2	0	1	0.33	-0.166	4
x_1	1	0	-0.165	0.333	10
Z	0	0	0.99	2.002	132

We get $x_1 = 10$, $x_2 = 4$, then the optimal solution $Z = 132$.

2.12.2 Big-M Method

Step 1. Express the problem in the standard from.

Step 2. Add non- negative artificial variable to the left side of each of the equations corresponding to constraints of the type $\geq$ or $=$. When artificial variables are added, it causes violation of the corresponding constraints. This difficulty is removed by introducing a condition which ensures that variables will be zero in the final solution (provided the solution of the problem exists).

On the other hand, if the problem does not have a solution, at least one of the artificial variables will appear in the final solution with positive value. This is achieved by assigning a very large price (per unit penalty) to these variables in the objective function. Such large will be designated by –M for maximization problems (+M for minimizing problem), where M>0.

Step 3- in the last, use the artificial variables for the starting solution and proceed with the usual simplex routine until the optimal solution is obtained [37].

Example 2.12.2 Extract the optimal solution to the problem of linear programming.

$$\text{Min } Z = 4x_1 + x_2$$

s.t

$$3x_1 + x_2 = 3$$
$$4x_1 + 3x_2 \geq 6$$
$$x_1 + 2x_2 \leq 40$$
$$x_1, x_2 \geq 0$$

$$\text{Min } Z = 4x_1 + x_2 + MR_1 + MR_2 + 0S_1 + 0S_2$$

s.t

$$3x_1 + x_2 + R_1 = 3$$
$$4x_1 + 3x_2 - S_1 + R_2 = 6$$
$$x_1 + 2x_2 + S_2 = 40$$
$$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0$$

Min $Z = 4x_1 + x_2 + M(3 - 3x_1 - x_2) + M(6 - 4x_1 - 3x_2 + S_1) + 0S_1 + 0S_2$

Min $Z = 4x_1 + x_2 + 3M - 3Mx_1 - Mx_2 + 6M - 4Mx_1 - 3Mx_2 + MS_1 + 0S_2$

Min $Z = (4 - 3M - 4M)x_1 + (1 - M - 3M)x_2 + 3M + 6M + MS_1 + 0S_2$

Min $Z = (4 - 7M)x_1 + (1 - 4M)x_2 + 9M + MS_1 + 0S_2$

Min $Z - (4 - 7M)x_1 - (1 - 4M)x_2 - MS_1 - 0S_2 = 9M$

s.t

$$3x_1 + x_2 + R_1 = 3$$
$$4x_1 + 3x_2 - S_1 + R_2 = 6$$
$$x_1 + 2x_2 + S_2 = 40$$
$$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0$$

Table 2.6: An initial value of Objective function and constraints for Example 2.12.2

	↓ x_1	x_2	S_1	S_2	R_1	R_2	R.H. S
← R_1	$\boxed{3}$	1	0	0	1	0	3
R_2	4	3	-1	0	0	1	6
S_2	1	2	0	1	0	0	40
Z	-4+7M	-1+4M	-M	0	0	0	9M

Table2.7: New values of Objective function and constraints for Example 2.12.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H. S
x_1	1	0	1/5	0	3/5	-1/5	3/5
x_2	0	1	-3/5	0	-4/5	3/5	6/5
S_2	0	0	1	1	1	-1	37
Z	0	0	1/5	0	8/5-M	1/5-M	18/5

Table2.8: New values of Objective function and constraints and the optimal solution for Example 2.12.2

	x_1	$\downarrow x_2$	S_1	S_2	R_1	R_2	R.H. S
x_1	1	1/3	0	0	1/3	0	1
$\leftarrow R_2$	0	$\boxed{5/3}$	-1	0	-4/3	1	2
S_2	0	5/3	0	1	-1/3	0	39
Z	0	1/3+5/3M	-M	0	3/4-7/3M	0	4+2M

$$x_1 = \frac{3}{5}\,, x_2 = \frac{6}{5}\ \,, S_1 = 0\,, S_2 = 37\,, R_1 = 0\,, R_2 - 0\,, Z = 18/5$$

2.12.3. Two Phase Method

The procedure of removing artificial variables is achieved in phase-I of the solution and phase-II is required to get an optimal solution. As the solution of LPP is calculated in two phases, it is known as Two-Phase Simplex Method.

Phase I. In this particular phase, the simplex method is applied to an exclusively constructed auxiliary linear programming problem leading to a final simplex table consisting of a basic feasible solution to the original problem.

Step 1. A lot a cost -1 to each artificial variable and a cost 0 to all the other variables in the objective function.

Step 2. Make the Auxiliary LPP in which the new objective function Z* is to be maximized subject to the specified set of constraints.

Step 3. Work out the auxiliary problem through simplex method until either of the following three possibilities do occur

i. Max Z* < 0 and at least one artificial vector seems in the optimum basis at a positive level ($\Delta_j \geq 0$). In this case, given problem does not have any feasible solution.

ii. Max Z* = 0 and at least one artificial vector seems in the optimum basis at a zero level. In this case one needs to proceed to phase-II.

iii. Max Z* = 0 and no one artificial vector seems in the optimum basis. In this case one also needs to proceed for phase-II.

Phase II - Now allocate the actual cost to the variables in the objective function and a zero cost to each artificial variable that seems in the basis at the zero level. This new objective function is at present maximized by simplex method subject to the given constraints.

Simplex method is practically applied to the modified simplex table achieved at the end of phase I, until an optimum basic feasible solution has been reached. The artificial variables which are non-basic at the finish of phase-I are removed [37].

Example 2.12.3 Extract the optimal solution to the problem of linear programming.

$$\max Z = 3x_1 - x_2$$

S.t

$$2x_1 + x_2 \geq$$
$$x_1 + 3x_2 \leq 2$$
$$x_2 \leq 4$$
$$x_1 \ ,x_2 \geq 0$$

Phase I

$$\max r = -R_1$$

s.t

$$2x_1 + x_2 - S_1 + R_1 = 2 \qquad \Longrightarrow \quad R_1 = 2 - 2x_1 - x_2 + S_1$$
$$x_1 + 3x_2 + S_2 = 2$$
$$x_2 + S_3 = 4$$
$$x_1, x_2, S_1, S_2, S_3, R_1 \geq 0$$
$$\max r = -2 + 2x_1 + x_2 - S_1$$
$$\max r - 2x_1 - x_2 + S_1 = -2$$

Table2.9: An initial value of the new Objective function and constraints for Example 2.12.3

	$\downarrow x_1$	x_2	S_1	S_2	S_3	R_1	R.H. S
$\leftarrow R_1$	$\boxed{2}$	1	-1	0	0	1	2
S_2	1	3	0	1	0	0	2
S_3	0	0	0	0	1	0	4
r	-2	-1	0	0	0	0	-2

Table 2.10: New values of the new Objective function and constraints and the value of $r = 0$ for Example 2.12.3

	x_1	x_2	S_1	S_2	S_3	R_1	R.H. S
x_1	1	1/2	-1/2	0	0	1/2	1
S_2	0	5/2	1/2	1	0	-1/2	1
S_3	0	0	0	0	1	0	4
r	0	0	0	0	0	1	0

note that the value of the $r = 0$, so move on to the next phase.

Phase II

Table2.11: Original objective function and constraints for Example 2.12.3

	x_1	x_2	S_1	S_2	S_3	R.H. S
x_1	1	1/2	-1/2	0	0	1
S_2	0	5/2	1/2	1	0	1
S_3	0	1	0	0	1	4
Z	-3	1	0	0	0	0

Table2.12: New values of objective function and constraints for Example 2.12.3

	x_1	x_2	$\downarrow S_1$	S_2	S_3	R.H. S
x_1	1	1/2	-1/2	0	0	1
$\leftarrow S_2$	0	5/2	$\boxed{1/2}$	1	0	1
S_3	0	1	0	0	1	4
Z	0	5/2	-3/2	0	0	3

Table2.13: New values of objective function and constraints and the optimal solution for Example 2.12.3

	x_1	x_2	S_1	S_2	S_3	R.H. S
x_1	1	3	0	1	0	2
S_1	0	5	1	2	0	2
S_2	0	1	0	0	1	4
Z	0	10	0	3	0	6

Now, note that every value in previous table is positive and zero then we get

$$\therefore\ x_1 = 2\,, x_2 = 0 \longrightarrow \max Z = 6$$

Chapter Three

The Optimized DL-Public Key Cryptosystems

The Optimized DL-Public Key Cryptosystems

3.1 Introduction

This chapter first discusses the discrete logarithm DL public key cryptosystem, Diffie-Hellman Key exchange and El-Gamal public key cryptosystem. Examples are given to explain these cryptosystems. modified public key cryptosystems are proposed using the optimized problem. optimized DL public key cryptosystems are discussed. security considerations on these optimized DL public key cryptosystems are determined.

3.2 The Optimized DL-Public Key Cryptosystems

This section proposes a new version of the DHKE and EPKC using the optimization problem, through some methods that are employed for this task.

3.2.1 The Optimized Diffie-Hellman Key Exchange Using Graphical Method

A new version of the DHKE is proposed in this work. This version depended on the new optimization problem that is formed based on the Optimized Discrete Logarithm Functions (ODLFs) which is given in the next definition.

Definition 3.2.1.1. (The Optimized Discrete Logarithm Function). Let g be a generator element of a prime field F_p and let h be a nonzero element of F_p. The ODLF is a function to find the exponent $a_1x_1+ a_2x_2$ such that

$$\text{max (or min) } g^{a_1x_1+a_2x_2} \equiv h(mod\, p),$$

where $a_1x_1+ a_2x_2$ is a bivariate function with $a_1, a_2 \in F_p$ and the integers $x_1, x_2 \geq 0$.

The Optimized Diffie-Hellman Key Exchange (ODHKE) is explained as follows. With a public domain parameter, which are a large prime p and a generator element g of a prime field F_p, Alice and Bob choose their secret keys $a = a_1x_1 + a_2x_2$ and $b = b_1x_1 + b_2x_2$ respectively. They use their shared secret constraints that are computed using the original DHKE and explained in section (2.8.1), which are given by

$$\begin{aligned}
&c_{11}x_1 + c_{12}x_2 \le s_1\\
&c_{21}x_2 + c_{22}x_2 \le s_2\\
&\vdots\\
&c_{1n}x_1 + c_{1n}x_2 \le s_n\\
&x_1, x_2 \ge 0.
\end{aligned}$$

The ODLFs here are applied the graphical method. Alice and Bob used their discrete logarithm objective functions

$max\ or\ min\, A \equiv g^{a_1x_1+a_2x_2}(mod\, p)$ and $max\ or\ min\, B \equiv g^{b_1x_1+b_2x_2}(mod\, p),$ (3,8)

Respectively with the same shared secret constraint $c_{11}x_1 + c_{12}x_2 = s_1$, one can find two points P_1 and P_2 that are computed by supposing $x_1 = 0$ to get $x_2 = s_1/c_{12} = t_1$ and if $x_2 = 0$ then $x_1 = s_1/c_{11} = t_2$, so the points $P_1 = (0, t_1)$ and $P_2 = (t_2, 0)$ respectively. Similar computations are done with second, third constraints and so on. There is a line L_1 passes through the points P_1 and P_2 that are resulted from the first constraint. Also, a line L_2 passes through two points resulted based on the second constraint. Similarly, there are lines pass through all other points that are resulted from other constraints. The intersection between these lines determines the region of the feasible solutions as shown in Figure (3.1). The intersection point D is computed. Based on the points that determines the shadow region which is a region of the feasible solutions, Alice and Bob can compute their public keys A and B. In other words, using the points $P_{1,}$ D, P_2 and C, Alice can compute the value of her public key A that are corresponded to these points. The optimal value is determined based on the determination of the objective function, maximum or minimum. Bob, in similar way, using the same points of the feasible region to compute the values of his public key *B*. The optimal value of *B* is determined. After the computations of the keys *A* and *B*, Alice sends *A* to Bob and Bob sends *B* to Alice.

Now, Alice computes A' as her shared secret key by $A^{'} \equiv B^{a_1x_1+a_2x_2}(mod\, p)$. In similar way, Bob computes his shared secret key by $B^{'} \equiv A^{b_1x_1+b_2x_2}(mod\, p)$. So, the shared secret key between Alice and Bob is

$$A^{'}(mod\, p) \equiv B^{'}(mod\, p).$$

$$A' \equiv B^{a_1x_1+a_2x_2} \equiv (g^{b_1x_1+b_2x_2})^{a_1x_1+a_2x_2} \equiv (g^{a_1x_1+a_2x_2})^{b_1x_1+b_2x_2}$$
$$\equiv A^{b_1x_1+b_2x_2} \equiv B'$$

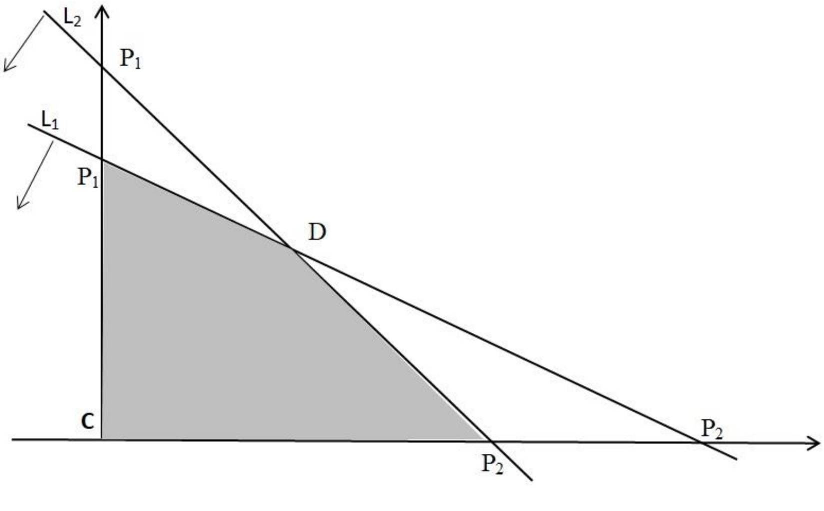


Figure 3.1. The region of the feasible solutions based on intersection point *D*.

Example 3.2.1.1 Let $p = 71$ be a prime number. Suppose $g = 7$ is a generator element over *F*17. Alice chooses her secret key $a = 5x_1 + 7x_2$ and Bob chooses his secret key $b = 3x_1 + x_2$ and they use their shared secret constrents that are given by

$$\begin{aligned} 2x_1 + 3x_2 &\leq 18\,, \\ x_1 + \ x_2 &\leq 8\,, \\ x_2 &\leq 4\,, \\ x_1, x_2 &\geq 0. \end{aligned}$$

The proposed ODHKE using the graphical method is discussed as follows

Alice and Bob compute her and his public keys A and B by max $A = g^{5x_1+7x_2}$ (mod 71) and max $B = g^{3x_1+x_2}$ (mod 71) respectively with the same shared secret constraints.

Both of them use the first constraint to find two points, so if $x_2 = 0$ then $2x_1 + 3x_2 = 18$ results $x_1 = 9$, so $P_1 = (9,0)$. And if $x_1 = 0$ then $x_2 = 6$, so $P_2 = (0,6)$

With the second constraint $x_1 + x_2 = 8$, if $x_1 = 0$ then $x_2 = 8$ and $P_1 = (0,8)$ and if $x_2 = 0$ then $x_1 = 8$, so $P_2 = (8,0)$.

Using two resulted points from first constraint, there is a line L_1 cross through them. In similar way, a line L_2 passes through other two points (0,8) and (8,0). And a line L_3 passes through a point (0,4) that is resulted from a third constraint $x_2 \leq 4$ with $x_1 = 0$. The intersection between these lines determines the region of the feasible solutions as shown in Figure (3.2).

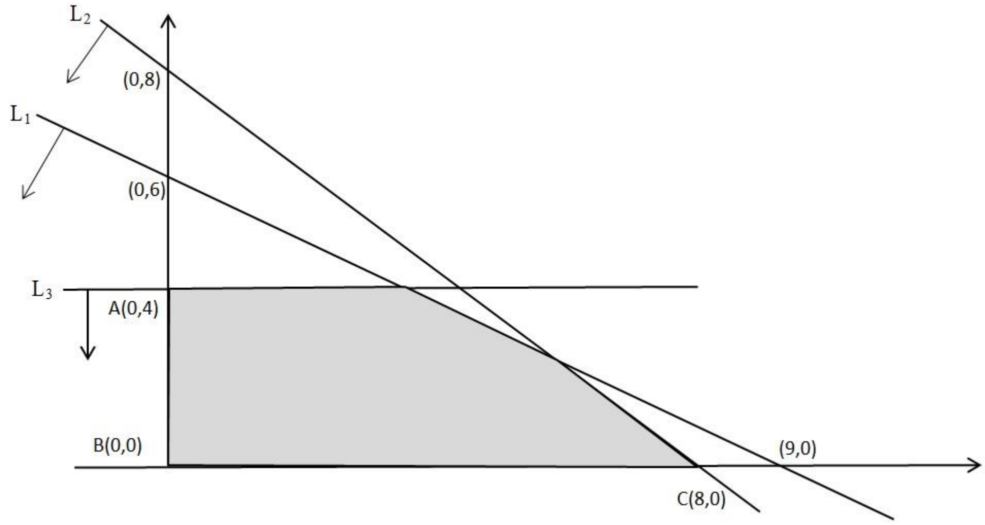


Figure 3.2. The region of the feasible solutions of three lines L$_1$, L$_2$ and L$_3$.

The intersection point D is computed by

$$\begin{array}{c} 2x_1 + 3x_2 = 18 \\ \left. x_1 + x_2 = 8 \right\} * (-2) \\ \hline \end{array}$$

$$\begin{array}{c} 2x_1 + 3x_2 = 18 \\ -2x_1 \mp 2x_2 = -16 \\ \hline \end{array}$$

$$x_2 = 2.$$

So, $x_1 = 6$. Thus, the intersection points $D = (6,2)$ as shown in Figure (3.3). Another intersection point *E* should be computed.

This computing is done as follows. Based on the substitution of the third constraint in first one, namely $2x_1 + 3(4) = 18$, so, *x*1=3. The second intersection point $E = (3,4)$, as shown in Figure (3,3).

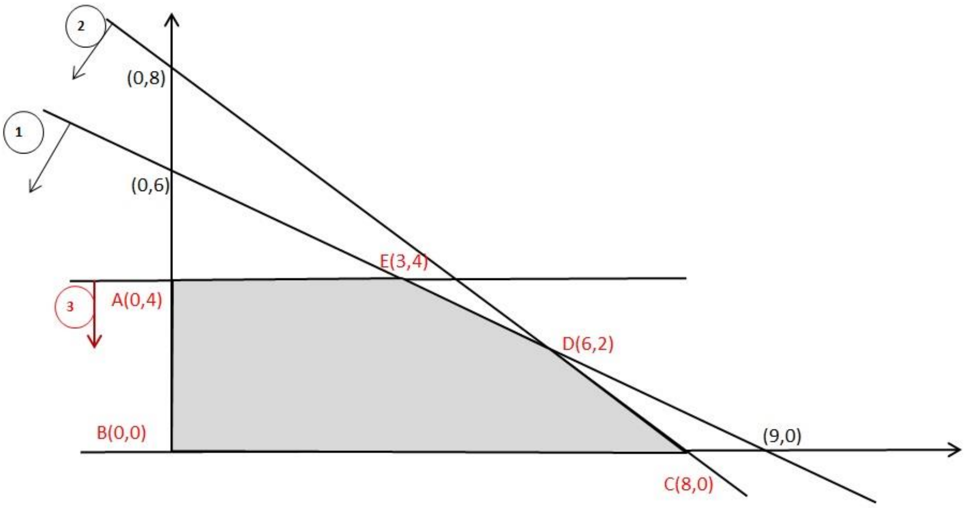


Figure 3.3**.** The region of the feasible solutions based on intersection points

D = (6,2) and *E* = (3,4).

Based on the points that determines the region of feasible solutions, Alice and Bob can compute their public keys *A* and *B* as shown in Table (3.1).

Table 3.1. The points of the optimal solution of Alice and Bob for Example 3.2.1.1

point	$A \equiv 7^{5x_1+7x_2}$ (Mod 71)	point	$B \equiv 7^{3x_1+x_2}$ (mod 71)
$A = (0,4)$	5	$A = (0,4)$	58 → optimal values
$B = (0,0)$	1	$B = (0,0)$	1
$C = (8,0)$	20	$C = (8,0)$	16
$D = (6,2)$	24	$D = (6,2)$	37
$E = (3,4)$	44 → optimal values	$E = (3,4)$	28

So, $A \equiv 44$ (mod 71) with a point (x = 3, y = 4) and $B \equiv 58$ (mod 71) with a point (x = 0 , y = 4). Now Alice sends A to Bob and Bob sends B to Alice. After that, Alice computes her shared secret key A' by

$$\begin{aligned}
A' &\equiv (B)^{5x_1+7x_2} && (mod\ 71) \\
&\equiv (58)^{5(3)+7(4)} && (mod\ 71) \\
&\equiv (58)^{43} && (\,mod\ 71) \\
&\equiv (58)(58)^{42} && (mod\ 71) \\
&\equiv (58)(58^{6})^{7} && (mod\ 71) \\
&\equiv (58)(16)^{7} && (mod\ 71) \\
&\equiv (58)(5) && (mod\ 71) \\
&\equiv 6 && (mod\ 71).
\end{aligned}$$

Whereas Bob computes his shared secret key B' by

$$\begin{aligned}
B' &\equiv (A)^{3x_1+x_2} && (mod\ 71) \\
&\equiv (44)^{3(0)+4} && (mod\ 71) \\
&\equiv (44)^{4} && (\,mod\ 71) \\
&\equiv 6 && (mod\ 71).
\end{aligned}$$

So, the shared secret Key between Alice and Bob is

$$A' \equiv B' \equiv 6 \quad (mod\ 71)$$

Example 3.2.1.2 Let $p = 47$ be a prime number. Suppose $g = 3$ is a generator element over F_{47}. Alice chooses her secret key $a = 10x_1+8x_2$, and Bob chooses his secret key b= $7x_1$+$3x_2$ and they use their shared secret constraints that are given by

$$\begin{gathered} 6x_1 + 2x_2 \geq 12 \\ 2x_1 + 2x_2 \geq 8 \\ 6x_1 + 4x_2 \geq 18 \\ x_1, x_2 \quad \geq 0 \end{gathered}$$

Namely,

Alice

$Min\ A = g^{10x_1+8x_2}$ mod 47

$$\begin{aligned} S.t \quad 6x_1 + 2x_2 &\geq 12 \\ 2x_1 + 2x_2 &\geq 8 \\ 6x_1 + 4x_2 &\geq 18 \\ x_1,\ x_2 &\geq\ 0 \end{aligned}$$

Bob

$Min\ B = g^{7x_1+3x_2}$ mod 47

$$\begin{aligned} S.t \quad 6x_1 + 2x_2 &\geq 12 \\ 2x_1 + 2x_2 &\geq 8 \\ 6x_1 + 4x_2 &\geq 18 \\ x_1,\ x_2 &\geq\ 0 \end{aligned}$$

For the first constraint

$6x_1 + 2x_2 = 12 \rightarrow x_1 = 0 \rightarrow x_2 = 6,$ so, the first point is (0,6) and

$\rightarrow x_2 = 0 \rightarrow x_1 = 2,$ the second point is (2.0).

For the second constraint

$2x_1 + 2x_2 = 8 \rightarrow x_1 = 0 \rightarrow x_2 = 4$,so, the first point is (0,4) and

$\rightarrow x_2 = 0 \rightarrow x_1 = 4,$ the second point is (4,0).

Now, with the third constraint

$6x_1 + 4x_2 = 18 \rightarrow x_1 = 0 \rightarrow x_2 = 4.5,$ so, the first point (0.4,5) and

$\rightarrow x_2 = 0 \rightarrow x_1 = 3,$ the second point is (3,0).

Based on these points, the region of feasible solutions is determined as shown in Figure (3.4).

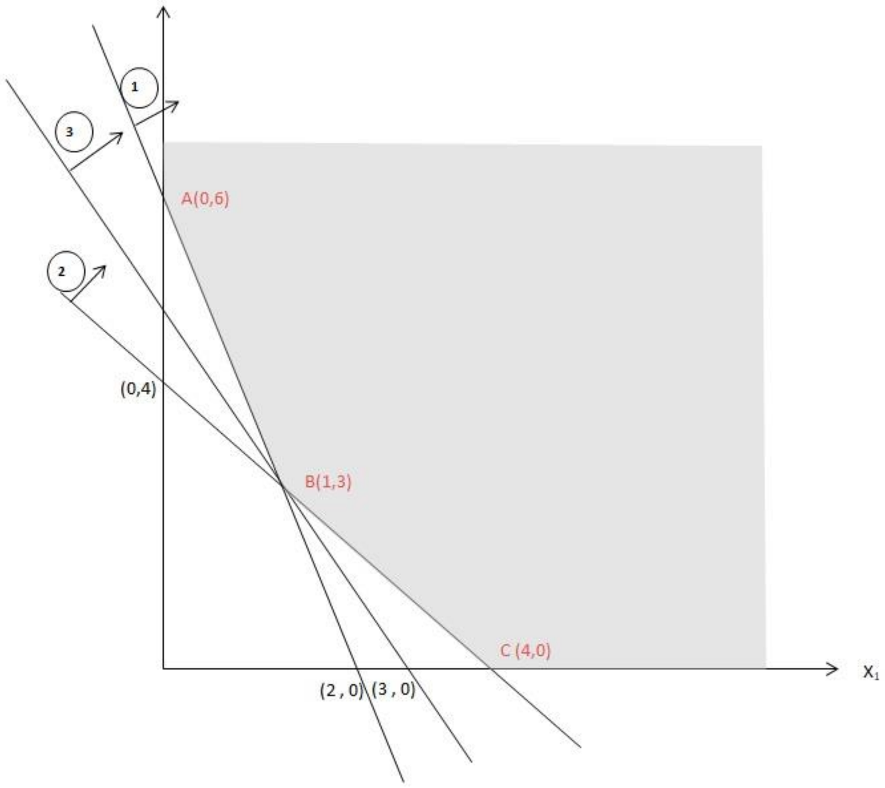


Figure 3.4. Feasible solutions and intersection point $B = (1,3)$

In figure (3.4), any two constraints are chosen from three intersection constraints to get point *B*. The computation of a point *B* is done by

$$\begin{array}{c} 6x_1 + 2x_2 = 12 \quad \rightarrow c_1 \\ \underline{\mp 6x_1 \mp 4x_2 = \mp 18 \rightarrow c_2} \end{array}$$

$$x_2 = 3 \rightarrow x_1 = 1 \ \rightarrow B(1,3).$$

Alice and Bob can compute their public keys *A* and *B* as given in Table (3.2).

Table 3.2. The points of optimal solution for Alice and Bob for Example 3.2.1.2

point	$\min A \equiv 3^{10x_1+8x_2}$ (Mod 47)	point	$\min B \equiv 7^{3x_1+x_2}$ (Mod 47)
A=(0,6)	9	*A*= (0,6)	6 → optimal values
B=(1,3)	4	*B*= (1,3)	32
C=(4,0)	2 → optimal values	*C*= (4,0)	8

So, $A \equiv 2$ (mod 47) with a point (*x* = 4, *y* = 0) and $B \equiv 6$ (mod 47) with a point (*x* = 0,*y* = 6).

Alice sends *A* to Bob and Bob sends *B* to Alice.

Now, Alice computes

$$\begin{aligned} A' &\equiv (B)^{10x_1+8x_2} && (mod\ \ 47) \\ &\equiv (6)^{40} && (mod\ \ 47) \\ &\equiv (6^4)^{10} && (\,mod\ 47) \\ &\equiv 25 && (mod\ 47), \end{aligned}$$

And Bob computes

$$\begin{aligned} B' &\equiv (A)^{7x_1+3x_2} && (mod\ \ 47) \\ &\equiv (2)^{7x_1+3x_2} && (mod\ \ 47) \\ &\equiv (2)^{18} && (\,mod\ 47) \\ &\equiv 25 && (mod\ 47). \end{aligned}$$

So, $A' \equiv B' \equiv 25\ \ (mod\ 47).$

3.2.2 The Optimized El-Gamal Public Key Cryptosystem

A new version of the EPKC is proposed in this work. This version depends on the new optimization problem that is formed based on the ODLF.

The optimized El-Gamal puplic key cryptosystem (OEPKC) is explained as follows. With a public domain parameters, which are a large prime p and a generator element g of a prime field F_p, Alice chooses her secret key $a = a_1 x_1 + a_2 x_2$. She uses shared secret constraints that is computed using the original (EPKC) which are given by

$$\begin{gathered} c_{11}x_1 + c_{12}x_2 \le s_1 \\ c_{21}x_2 + c_{22}x_2 \le s_2 \\ \vdots \\ c_{1n}x_1 + c_{1n}x_2 \le s_n \\ x_1, x_2 \ge 0. \end{gathered}$$

The ODLF here are applied the graphical method to do our proposition. Alice used her (ODLFs)

$$max\, or(\, min)\, A \equiv g^{a=a_1x_1+a_2x_2}(mod\, p).$$

With a first shared secret constraint $c_{11}x_1 + c_{12}x_2 = s_1$, two points P_1 and P_2 are computed by supposing $x_1 = 0$ to get $x_2 = s_1/c_{12} = t_1$ and if $x_2 = 0$ then $x_1 = s_1/c_{11} = t_2$, so the points P_1=(0, t_1) and P_2= (t_2, 0) respectively. Similar computations are done with second, third constraints and so on. There is a line L_1 passes through the points P_1 and P_2 that are resulted from the first constraint. Also, a line L_2 passes through resulted two points based on the second constraint. Similarly, there are lines pass through all other points that are resulted from other constraints. The intersection between these lines determines the region of the feasible solutions as shown in Figure (3.1). The intersection point D is computed. Based on the points that determines the shadow region which is a region of the feasible solutions, Alice can compute her public key A. In other words, using the points P_1 , D , P_2 and C, Alice can compute the value of her public key A that are corresponded to these points. The optimal value is determined based on the determination of the objective function.

Now, Bob wants to encrypt a plaintext M and send it to Alice. So, he select M from the range [2,p-1]. Also, he selects an integer $k = bx_1 + bx_2 \ni k \text{ such that} [1, p-1]$. He computes C_1 with a first shared secret constraint $c_{11}x_1 + c_{12}x_2 = s_1$ through determining two points P_1 and P_2 that are computed by supposing $x_1 = 0$ to get $x_2 = s_1/c_{12} = t_1$ and $x_2 = 0$ then $x_1 = s_1/c_{11} = t_2$, so the points P_1= (0, t_1) and P_2 = (t_2, 0) respectively. Similar computations are done with second, third constraints and so on. There is a line L_1 pass through the points P_1 and P_2 that are resulted from the first constraint. Also, a line L_2 pass through two points resulted based on the second constraint. Similarly, there are lines pass through all other points that are resulted from other constraints. The intersection between these lines determines the region of the feasible solutions. The intersection between these lines determines the region of the feasible solutions as shown in Figure (3.1). The intersection point D is computed. Based on the points that determines the shadow region which is a region of the feasible solutions, Bob can compute value of C_1. In other words, using the points P_1 , D , P_2 and C, Bob can compute the value of C_1 that are corresponded to these points. The optimal value is determined based on the determination of the objective function. Bob uses Alice's public key A to compute C_2, A pair of the ciphertext is computed by (C_1, C_2)

$$\text{max or (min) } C_1 \equiv g^{k=b_1x_1+b_2x_2} \pmod p$$

and

$$max \text{ or } (min)\ C_2 \equiv M(max\ or\ min\ A)^{k=b_1x_1+b_2x_2}(mod\ p).$$

sends the pair of the ciphertext (C_1, C_2) to Alice.

Upon Alice receiving the ciphertext (C_1, C_2), some steps have been calculated by her to recover the plaintext M. She first computes the value X through the following relation. $X \equiv C_1^{a_1x_1+a_2x_2} \pmod p$

Also, she computes the invers value $X^{-1} \pmod p$ of X . One can use Extended Euclidean algorithm (EEA), as shown in section (2.5) in Chapter (2), for computing the inverse element modulo p. Finally, she computes the relation $X^{-1} * C_2 \equiv M \pmod p$ to recover a plaintext M.

$$\begin{aligned} C_2 * (C_1^a)^{-1} &\equiv M(\text{max}\, or \text{ min}\, A)^{k=b_1x_1+b_2x_2} * ((g^{k=b_1x_1+b_2x_2})^{a=a_1x_1+a_2x_2})^{-1} \\ &\equiv M(g^{a=a_1x_1+a_2x_2})^{k=b_1x_1+b_2x_2} * (g^{a=a_1x_1+a_2x_2})^{k=b_1x_1+b_2x_2})^{-1} \\ &\equiv M \end{aligned}$$

Example 3.3.2.1 Let p = 29 be a prime number. Suppose g = 5 is a generator element over F_{29}. Alice chooses her secret key $a = 3x_1 + 4x_2$. Bob selects his plaintext M =9. And he chooses the ephemeral key $k = x_1 + x_2$. And they used a private share secret constrant

$$x_1 + x_2 \leq 4,$$
$$x_1 + 3x_2 \leq 6,$$
$$x_1, x_2 \geq 0.$$

The proposed optimized El-Gamal puplic key cryptosystem using the graphical method is discussed as follows. Alice computes her public key *A* through using the ODLF

$$\max A \equiv g^{5x_1+7x_2} \pmod{29}.$$

With the same shared secret constraints. She uses the first constraint to find two points, so if $x_1 = 0$ then $x_1+x_2 = 4$ results $x_2 = 4$, so $P_1 =(0,4)$ and if $x_2 = 0$ then $x_1= 4$, therefore a resulted point is $P_2 = (4,0)$. Now, with the second constraint $x_1+ 3x_2 = 6$, if $x_1 = 0$ then $x_2 = 2$, therefor $P_1 = (0,2)$ and if $x_2 = 0$ then $x_1 = 6$, therefore, $P_2 = (6,0)$. With two resulted points from first constraint, there is a line L_1 crosses through them. In similar way, a line L_2 passes through other two points (0,2) and (6,0). The intersection between these lines determines the region of the feasible solutions as shown in Figure (3.5).

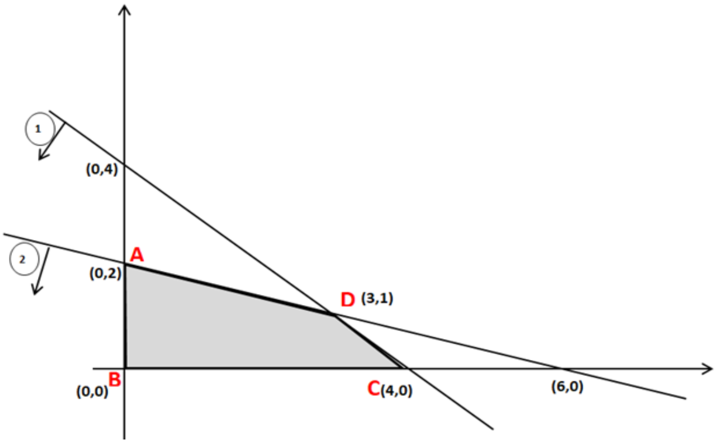


Figure 3.5. The region of the feasible solutions based on intersection points *D*=(3,1).

The intersection point *D* is computed by

$$\left.\begin{array}{r} x_1 + x_2 = 4 \\ x_1 + 3x_2 = 6 \end{array}\right\} = \left.\begin{array}{r} x_1 + x_2 = 4 \\ -x_1 \mp 3x_2 = -6 \end{array}\right\} \text{so, } x_2 = 1 \text{ and } x_1 = 3.$$

Thus, the intersection point D = (3,1) as shown in Figure (3.5). Alice can compute her public key A as shown in Table (1). So, $A \equiv 20 \pmod{29}$ with a point (0,2).

Table 3.3. The point of the optimal solution of Alice for Example 3.3.2.1

point	$A \equiv g^{3x_1+5x_2}\ (mod\,2\,9)$
A=(0,2)	20 → Optimal value
B=(0,0)	1
C=(4,0)	7
D=(3,1)	0

Now for encryption process, Bob selects his plaintext M =9. And he chooses the ephemeral key $k = x_1 + x_2$. He used shared secret constraints. He computes C_1 that given by

$$\max C_1 \equiv 5^{x_1+x_2} \pmod{29}.$$

With the same shared secret constraint. He uses the first constraint to find two points, so if x_1 = 0 then x_1+x_2 = 4 results x_2 = 4, so P_1 =(0,4) and if x_2 = 0 then x_1= 4, therefore a resulted point is P_2 = (4,0). Now, with the second constraint x_1+ 3x_2 = 6, if x_1 = 0 then x_2 = 2, therefor P_1 = (0,2) and if x_2 = 0 then x_1 = 6, therefore, P_2 = (6,0). With two resulted points from first constraint, there is a line L_1 crosses through them. In similar way, a line L_2 passes through other two points (0,2) and (6,0). The intersection between these lines determines the region of the feasible solutions as shown in Table (3.4). So, $C_1 \equiv 25 \pmod{29}$ by the optimal point (0,2).

Table 3.4 The point of the optimal solution of C_1 for Example 3.3.2.1

Point	$C_1 \equiv g^{x_1+x_2}$ (mod 29)
A=(0,2)	25 → optimal value
B=(0,0)	1
C=(4,0)	16
D=(3,1)	16

And he computes

$$C_2 \equiv M \times (max\, A)^{x_1+x_2} (mod\, 2\, 9)$$

$$\equiv 9.(20)^2 \qquad (mod\ 29)$$

$$\equiv 4 \qquad (mod\ 29)$$

$$(C_1, C_2) \to (25{,}4)$$

Then Bob sends (C_1, C_2) to Alice.

When Alice receives code (C_1, C_2) and she wants to decrypt it and recover the original plaintext M by

$$M \equiv (C_1^a)^{-1}(C_2) \qquad (mod\ 29)$$

$$\equiv (25^{3x_1+5x_2})^{-1}(4) \quad (mod\, 2\, 9)$$

$$\equiv (25^{10})^{-1}(4) \qquad (mod\, 2\, 9)$$

$$\equiv (23)^{-1}(4) \qquad (mod\, 2\, 9)$$

$$\equiv 24(4) \qquad (mod\, 2\, 9)$$

$$\equiv 9 \qquad (mod\, 2\, 9).$$

Example 3.3.2.2 Let p = 71 be a prime number. Suppose $g = 7$ is a generator element over F_{71}. Alice chooses her secret key $a = 2x + 5y$. Bob selects his plaintext M = 20. And he chooses the ephemeral key $k = x + 3y$ such that $k \in [1, 70]$. And they used a private share secret constrant

$$x + 2y \geq 10$$

$$3x + 4y \leq 24$$

$$x, y \geq 0$$

Alice computes her public key A through using the ODLF

$$\text{Min} \quad A \equiv g^{2x+5y} \,(\text{mod}\, 71)$$

$$\text{s.t}$$

$$x + 2y \geq 10$$

$$3x + 4y \leq 24$$

$$x, y \geq 0$$

For the first constraint

$$x + 2y = 10 \rightarrow x = 0 \rightarrow y = 5, \text{ so, the first point is } (0,5)$$

$$\text{and} \quad \rightarrow y = o \rightarrow x = 10, \text{ the second point is } (10,0).$$

for the second constraint

$$3x + 4y = 24 \rightarrow x = 0 \rightarrow y = 6, \text{ so, the first point is } (0,6)$$

$$\text{and} \quad \rightarrow y = o \rightarrow x = 8, \text{ the second point is } (8,0)$$

Basd on these points, the region of feasible solution is determind as shown in figure (3.6).

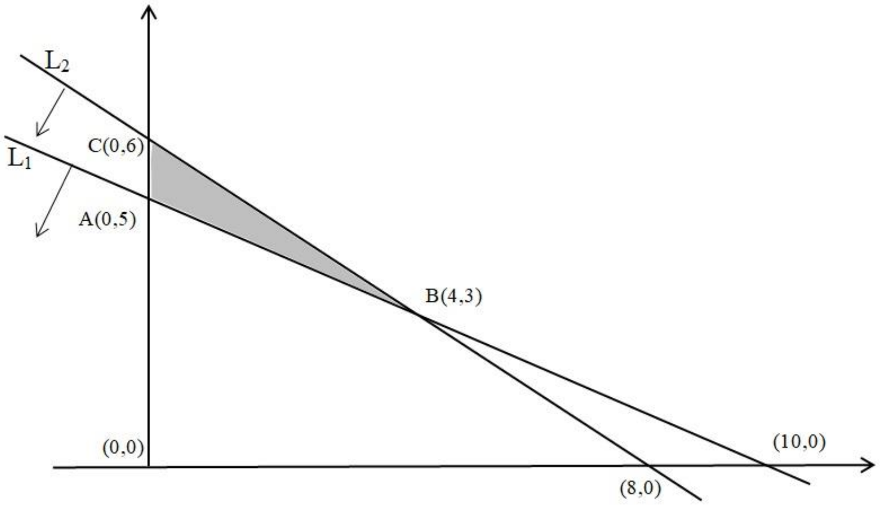


Figure 3.6. Feasible solution and intersection point $p = (4,3)$.

In figure (3.4), from intersection two constraints to get point *B*. The computation of a point *B* is done by to find intersection point *B*

$$\begin{array}{l} x + 2y = 10] \times 2 \\ \underline{3x + 4y = 24} \end{array}$$

$$\begin{array}{l} 2x + 4y = 20 \\ \underline{\mp 3x \mp 4y = \mp 24} \end{array}$$

$$\begin{array}{l} -x = -4 \\ x = 4 \rightarrow y = 3 \rightarrow B(4,3) \end{array}$$

Alice can compute her public keys *A* as given in Table (3.5).

Table 3.5. The points of optimal solution for Alice for Example 3.3.2.2

points	$A = 7^{2x+5y} \bmod 71$
A (0,5)	45
B (4,3)	53
C (0,6)	32 → *optimal*

So, min $A \equiv 32 \pmod{71}$ with a point ($x = 0$, $y = 6$)

Now for encryption process, Bob compute C_1 that given by

$$\text{Min} \quad C_1 \equiv 7^{x+3y} \bmod 71$$

S.t

$$x + 2y \geq 10$$
$$3x + 4y \leq 24$$
$$x, y \geq 0$$

For the first constraints

$$x + 2y = 10 \rightarrow x = 0 \rightarrow y = 5, \text{ so, the first point is } (0,5)$$

$$\text{and} \quad \rightarrow y = o \rightarrow x = 10, \text{ the second point is } (10,0).$$

for the second constraint

$$3x + 4y = 24 \rightarrow x = 0 \rightarrow y = 6, \text{ so, the first point is } (0,6)$$

$$\text{and} \quad \rightarrow y = o \rightarrow x = 8, \text{ the second point is } (8,0)$$

Bob can compute his public keys *B* as given in Table (3.6).

Table 3.6. The points of optimal solution for Bob for Example 3.3.2.2

points	$C_1 = 7^{x+3y} \bmod 71$
A (0,5)	23
B (4,3)	14
C (0,6)	8 → *optimal*

So, min $C_1 \equiv 8$ (mod 71) with a point ($x = 0, y = 6$)

And he computes

$$\begin{aligned} C_2 &\equiv M(min\,A)^k(mod\,p) \\ &\equiv 20(32)^{x+3y}(mod\,7\,1) \\ &\equiv 20(32)^{18}(mod\,7\,1) \\ &\equiv 20(48)(mod\,7\,1) \\ &\equiv 37 \qquad (mod\,7\,1) \\ &(C_1, C_2)\text{is}(8{,}37) \end{aligned}$$

Then Bob sends (C_1, C_2) to Alice

When Alice receives code (C_1, C_2) and she wants to decrypt it and recover the original plaintext M by

$$\begin{aligned} M &\equiv (C_1^a)^{-1}C_2 && (\text{mod}\,p) \\ &\equiv (8^{2x+5y})^{-1}(37) && (\text{mod}\,71) \\ &\equiv (8^{30})^{-1}(37) && (\text{mod}\,71) \\ &\equiv (48)^{-1}(37) && (\text{mod}\,71) \\ &\equiv 37(37) && (\text{mod}\,71) \\ &\equiv 20 && (\text{mod}\,71) \end{aligned}$$

3.2.3 The Optimized Diffie-Hellman Key Exchange Using the Simplex Method

This section explains the ODHKE using the simplex method. But before explaining that, another definition of the optimized discrete logarithm functions (ODLFs) is proposed as follows.

Definition 3.3.3.1. (The Optimized Discrete Logarithm Function). Let g be a generator element of a prime field F_p and let h be a nonzero element of F_p. The ODLF is a function to find the exponent *max* or *min* a_1x_1+ a_2x_2 such that

$$(g)^{max\ or\ min(a_1x_1+a_2x_2)} \equiv h(mod\,p),$$

where a_1x_1+ a_2x_2 is a bivariate function with $a_1, a_2 \in F_p$ and the integers $x_1, x_2 \geq 0$.

The Optimized Diffie-Hellman Key Exchange (ODHKE) using the simplex method is explained as follows. With a public domain parameter, which are a large prime p and a generator element g of a prime field F_p, Alice and Bob choose their secret keys $a = a_1x_1 + a_2x_2$ and $b = b_1x_1 + b_2x_2$ respectively. They use their shared secret constraints that are computed using the original DHKE and explained in section (2.8.1), which are given by

$$\begin{gathered} c_{11}x_1 + c_{12}x_2 \leq b_1 \\ c_{21}x_2 + c_{22}x_2 \leq b_2 \\ \vdots \\ c_{1n}x_1 + c_{1n}x_2 \leq b_n \\ x_1, x_2 \geq 0. \end{gathered}$$

The ODLFs here are applied the simplex method. Alice and Bob used their discrete logarithm objective functions

Each of them begins by converting the problem to the standard form (adding slackness variables):

$$A \equiv g^{\max\ or\ \min\ (a = a_1x_1 + a_2x_2 + 0S_n)} \pmod p$$

s.t

$$\begin{gathered} c_{11}x_1 + c_{12}x_2 + S_1 = b_1 \\ c_{21}x_2 + c_{22}x_2 + S_2 = b_2 \\ \vdots \\ c_{1n}x_1 + c_{1n}x_2 + S_n = b_n \\ x_1, x_2, S_1, S_2, \cdots, S_n \geq 0. \end{gathered}$$

$$B \equiv g^{max\ or\ min\ (b = b_1x_1 + b_2x_2 + 0S_n)} \pmod p$$

s.t

$$c_{11}x_1 + c_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 + S_2 = b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 + S_n = b_n$$
$$x_1, x_2, S_1, S_2, \cdots, S_n \geq 0.$$

Setting the objective function to zero

$$A \equiv g^{max\ or\ min\ (a - a_1x_1 - a_2x_2 - 0S_n = 0)} \pmod{p}$$
$$B \equiv g^{max\ or\ min\ (b - b_1x_1 - b_2x_2 - 0S_n = 0)} \pmod{p}$$

s.t

$$c_{11}x_1 + c_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 + S_2 = b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 + S_n = b_n$$
$$x_1, x_2, S_1, S_2, \cdots, S_n \geq 0.$$

Each of them will form an initial table for its own objective function and shared secret constraints

Table 3.7 An initial values of objective function and constraints for Alice for method 3.3.3

	x_1	x_2	S_1	S_2	…	S_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	b_1
S_2	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S_n	c_{n1}	c_{n2}	0	0	…	1	b_n
a	$-a_1$	$-a_2$	0	0	…	0	0

Table 3.8 An initial values of objective function and constraints for Bob for method 3.3.3

	x_1	x_2	S_1	S_2	…	S_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	b_1
S_2	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S_n	c_{n1}	c_{n2}	0	0	…	1	b_n
b	$-b_1$	$-b_2$	0	0	…	0	0

Then they extract the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then they extract the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then they find the commonality between the intrinsic and extrinsic variable.

All the elements of the external variable are divided by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1)) (New interior variable elements)

And they continue until all the values of the table change and get a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max and negative or zeros if they are min, then this means that they have reached the optimal solution, but if not, we return to the first step and so on until we reach to the optimal solution, Alice and Bob take a value (max or min a), (max or min b) and put it in their function to extract the declared key

$$A \equiv g^{\max or \min(a)} \quad (\bmod p)$$

$$B \equiv g^{\max or \min(b)} \quad (\bmod p)$$

Then the declared keys are exchanged, and they both compute the cipher using their secret key extracted from the simplex method

$$A' \equiv B^{\max or \min(a)} \quad (\bmod p)$$

$$B' \equiv A^{\max or \min(b)} \quad (\mathrm{mod} p)$$

$$A' \equiv B^{\max or \min a=a_1x_1+a_2x_2} \equiv (g^{\max or \min b=b_1x_1+b_2x_2})^{\max or \min a=a_1x_1+a_2x_2}$$

$$\equiv (g^{\max or \min a=a_1x_1+a_2x_2})^{\max or \min b=b_1x_1+b_2x_2}$$

$$\equiv A^{\max or \min b=b_1x_1+b_2x_2} \equiv B'$$

Share secret key between Alice and Bob $B'\ (mod\ p) \equiv A'(mod\ p)$

Example 3.3.3.1 Alice and Bob agree to use the prime p =23 and the generator element g =11 and Alice choose her secret key $a = 10x_1 + 6x_2$ and Bob chooses his secret key $b = 3x_1 + 5x_2$ and both of them uses the private share constraints that are

$$5x_1 + 3x_2 \leq 30$$
$$x_1 + 2x_2 \leq 18$$
$$x_1, x_2 \geq 0$$

Alice and Bob used their discrete logarithm objective functions

Alice	Bob
$A \equiv g^{\max a=10x_1+6x_2} (\mathrm{mod}\, 23)$	$B \equiv g^{\max b=3x_1+5x_2} (\mathrm{mod}\, 23)$
$S.to$	$S.to$
$5x_1 + 3x_2 \leq 30$	$5x_1 + 3x_2 \leq 30$
$x_1 + 2x_2 \leq 18$	$x_1 + 2x_2 \leq 18$
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$

Each of Alice and Bob begins by converting the problem to the standard form (adding slackness variables). And, setting the objective function to zero

$$A \equiv 11^{max\, a-10x_1-6x_2-0S_1-0S_2=0}(mod\ 2\ 3)$$

$S.t$

$5x_1 + 3x_2 + S_1 = 30$

$x_1 + 2x_2 + S_2 = 18$

$x_1, x_2, S_1, S_2 \geq 0$

$$B \equiv 11^{max\, b-3x_1-5x_2-0S_1-0S_2=0}(mod\, 2\, 3)$$

$S.t$

$5x_1 + 3x_2 + S_1 = 30$

$x_1 + 2x_2 + S_2 = 18$

$x_1, x_2, S_1, S_2 \geq 0$

Each one of them will form an initial table for its own objective function and shared secret constraints

Table 3.9 An initial values of objective function and constraints for Alice for Example 3.3.3.1

	↓x_1	x_2	S_1	S_2	R.H.S
←S_1	$\boxed{5}$	3	1	0	30
S_2	1	2	0	1	18
a	-10	-6	0	0	0

Table 3.10 An initial values of objective function and constraints for Bob for Example 3.3.3.1

	x_1	↓x_2	S_1	S_2	R.H.S
S_1	5	3	1	0	30
←S_2	1	$\boxed{2}$	0	1	18
b	-3	-5	0	0	0

Then Alice extracts the input variable x_1 that corresponds to the largest negative value in the objective function,

Then Bob extracts the input variable x_2 that corresponds to the largest negative value in the objective function, then she extracts the output variable S$_1$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (5)

dividing S_1 by 5 to get
inside variable x_1

to get new S$_2$- row =

old S$_2$-row + (-1)(new x_1- row)

$$\begin{array}{llllll} S_2 & & & & & \\ & 1 & 2 & 0 & 1 & 18 \\ +(-1)(1 & & \tfrac{3}{5} & \tfrac{1}{5} & 0 & 6) \\ \hline S_2 & 0 & \tfrac{7}{5} & \tfrac{-1}{5} & 1 & 12 \end{array}$$

To get new a – row =

Old a – row + (+10) (new x_1- row)

$$\begin{array}{llllll} a & -10 & -6 & 0 & 0 & 0 \\ (+10)(1 & \tfrac{3}{5} & \tfrac{1}{5} & 0 & 6 & \\ \hline & 0 & 0 & 2 & 0 & 60 \end{array}$$

then he extracts the output variable S$_2$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

the common element is (2)

dividing S_2 by 2 to get
inside variable x_2

to get new S$_2$ – row =

old S$_1$-row + (-3)(new x_2- row)

$$\begin{array}{llllll} S_1 & & & & & \\ & 5 & 3 & 1 & 0 & 30 \\ (-3)(\tfrac{1}{2} & 1 & 0 & \tfrac{1}{2} & 9) & \\ \hline S_1 & \tfrac{7}{3} & 0 & 1 & \tfrac{-3}{2} & 3 \end{array}$$

To get new b – row =

Old b – row + (+5) (new x_2- row)

$$\begin{array}{llllll} b & -3 & -5 & 0 & 0 & 0 \\ (+5)(\tfrac{1}{2} & 1 & 0 & \tfrac{1}{2} & 9 & \\ \hline & \tfrac{5}{2} & 0 & 5 & \tfrac{5}{2} & 45 \end{array}$$

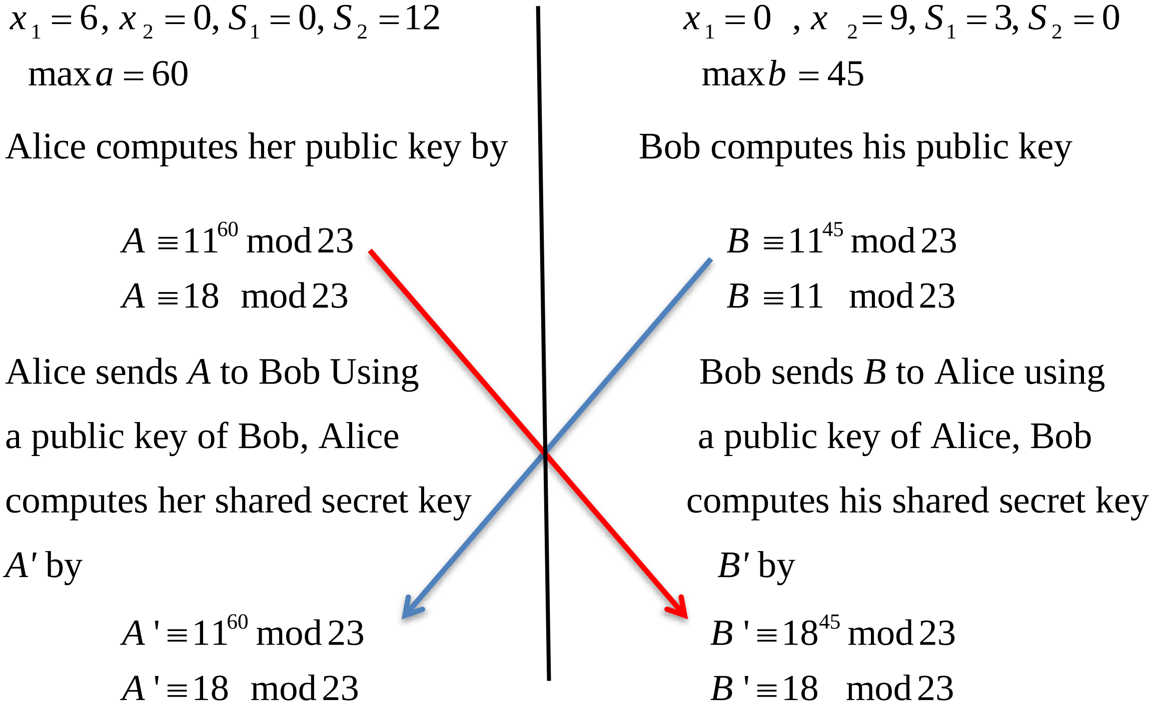
Table 3.11 New values of objective function and constraint for Alice for Example 3.3.3.1

	x_1	x_2	S_1	S_2	R.H.S
x_1	1	$\frac{3}{5}$	$\frac{1}{5}$	0	6
S_2	0	$\frac{7}{5}$	$\frac{-1}{5}$	1	12
a	0	0	2	0	60

Table 3.12 New values of objective function and constraint for Bob for Example 3.3.3.1

	x_1	x_2	S_1	S_2	R.H.S
S_1	$\frac{7}{3}$	0	1	$\frac{-3}{2}$	3
x_2	$\frac{1}{2}$	1	0	$\frac{1}{2}$	9
b	$\frac{5}{2}$	0	5	$\frac{5}{2}$	45

Now notice that every value in Previous tables are positive and zero then we get



$x_1 = 6, x_2 = 0, S_1 = 0, S_2 = 12$

$\max a = 60$

Alice computes her public key by

$$A \equiv 11^{60} \bmod 23$$
$$A \equiv 18 \mod 23$$

Alice sends A to Bob Using a public key of Bob, Alice computes her shared secret key A' by

$$A' \equiv 11^{60} \bmod 23$$
$$A' \equiv 18 \mod 23$$

$x_1 = 0 \ , x \ _2 = 9, S_1 = 3, S_2 = 0$

$\max b = 45$

Bob computes his public key

$$B \equiv 11^{45} \bmod 23$$
$$B \equiv 11 \mod 23$$

Bob sends B to Alice using a public key of Alice, Bob computes his shared secret key B' by

$$B' \equiv 18^{45} \bmod 23$$
$$B' \equiv 18 \mod 23$$

Example 3.3.3.2 Alice and Bob agree to use the prime p =17 and the general element g = 3 and Alice chooses her secret key $a = 2x_2 - 2x_2 - 6x_3$ and Bob choose his secret key $b = x_1 - x_2 - 3x_3$ and both of them use the private share constraints that are

$$\begin{aligned}
-8x_1 + 4x_2 - 2x_3 &\le 4 \\
4x_1 - 2x_2 - 2x_3 &\le 2 \\
6x_1 \qquad + 2x_3 &\le 10 \\
x_1, x_2, x_3 &\ge 0.
\end{aligned}$$

$$A \equiv 3^{min\ a=2x_1-2x_2-6x_3}(mod\ 1\ 7)$$

$$S.t$$

$$-8x_1+4x_2-2x_3 \leq 4$$

$$4x_1-2x_2-2x_3 \leq 2$$

$$6x_1+2x_3 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

$$B \equiv 3^{min\ b=x_1-x_2-3x_3}(mod\ 1\ 7)$$

$$S.t$$

$$-8x_1+4x_2-2x_3 \leq 4$$

$$4x_1-2x_2-2x_3 \leq 2$$

$$6x_1+2x_3 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

Alice and Bob begin adding slackness variables, and setting the objective function to zero

$$A \equiv 3^{min\ a-2x_1+2x_2+6x_3-0S_1-0S_2-0S_3=0}(mod\ 1\ 7)$$

$$S.t$$

$$-8x_1+4x_2-2x_3+S_1 = 4$$

$$4x_1-2x_2-2x_3+S_2 = 2$$

$$6x_1+2x_3+S_3 = 10$$

$$x_1, x_2, x_3, S_1, S_2, S_3 \geq 0$$

$$B \equiv 3^{min\ b-x_1+x_2+3x_3-0S_1-0S_2-0S_3=0}(mod\ 1\ 7)$$

$$S.t$$

$$-8x_1+4x_2-2x_3+S_1 = 4$$

$$4x_1-2x_2-2x_3+S_2 = 2$$

$$6x_1+2x_3+S_3 = 10$$

$$x_1, x_2, x_3, S_1, S_2, S_3 \geq 0$$

Table 3.13 An initial values of objective function and constraints for Alice for Example 3.3.3.2

	x_1	x_2	$\downarrow x_3$	S_1	S_2	S_3	R.H.S
S_1	-8	4	1	0	0	0	4
S_2	4	-2	-2	0	1	0	2
$\leftarrow S_3$	6	0	$\boxed{2}$	0	0	1	10
a	-2	2	6	0	0	0	0

Table 3.14 An initial values of objective function and constraints for Bob for Example 3.3.3.2

	x_1	x_2	$\downarrow x_3$	S_1	S_2	S_3	R.H.S
S_1	-8	4	-2	1	0	0	4
S_2	4	-2	-2	0	1	0	2
$\leftarrow S_3$	6	0	$\boxed{2}$	0	0	1	10
b	-1	1	3	0	0	0	0

x_3 is input variable

S_3 is output variable

The common element is 2

x_3 is input variable

S_3 is output variable

the common element is 2

Table 3.15 New values of objective function and constraint for Alice for Example 3.3.3.2

	x_1	$\downarrow x_2$	x_3	S_1	S_2	S_3	R.H.S
$\leftarrow S_1$	-2	$\boxed{4}$	0	1	0	0	14
S_2	10	-2	0	0	0	1	12
x_3	3	0	1	0	0	$\frac{1}{2}$	5
a	-20	2	0	0	0	-3	-30

Table 3.17 New values of objective function and constraint for Bob for Example 3.3.3.2

	x_1	$\downarrow x_2$	x_3	S_1	S_2	S_3	R.H.S
$\leftarrow S_1$	-2	$\boxed{4}$	0	1	0	1	14
S_2	10	-2	0	0	0	1	12
x_3	3	0	1	0	0	$\frac{1}{2}$	10
b	-10	1	0	0	0	$\frac{-3}{2}$	-15

x_2 is input variable

S_1 is output variable

The common element is 4

x_2 is input variable

S_1 is output variable

the common element is 4

Table 3.18 New values of objective function and constraint for Alice for Example 3.3.3.2

	x_1	x_2	x_3	S_1	S_2	S_3	R.H.S
x_2	$\frac{-1}{2}$	1	0	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{7}{2}$
S_2	9	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	19
x_3	3	0	1	0	0	$\frac{1}{2}$	5
a	-19	0	0	$\frac{1}{2}$	0	$\frac{-7}{2}$	-37

Table 3.19 New values of objective function and constraint for Bob for Example 3.3.3.2

	x_1	x_2	x_3	S_1	S_2	S_3	R.H.S
x_2	$\frac{-1}{2}$	1	0	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{7}{2}$
S_2	9	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	19
x_3	3	0	1	0	0	$\frac{1}{2}$	5
b	$\frac{-1}{2}$	0	0	$\frac{-1}{4}$	0	$\frac{-7}{4}$	$\frac{-23}{2}$

Now notice that every value in previous tables is negative and zero then we get

$x_1 = 0,\ \ x_2 = 7/2\ , x_3 = 5, S_1 = 0, S_2 = 19,$
$S_3 = 0$

$$\min a = -37$$

$$A \equiv 3^{-37} \qquad (\mathrm{mod}17)$$

$$A \equiv (3^{37})^{-1} \qquad (\mathrm{mod}17)$$

$$A \equiv \left((3^6)\cdot 3\right)^{-1} (\mathrm{mod}17)$$

$$A \equiv \left((15)^6 \cdot 3\right)^{-1} (\mathrm{mod}17)$$

$$A \equiv (13 \cdot 3)^{-1} \qquad (\mathrm{mod}17)$$

$$A \equiv (5)^{-1} \qquad (\mathrm{mod}17)$$

$(17,5)$
$17 = 5(3) + 2$

$$B \equiv \left(3^7 \cdot (3^2)^{100}\right)^{-1} (\mathrm{mod}17)$$

$$B \equiv \left(11 \cdot (9^4)^{25}\right)^{-1} \ (\mathrm{mod}17)$$

$= 5(7) - 17(2)$
$5 \times 7 = 1\ (\mathrm{mod}\ 17)$

$x_1 = 0,\ \ x_2 = 7/2\ , x_3 = 5, S_1 = 0, S_2 = 19,$
$S_3 = 0$

$$\min b = -23/2$$

$$B \equiv 3^{-\frac{23}{2}} \quad (\mathrm{mod}17)$$

$$B \equiv 3^{23 \cdot 2^{-1}} (\mathrm{mod}17)$$

$(17,2)$

$17 = 5(8) + 1$
$1 = 17 - 2(8)$
$-8 + 17 = 9$
$9 \times 2 = 1\ (\mathrm{mod}\ 17)$

$$B \equiv (3^{23 \cdot 9})^{-1} \qquad (\mathrm{mod}17)$$

$$B \equiv (3^{207})^{-1} \qquad (\mathrm{mod}17)$$

$5 = 2(2) + 1$
$1 = 5 - 2(2)$
$= 5 - 17(2) + 5(6)$

$$B \equiv \left(11 \cdot (16^5)^5\right)^{-1} (\mathrm{mod}17)$$

$$B \equiv (11 \cdot 16)^{-1} \qquad (\mathrm{mod}17)$$

Alice sends A to Bob Using a public key of Bob, Alice computes her shared secret key A' by

$$A \equiv 7 \quad (\text{mod}17)$$

$$A' \equiv (3)^{-37} (\text{mod}17)$$

$$A' \equiv 7 \quad (\text{mod}17)$$

$$B \equiv (6)^{-1} \quad (\text{mod}17)$$

(6,17)

$$17 = 6(2) + 5$$

$$6 = 5(1) + 1$$

$$1 = 6 - 5(1)$$

$$= 6 - 17 + 6(2)$$

$$= 6(3) - 17$$

$$6 \times 3 = 1\ (\text{mod}17)$$

Bob sends B to Alice Using a public key of Alice, Bob computes his shared secret key B' by

$$B \equiv 3 (mod\,17)$$

$$B' \equiv (7)^{-\frac{23}{2}} (mod\,17)$$

$$B' \equiv 7^{-23 \cdot 2^{-1}} (mod\,17)$$

$$B' \equiv 7^{-23 \cdot 9} (mod\,17)$$

$$B' \equiv \left(12 \cdot (15^4)^{25}\right)^{-1} (\text{mod}17)$$

$$B' \equiv \left(12 \cdot (16^5)^5\right)^{-1} (\text{mod}17)$$

$$B' \equiv (12 \cdot 16)^{-1} \quad (\text{mod}17)$$

$$B' \equiv \left(7^{207}\right)^{-1} \quad (\text{mod}17)$$

$$B' \equiv \left(7^7 \cdot (7^2)^{100}\right)^{-1} (\text{mod}17)$$

$$B' \equiv (5)^{-1} \quad (\text{mod}17)$$

$$B' \equiv 7 \quad (\text{mod}17)$$

3.2.4. The Optimized El-Gamal Public Key cryptosystem using the simplex method

The optimized El-Gamal puplic key cryptosystem (OEPKC) is explained as follows. With a public domain parameters, which are a large prime p and a generator element g of a prime field F_p, Alice chooses her secret key max or min

$a = a_1 x_1 + a_2 x_2$. She uses shared secret constraints that is computed using the original (EPKC) which are given by

The ODLF here are applied the Simplex method to do our proposition. Alice used her (ODLFs).

$$A \equiv g^{max\, or\, min(a=a_1x_1+a_2x_2)} \quad (mod\, p).$$

s.t

$$c_{11}x_1 + c_{12}x_2 \leq b_1$$
$$c_{21}x_2 + c_{22}x_2 \leq b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 \leq b_n$$
$$x_1, x_2 \geq 0.$$

Alice converts the problem to the standard form (adding slack varibles)

$$A \equiv g^{\max\, or\, \min\, (a=a_1x_1+a_2x_2+0S_n)} \quad (\bmod p)$$

s.t

$$c_{11}x_1 + a_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 + S_2 = b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 + S_n = b_n$$
$$x_1, x_2, S_1, S_2, \cdots, S_n \geq 0.$$

Then, Alice zeroes the objective function

$$A \equiv g^{max\, or\, \min\, (a-a_1x_1-a_2x_2-0S_n=0)} \quad (\bmod p)$$

s.t

$$c_{11}x_1 + c_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 + S_2 = b_2$$
$$\vdots$$

$$c_{1n}x_1 + c_{1n}x_2 + S_n = b_n$$

$$x_1, x_2, S_1, S_2, \cdots, S_n \geq 0.$$

Alice will make form an initial table for its own objective function and shared secret constraints

Table 3.20 An initial values of objective function and constraints for Alice for method 3.3.4

	x_1	x_2	S_1	S_2	…	S_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	b_1
S_2	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S_n	c_{n1}	c_{n2}	0	0	…	1	b_n
A	$-a_1$	$-a_2$	0	0	…	0	0

Then Alice extracts the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then she extracts the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then she finds the commonality between the intrinsic and extrinsic variables.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1))

(New interior variable elements)

And she continue until all the values of the table change and she get a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max or, negative and zeros if they are min, then this means that she has reached the optimal solution, but if not, she return to the first step and so on until we reach to the optimal solution, Alice takes a value (max or min a) and put it in her function to extract the declared key

$$A \equiv g^{\max or \min a} \quad (mod\, p)$$

Now, Bob wants to encrypt a plaintext *M* and sent it to Alice. So, he select *M* from the range [2,*p*-1]. Also, he selects an integer $k = bx_1 + bx_2$ such that $[1, p-1]$ He computes C_1.

$$C_1 \equiv g^{max\, or\, min(k=k_1x_1+k_2x_2)}(mod\, p).$$

s.t

$$c_{11}x_1 + c_{12}x_2 \leq b_1$$
$$c_{21}x_2 + c_{22}x_2 \leq b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 \leq b_n$$
$$x_1, x_2 \geq 0.$$

Bob converts the problem to the standard form (adding slack varibles)

$$C_1 \equiv g^{max\ or\ min\ (k=k_1x_1+k_2x_2+0S_n)} \quad (\mathrm{mod}\, p)$$

s.t

$$c_{11}x_1 + c_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 + S_2 = b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 + S_n = b_n$$
$$x_1, x_2, S_1, S_2, \cdots, S_n \geq 0.$$

Then, Bob zeroes the objective function

$$C_1 \equiv g^{max\ or\ min\ (k-k_1x_1-k_2x_2-0S_n=0)} \quad (\mathrm{mod}\, p)$$

s.t

$$c_{11}x_1 + c_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 + S_2 = b_2$$

$$\vdots$$

$$c_{1n}x_1 + c_{1n}x_2 + S_n = b_n$$

$$x_1, x_2, S_1, S_2, \cdots, S_n \geq 0.$$

Bob will make form an initial table for its own objective function and shared secret constraints

Table 3.21: An initial value of objective function and constraints for Bob for method 3.3.4

	x_1	x_2	S_1	S_2	…	S_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	b_1
S_2	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S_n	c_{n1}	c_{n2}	0	0	…	1	b_n
k	$-k_1$	$-k_2$	0	0	…	0	0

Then Bob extracts the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then he extracts the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then he finds the commonality between the intrinsic and extrinsic variables.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1))

(New interior variable elements)

And he continue until all the values of the table change and he get a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max or, negative and zeros if they are min, then this means that he has reached the optimal solution, but if not, he return to the first step and so on until we reach to the optimal solution, Bob takes a value (max or min k) and put it in her function to extract the declared key

$$C_1 \equiv g^{\max or \min(k)} \quad (mod\, p)$$

Bob uses Alice's public key A to compute C_2 a pair of the ciphertext is (C_1,C_2) computed by

$$C_1 \equiv g^{max\, or\, min(k=k_1x_1+k_2x_2)} \quad (mod\, p)$$

and

$$C_2 \equiv M(A)^{max\, or\, min(k=k_1x_1+k_2x_2)} \quad (mod\, p).$$

sends the pair of the ciphertext (C_1,C_2) to Alice.

Upon Alice receiving the ciphertext (C_1,C_2), some steps have been calculated by her to recover the plaintext M. She first computes the value X through the following relation. $X \equiv C_1^{a_1x_1+a_2x_2} \pmod{p}$

Also, she computes the invers value $X^{-1} \pmod{p}$ of X . One can use extended Euclidean algorithm (EEA), as shown in section (2.5) in Chapter (2), for computing the inverse element modulo p. Finally, she computes the relation $X^{-1} * C_2 \equiv M \pmod{p}$ to recover a plaintext M.

$$\begin{aligned} C_2 * (C_1^a)^{-1} &\equiv M(A)^{max\, or\, min(k=k_1x_1+k_2x_2)} \\ &* ((g^{max\, or\, min(k=k_1x_1+k_2x_2)})^{max\, or\, min(a=a_1x_1+a_2x_2)})^{-1} \\ &\equiv M(g^{max\, or\, min(a=a_1x_1+a_2x_2)})^{max\, or\, min(k=k_1x_1+k_2x_2} \\ &* ((g^{max\, or\, min(a=a_1x_1+a_2x_2)})^{max\, or\, min(k=k_1x_1+k_2x_2})^{-1} \equiv M \end{aligned}$$

Example 3.3.4.1 Alice and Bob agree to use the prime p =89 and the generator element $g = 5$ and Alice chooses her secret key $a = 5x_1 + 10x_2$ and Bob selects his plaintext $M = 12$ and chooses ephemeral secret key $k = 3x_1 + 2x_2$ and both use private share secret constraints that are

$$3x_1 + 6x_2 \leq 12$$

$$2x_1 + 8x_2 \leq 24$$

$$4x_1 + 5x_2 \leq 20$$

$$x_1, x_2 \geq 0$$

First Alice computes her public key *A* by

$$A \equiv g^{max\, a=5x_1+10x_2} (mod \quad 89)$$

$$S.t$$

$$3x_1 + 6x_2 \leq 12$$

$$2x_1 + 8x_2 \leq 24$$

$$4x_1 + 5x_2 \leq 20$$

$$x_1, x_2 \geq 0$$

Alice converts the problem to the standard form (adding slack varibles), and, setting the objective function to zero

$$A \equiv g^{max\, a-5x_1-10x_2-0S_1-0S_2-0S_3=0} \quad (mod\, 8\, 9)$$

$$s.to$$

$$3x_1 + 6x_2 + S_1 = 12$$

$$2x_1 + 8x_2 + S_2 = 24$$

$$4x_1 + 5x_2 + S_3 = 20$$

$$x_1, x_2, S_1, S_2, S_3 \geq 0$$

Alice will make form an initial table for its own objective function and shared secret constraints

Table 3.22 An initial values of objective function and constraints for Alice for Example 3.3.4.1

	x_1	↓ x_2	S_1	S_2	S_3	R.H.S
← S_1	3	$\boxed{6}$	1	0	0	12
S_2	2	8	0	1	0	24
S_3	4	5	0	0	1	20
a	-5	-10	0	0	0	0

Then she extracts the input variable x_2 that corresponds to the largest negative value in the objective function

then she extracts the output variable S$_1$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (6) dividing S_1 by 6 to get inside variable x_2

to get new S$_2$ - row = old S$_2$-row + (-8)(new x_2- row)

to get new S$_3$- row = old S$_3$-row + (-5)(new x_2- row)

to get new a - row = old a-row + ($+10$)(new x_2- row)

Table 3.23: New values of objective function and constraints for Alice for Example 3.3.4.1

	x_1	x_2	S_1	S_2	S_3	R.H.S
x_2	$\frac{1}{2}$	1	$\frac{1}{6}$	0	0	2
S_2	-2	0	$\frac{-4}{3}$	1	0	8
S_3	$\frac{3}{2}$	0	$\frac{-5}{6}$	0	1	10
a	0	0	$\frac{5}{3}$	0	0	20

Now notice that every value in previous tables is negative and zero then we get

$$x_1 = 0, x_2 = 2, x = 0, S_1 = 0, S_2 = 8, S_3 = 10$$

$$max\, a = 20,$$

Thus, Alice's public key is computed by

$$A \equiv 5^{20} \quad (\text{mod}\,89)$$
$$A \equiv (10)^{4} \ (\text{mod}\,89)$$
$$A \equiv 32 \quad (\text{mod}\,89).$$

Now, for encrypting a message M, Bob computes the pair of the ciphertext (C_1, C_2) by

$$C_1 \equiv 5^{max\,K=3x_1+2x_2}(mod\,8\,9)$$
$$S.t$$
$$3x_1 + 6x_2 \leq 12$$
$$2x_1 + 8x_2 \leq 24$$
$$4x_1 + 5x_2 \leq 20$$
$$x_1, x_2 \geq 0$$

Bob converts the problem to the standard form (adding slack varibles), and, setting the objective function to zero

$$C_1 \equiv 5^{max\,K-3x_1-2x_2-0S_1-0S_2-0S_3=0}(mod\,8\,9)$$
$$s.t$$
$$3x_1 + 6x_2 + S_1 = 12$$
$$2x_1 + 8x_2 + S_2 = 24$$
$$4x_1 + 5x_2 + S_3 = 20$$
$$x_1, x_2, S_1, S_2, S_3 \geq 0$$

Bob will make form an initial table for its own objective function and shared secret constraints

Table 3.24: An initial value of objective function and constraints for Bob for Example 3.3.4.1

	$\downarrow x_1$	x_2	S_1	S_2	S_3	R.H.S
$\leftarrow S_1$	$\boxed{3}$	6	1	0	0	12
S_2	2	8	0	1	0	24
S_3	4	5	0	0	1	20
k	-3	-2	0	0	0	0

Then he extracts the input variable x_1 that corresponds to the largest negative value in the objective function

Then he extracts the output variable S_1, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (3) dividing S_1 by 3 to get inside variable x_1

to get new S_2 - row = old S_2-row + (-2)(new x_1- row)

to get new S_2- row = old S_2-row + (-4)(new x_1- row)

to get new k - row = old k -row + ($+3$)(new x_1- row)

Table 3.25: New values of objective function and constraints for Bob for Example 3.3.4.1

	x_1	x_2	S_1	S_2	S_3	R.H.S
x_1	1	2	$\frac{1}{3}$	0	0	4
S_2	0	4	$\frac{-2}{3}$	1	0	16
S_3	0	-3	$\frac{-4}{3}$	0	1	4
k	0	4	1	0	0	12

Now notice that every value in Previous tables is positive and zero then we get

$$x_1 = 4, x_2 = 0, S_1 = 0, S_2 = 16, S_3 = 4$$
$$\max k = 12$$

$$C_1 \equiv 5^{12} \quad (mod\, 8\,9)$$
$$C_1 \equiv 8 \quad (mod\, 8\,9),$$
$$and$$
$$C_2 \equiv MA^{max\,K} \quad (mod\, 8\,9)$$
$$C_2 \equiv 12 \cdot (32)^{12} \quad (mod\, 8\,9)$$

$$C_2 \equiv 12 \cdot 32 \quad (mod\, 8\,9)$$
$$C_2 \equiv 28 \quad (mod\, 8\,9).$$

The ciphertext is $(C_1, C_2) = (8,28)$ which is sent to Alice by Bob.

Alice receives the ciphertext $(C_1, C_2) = (8,28)$, she computes the plaintext M by

$$M \equiv ({C_1}^{max\,a})^{-1}(C_2)(mod\, 8\,9)$$
$$\equiv (8^{20})^{-1}(28)(mod\, 8\,9)$$
$$\equiv 64 \cdot 28(mod\, 8\,9)$$
$$\equiv 12(mod\, 8\,9).$$

Example 3.3.4.2 Alice and Bob agree to use the prime $p = 71$ and the generator element $g = 33$ and Alice choose secret key $a = -2x_1 + x_2 - x_3$ and Bob select his plaintext $M = 5$ and chooses ephemeral secret key $k = -5x_1 + 3x_2 - 2x_3$ and both of them use private share secret constraint that are

$$x_1 + x_2 + x_3 \leq 6$$
$$x_1 + 2x_2 + \leq 4$$
$$x_2 \leq 2$$
$$x_1, x_2, x_3 \geq 0$$

First Alice computes her public key A by

$$A \equiv 33^{\min a = -2x_1 + x_2 - x_3} \text{ (mod 71)}$$

s.t

$$x_1 + x_2 + x_3 \leq 6$$
$$x_1 + 2x_2 + \leq 4$$
$$x_2 \leq 2$$
$$x_1, x_2, x_3 \geq 0$$

Alice converts the problem to the standard form (adding slack varibles), and, setting the objective function to zero

$$A \equiv 33^{min\, a+2x_1-x_2+x_3-0S_1-0S_2-0S_3=0}\; mod\; 7$$

$$s.t$$

$$x_1 + x_2 + x_3 + S_1 = 6$$
$$x_1 + 2x_2 + S_2 = 4$$
$$x_2 + S_3 = 2$$
$$x_1, x_2, x_3, S_1, S_2, S_3 \geq 0$$

Table 3.26: An initial value of objective function and constraints for Alice for Example 3.3.4.2

	$\downarrow x_1$	x_2	x_3	S_1	S_2	S_3	R.H.S
S_1	1	1	1	1	0	0	6
$\leftarrow S_2$	$\boxed{1}$	2	0	0	1	0	4
S_3	0	1	0	0	0	1	2
a	2	-1	-1	0	0	0	0

x_1 is input variable S_2 is output variable and the common element is 1

Table 3.27: New values of objective function and constraints for Alice for Example 3.3.4.2

	x_1	x_2	$\downarrow x_3$	S_1	S_2	S_3	R.H.S
$\leftarrow S_1$	0	-1	$\boxed{1}$	1	-1	0	2
S_2	1	2	0	0	1	0	4
S_3	0	1	0	0	0	1	2
A	0	-5	1	0	-2	0	-8

x_3 is input variable S_1 is output variable and the common element is 1

Table 3.28: New values of objective function and constraints for Alice for Example 3.3.4.2

	x_1	x_2	x_3	S_1	S_2	S_3	R.H.S
x_3	0	-1	1	1	-1	0	2
x_1	1	2	0	0	1	0	4
S_3	0	1	0	0	0	1	2
A	0	-4	0	-1	-1	0	-10

Now notice that every value in Previous tables is positive and zero then we get

$$x_1 = 4, x_2 = 0, x_3 = 2, S_1 = 0, S_2 = 0, S_3 = 2$$
$$\min a = -10$$

$$A \equiv (33)^{-10} (mod\ 7\ 1)$$
$$\equiv (33^{10})^{-1} (mod\ 7\ 1)$$
$$\equiv (45)^{-1} (mod\ 7\ 1)$$
$$\equiv 30 (mod\ 7\ 1)$$

Now, for encrypting a message M, Bob computes the pair of the ciphertext (C_1, C_2) by

$$C_1 \equiv 33^{min\, K=-5x_1+3x_2-2x_3} (mod\ 7\ 1)$$

$$S.t$$

$$x_1 + x_2 + x_3 \leq 16$$

$$x_1 + 2x_2 \leq 4$$

$$x_2 \leq 2$$

$$x_1, x_2, x_3 \geq 0$$

Bob converts the problem to the standard form (adding slack varibles), and, setting the objective function to zero

$$C_1 \equiv 33^{min\, K+5x_1-3x_2+2x_3-0S_1-0S_2-0S_3=0} (mod\ 7\ 1)$$

$$S.t$$

$$x_1 + x_2 + x_3 + S_1 = 16$$

$$x_1 + 2x_2 + S_2 = 4$$

$$x_2 + S_3 = 2$$

$$x_1, x_2, x_3, S_1, S_2, S_3 \geq 0$$

Table 3.29: An initial value of objective function and constraints for Bob for Example 3.3.4.2

	$\downarrow x_1$	x_2	x_3	S_1	S_2	S_3	R.H.S
S_1	1	1	1	1	0	0	6
$\leftarrow S_2$	$\boxed{1}$	2	0	0	1	0	4
S_3	0	1	0	0	0	1	2
K	5	-3	2	0	0	0	0

x_1 is input variable S$_2$ is output variable and the common element is 1

Table 3.30: New values of objective function and constraints for Bob for Example 3.3.4.2

	x_1	x_2	$\downarrow x_3$	S_1	S_2	S_3	R.H.S
$\leftarrow S_1$	0	-1	$\boxed{1}$	1	-1	0	2
x_1	1	2	0	0	1	0	4
S_3	0	1	0	0	0	1	2
k	0	-13	2	0	-5	0	-20

x_1 is input variable S_2 is output variable and the common element is 1

Table 3.31: New values of objective function and constraints for Bob for Example 3.3.4.2

	x_1	x_2	x_3	S_1	S_2	S_3	R.H.S
x_3	0	-1	1	1	-1	0	2
x_1	1	2	0	0	1	0	4
S_3	0	1	0	0	0	1	2
K	0	-11	0	-2	-3	0	-16

Now notice that every value in previous tables is negative and zero then we get

$$x_1 = 4, x_2 = 0, x_3 = 2, S_1 = 0, S_2 = 0, S_3 = 2$$
$$\min K = -16$$

$$\begin{aligned} C_1 &\equiv (33)^{-16} \pmod{71} \\ &\equiv (33^{16})^{-1} \pmod{71} \\ &\equiv (49)^{-1} \pmod{71} \\ &\equiv 29 \quad (mod\ 71) \end{aligned}$$

and

$$C_2 \equiv M \cdot A^{\min K} (mod\ 71)$$

$$
\begin{aligned}
&\equiv 5\cdot(30)^{-16}(mod\;71)\\
&\equiv 5\cdot((30^{4})^{4})^{-1}(mod\;7)\\
&\equiv 5(48)^{-1}\quad(mod\;71)\\
&\equiv 5\cdot(37)\quad(mod\;71)\\
&\equiv 43\qquad\quad(mod\;71)
\end{aligned}
$$

The ciphertext is $(C_1, C_2) = (29{,}43)$ which is sent to Alice by Bob.

Alice receives the ciphertext $(C_1, C_2) = (29{,}43)$, she computes the plaintext M by

$$
\begin{aligned}
M &\equiv (C_1^{\min a})^{-1}\cdot(C_2)\quad(\mathrm{mod}\,71)\\
&\equiv (29^{-10})^{-1}(43)\quad(\mathrm{mod}\,71)
\end{aligned}
$$

$$
\begin{aligned}
M &\equiv \left((29^{10})^{-1}\right)^{-1}(43)\quad(\mathrm{mod}\,71)\\
&\equiv (29^{10})(43)\qquad(\mathrm{mod}\,71)\\
&\equiv (48)(43)\qquad(\mathrm{mod}\,71)\\
&\equiv 5\qquad\qquad(\mathrm{mod}\,71)
\end{aligned}
$$

Chapter Four

Big-*M* and Two-Phase Optimization Methods for the DL-Public Key Cryptosystems

Big-*M* and Two-Phase Optimization Methods for the DL-Public Key Cryptosystems

4.1 Introduction

This chapter discusses using two optimization methods that are called big-*M* and two-phase methods the DL - public key cryptosystems, Diffie-Hellman Key exchange and El-Gamal public key cryptosystem. Examples are given to explain these cryptosystems. The modified public key cryptosystems are proposed using the optimized problem. The optimized DL public key cryptosystems are discussed. The security considerations on these optimized DL public key cryptosystems are determined.

4.2 Other Optimized DL-Public Key Cryptosystems

This section proposes other new versions of the DHKE and EPKC using optimization problem through using two methods. The first one is Big-*M* method and second one Tow phases Method. These methods are explained as follows.

4.2.1 The Optimized Diffie-Hellman using Big-*M* Method

Using Definition (3.3.3.1) of the ODLFs that is defined as new concept in Chapter (3), the Optimized Diffie-Hellman Key Exchange (ODHKE) is explained as follows. With a public domain parameter, which are a large prime p and a generator element g of a prime field F_p, Alice and Bob choose their secret keys $a = a_1x_1 + a_2x_2$ and $b = b_1x_1 + b_2x_2$ respectively. They use their shared secret constraints that are computed using the original DHKE and explained in section (2.8.1), which are given by

$$\begin{aligned}&c_{11}x_1+c_{12}x_2\le b_1\\&c_{21}x_2+c_{22}x_2\ge b_2\\&\qquad\vdots\\&c_{1n}x_1+c_{1n}x_2=b_n\\&x_1,x_2\ge 0.\end{aligned}$$

The ODLFs here are applied the big-M method. Alice and Bob used their discrete logarithm objective functions

Each of them (namely, Alice and Bob) begins by converting the problem to the standard form. They are adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$A\equiv g^{max\ or\ \min\ (\ a=a_1x_1+a_2x_2+0S_n\mp MR_n)}\quad (\bmod p)$$

s.t

$$\begin{aligned}&c_{11}x_1+c_{12}x_2+S_1=b_1\\&c_{21}x_2+c_{22}x_2-S_2+R_1=b_2\\&\qquad\vdots\\&c_{1n}x_1+c_{1n}x_2+R_n=b_n\\&x_1,x_2,S_1,S_2,\cdots,S_n,R_1,R_2,\cdots,\mathrm{R}_n\ge 0.\end{aligned}$$

$$B\equiv g^{max\ or\ min\ (b=b_1x_1+b_2x_2+0S_n\mp MR_n)}\quad (\bmod p)$$

s.t

$$\begin{aligned}&c_{11}x_1+c_{12}x_2+S_1=b_1\\&c_{21}x_2+c_{22}x_2-S_2+R_1=b_2\\&\qquad\vdots\\&c_{1n}x_1+c_{1n}x_2+R_n=b_n\\&x_1,x_2,S_1,S_2,\cdots,S_n,R_1,R_2,\cdots,\mathrm{R}_n\ge 0.\end{aligned}$$

Setting the objective function to zero

$$A \equiv g^{max\ or\ \min\ (a-a_1x_1-a_2x_2-0S_n=\mp M)} \pmod{p}$$

$$B \equiv g^{max\ or\ \min\ (\ b-b_1x_1-b_2x_2-0S_n=\mp M)} \pmod{p}$$

s.t

$$c_{11}x_1+c_{12}x_2+S_1=b_1$$

$$c_{21}x_2+c_{22}x_2-S_2+R_1=b_2$$

$$\vdots$$

$$c_{1n}x_1+c_{1n}x_2+R_n=b_n$$

$$x_1,x_2,S_1,S_2,\cdots,S_n,R_1,R_2,\cdots,\mathrm{R}_n\geq 0.$$

Each one of them will form a prime table for its own objective function and shared secret constraints

Table 4.1: An initial value of objective function and constraints for Alice for method 4.2.1

	x_1	x_2	S_1	S_2	…	S_n	R_1	R_2	…	R_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	0	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	0	0	…	1	b_n
a	$-a_1$	$-a_2$	0	$-x_2M$	…	$-x_nM$	0	0	…	0	$\mp M$

Table 4.2: An initial value of objective function and constraints for Bob for method 4.2.1

	x_1	x_2	S_1	S_2	…	S_n	R_1	R_2	…	R_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	0	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	0	0	…	1	b_n
b	$-b_1$	$-b_2$	0	$-x_2M$	…	$-x_nM$	0	0	…	0	$\mp M$

Then they extract the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then they extract the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then they find the commonality between the intrinsic and extrinsic variable.

All the elements of the output variable are divided by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1)) (New interior variable elements).

Continue until all the values of the table change and we get a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max and negative and zeros if they are min, then this means that they have reached the optimal solution, but if not, we return to the first step and so on until we reach to the optimal solution, Alice and Bob take a value (max or min a), (max or min b) and put it in their function to extract the declared key

$$A \equiv g^{\max or \min{(a)}} \quad (\bmod p)$$

$$B \equiv g^{\max or \min(b} \quad (\bmod p)$$

Then the declared keys are exchanged, and they both compute the cipher using their secret key extracted from the big-M method

$$A' \equiv B^{\max or \min(a)} \quad (\bmod p)$$

$$B' \equiv A^{\max or \min(b)} \quad (\mathrm{mod} p)$$

$$A' \equiv B^{\max or \min a=a_1x_1+a_2x_2} \equiv (g^{\max or \min b=b_1x_1+b_2x_2})^{\max or \min a=a_1x_1+a_2x_2}$$

$$\equiv (g^{\max or \min a=a_1x_1+a_2x_2})^{\max or \min b=b_1x_1+b_2x_2}$$

$$\equiv A^{\max or \min b=b_1x_1+b_2x_2} \equiv B'$$

Share secret key between Alice and Bob $B'\ (mod\ p) \equiv A'(mod\ p)$

Example 4.2.1.1 Alice and Bob agree to use the prime $p = 23$ and the generator element $g = 11$ and Alice chooses her secret key $a = 3x_1 - x_2$ and Bob chooses his secret key $b = x_1 + x_2$ and both use the private share secret constrents which are

$$\begin{gathered} 2x_1 + x_2 \geq 2 \\ x_1 + 3x_2 \leq 3 \\ x_2 \leq 4 \end{gathered}$$

Alice first computes her public key A by

$$A \equiv (11)^{max\, a=3x_1-x_2} (mod\, 2\, 3)$$

$$S.t$$

$$\begin{aligned} &2x_1 + x_2 \geq 2 \\ &x_1 + 3x_2 \leq 3 \\ &x_2 \leq 4 \\ &x_1, x_2 \geq 0. \end{aligned}$$

Alice begins by converting the problem to the standard form they are adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$A \equiv (11)^{max\, a=3x_1-x_2-MR_1+0S_1+0S_2+0S_3} (mod\, 2\, 3)$$

$$\begin{aligned} &S.t \\ &2x_1 + x_2 - S_1 + R_1 = 2 \rightarrow R_1 = 2 - 2x_1 - x_2 + S_1 \\ &x_1 + 3x_2 + S_2 = 3 \\ &x_1 + S_3 = 4 \end{aligned}$$

$$x_1, x_2, S_1, S_2, S_3, R_1 \geq 0.$$

With more details, computing A is done as follows.

$$\begin{aligned} A &\equiv (11)^{\max a=3x_1-x_2-M(2-2x_2+S_1)+0S_2+0S_3} && (mod\ 23) \\ &\equiv (11)^{\max a=3x_1-x_2-2M+x_1M+x_2M+S_1M+0S_2+0S_3} && (mod\ 23) \end{aligned}$$

$$\equiv (11)^{\max a=(3-2M)x_1+(-1+M)x_2+S_1M+0S_2+0S_3-2M} \quad (mod23)$$

Setting the objective function to zero

$$A \equiv (11)^{\max a+(-3+2M)x_1+(1-M)x_2-S_1M-0S_2-0S_3=-2M} \quad (mod\ 23)$$

S.t

$$\begin{aligned} 2x_1 + x_2 - S_1 + R_1 &= 2 \\ x_1 + 3x_2 + S_2 \qquad &= 3 \\ x_2 \qquad\qquad + S_3 &= 4 \\ x_1, x_2, S_1, S_2, S_3, R_1 &\geq 0 \end{aligned}$$

Table 4.3: An initial value of objective function and constraints for Alice For Example 4.2.1.1

	$\downarrow x_1$	x_2	S_1	S_2	S_3	R_1	R.H.S
$\leftarrow S_1$	$\boxed{2}$	1	-1	0	0	1	2
S_2	1	3	0	1	0	0	3
S_3	0	1	0	0	1	0	4
a	-3-2M	1-M	M	0	0	0	-2M

Then she extracts the input variable x_1 that corresponds to the factor largest negative M value in the objective function,

then she extracts the output variable S$_1$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (2) dividing S_1 by 2 to get inside variable x_1, to get

new S$_2$- row = old S$_2$-row + (-1)(new x_1- row)

$$\begin{array}{r} old\ S_2\ (1 \quad 3 \quad 0 \quad 1 \quad 0 \quad 0 \quad 3) \\ \underline{+(-1)(1 \quad \tfrac{1}{2} \quad \tfrac{-1}{2} \quad 0 \quad 0 \quad \tfrac{1}{2} \quad 1)} \\ new\ S_2\ (0 \quad \tfrac{5}{2} \quad \tfrac{1}{2} \quad 1 \quad 0 \quad \tfrac{-1}{2} \quad 2) \end{array}$$

to get new S$_3$- row = old S$_3$-row + (0)(new x_1- row)

Factor is zero so, the value of old S$_3$ will not change

to get new a - row = old a -row + $(3 + 2M)$(new x_1- row)

$$\begin{array}{llllllll}
old\ a & (-3-2M & 1-M & M & 0 & 0 & 0 & -2M\,) \\
\underline{+(3+2M\,)(1} & \underline{\quad} & \underline{{}^1\!/\!_2} & \underline{{}^{-1}\!/\!_2} & \underline{0} & \underline{0} & \underline{{}^1\!/\!_2} & \underline{1)} \\
\text{new}\,a & (0 & {}^5\!/\!_2 & {}^{-3}\!/\!_2 & 0 & 0 & {}^3\!/\!_2+M & 3)
\end{array}$$

Table 4.4: New values of objective function and constraint for Alice For Example 4.2.1.1

	x_1	x_2	$\downarrow S_1$	S_2	S_3	R_1	R.H.S
x_1	1	$1/2$	$-1/2$	0	0	$1/2$	1
$\leftarrow S_2$	0	$5/2$	$\boxed{1/2}$	1	0	$-1/2$	2
S_3	0	1	0	0	1	0	4
a	0	$5/2$	$-3/2$	0	0	$3/2+M$	3

Then she extracts the input variable S_1 that corresponds to the factor largest negative M value in the objective function,

then she extracts the output variable S$_2$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (1/2) dividing S$_2$ by (1/2) to get inside variable S$_1$, to get new x_1 - row = old x_1-row + ($\frac{1}{2}$)(new S_1- row)

S$_3$ Factor is zero so, the value of old S$_3$ will not change

to get new a - row = old a-row + ($\frac{3}{2}$)(new S_1- row)

Table 4.5: New values of objective function and constraint for Alice For Example 4.2.1.1

	x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
x_1	1	3	0	1	0	0	3
S_1	0	5	1	2	0	-1	4
S_3	0	1	0	0	1	0	4
a	0	10	0	3	0	M	9

Now notice that every value in previous tables is positive and zero then we get

$$x_1 = 3, x_2 = 0, S_1 = 4, S_2 = 0, S_3 = 4, R_1 = 0$$
$$\max a = 9.$$

Therefore,

$$A \equiv 11^9 \bmod 23$$
$$\equiv 19 \bmod 23$$

On the same time, Bob also computes his public key by

$$B \equiv (11)^{max\, b = x_1 + x_2} (mod\, 2\, 3)$$
$$S.to$$
$$2x_1 + x_2 \geq 2$$
$$x_1 + 3x_2 \leq 3$$
$$x_2 \leq 4$$
$$x_1, x_2 \geq 0.$$

Bob begins by converting the problem to the standard form they are adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$B = (11)^{max\, b = x_1 + x_2 - MR_1 + 0S_1 + 0S_2 + 0S_3} (mod\, 2\, 3)$$
$$S.t$$
$$2x_1 + x_2 - S_1 + R_1 = 2 \rightarrow R_1 = 2 - 2x_1 - x_2 + S_1$$
$$x_1 + 3x_2 + S_2 = 3$$

$$x_2 + S_3 = 4$$

$$x_1, x_2, S_2, S_2, S_3, R_1 \geq 0$$

with more details, computing B can be done by

$$\boldsymbol{B \equiv (11)^{max\, a = x_1 + x_2 - M(2 - 2x_1 - x_2 + S_1) + 0S_2 + 0S_3} (mod\, 2\, 3)}$$

$$\boldsymbol{\equiv (11)^{max\, a = x_1 + x_2 - 2M + 2x_1 M + x_2 M + S_1\, M + 0S_2 + 0S_3} (mod\, 2\, 3)}$$

$$\boldsymbol{\equiv (11)^{max\, a = (1+2M)x_1 + (-1-M)x_2 + S_1 M + 0S_2 + 0S_3 - 2M} (mod\, 2\, 3)}$$

Setting the objective function to zero

$$B \equiv (11)^{max\, a + (-1-2M)x_1 + (-1-M)x_2 - S_1 M - 0S_2 - 0S_3 = -2M} (mod\, 2\, 3)$$

$$S.t$$

$$2x_1 + x_2 - S_1 + R_1 = 2$$

$$x_1 + 3x_2 + S_2 = 3$$

$$x_2 + S_3 = 4$$

$$x_1, x_2, S_2, S_2, S_3, R_1 \geq 0$$

Table 4.6: An initial value of objective function and constraints for Bob for Example 4.2.1.1

	$\downarrow x_1$	x_2	S_1	S_2	S_3	R_1	R.H.S
$\leftarrow R_1$	$\boxed{2}$	1	-1	0	0	1	2
S_2	1	3	0	1	0	0	3
S_3	0	1	0	0	1	0	4
B	-1-2M	-1-M	M	0	0	0	-2M

Then he extracts the input variable x_1 that corresponds to the factor largest negative M value in the objective function, then she extracts the output variable R₁, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (2) dividing S_1 by (2) to get inside variable x_1, to get

new S₂- row = old S₂-row + (-1)(new x_1- row)

$$\begin{array}{lccccccc} old\ S_2 & (1 & 3 & 0 & 1 & 0 & 0 & 3) \\ +(-1) & (1 & \tfrac{1}{2} & \tfrac{-1}{2} & 0 & 0 & \tfrac{1}{2} & 1) \\ \hline new\ S_2 & (0 & \tfrac{5}{2} & \tfrac{1}{2} & 1 & 0 & \tfrac{-1}{2} & 2) \end{array}$$

to get new S₃- row = old S₃-row + (0)(new x_1- row)

Factor is zero so, the value of old S₃ –row will not change

to get new b - row = old b -row + $(1+2M)$(new x_1- row)

$$\begin{array}{lccccccc} old\ b & (-1-2 & -1-M & M & 0 & 0 & 0 & -2M) \\ +(1+2M) & (1 & \tfrac{1}{2} & \tfrac{-1}{2} & 0 & 0 & \tfrac{1}{2} & 1) \\ \hline \text{new}b & (0 & \tfrac{1}{2} & \tfrac{-1}{2} & 0 & 0 & \tfrac{1}{2}+M & 1) \end{array}$$

Table 4.7: New values of objective function and constraint for Bob for Example 4.2.1.1

	x_1	x_2	$\downarrow S_1$	S_2	S_3	R_1	R.H.S
x_1	1	$1/2$	$-1/2$	0	0	$1/2$	1
$\leftarrow S_2$	0	$5/2$	$\boxed{1/2}$	1	0	$-1/2$	2
S_3	0	1	0	0	1	0	4
b	0	$1/2$	$-1/2$	0	0	$1/2+M$	1

Then he extracts the input variable S_1 that corresponds to the factor largest negative M value in the objective function,

then she extracts the output variable S₂, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (1/2) dividing S₂ by 1/2 to get inside variable S₁, to get

new x_1- row = old x_1-row + ($1/2$)(new S_1- row)

S_3 Factor is zero so, the value of old S_3 –row will not change

to get new *b*- row = old *b*-row + (1/2)(new S_1- row)

Table 4.8: New values of objective function and constraint for Bob for Example 4.2.1.1

	x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
x_1	1	3	0	1	0	0	3
S_1	0	5	1	2	0	-1	4
S_3	0	1	0	0	1	0	4
b	0	3	0	1	0	M	3

Now notice that every value in previous tables is positive and zero then we get

$$x_1 = 3, x_2 = 0, S_1 = 4, S_2 = 0, S_3 = 4, R_1 = 0$$
$$\max b = 3$$

So,

$$B \equiv (11)^3 \pmod{23}$$
$$\equiv 20 \pmod{23}$$

Alice sends *A* to Bob using a public key of Bob, Alice computes her shared secret key by

Bob sends *B* to Alice using a public key of Alice, Bob computes his shared secret key by

$$A \equiv 19 \pmod{23}$$
$$A' \equiv 20^9 \pmod{23}$$
$$\equiv (20^3)^3 \pmod{23}$$
$$\equiv 5 \pmod{23}$$

$$\equiv$$

$$B \equiv 20 \pmod{23}$$
$$B' \equiv 19^3 \pmod{23}$$
$$\equiv 5 \pmod{23}$$

S0,

$$A' \equiv B' \equiv 5 \quad (\mathit{mod}\ 23).$$

Example 4.2.1.2. Alice and Bob agree to use the prime *p*=17 and the generator element $g = 3$ and Alice chooses her secret key $a = 3x_1 + 4x_2$ and Bob chooses his secret key $b = 2x_1 + 2x_2$ and both use the private share secret constraints and that are

$$\begin{aligned} &x_1 + 3x_2 \geq 6 \\ &x_1 + x_2 \geq 4 \\ &x_1, x_2 \geq 0 \end{aligned}$$

Alice first computes her public key *A* by

$$\begin{aligned} &A \equiv 3^{min\, a=3x_1+4x_2} (mod\, 17) \\ &S.t \\ &\qquad x_1 + 3x_2 \geq 6 \\ &\qquad x_1 + x_2 \geq 4 \\ &\qquad x_1, x_2 \geq 0 \end{aligned}$$

Alice begins by converting the problem to the standard form they are adding slackness variables and adding artificial variables

$$\begin{aligned} &A \equiv 3^{min\, a=3x_1+4x_2+MR_1+MR_2+0S_1+0S_2} (mod \quad 17) \\ &s.t \\ &x_1 + 3x_2 - S_1 + R_1 = 6 \rightarrow R_1 = 6 - x_1 - 3x_2 + S_1 \\ &x_1 + x_2 - S_2 + R_2 = 4 \rightarrow R_2 = 4 - x_1 - x_2 + S_2 \\ &x_1, x_2, S_1, S_2, R_1, R_2 \geq 0 \end{aligned}$$

With more details, computing *A* is done as follows.

$$\begin{aligned} A &\equiv (3)^{\min a=3x_1+4x_2+6M-x_1M-3x_2M+S_1M+4M-x_1M-x_2M+S_2M} \quad (\text{mod}\, 17) \\ &\equiv (3)^{\min a=(3-2M)x_1+(4-4M)x_2+S_1M+S_2M+10M} \quad (\text{mod}\, 17) \end{aligned}$$

Setting the objective function to zero

$$A \equiv (3)^{min\, a+(-3+2M)x_1+(-4+4M)x_2-S_1M-S_2M=10M} (mod\, 17)$$

$s.t$

$x_1 + 3x_2 - S_1 + R_1 = 6 \rightarrow R_1 = 6 - x_1 - 3x_2 + S_1$

$x_1 + x_2 - S_2 + R_2 = 4 \rightarrow R_2 = 4 - x_1 - x_2 + S_2$

$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0$

Table 4.9: An initial value of objective function and constraints for Alice for Example 4.2.1.2

	x_1	$\downarrow x_2$	S_1	S_2	R_1	R_2	R.H.S
$\leftarrow R_1$	1	$\boxed{3}$	-1	0	1	0	6
S_2	1	1	0	-1	0	1	4
a	-3+2M	-4+4M	-M	-M	0	0	10M

x_2 is input variable, R$_1$ is output variable and common element is 3

Table 4.10: New values of objective function and constraint for Alice for Example 4.2.1.2

	$\downarrow x_1$	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	$^1/_3$	1	$^{-1}/_3$	0	$^1/_3$	0	2
$\leftarrow R_2$	$\boxed{^2/_3}$	0	$^1/_3$	-1	$^{-1}/_3$	1	2
a	$\frac{-5}{2} + \frac{2}{3}M$	0	$\frac{-4}{3} + \frac{1}{3}M$	-M	$\frac{4}{3} - \frac{4}{3}M$	0	1

x_1 is input variable, R$_2$ is output variable and common element is (2/3)

Table 4.11: New values of objective function and constraint for Alice for Example 4.2.1.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	0	1	$^{-1}/_2$	$^1/_2$	$^2/_3$	$^{-1}/_2$	1
x_1	1	0	$^1/_2$	$^{-3}/_2$	$^{-1}/_2$	$^3/_2$	3
a	0	0	$^{-5}/_3$	$^{-5}/_2$	$\frac{1}{2} - M$	$\frac{5}{2} - M$	13

Now notice that every value in Previous tables is negative and zero then we get

$$x_1 = 3, x_2 = 1, S_1 = 0, S_2 = 0, R_1 = 0, R_2 = 0, min\, a = 13$$

Therefore,

$$A \equiv 3^{13}\, mod\, 17$$

$$\equiv 12\, mod\, 17$$

On the same time, Bob also computes his public key by

$$B \equiv 3^{min\, b=2x_1+2x_2} (mod\, 17)$$

$$S.t$$

$$x_1 + 3x_2 \geq 6$$

$$x_1 + x_2 \geq 4$$

$$x_1, x_2 \geq 0$$

Bob begins by converting the problem to the standard form they are adding slackness variables and adding artificial variables

$$B \equiv 3^{min\, b=2x_1+2x_2+MR_1+MR_2} (mod\, 17)$$

$$S.t$$

$$x_1 + 3x_2 - S_1 + R_1 = 6 \rightarrow R_1 = 6 - x_1 - 3x_2 + S_1$$

$$x_1 + x_2 - S_2 + R_2 = 4 \rightarrow R_2 = 4 - x_1 - x_2 + S_2$$

$$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0$$

With more details, computing *B* can be done by

$$B \equiv (3)^{\min b=2x_1+2x_2+6M-x_1M-3x_2M+S_1M+4M-x_1M-x_2M+S_1m} \quad (\text{mod } 17)$$

$$\equiv (3)^{\min b=(2-2M)x_1+(-2+4M)x_2+S_1M+S_2M+10M} \quad (\text{mod } 17)$$

Setting the objective function to zero

$$B \equiv (3)^{\min b+(-2+2M)x_1+(-2+4M)x_2-S_1M-S_2M=10M} \quad (\text{mod } 17)$$

Table 4.12: An initial value of objective function and constraints for Bob for Example 4.2.1.2

	x_1	$\downarrow x_2$	S_1	S_2	R_1	R_2	R.H.S
$\leftarrow R_1$	1	$\boxed{3}$	-1	0	1	0	6
R_2	1	1	0	-1	0	1	4
b	-2+2*M*	-2+4*M*	-*M*	-*M*	0	0	10*M*

x_2 is input variable, R_1 is output variable and common element is (3)

Table 4.13: New values of objective function and constraint for Bob for Example 4.2.1.2

	$\downarrow x_1$	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	$^1/_3$	1	$^{-1}/_3$	0	$^1/_3$	0	2
$\leftarrow R_2$	$\boxed{^2/_3}$	0	$^1/_3$	-1	$^{-1}/_3$	1	2
b	$\frac{-4}{3}+\frac{2}{3}$	0	$\frac{-2}{3}+\frac{1}{3}M$	-M	$\frac{2}{3}-\frac{4}{3}M$	0	4+2M

x_1 is input variable, R_2 is output variable and common element is (2/3)

Table 4.14: New values of objective function and constraint for Bob for Example 4.2.1.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	0	1	$^{-1}/_2$	$^1/_2$	$^2/_3$	$^{-1}/_2$	1
x_1	1	0	$^1/_2$	$^{-3}/_2$	$^{-1}/_2$	$^3/_2$	3
b	0	0	0	-2	-M	2-M	8

Now notice that every value in previous tables is negative and zero then we get

$$x_1 = 3, x_2 = 1, S_1 = 0, S_2 = 0, R_1 = 0, R_2 = 0,$$
$$\min a = 8$$

So, $$B \equiv (3)^8 (mod\ 17)$$
$$\equiv 16 (mod\ 17)$$

Alice sends A to Bob using a public key of Bob, Alice computes her shared secret key by

$$\begin{aligned} A &\equiv 12 && (\bmod 17) \\ A' &\equiv 16^{13} && (\bmod 17) \\ &\equiv (16^6)^2 \cdot 16 && (\bmod 17) \\ &\equiv (1)^2 \cdot 16 && (\bmod 17) \\ &\equiv 16 && (\bmod 17) \end{aligned}$$

Bob sends B to Alice using a public key of Alice, Bob computes his shared secret key by

$$\begin{aligned} B &\equiv 16 && (\bmod 17) \\ B' &\equiv 12^8 && (\bmod 17) \\ &\equiv 16 && (\bmod 17) \end{aligned}$$

4.2.2 The Optimized El-Gamal Public Key Cryptosystem using the Big-*M* method

Using Definition (3.3.3.1) of the ODLFs that is defined as new concept in Chapter (3), the Optimized El-Gamal public key cryptosystem (OEPKC) is explained as follows. With a public domain parameter, which are a large prime *p* and a generator element *g* of a prime field F_p, Alice chooses her secret keys $a = a_1 x_1 + a_2 x_2$. she uses their shared secret constraints that are computed using the original GPKC and explained in section (2.8.2), which are given by

$$\begin{aligned} & c_{11} x_1 + c_{12} x_2 \le b_1 \\ & c_{21} x_2 + c_{22} x_2 \ge b_2 \\ & \qquad \vdots \\ & c_{1n} x_1 + c_{1n} x_2 = b_n \\ & x_1, x_2 \ge 0. \end{aligned}$$

The ODLFs here are applied the Big-*M* method. Alice used her discrete logarithm objective functions

Alice begins by converting the problem to the standard form she is adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$A \equiv g^{max\ or\ \min\ (\ a=a_1x_1+a_2x_2+0S_n\mp MR_n)} \quad (\bmod p)$$

s.t

$$\begin{aligned} & c_{11} x_1 + c_{12} x_2 + S_1 = b_1 \\ & c_{21} x_2 + c_{22} x_2 - S_2 + R_1 = b_2 \\ & \qquad \vdots \\ & c_{1n} x_1 + c_{1n} x_2 + R_n = b_n \\ & x_1, x_2, S_1, S_2, \cdots, S_n, R_1, R_2, \cdots, \mathrm{R}_n \ge 0. \end{aligned}$$

Setting the objective function to zero

$$A \equiv g^{max\ or\ min\ (a-a_1x_1-a_2x_2-0S_n=\mp M)} \quad (\bmod p)$$

s.t

$$c_{11}x_1 + c_{12}x_2 + S_1 = b_1$$
$$c_{21}x_2 + c_{22}x_2 - S_2 + R_1 = b_2$$
$$\vdots$$
$$c_{1n}x_1 + c_{1n}x_2 + R_n = b_n$$
$$x_1, x_2, S_1, S_2, \cdots, S_n, R_1, R_2, \cdots, \mathrm{R}_n \geq 0.$$

Alice will form a prime table for its own objective function and shared secret constraints

Table 4.15: An initial value of objective function and constraints for Alice for method 4.2.2

	x_1	x_2	S$_1$	S$_2$	…	S$_n$	R$_1$	R$_2$	…	R$_n$	R.H.S
S$_1$	c_{11}	c_{12}	1	0	…	0	0	0	…	0	b_1
R$_2$	c_{21}	c_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R$_n$	c_{n1}	c_{n2}	0	0	…	1	0	0	…	1	b_n
a	-a_1	-a_2	0	$-x_2M$	…	$-x_nM$	0	0	…	0	$\mp M$

Alice extracts the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then she extracts the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then she finds the commonality between the intrinsic and extrinsic variable.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S$_1$ – row = old S$_1$ – row + (input variable* (-1) (New interior variable elements)

And she continue until all the values of the table change and she gets a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max and negative and zeros if they are min, then this means that she have reached the optimal solution, but if not, we return to the first step and so on until we reach to the optimal solution, Alice takes a value (max or min *a*), and put it in her function to extract the declared key

$$A \equiv g^{\max or \min a} \quad (\bmod p)$$

Now, Bob wants to encrypt a plaintext *M* and sent it to Alice. So, he select *M* from the range [2,*p*-1]. Also, he selects an integer $k = bx_1 + bx_2$ such that $[1, p-1]$ He computes C_1.

$$C_1 \equiv g^{max\ or\ min(k=k_1x_1+k_2x_2)} (mod\, p).$$

s.t

$$\begin{aligned} & c_{11}x_1 + c_{12}x_2 \leq b_1 \\ & c_{21}x_2 + c_{22}x_2 \geq b_2 \\ & \qquad \vdots \\ & c_{1n}x_1 + c_{1n}x_2 = b_n \\ & x_1, x_2 \geq 0. \end{aligned}$$

The ODLFs here are applied the Big-*M* method. Bob used his discrete logarithm objective functions

Bob begins by converting the problem to the standard form he is adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$C_1 \equiv g^{\max \text{ or } \min\, (\, k=k_1x_1+kx_2+0S_n \mp MR_n)} \quad (\bmod p)$$

s.t

$$\begin{gathered}c_{11}x_1+c_{12}x_2+S_1=b_1\\ c_{21}x_2+c_{22}x_2-S_2+R_1=b_2\\ \vdots\\ c_{1n}x_1+c_{1n}x_2+R_n=b_n\\ x_1,x_2,S_1,S_2,\cdots,S_n,R_1,R_2,\cdots,\mathrm{R}_n\geq 0.\end{gathered}$$

Setting the objective function to zero

$$B\equiv g^{max\ or\ \min\,(b-b_1x_1-b_2x_2-0S_n=\mp M)}\quad(\bmod p)$$

s.t

$$\begin{gathered}c_{11}x_1+c_{12}x_2+S_1=b_1\\ c_{21}x_2+c_{22}x_2-S_2+R_1=b_2\\ \vdots\\ c_{1n}x_1+c_{1n}x_2+R_n=b_n\\ x_1,x_2,S_1,S_2,\cdots,S_n,R_1,R_2,\cdots,\mathrm{R}_n\geq 0.\end{gathered}$$

Alice will form a prime table for its own objective function and shared secret constraints

Table 4.16: An initial value of objective function and constraints for Bob for method 4.2.2

	x_1	x_2	S_1	S_2	…	S_n	R_1	R_2	…	R_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	0	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	0	0	…	1	b_n
b	$-b_1$	$-b_2$	0	$-x_2M$	…	$-x_nM$	0	0	…	0	$\mp M$

Bob extracts the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then he extracts the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then he finds the commonality between the intrinsic and extrinsic variable.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1) (New interior variable elements)

And he continues until all the values of the table change and he gets a table for a new solution if all the elements in the line of the objective function are positive and zeros if they are max and negative and zeros if they are min, then this means that he have reached the optimal solution, but if not, we return to the first step and so on until we reach to the optimal solution, Bob takes a value (max or min a), and put it in his function to extract the declared key

$$C_1 \equiv g^{\max \text{ or } \min(k)} \quad (\bmod p)$$

Bob uses Alice's public key A to compute C_2 a pair of the ciphertext is (C_1, C_2) computed by

$$C_1 \equiv g^{max\ or\ min(k=k_1x_1+k_2x_2)}(mod\, p)$$

and

$$C_2 \equiv M(A)^{max\ or\ min(\ k=k_1x_1+k_2x_2)}(mod\, p).$$

sends the pair of the ciphertext (C_1, C_2) to Alice.

Upon Alice receiving the ciphertext (C_1, C_2), some steps have been calculated by her to recover the plaintext M. She first computes the value X through the following relation. $X \equiv C_1^{a_1x_1+a_2x_2} \ (\bmod p)$

Also, she computes the invers value $X^{-1}(\bmod p)$ of X . One can use extended Euclidean algorithm (EEA), as shown in section (2.5) in Chapter (2), for computing the inverse element modulo p. Finally, she computes the relation $X^{-1} * C_2 \equiv M \ (\bmod p)$ to recover a plaintext M.

$$
\begin{aligned}
C_2 * (C_1^a)^{-1} &\equiv M(A)^{max\, or\, min(k=k_1x_1+k_2x_2)} \\
&* ((g^{max\, or\, min(k=k_1x_1+k_2x_2)})^{max\, or\, min(a=a_1x_1+a_2x_2)})^{-1} \\
&\equiv M(g^{max\, or\, min(a=a_1x_1+a_2x_2)})^{max\, or\, min(k=k_1x_1+k_2x_2} \\
&* ((g^{max\, or\, min(a=a_1x_1+a_2x_2)})^{max\, or\, min(k=k_1x_1+k_2x_2)})^{-1} \equiv M
\end{aligned}
$$

Example 4.2.2.1 Alice and Bob agree to use the prime $p = 353$ and the generator element $g = 3$ and Alice chooses her secret key $a = 6x_1 + 4x_2$ and Bob selects his plaintext *M*=100 and he chooses secret ephemeral key $\boldsymbol{k} = \boldsymbol{3x_1 + 2x_2}$ and uses his private share secret constraints

$$
\begin{aligned}
2x_1 + 3x_2 &\leq 30 \\
3x_1 + 2x_2 &\leq 24 \\
x_1 + \quad x_2 &\geq 3 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

Alice first computes her public key *A* by

$$
\begin{aligned}
&A \equiv (3)^{max\; a=6x_1+4x_2} (mod\, 353) \\
&S.t \\
&2x_1 + 3x_2 \leq 30 \\
&3x_1 + 2x_2 \leq 24 \\
&x_1 + x_2 \geq 3 \\
&x_1, x_2 \geq 0
\end{aligned}
$$

Alice begins by converting the problem to the standard form she is adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$
\begin{aligned}
&A \equiv (3)^{max\; a=6x_1+4x_2+0S_1+0S_2+0S_3-MR_1} (mod\, 353) \\
&S.t \\
&2x_1 + 3x_2 + S_1 = 30
\end{aligned}
$$

$$3x_1 + 2x_2 + S_2 = 24$$

$$x_1 + x_2 - S_3 + R_1 = 3 \rightarrow R_1 = 3 - x_1 - x_2 + S_3$$

$$x_1, x_2, S_1, S_2, S_3, R_1 \geq 0$$

With more details, computing A is done as follows.

$$A \equiv (3)^{\max a = 6x_1 + 4x_2 - 3M + x_1 M + x_2 M - S_3 M + 0S_1 + 0S_2} \pmod{353}$$

$$A \equiv (3)^{\max a = (6+M)x_1 + (4+M)x_2 - 0S_1 - 0S_2 - S_3 M - 3M} \pmod{353}$$

Setting the objective function to zero

$$A \equiv (3)^{\max a + (-6-M)x_1 + (-4-M)x_2 - 0S_1 - 0S_2 + S_3 M = -3M} \pmod{353}$$

$$S.t$$

$$2x_1 + 3x_2 + S_1 = 30$$

$$3x_1 + 2x_2 + S_2 = 24$$

$$x_1 + x_2 - S_3 + R_1 = 3 \rightarrow R_1 = 3 - x_1 - x_2 + S_3$$

$$x_1, x_2, S_1, S_2, S_3, R_1 \geq 0$$

Alice will form a prime table for its own objective function and shared secret constraints

Table 4.17: An initial value of objective function and constraints for Alice for Example 4.2.2.1

	↓ x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
S_1	2	3	1	0	0	0	30
S_2	3	2	0	1	0	0	24
← R_1	$\boxed{1}$	1	0	0	-1	1	3
a	-6-M	-4-M	0	0	M	0	-3M

Then she extracts the input variable x_1 that corresponds to the factor largest negative M value in the objective function, then she extracts the output variable R$_1$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (1) dividing R$_1$ by (1) to get inside variable x_1, to get

new S$_1$ - row = old S$_1$-row + (-2)(new x_1- row)

$$\begin{array}{lrrrrrrr}
old\ S_1 & (2 & 3 & 1 & 0 & 0 & 0 & 30) \\
+(-2)(1 & 1 & 0 & 0 & -1 & 1 & & 3) \\
\hline
\text{new}\,S_1 & (0 & 1 & 1 & 0 & 2 & -2 & 24)
\end{array}$$

to get new S$_2$ - row = old S$_2$ -row + (-3)(new x_1- row)

$$\begin{array}{lrrrrrrr}
old\ S_2 & (3 & 2 & 0 & 1 & 0 & 0 & 24) \\
+(-3)(1 & 1 & 0 & 0 & -1 & 1 & & 3) \\
\hline
\text{new}\,S_2 & (0 & -1 & 0 & 1 & 3 & -3 & 15)
\end{array}$$

to get new a - row = old a -row + $(6+M)$(new x_1- row)

$$\begin{array}{lrrrrrrr}
old\ a & (-6-M & -4-M & 0 & 0 & M & 0 & -3M) \\
+(6+M)(& 1 & 1 & 0 & 0 & -1 & 1 & 3) \\
\hline
\text{new}\,a & (\ \ 0 & 2 & 0 & 0 & -6 & 6+M & 18)
\end{array}$$

Table 4.18: New values of objective function and constraint for Alice Example 4.2.2.1

	x_1	x_2	S_1	S_2	$\downarrow S_3$	R_1	R.H.S
S_1	0	1	1	0	2	-2	24
$\leftarrow S_2$	0	-1	0	1	$\boxed{3}$	-3	15
x_1	1	1	0	0	-1	1	3
a	0	2	0	0	- 6	6+M	18

Then she extracts the input variable S_3 that corresponds to the factor largest negative M value in the objective function, then she extracts the output variable S$_2$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (3) dividing S$_2$ by (3) to get inside variable S$_3$,

Table 4.19: New values of objective function and constraint for Alice for Example 4.2.2.1

	x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
S_1	0	$^5/_3$	1	$^{-2}/_3$	0	0	14
S_3	0	$^{-1}/_3$	0	$^1/_3$	1	-1	5
x_1	1	$^2/_3$	0	$^1/_3$	0	0	8
a	0	0	0	2	0	M	48

Now notice that every value in Previous tables is positive and zero then we get

$$x_1 = 8\,, x_2 = 0\,, S_1 = 14\,, S_2 = 0\,, S_3 = 5\,, R_1 = 0$$
$$\max a = 48$$

Therefore,

$$\begin{aligned} A &\equiv (3)^{48} \pmod{353} \\ &\equiv ((3)^6)^8 \pmod{353} \\ &\equiv ((23)^4)^2 \pmod{353} \\ &\equiv (265^2) \pmod{353} \\ &\equiv 331 \pmod{353} \end{aligned}$$

Bob encrypts his message M through computing the ciphertext pair (C_1, C_2) by

$$C_1 \equiv (3)^{\max 3x_1+2x_2} \pmod{353}$$
$$S.to$$
$$2x_1 + 3x_2 \leq 30$$
$$3x_1 + 2x_2 \leq 24$$
$$x_1 + \ x_2 \geq 3$$
$$x_1\,, x_2 \ \geq 0$$

So,

Bob begins by converting the problem to the standard form he is adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function

$$C_1 \equiv (3)^{\max a=3x_1+2x_2+0S_1+0S_2+0S_3-MR_1} \pmod{353}$$

$$\begin{aligned} &S.t \\ &2x_1 + 3x_2 + S_1 = 30 \\ &3x_1 + 2x_2 + S_2 = 24 \\ &x_1 + x_2 - S_3 + R_1 = 3 \rightarrow R_1 = 3 - x_1 - x_2 + S_3 \\ &x_1 , x_2 , S_1 , S_2 , S_3 , R_1 \ \geq 0 \end{aligned}$$

With more details, computing C_1 is done as follows.

$$C_1 \equiv (3)^{\max k=3x_1+2x_2-3M+x_1M+x_2M-S_3M+0S_1+0S_2} \pmod{353}$$

$$\equiv (3)^{max\ k=(3-M)x_1+(-2+M)x_2+0S_1+0S_2-S_3M-3M} \pmod{353}$$

Setting the objective function to zero

$$C_1 \equiv (3)^{max\ k+(-3-M)x_1+(-2-M)x_2-0S_1-0S_2+S_3M=-3M} \pmod{353}$$

Bob will form a prime table for its own objective function and shared secret constraints

Table 4.20: An initial value of objective function and constraints for Bob for Example 4.2.2.1

	↓ x_1	x_2	S_1	S$_2$	S$_3$	R$_1$	R.H.S
S_1	2	3	1	0	0	0	30
S_2	3	2	0	1	0	0	24
← R_1	$\boxed{1}$	1	0	0	-1	1	3
K	-3-*M*	-2-*M*	0	0	*M*	0	-3*M*

Then he extracts the input variable x_1 that corresponds to the factor largest negative M value in the objective function,

then he extracts the output variable R$_1$, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (1) dividing S_1 by (1) to get inside variable x_1, to get

new S$_1$ - row = old S$_1$-row + (-2)(new x_1- row)

$$\begin{array}{lccccccc} old\ S_1 & (2 & 3 & 1 & 0 & 0 & 0 & 30) \\ +(-2)(& 1 & 1 & 0 & 0 & -1 & 1 & 3) \\ \hline \text{new}\,S_1 & (0 & 1 & 1 & 0 & 2 & -2 & 24) \end{array}$$

to get new S$_2$ - row = old S$_2$ -row + (-3)(new x_1- row)

$$\begin{array}{lccccccc} old\ S_2 & (3 & 2 & 0 & 1 & 0 & 0 & 24) \\ +(-3)(& 1 & 1 & 0 & 0 & -1 & 1 & 3) \\ \hline \text{new}\,S_2 & (0 & -1 & 0 & 1 & 3 & -3 & 15) \end{array}$$

to get new k - row = old k -row + ($3 + M$)(new x_1- row)

$$\begin{array}{lccccccc} old\ k & (-3-M & -2-M & 0 & 0 & M & 0 & -3M\,) \\ +(3+M\,)(& 1 & 1 & 0 & 0 & -1 & 1 & 3) \\ \hline \text{new}\,k & (\ \ 0 & 1 & 0 & 0 & -3 & 3+M & 9) \end{array}$$

Table 4.21: New values of objective function and constraint for Bob for Example 4.2.2.1

	x_1	x_2	S_1	S_2	$\downarrow S_3$	R_1	R.H.S
S_1	0	1	1	0	2	-2	24
$\leftarrow S_2$	0	-1	0	1	$\boxed{3}$	-3	15
x_1	1	1	0	0	-1	1	3
k	0	1	0	0	- 3	3+M	9

Then he extracts the input variable S_3 that corresponds to the factor largest negative *M* value in the objective function,

then he extracts the output variable S_2, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (3) dividing S_2 by (3) to get inside variable S_3,

Table 4.22: New values of objective function and constraint for Bob for Example 4.2.2.1

	x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
S_1	0	$^3/_5$	1	$^{-2}/_3$	0	0	14
S_3	0	$^{-1}/_3$	0	$^1/_3$	1	-1	5
x_1	1	$^2/_3$	0	$^1/_3$	0	0	8
K	0	0	0	1	0	M	24

Now notice that every value in Previous tables is positive and zero then we get

$$x_1 = 8\,, x_2 = 0\,, S_1 = 14\,, S_2 = 0\,, S_3 = 5\,, R_1 = 0$$
$$\max k = 24$$

$$\begin{aligned} C_1 &\equiv (3)^{24} \pmod{353} \\ &\equiv (3^6)^4 \pmod{353} \\ &\equiv (23)^4 \pmod{353} \\ &\equiv 265 \pmod{353} \end{aligned}$$

and

$$\begin{aligned} C_2 &\equiv M \cdot (A^{\max k}) \pmod{353} \\ &\equiv 100 \cdot (331^{24}) \pmod{353} \\ &\equiv 100 \cdot (331^6)^4 \pmod{353} \end{aligned}$$

$$\begin{aligned}
&\equiv 100\cdot(295)^2)^4 && (\text{mod } 353)\\
&\equiv 100\cdot(187)^4 && (\text{mod } 353)\\
&\equiv 100\cdot 131 && (\text{mod } 353)\\
&\equiv 39 && (\text{mod } 353)
\end{aligned}$$

Thus, the ciphertext (C_1, C_2) is $(265{,}39)$.

Alice receives the ciphertext (C_1, C_2) and recover the original plaintext M by

$$\begin{aligned}
M &\equiv (C_1^{\max a})^{-1}\cdot(C_2) && (\text{mod } 353)\\
&\equiv (265^{48})^{-1}\cdot(C_2) && (\text{mod } 353)\\
&\equiv \left(\left(\left((265)^2\right)^3\right)^8\right)^{-1}(39) && (\text{mod } 353)\\
&\equiv \left(\left((331)^3\right)^8\right)^{-1}(39) && (\text{mod } 353)\\
&\equiv \left(\left((295)^2\right)^4\right)^{-1}(39) && (\text{mod } 353)\\
&\equiv \left((187)^4\right)^{-1}(39) && (\text{mod } 353)\\
&\equiv (131)^{-1}(39) && (\text{mod } 353)\\
&\equiv (256)\,(39) && (\text{mod } 353)\\
&\equiv 100 && (\text{mod } 353)
\end{aligned}$$

Example 4.2.2.2 Alice and Bob agree to use the prime p =19 and the generator element $g = 10$ and Alice chooses her secret key $a = 2x_1 + x_2$ and Bob selects his plaintext M=15 and he chooses his secret ephemeral key $k = x_1 + x_2$ and uses his private share secret constraints

$$\begin{aligned}
&x_1 + 3x_2 \geq 30\\
&4x_1 + 2x_2 \geq 40\\
&x_1, x_2 \geq 0
\end{aligned}$$

Alice First computes her public key A by

$$A \equiv (10)^{min\, a=2x_1+x_2} (mod\ 19)$$

$S.t$

$$x_1 + 3x_2 \geq 30$$
$$4x_1 + 2x_2 \geq 40$$
$$x_1, x_2 \geq 0$$

Alice begins by converting the problem to the standard form she is adding slackness variables. And adding artificial variables and she added to the objective function

$$A \equiv (10)^{min\, a=2x_1+x_2+0S_1+0S_2+MR_1+MR_2} (mod\ 19)$$

$s.t$

$x_1 + 3x_2 - S_1 + R_1 = 30 \rightarrow R_1 = 30 - x_1 - 3x_2 + S_1$

$4x_1 + 2x_2 - S_2 + R_2 = 40 \rightarrow R_2 = 40 - 4x_1 - 2x_2 + S_2$

$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0$

With more details, computing *A* is done as follows.

$$A \equiv (10)^{min\, a=2x_1+x_2+30M-x_1M-3x_2M+S_1M+40M-2x_2M+S_2M} (mod\ 19)$$
$$\equiv (10)^{min\, a=(2-5M)x_1+(1-5M)x_2+S_1M+S_2M+70M} (mod\ 19)$$

Setting the objective function to zero

$$A \equiv (10)^{min\, a+(-2+5M)x_1+(-1+5M)x_2-S_1M-S_2M=70M} (mod\ 19)$$

$s.t$

$x_1 + 3x_2 - S_1 + R_1 = 30 \rightarrow R_1 = 30 - x_1 - 3x_2 + S_1$

$4x_1 + 2x_2 - S_2 + R_2 = 40 \rightarrow R_2 = 40 - 4x_1 - 2x_2 + S_2$

$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0$

Alice will form a prime table for its own objective function and shared secret constraints

Table 4.23: An initial value of objective function and constraints for Alice for Example 4.2.2.2

	x_1	$\downarrow x_2$	S_1	S_2	R_1	R_2	R.H.S
$\leftarrow R_1$	1	$\boxed{3}$	-1	0	1	0	30
R_2	4	2	0	-1	0	1	40
a	-2+5M	-1+5M	-M	-M	0	0	70M

x_2 is input variable, R₁ is output, common element is (3)

Table 4.24: New values of objective function and constraint for Alice for Example 4.2.2.2

	$\downarrow x_1$	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	$^1/_3$	1	$^{-1}/_3$	0	$^1/_3$	0	10
$\leftarrow R_2$	$\boxed{^{10}/_3}$	0	$^2/_3$	-1	$^{-2}/_3$	1	20
a	$\frac{-5}{3}+\frac{10}{3}M$	0	$\frac{-1}{3}+\frac{2}{3}M$	-M	$\frac{1}{3}-\frac{5}{3}M$	0	10+20M

x_1 is input variable, R₂ is output, common element is (10/3)

Table 4.25: New values of objective function and constraint for Alice for Example 4.2.2.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	0	1	$^{-6}/_{15}$	$^{-1}/_{10}$	$^6/_{15}$	$^{-1}/_{10}$	8
x_1	1	0	$^1/_5$	$^{-3}/_{10}$	$^{-1}/_5$	$^3/_{10}$	6
a	0	0	0	$^{-1}/_2$	$-M$	$\frac{1}{2}-M$	20

Now notice that every value in Previous tables is negative and zero then we get

$$x_1 = 6 , x_2 = 8 , S_1 = 0 , S_2 = 0 , R_1 = 0 , R_2 = 0$$
$$\min a = 20.$$

Therefore,

$$\begin{aligned} A &\equiv (10)^{20} \quad (\text{mod } 19) \\ &\equiv (5) \quad (\text{mod } 19) \end{aligned}$$

Bob encrypts his message *M* through computing the ciphertext pair (C_1, C_2) by

$$C_1 \equiv (10)^{min\, k = x_1 + x_2} (mod\ 19)$$
$$S.to$$

$$x_1 + 3x_2 \geq 30$$
$$4x_1 + 2x_2 \geq 40$$
$$x_1, x_2 \geq 0.$$

Bob begins by converting the problem to the standard form he is adding slackness variables. And adding artificial variables and he added to the objective function

$$C_1 \equiv (10)^{mink = x_1 + x_2 + 0S_1 + 0S_2 + MR_1 + MR_2} (mod\ 19)$$
$$S.t$$
$$x_1 + 3x_2 - S_1 + R_1 = 30 \rightarrow R_1 = 30 - x_1 - 3x_2 + S_1$$
$$4x_1 + 2x_2 - S_2 + R_2 = 40 \rightarrow R_2 = 40 - 4x_1 - 2x_2 + S_2$$
$$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0.$$

With more details, computing C_1 can be done by

$$C_1 \equiv$$
$$(10)^{min\, k = x_1 + x_2 + 30M - x_1 M - 3x_2 M + S_1 M + 40M - 4x_1 M - 2x_2 M + S_2 M} (mod\ 19)$$
$$\equiv (10)^{min\, k = (1-5M)x_1 + (1-5M)x_2 + S_1 M + S_2 M + 70M} (mod\ 19)$$

Setting the objective function to zero

$$C_1 \equiv (10)^{min\, k + (-1+5M)x_1 + (-1+5M)x_2 - S_1 M - S_2 M = 70M} (mod\ 19)$$
$$S.t$$
$$x_1 + 3x_2 - S_1 + R_1 = 30 \rightarrow R_1 = 30 - x_1 - 3x_2 + S_1$$
$$4x_1 + 2x_2 - S_2 + R_2 = 40 \rightarrow R_2 = 40 - 4x_1 - 2x_2 + S_2$$
$$x_1, x_2, S_1, S_2, R_1, R_2 \geq 0.$$

Bob will form a prime table for its own objective function and shared secret constraints

Table 4.26: An initial value of objective function and constraints for Bob for Example 4.2.2.2

	x_1	↓ x_2	S_1	S_2	R_1	R_2	R.H.S
← R_1	1	$\boxed{3}$	-1	0	1	0	30
R_2	4	2	0	-1	0	1	40
k	-1+5M	-1+5M	-M	-M	0	0	70M

x_2 is input variable, R$_1$ is output, common element is (3)

Table 4.27: New values of objective function and constraint for Bob for Example 4.2.2.2

	↓ x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	$^1/_3$	1	$^{-1}/_3$	0	$^1/_3$	0	10
← R_2	$\boxed{^{10}/_3}$	0	$^2/_3$	-1	$^{-2}/_3$	1	20
k	$\frac{-2}{3}+\frac{10}{3}M$	0	$\frac{-1}{3}+\frac{2}{3}M$	-M	$\frac{1}{3}-\frac{5}{3}M$	0	10+20M

x_1 is input variable, R$_2$ is output, common element is (10/3)

Table 4.28: New values of objective function and constraint for Bob for Example 4.2.2.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	0	1	$^{-6}/_{15}$	$^{-1}/_{10}$	$^6/_{15}$	$^{-1}/_{10}$	8
x_1	1	0	$^1/_5$	$^{-3}/_{10}$	$^{-1}/_5$	$^3/_{10}$	6
k	0	0	0	$^{-1}/_5$	$\frac{1}{5}-M$	$\frac{1}{5}-M$	14

Now notice that every value in previous tables is negative and zero then we get

$$x_1=6\,,\ x_2=8\,,S_1=0\,,S_2=0\,,R_1=0\,,R_2=0$$
$$\min k\ =14$$

So,

$$\begin{aligned} C_1 &\equiv (10)^{14} && \pmod{19} \\ &\equiv (15)^2 && \pmod{19} \\ &\equiv 16 && \pmod{19} \end{aligned}$$

and

$$\begin{aligned} C_2 &\equiv M \ \cdot (A^{\min k}) && \pmod{19} \\ &\equiv 15 \cdot (5^{14}) && \pmod{19} \\ &\equiv 15 \cdot 9 && \pmod{19} \\ &\equiv 2 && \pmod{19} \end{aligned}$$

Thus, the ciphertext (C_1, C_2) is $(16, 2)$

Alice receives the ciphertext (C_1, C_2) and recover the original plaintext *M* by

$$\begin{aligned} M &\equiv (C_1^{\min a})^{-1}(C_2) && \pmod{19} \\ &\equiv (C_1^{20})^{-1}(2) && \pmod{19} \\ &\equiv \left(\left((16)^4\right)^5\right)^{-1}(2) && \pmod{19} \\ &\equiv \left(5^5\right)^{-1}(2) && \pmod{19} \\ &\equiv \left(9\right)^{-1}(2) && \pmod{19} \\ &\equiv \left(9\right)(2) && \pmod{19} \\ &\equiv \left(17\right)(2) && \pmod{19} \\ &\equiv 15 && \pmod{19} \end{aligned}$$

4.2.3 The Optimized Diffie-Hellman using Two Phase Method

Based on Definition (3.3.3.1) of the ODLFs that is defined as new concept in Chapter (3), the Optimized Diffie-Hellman Key Exchange (ODHKE) is explained as follows. With a public domain parameter, which are a large prime *p* and a generator element *g* of a prime field F_p, Alice and Bob choose their secret keys $a = a_1 x_1 + a_2 x_2$ and $b = b_1 x_1 + b_2 x_2$ respectively. They use their shared secret constraints that are computed using the original DHKE and explained in section (2.8.1), which are given by

$$\begin{array}{l} c_{11} x_1 + c_{12} x_2 \le b_1 \\ c_{21} x_2 + c_{22} x_2 \ge b_2 \\ \qquad \vdots \\ c_{1n} x_1 + c_{1n} x_2 = b_n \\ x_1, x_2 \ge 0. \end{array}$$

The ODLFs here are applied the Two-Phase Method. Alice and Bob used their discrete logarithm objective functions

Phase I

Alice and Bob begin by adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And they are adding artificial variables to constraints of type bigger than and equal (≥) or equal (=).

They define a new objective function r that depends on only the artificial variables (the sum of the artificial variables if the function is min). The sum of subtracting the artificial variables if the function is max

$$A \equiv g^{max\ or\ min\ (r = \mp R_n)} \qquad (\bmod p)$$

$$B \equiv g^{max\ or\ \min\ (\ r = \mp R_n)} \qquad (\bmod p)$$

s.t

$$\begin{array}{l} c_{11} x_1 + c_{12} x_2 + S_1 = b_1 \\ c_{21} x_2 + c_{22} x_2 - S_2 + R_1 = b_2 \\ \qquad \vdots \\ c_{1n} x_1 + c_{1n} x_2 + R_n = b_n \\ x_1, x_2, S_1, S_2, \cdots, S_n, R_1, R_2, \cdots, \mathrm{R}_n \ge 0. \end{array}$$

Alice and Bob will form a prime table for its own objective function and shared secret constraints

Table 4.29: An initial values of new objective function and constraints for Alice and Bob for method4.2.3

	x_1	x_2	S_1	S_2	…	S_n	R_1	R_2	…	R_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	0	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	0	0	…	1	b_n
r	$\mp r_1$	$\mp r_2$	0	0	…	0	0	0	…	0	$\mp b$

They extract the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then, they extract the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then they find the commonality between the intrinsic and extrinsic variable.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1) (New interior variable elements)

And for there to be a solution to the model, we must conclude that the value of $r = 0$, otherwise there is no solution to the model

Phase II

Alice and Bob move to the second Phase only in the event that $r = 0$, and the second Phase begins by deleting the artificial variables from the table of the first Phase. Then they complete the solution by substituting the value of the variables into the original objective function and then finding them

Table 4.30: Original objective function and constraint of Alice for method 4.2.3

	x_1	x_2	S_1	S_2	…	S_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	b_n
A	$\mp a_1$	$\mp a_2$	0	0	…	0	0

Table 4.31: Original objective function and constraint of Bob for method 4.2.3

	x_1	x_2	S_1	S_2	…	S_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	b_n
b	$\mp b_1$	$\mp b_2$	0	0	…	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss Gordon method to get the new table.

And they continue until all the values of the table change and we get a table for a new solution. If all the elements in the line of the objective function are positive and zeros if they are max and negative and zeros if they are min, then this means that they have reached the optimal solution. If not, we return to the first step and so on until we reach the optimal solution. Alice and Bob take a value (max or min a) and put it in their function to extract the declared key.

$$A \equiv g^{\max or \min(a)} \pmod p$$

$$B \equiv g^{\max or \min(b)} \pmod p$$

Then the declared keys are exchanged, and they both compute the cipher using their secret key extracted from the simplex method

$$A' \equiv B^{\max or \min(a)} \pmod p$$

$$B' \equiv A^{\max or \min(b)} \quad (\text{mod}p)$$

$$A' \equiv B^{\max or \min a=a_1x_1+a_2x_2} \equiv (g^{\max or \min b=b_1x_1+b_2x_2})^{\max or \min a=a_1x_1+a_2x_2}$$

$$\equiv (g^{\max or \min a=a_1x_1+a_2x_2})^{\max or \min b=b_1x_1+b_2x_2}$$

$$\equiv A^{\max or \min b=b_1x_1+b_2x_2} \equiv B'$$

Share secret key between Alice and Bob $B'\ (mod\ p) \equiv A'(mod\ p)$

Example 4.2.3.1 Alice and Bob agree to use the prime $p = 17$and the generator element $g = 3$ and Alice chooses her secret key $a = 5x_1 - 4x_2 + 3x_3$ and Bob chooses his secret key $b = 3x_1 - 2x_2 + x_3$ and both use the private share secret constraints which are

$$\begin{gathered} 2x_1 \ + x_2 \ - 6x_3 = 20 \\ 6x_1 + 5x_2 + 10x_3 \leq 76 \\ 8x_1 - 3x_2 \ + 6x_3 \leq 50 \\ x_1\,, x_2\,, x_3 \ \geq 0 \end{gathered}$$

Alice and Bob compute their public key *A* and *B*

Alice

$A \equiv 3^{\max a=5x_1-4x_2+3x_3}\ mod\ 17$

S.t

$$\begin{gathered} 2x_1 \ + x_2 \ - 6x_3 = 20 \\ 6x_1 + 5x_2 + 10x_3 \leq 76 \\ 8x_1 - 3x_2 \ + 6x_3 \leq 50 \\ x_1\,, x_2\,, x_3 \ \geq 0 \end{gathered}$$

Bob

$B \equiv 3^{\max b=3x_1-2x_2+x_3}\ mod\ 17$

S.t

$$\begin{gathered} 2x_1 \ + x_2 \ - 6x_3 = 20 \\ 6x_1 + 5x_2 + 10x_3 \leq 76 \\ 8x_1 - 3x_2 \ + 6x_3 \leq 50 \\ x_1\,, x_2\,, x_3 \ \geq 0 \end{gathered}$$

Phase I

Alice and Bob begin by adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And they are adding artificial variables to constraints of type bigger than and equal (≥) or equal (=).

They define a new objective function r that depends on only the artificial variables. (The sum of subtracting the artificial variables)

$$A \equiv 3^{\max r=-R_1} \; mod\ 17$$

s.t

$$\begin{gathered}2x_1 + x_2 - 6x_3 + R_1 \quad = 20\\ 6x_1 + 5x_2 + 10x_3 + S_1 = 76\\ 8x_1 - 3x_2 \; + 6x_3 + S_2 = 50\\ x_1\,, x_2\,, x_3, S_1\,, S_2\,, R_1 \geq 0\end{gathered}$$

$$B \equiv 3^{\max r=-R_1} mod\ 17$$

s.t

$$\begin{gathered}2x_1 + x_2 - 6x_3 + R_1 \quad = 20\\ 6x_1 + 5x_2 + 10x_3 + S_1 = 76\\ 8x_1 - 3x_2 \; + 6x_3 + S_2 = 50\\ x_1\,, x_2\,, x_3, S_1\,, S_2\,, R_1 \geq 0\end{gathered}$$

With more details, computing *A* and *B* can be done by

$$A \equiv 3^{\max r=-20+2x_1+x_2-6x_3} \; mod\ \ 17$$

Setting the objective function to zero

$$A \equiv 3^{max\, r-2x_1-x_2+6x_3=-20} \; mod\ \ 17$$

S.t

$$\begin{gathered}2x_1 + x_2 - 6x_3 + R_1 = 20\\ 6x_1 + 5x_2 + 10x_3 + S_1 = 76\\ 8x_1 - 3x_2 \; + 6x_3 + S_2 = 50\\ x_1\,, x_2\,, x_3, S_1\,, S_2\,, R_1 \geq 0\end{gathered}$$

$$B \equiv 3^{max\, r=-20+2x_1+x_2-6x_3} \; mod\ \ 17$$

Setting the objective function to zero

$$B \equiv 3^{max\, r-2x_1-x_2+6x_3=-20}\ mod\ \ 17$$

s.t

$$\begin{gathered}2x_1 + x_2 - 6x_3 + R_1 = 20\\ 6x_1 + 5x_2 + 10x_3 + S_1 = 76\\ 8x_1 - 3x_2 \ \ + 6x_3 + S_2 = 50\\ x_1\ , x_2\ , x_3 , S_1\ , S_2\ , R_1 \geq 0\end{gathered}$$

Alice and Bob will form a prime table for its own objective function and shared secret constraints

Table 4.32: An initial value of new objective function and constraints for Alice and Bob for Example 4.2.3.1

	↓ x_1	x_2	x_3	S_1	S_2	R_1	R.H.S
R_1	2	1	-6	0	0	1	20
S_1	6	5	10	1	0	0	76
← S_2	$\boxed{8}$	-3	6	0	1	0	50
r	-2	-1	6	0	0	0	-20

Then they extract the input variable x_1 that corresponds to the factor largest negative *M* value in the objective function,

then they extract the output variable S_2, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). The common element is (8) dividing S_2 by (8) to get inside variable x_1, to get new R_1- row = old R_1-row + (-2)(new x_1- row)

$$\begin{array}{l} old\ \ R_1\ \ (2 \quad 1 \quad -6 \quad 0 \quad 0 \quad 1 \quad 20) \\ \underline{\ \ +(-2)(1 \quad {}^{-3}\!/_{8} \quad {}^{3}\!/_{4} \quad 0 \quad {}^{1}\!/_{8} \quad 0 \quad {}^{25}\!/_{2})} \\ new\ \ R_1\ (0 \quad {}^{7}\!/_{4} \quad {}^{-15}\!/_{2} \quad 0 \quad {}^{-1}\!/_{4} \quad 1 \quad {}^{15}\!/_{2}) \end{array}$$

to get new S_1- row = old S_1-row + (-6)(new x_1- row)

$$\begin{array}{lccccccc}
old\ S_1 & (6 & 5 & 10 & 1 & 0 & 0 & 76) \\
+(-6)(1 & {}^{-3}\!/_{8} & {}^{3}\!/_{4} & 0 & {}^{1}\!/_{8} & 0 & {}^{25}\!/_{2}) & \\ \hline
new\ S_1 & (0 & {}^{20}\!/_{4} & {}^{11}\!/_{2} & 1 & {}^{-3}\!/_{4} & 0 & {}^{77}\!/_{2})
\end{array}$$

to get new r - row = old r -row + (2)(new x_1- row)

$$\begin{array}{lccccccc}
old\ r & (-2 & -1 & 6 & 0 & 0 & 0 & -20) \\
+(2)(1 & {}^{-3}\!/_{8} & {}^{3}\!/_{4} & 0 & {}^{1}\!/_{8} & 0 & {}^{25}\!/_{2}) & \\ \hline
\text{new}\, r & (0 & {}^{-7}\!/_{4} & {}^{15}\!/_{2} & 0 & {}^{1}\!/_{4} & 0 & {}^{-15}\!/_{2})
\end{array}$$

Table 4.33: New values of objective function and constraint for Alice and Bob for Example 4.2.3.1

	x_1	$\downarrow x_2$	x_3	S_1	S_2	R_1	R.H.S
$\leftarrow R_1$	0	$\boxed{7/4}$	-15/2	0	-1/4	1	15/2
S_1	0	20/4	11/2	1	-3/4	0	77/2
x_1	1	-3/8	3/4	0	1/8	0	25/2
r	0	-7/4	15/2	0	1/4	0	-15/2

x_2 is input variable and R_1 is output variable and common element (7/4)

Table 4.34: New values of objective function and constraint for Alice and Bob for Example 4.2.3.1

	x_1	x_2	x_3	S_1	S_2	R_1	R.H.S
x_2	0	1	-30/7	0	-1/7	4/7	30/7
S_1	0	0	256/7	1	2/7	-29/7	52/7
x_1	1	0	-6/7	0	1/14	3/4	55/7
r	0	0	0	0	0	1	0

they note that the value of the $r = 0$, so they move on to the next Phase.

The second Phase begins by deleting the artificial variables from the table of the first Phase. Then they complete the solution by substituting the value of the variables into the original objective function and then finding them

Phase II

Table 4.35: Original objective function for Alice for Example 4.2.3.1

	x_1	x_2	x_3	S_1	S_2	R.H.S
x_2	0	1	-30/7	0	-1/7	30/7
S_1	0	0	256/7	1	2/7	52/7
x_1	1	0	-6/7	0	1/14	55/7
a	-5	4	-3	0	0	0

Table 4.36: Original objective function for Bob for Example 4.2.3.1

	x_1	x_2	x_3	S_1	S_2	R.H.S
x_2	0	1	-30/7	0	-1/7	30/7
S_1	0	0	256/7	1	2/7	52/7
x_1	1	0	-6/7	0	1/14	55/7
b	-3	2	-1	0	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss Gordon method to get the new table.

New a - row

$$= old\ a - row + (5 * x_1 - row) + (-4 * x_2 - row)$$

New a-row

= (-5   4   -3   0   0   0)

+(5   0   -30/7   0   5/14   275/7)

+(0 -4   120/7   0   4/7   -120/7)

(0 0   69/7   0   13/14   155/7)

new b - row

$$= old\ b - row + (3 * x_1 - row) + (-2 * x_2 - row)$$

new b-row

= (-3   2   -1   0   0   0)

+(3   0   -30/7   0   5/14   275/7)

+(0   -2   60   0   2/7   -60/7)

(0   0   37/7   0   9/14   215/7)

Table 4.37: New values of objective function and constraint for Alice for Example 4.2.3.1

	x_1	x_2	x_3	S_1	S_2	R.H.S
x_2	0	1	-30/7	0	-1/7	30/7
S₁	0	0	256/7	1	2/7	52/7
x_1	1	0	-6/7	0	1/14	55/7
a	0	0	69/7	0	13/14	155/7

Table 4.38: New values of objective function and constraint for Bob for Example 4.2.3.1

	x_1	x_2	x_3	S_1	S_2	R.H.S
x_2	0	1	-30/7	0	-1/7	30/7
S₁	0	0	256/7	1	2/7	52/7
x_1	1	0	-6/7	0	1/14	55/7
b	0	0	37/7	0	9/14	215/7

Now notice that every value in previous tables is positive and zero then we get

$x_1 = 55/7 \ , \ x_2 = 30/7$

$Max\ a = 155/7$

$$A \equiv 3^{\frac{155}{7}} \pmod{17}$$
$$\equiv 3^{155 \cdot 7^{-1}} \pmod{17}$$
$$7^{-1} = 78 * 5 = 1 \pmod{17}$$

$$\equiv 3^{155 \cdot 5} \pmod{17}$$
$$\equiv 3^{775} \pmod{17}$$

$$\equiv (3^7)^{100} \cdot (3^{72}) \cdot 3^3 \mod 17$$
$$\equiv (11^4)^{25} \cdot (3^8)^9 \cdot 10 \pmod{17}$$

$$\equiv (4^5)^5 \cdot (16)^9 \cdot 10 \pmod{17}$$
$$\equiv (4^4) \cdot 16^3 \cdot 10 \pmod{17}$$

$x_1 = 55/7 \ , \ x_2 = 30/7$

$Max\ b = 215/7$

$$B \equiv 3^{\frac{215}{7}} \pmod{17}$$
$$\equiv 3^{215 \cdot 7^{-1}} \pmod{17}$$
$$7^{-1} = 7 * 5 = 1 \pmod{17}$$

$$\equiv 3^{215 \cdot 5} \pmod{17}$$
$$\equiv 3^{1075} \pmod{17}$$

$$\equiv (3^{10})^{100} \cdot 3^{72} \cdot 3^3 \pmod{17}$$
$$\equiv (8^{10})^{10} \cdot 16 \cdot 10 \pmod{17}$$

$$\equiv (13)^{10} \cdot 160 \pmod{17}$$
$$\equiv (13)^2 \cdot 160 \pmod{17}$$

$$\equiv 4 \cdot 16 \cdot 10 \qquad (\text{mod } 17)$$

Alice sends A to Bob using a public key of Bob, Alice computes her shared secret key by

$$A \equiv 11 \qquad (\text{mod } 17)$$

$$A' \equiv B^{\max a} \quad (mod\ 17)$$

$$\begin{aligned}
&\equiv 10^{775} && (\text{mod } 17)\\
&\equiv (10^{7})^{100} \cdot (10)^{72} \cdot (10)^{3} && (\text{mod } 17)\\
&\equiv (5^{10})^{10} \cdot (10^{8})^{9} \cdot 14 && (\text{mod } 17)\\
&\equiv 9^{10} \cdot 16^{9} \cdot 14 && (\text{mod } 17)\\
&\equiv 13 \cdot (16^{3})^{3} \cdot 14 && (\text{mod } 17)\\
&\equiv 13 \cdot 16 \cdot 14 && (\text{mod } 17)\\
&\equiv 5 && (\text{mod } 17)
\end{aligned}$$

$$\equiv 16 \cdot 160 \qquad \text{mod } 17$$

Bob sends B to Alice using a public key of Alice, Bob computes his shared secret key by

$$B \equiv 10 \qquad \text{mod } 17$$

$$B' \equiv A^{\max b} \quad (mod\ 17)$$

$$\begin{aligned}
&\equiv 11^{1075} && (\text{mod } 17)\\
&\equiv (11)^{1000} \cdot (11)^{72} \cdot 11^{3} && (\text{mod}17)\\
&\equiv (11^{10})^{100} \cdot (11^{8})^{9} \cdot 5 && (\text{mod } 17)\\
&\equiv ((11^{5})^{2})^{100} \cdot (16^{3})^{3} \cdot 5 && (\text{mod } 17)\\
&\equiv (11^{10})^{100} \cdot (11^{8})^{9} \cdot 5 && (\text{mod } 17)\\
&\equiv ((11^{5})^{2})^{100} \cdot (16^{3})^{3} \cdot 5 && (\text{mod } 17)\\
&\equiv (16^{5})^{5} \cdot 12 && (\text{mod } 17)\\
&\equiv 16 \cdot 12 && (\text{mod } 17)\\
&\equiv 5 && (\text{mod } 17)
\end{aligned}$$

Example 4.2.3.2 Alice and Bob agree to use the prime $p = 71$ and the generator element $g = 7$ and Alice chooses her secret key $a = 10x_1 + 8x_2$ and Bob chooses his secret key $b = 7x_1 + 6x_2$ and both use the private share secret constraints which are

$$\begin{aligned}
2x_1 + 3x_2 &\geq 18\\
3x_1 \qquad &\geq 9\\
x_1, x_2 \ &\geq 0
\end{aligned}$$

Alice and Bob used discrete logarithm objective function

Alice

$$A \equiv (7)^{\min a = 10x_1 + 8x_2} \quad (\text{mod } 71)$$

S.t

Bob

$$B \equiv (7)^{min\ b = 7x_1 + 6x_2} \quad (\text{mod } 71)$$

S.t

$$\begin{aligned}2x_1 + 3x_2 &\geq 18\\ 3x_1 \qquad &\geq 9\\ x_1\ ,x_2\ &\geq 0\end{aligned}$$

$$\begin{aligned}2x_1 + 3x_2 &\geq 18\\ 3x_1 \qquad &\geq 9\\ x_1\ ,x_2\ &\geq 0\end{aligned}$$

Phase I

Alice and Bob begin by adding slackness variables and they are adding artificial variables.

They define a new objective function *r* that depends on only the artificial variables.

$A \equiv (7^{)\min r=R_1+R_2}$ mod 71

S.t

$$\begin{aligned}&2x_1 + 3x_2 - S_1 + R_1 = 18\\ &3x_1 - S_2 + R_2 = 9\\ &x_1, x_2, S_1, S_2, R_1, R_2 \geq 0\end{aligned}$$

$B \equiv (7)^{\min r=R_1+R_2}\ (\text{mod } 71)$

S.t

$$\begin{aligned}&2x_1 + 3x_2 - S_1 + R_1 = 18\\ &3x_1 - S_2 + R_2 = 9\\ &x_1, x_2, S_1, S_2, R_1, R_2 \geq 0\end{aligned}$$

With more details, computing *A* and *B* can be done by

$A \equiv (7)^{\min r=-5x_1-3x_2-S_1+S_2+27}$
(mod 71)

Setting the objective function to zero

$A \equiv 7^{\min r+5x_1+3x_2-S_1-S_2=27}$
(mod 71)

S.t

$$\begin{aligned}&2x_1 + 3x_2 - S_1 + R_1 = 18\\ &3x_1 - S_2 + R_2 = 9\\ &x_1, x_2, S_1, S_2, R_1, R_2 \geq 0\end{aligned}$$

$B \equiv (7)^{\min r=-5x_1-3x_2+S_1+S_2+27}$
(mod71)

$B \equiv 7^{\min r+5x_1+3x_2+S_1+S_2=27}$
(mod 71)

S.t

$$\begin{aligned}&2x_1 + 3x_2 - S_1 + R_1 = 18\\ &3x_1 - S_2 + R_2 = 9\\ &x_1, x_2, S_1, S_2, R_1, R_2 \geq 0\end{aligned}$$

Alice and Bob will form a prime table for its own objective function and shared secret constraints

Table 4.39: An initial value of new objective function and constraints for Alice and Bob for Example 4.2.3.2

	$\downarrow x_1$	x_2	S_1	S_2	R_1	R_2	R.H.S
R_1	2	3	-1	0	1	0	18
$\leftarrow R_2$	$\boxed{3}$	0	0	-1	0	1	9
r	5	3	-1	-1	0	0	27

x_1 is input variable, R_2 is output variable and 3 is common element

Table 4.40: New values of objective function and constraint for Alice and Bob for Example 4.2.3.2

	x_1	$\downarrow x_2$	S_1	S_2	R_1	R_2	R.H.S
$\leftarrow R_1$	0	$\boxed{3}$	-1	2/3	1	-2/3	12
x_1	1	0	0	-1/3	0	1/3	3
r	0	3	-1	2/3	1	-5/3	12

x_2 is input variable, R_1 is output variable and 3 is common element

Table 4.41: New values of objective function and constraint for Alice and Bob for Example 4.2.3.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_2	0	1	-1/3	2/9	1/3	$-2/9$	4
x_1	1	0	0	-1/3	0	1/3	3
r	0	0	0	0	0	-1	0

Note that $r = 0$ so, move to the next phase

Phase II

Table 4.42: Original objective function for Alice for Example 4.2.3.2

	x_1	x_2	S_1	S_2	R.H.S
x_2	0	1	-1/3	2/9	4
x_1	1	0	0	-1/3	3
a	-10	-8	0	0	0

Table 4.43: Original objective function for Bob for Example 4.2.3.2

	x_1	x_2	S_1	S_2	R.H.S
x_2	0	1	-1/3	2/9	4
x_1	1	0	0	-1/3	3
b	-7	-6	0	0	0

Now make coefficient of x_1, x_2 equal to zero

New a-row

$$= old\ a - row + (10 * x_1 - row) \\ +(8 * x_2 - row)$$

New a-row

= (-10 -8 0 0 0)

+(10 0 0 -10/3 30)

+(0 8 -8/3 16/9 32)

(0 0 -8/3 -14/9 62)

new b-row

$$= old\ b - row + (7 * x_1 - row) \\ +(6 * x_2 - row)$$

new b-row

= (-7 -6 0 0 0)

+(7 0 0 -7/3 21)

+(0 6 -2 4/3 24)

(0 0 -2 -1 45)

Table 4.44: New values of objective function and constraint for Alice for Example 4.2.3.2

	x_1	x_2	S_1	S_2	R.H.S
x_2	0	1	-1/3	2/9	4
x_1	1	0	0	-1/3	3
a	0	0	-8/3	-14/9	62

Table 4.45: New values of objective function and constraint for Bob for Example 4.2.3.2

	x_1	x_2	S_1	S_2	R.H.S
x_2	0	1	-1/3	2/9	4
x_1	1	0	0	-1/3	3
b	0	0	-2	-1	45

Now, note that every value in previous tables is negative and zero, then we get

$x_1 = 3 \;, x_2 = 4$

$\min a = 62$

$$A \equiv 7^{62} \quad mod\ 71$$
$$A \equiv (7^6)^{10} \cdot (7)^2 \quad mod\ 71$$
$$A \equiv (2)^{10} \cdot 49 \quad mod\ 71$$
$$A \equiv 30 \cdot 49 \quad mod\ 71$$

Alice sends A to Bob using a public key of Bob, Alice computes her shared secret key by

$$A \equiv 50 \quad mod\ 71$$
$$A' \equiv (26)^{62} \quad mod\ 71$$
$$\equiv (26^6)^{10} \cdot (26)^2 \quad mod\ 71$$
$$\equiv (30^5)^2 \cdot 37 \quad mod\ 71$$
$$\equiv 37^2 \cdot 37 \quad mod\ 71$$
$$\equiv 20 \cdot 37 \quad mod\ 71$$
$$\equiv 30 \quad mod\ 71$$

$x_1 = 3 \,, x_2 = 4$

$\min b = 45$

$$B \equiv 7^{45} \quad mod\ 71$$
$$B \equiv (7^9)^5 \quad mod\ 71$$
$$B \equiv (51^3)^3 \quad mod\ 71$$
$$B \equiv (23)^3 \quad mod\ 71$$

Bob sends B to Alice using a public key of Alice, Bob computes his shared secret key by

$$B \equiv 26 \quad mod\ 71$$
$$B' \equiv (50)^{45} \quad mod\ 71$$
$$\equiv (50^5)^9 \quad mod\ 71$$
$$\equiv (32^3)^3 \quad mod\ 71$$
$$\equiv 37^3 \quad mod\ 71$$
$$\equiv 30 \quad mod\ 71$$

4.2.4 The Optimized El-Gamal Public Key Cryptosystem using the Two-Phase Method

Using Definition (3.3.3.1) of the ODLFs that is defined as new concept in Chapter (3), the Optimized El-Gamal Public Key Cryptosystem (OEPKC) is explained as follows. With a public domain parameter, which are a large prime *p* and a generator element g of a prime field F_p, Alice chooses her secret keys $a = a_1x_1 + a_2x_2$. she uses their shared secret constraints that are computed using the original EPKC and explained in section (2.8.2), which are given by

$$\begin{gathered} c_{11}x_1 + c_{12}x_2 \leq b_1 \\ c_{21}x_2 + c_{22}x_2 \geq b_2 \\ \vdots \\ c_{1n}x_1 + c_{1n}x_2 = b_n \\ x_1, x_2 \geq 0. \end{gathered}$$

The ODLFs here are applied the Two-Phase Method. Alice used her discrete logarithm objective functions

Phase I

Alice begins by adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And she is adding artificial variables to constraints of type bigger than and equal (≥) or equal (=).

she defines a new objective function *r* that depends on only the artificial variables (the sum of the artificial variables if the function is min). The sum of subtracting the artificial variables if the function is max

$$A \equiv g^{max\ or\ min\ (r = \mp R_n)} \qquad (\bmod p)$$

s.t

$$\begin{aligned}
&c_{11}x_1 + c_{12}x_2 + S_1 = b_1\\
&c_{21}x_2 + c_{22}x_2 - S_2 + R_1 = b_2\\
&\qquad\vdots\\
&c_{1n}x_1 + c_{1n}x_2 + R_n = b_n\\
&x_1, x_2, S_1, S_2, \cdots, S_n, R_1, R_2, \cdots, \mathrm{R}_n \geq 0.
\end{aligned}$$

Alice will form a prime table for its own objective function and shared secret constraints

Table 4.46: An initial value of new objective function and constraints for Alice for method 4.2.4

	x_1	x_2	S_1	S_2	…	S_n	R_1	R_2	…	R_n	R.H.S
S_1	c_{11}	c_{12}	1	0	…	0	0	0	…	0	b_1
R_2	c_{21}	c_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	c_{n1}	c_{n2}	0	0	…	1	0	0	…	1	b_n
r	$\mp r_1$	$\mp r_2$	0	0	…	0	0	0	…	0	$\mp b$

she extracts the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then, she extracts the output variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then she finds the commonality between the intrinsic and extrinsic variable.

Divides all the elements of the external variable by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1) (New interior variable elements)

And for there to be a solution to the model, we must conclude that the value of $r = 0$, otherwise there is no solution to the model

Phase II

Alice moves to the second phase only in the event that $r = 0$, and the second phase begins by deleting the artificial variables from the table of the first phase. Then she completes the solution by substituting the value of the variables into the original objective function and then finding them.

Table 4.47: Original objective function for Alice for method 4.2.4

	x_1	x_2	S$_1$	S$_2$	…	S$_n$	R.H.S
S$_1$	a_{11}	a_{12}	1	0	…	0	b_1
R$_2$	a_{21}	a_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
R$_n$	a_{n1}	a_{n2}	0	0	…	1	b_n
a	$\mp a_1$	$\mp a_2$	0	0	…	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss-Gordon method to get the new table.

And she continues until all the values of the table change and we get a table for a new solution. If all the elements in the line of the objective function are positive and zeros if they are max and negative or negative and zeros if they are min, then this means that she has reached the optimal solution. If not, we return to the first step and so on until we reach the optimal solution. Alice takes a value (max or min a) and put it in their function to extract the declared key.

$$A \equiv g^{\max or \min(a)} \quad (\bmod p)$$

Now, Bob wants to encrypt a plaintext *M* and sent it to Alice. So, he select *M* from the range [2,*p*-1]. Also, he selects an integer $k = bx_1 + bx_2$ such that $[1, p-1]$ He computes C_1.

$$C_1 \equiv g^{max\, or\, min(k=k_1x_1+k_2x_2)}(mod\, p).$$

s.t

$$
\begin{aligned}
&c_{11}x_1 + c_{12}x_2 \le b_1\\
&c_{21}x_2 + c_{22}x_2 \ge b_2\\
&\qquad\vdots\\
&c_{1n}x_1 + c_{1n}x_2 = b_n\\
&x_1, x_2 \ge 0.
\end{aligned}
$$

The ODLFs here are applied the Two-Phase Method. Bob used his discrete logarithm objective functions

Phase I

Bob begins by converting the problem to the standard form he is adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And adding artificial variables to constraints of type bigger than and equal (≥) or equal (=). It is added to the objective function.

He defines a new objective function *r* that depends on only the artificial variables (the sum of the artificial variables if the function is min). The sum of subtracting the artificial variables if the function is max

$$C_1 \equiv g^{max\ or\ min\ (r=\mp R_n)} \qquad (\bmod p)$$

s.t

$$
\begin{aligned}
&c_{11}x_1 + c_{12}x_2 + S_1 = b_1\\
&c_{21}x_2 + c_{22}x_2 - S_2 + R_1 = b_2\\
&\vdots\\
&c_{1n}x_1 + c_{1n}x_2 + R_n = b_n\\
&x_1, x_2, S_1, S_2, \cdots, S_n, R_1, R_2, \cdots, R_n \ge 0.
\end{aligned}
$$

And setting the objective function to zero

Bob will form a prime table for its own objective function and shared secret constraints

Table 4.48: An initial value of new objective function and constraints for Bob for method 4.2.4

	x_1	x_2	S_1	S_2	…	S_n	R_1	R_2	…	R_n	R.H.S
S_1	a_{11}	a_{12}	1	0	…	0	0	0	…	0	b_1
R_2	a_{21}	a_{22}	0	1	…	0	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	a_{n1}	a_{n2}	0	0	…	1	0	0	…	1	b_n
r	$\mp r_1$	$\mp r_2$	0	0	…	0	0	0	…	0	$\mp b$

Bob extracts the input variable that corresponds to the largest negative value in the objective function if it is max, and the largest positive value in the objective function if it is min, then, he extracts the outer variable, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output). Then he finds the commonality between the intrinsic and extrinsic variable.

All the elements of the external variable are divided by the common element and then changes the remaining rows according to the Gauss-Jordan method.

New S_1 – row = old S_1 – row + (input variable* (-1) (New interior variable elements)

And for there to be a solution to the model, we must conclude that the value of $r = 0$, otherwise there is no solution to the model

Phase II

Bob moves to the second phase only if $r = 0$, and the second phase begins by deleting the artificial variables from the table of the first phase. Then he completes the solution by substituting the value of the variables into the original objective function and then finding them.

Table 4.49: Original objective function for Bob for method 4.2.4

	x_1	x_2	S₁	S₂	…	Sₙ	R.H.S
S₁	c_{11}	c_{12}	1	0	…	0	b_1
R₂	c_{21}	c_{22}	0	1	…	0	b_2
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
Rₙ	c_{n1}	c_{n2}	0	0	…	1	b_n
k	$\mp b_1$	$\mp b_2$	0	0	…	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss Gordon-method to get the new table.

And he continues until all the values of the table change and we get a table for a new solution. If all the elements in the line of the objective function are positive and zeros if they are max and negative and zeros if they are min, then this means that they have reached the optimal solution. If not, we return to the first step and so on until we reach the optimal solution. Bob takes a value (max or min a) and put it in their function to extract the declared key.

$$C_1 \equiv g^{\max or \min(k)} \quad (\bmod p)$$

and

$$C_2 \equiv M(A)^{max\, or\, min(k=k_1x_1+k_2x_2)}(mod\, p).$$

sends the pair of the ciphertext (C_1,C_2) to Alice.

Upon Alice receiving the ciphertext (C_1,C_2), some steps have been calculated by her to recover the plaintext *M*. She first computes the value X through the following relation. $X \equiv C_1^{a_1x_1+a_2x_2} \ (\bmod p)$

Also, she computes the invers value $X^{-1}(\bmod p)$ of X . One can use extended Euclidean algorithm (EEA), as shown in section (2.5) in Chapter (2), for computing the inverse element modulo *p*. Finally, she computes the relation $X^{-1} * C_2 \equiv M \ (\bmod p)$ to recover a plaintext *M*.

$$\begin{aligned} C_2 * (C_1^a)^{-1} &\equiv M(A)^{max\, or\, min(k=k_1x_1+k_2x_2)} \\ &* ((g^{max\, or\, min(k=k_1x_1+k_2x_2)})^{max\, or\, min(a=a_1x_1+a_2x_2)})^{-1} \\ &\equiv M(g^{max\, or\, min(a=a_1x_1+a_2x_2)})^{max\, or\, min(k=k_1x_1+k_2x_2} \\ &* ((g^{max\, or\, min(a=a_1x_1+a_2x_2)})^{max\, or\, min(k=k_1x_1+k_2x_2)})^{-1} \equiv M \end{aligned}$$

Example 4.2.4.1 Alice and Bob agree to use the prime $p\ = 23$ and the generator element $g = 11$ and Alice chooses her secret key $a = 3x_1 - x_1$ and Bob selects his plaintext $M = 16$ and he chooses secret ephemeral key $k = 5x_1 - 4x_2$ and uses his private share secret constraints

$$\begin{aligned} 2x_1 + x_2 &\geq 2 \\ x_1 + 3x_2 &\leq 2 \\ x_2 &\leq 4 \\ x_1\ ,\ \ x_2 &\geq 0 \end{aligned}$$

Alice first computes her public key A by

$$A \equiv 11^{\max a=3x_1-x_2} \quad mod\ 23$$

S.t

$$\begin{aligned} 2x_1 + x_2 &\geq \\ x_1 + 3x_2 &\leq 2 \\ x_2 &\leq 4 \\ x_1\ ,\ \ \ \ \ x_2 &\geq 0 \end{aligned}$$

Phase I

Alice begins by adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And she is adding artificial variables to constraints of type bigger than and equal (≥) or equal (=).

she defines a new objective function *r* that depends on only the artificial variables (the sum of the artificial variables if the function is min). The sum of subtracting the artificial variables if the function is max

$$A \equiv 11^{\max r=-R_1}\ mod\ 23$$

s.t

$$2x_1 + x_2 - S_1 + R_1 = 2 \qquad \Longrightarrow \; R_1 = 2 - 2x_1 - x_2 + S_1$$
$$x_1 + 3x_2 + S_2 = 2$$
$$x_2 + S_3 = 4$$
$$x_1, x_2, S_1, S_2, S_3, R_1 \geq 0$$

With more details, computing *A*

$$A \equiv 11^{\max r = -2 + 2x_1 + x_2 - S_1} \quad mod\ 23$$

Setting objective function is zero

$$A \equiv 11^{\max r - 2x_1 - x_2 + S_1 = -2} \quad mod\ 23$$

Alice will form a prime table for its own objective function and shared secret constraints

Table 4.50: An initial value of new objective function and constraints for Alice for Example 4.2.4.1

	↓ x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
← R_1	$\boxed{2}$	1	-1	0	0	1	2
S_2	1	3	0	1	0	0	2
S_3	0	0	0	0	1	0	4
r	-2	-1	0	0	0	0	-2

Then she extracts the input variable x_1 that corresponds to the factor largest negative *M* value in the objective function,

then she extracts the output variable R_1, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (2) dividing R_1 by (2) to get inside variable x_1, to get

new S_2 - row = old S_2-row + (-1)(new x_1- row)

$$\begin{array}{rccccccc} old\ S_2 & (1 & 3 & 0 & 1 & 0 & 0 & 2) \\ +(-1)(1 & \tfrac{1}{2} & -\tfrac{1}{2} & 0 & 0 & \tfrac{1}{2} & 1) & \\ \hline \text{new}\, S_2 & (0 & \tfrac{5}{2} & \tfrac{1}{2} & 1 & 0 & -\tfrac{1}{2} & 1) \end{array}$$

to get new S$_3$ - row = old S$_3$ -row + (0) (new x_1- row)

factor is zero, so, the value of old S$_3$ will not change

to get new r - row = old r -row + (2) (new x_1- row)

$$\begin{array}{rccccccc} old\ r & (-2 & -1 & 0 & 1 & 0 & 0 & -2) \\ +(2)(1 & \tfrac{1}{2} & -\tfrac{1}{2} & 0 & 0 & \tfrac{1}{2} & 1) & \\ \hline \text{new}\, r & (0 & 0 & 0 & 0 & 0 & 1 & 0) \end{array}$$

Table 4.51: New values of objective function and constraint for Alice for Example 4.2.4.1

	x_1	x_2	S$_1$	S$_2$	S$_3$	R$_1$	R.H.S
x_1	1	1/2	-1/2	0	0	1/2	1
S$_2$	0	5/2	1/2	1	0	-1/2	1
S$_3$	0	0	0	0	1	0	4
r	0	0	0	0	0	1	0

She notes that the value of the $r = 0$, so she moves on to the next stage.

The second stage begins by deleting the artificial variables from the table of the first stage. Then they complete the solution by substituting the value of the variables into the original objective function and then finding them

Phase II

Table 4.52: Original objective function for Alice for Example 4.2.4.1

	x_1	x_2	S_1	S_2	S_3	R.H.S
x_1	1	1/2	-1/2	0	0	1
S_2	0	5/2	1/2	1	0	1
S_3	0	1	0	0	1	4
A	-3	1	0	0	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss Gordon method to get the new table.

*new a - row=old a – row + (3 * x_1 – row)*

new a - row = (-3 1 0 0 0 0)

+(3 3/2 -3/2 0 0 3)

(0 5/2 -3/2 0 0 3)

Table 4.53: New values of objective function and constraint for Alice for Example 4.2.4.1

	x_1	x_2	$\downarrow S_1$	S_2	S_3	R.H.S
x_1	1	1/2	-1/2	0	0	1
$\leftarrow S_2$	0	5/2	$\boxed{1/2}$	1	0	1
S_3	0	1	0	0	1	4
A	0	5/2	-3/2	0	0	3

S_1 is input variable, S_2 is output and (1/2) is common element

Table 4.54: New values of objective function and constraint for Alice for Example 4.2.4.1

	x_1	x_2	S_1	S2	S3	R.H.S
x_1	1	3	0	1	0	2
S_1	0	5	1	2	0	2
S2	0	1	0	0	1	4
A	0	10	0	3	0	6

Now, note that every value in previous table is positive and zero then we get

$$\therefore\ x_1 = 2\,, x_2 = 0 \longrightarrow \max a = 6$$

$$\begin{aligned} A &\equiv 11^6 \quad mod\ 23 \\ &\equiv 9 \qquad mod\ 23 \end{aligned}$$

Bob encrypts his message *M* through computing the ciphertext pair (C_1,C_2) by

$$C_1 \equiv 11^{\max k=5-4x_2} \quad mod\ 23$$

s.t

$$\begin{aligned} 2x_1 + x_2 &\geq \\ x_1 + 3x_2 &\leq 2 \\ x_2 &\leq 4 \\ x_1\ , \qquad x_2 &\geq 0 \end{aligned}$$

Phase I

Bob begins by adding slackness variables to constraints of type bigger and equal (≥) and less than and equal (≤). And he is adding artificial variables to constraints of type bigger than and equal (≥) or equal (=).

He defines a new objective function *r* that depends on only the artificial variables (the sum of the artificial variables if the function is min). The sum of subtracting the artificial variables if the function is max

$$C_1 \equiv 11^{\max r=-R_1} \quad mod\ 23$$

s.t

$$\begin{aligned} 2x_1 + x_2 - S_1 + R_1 &= 2 \qquad \Longrightarrow \quad R_1 = 2 - 2x_1 - x_2 + S_1 \\ x_1 + 3x_2 + S_2 &= 2 \\ x_2 + S_3 &= 4 \\ x_1, x_2, S_1, S_2, S_3, R_1 &\geq 0 \end{aligned}$$

With more details computing C$_1$

$$C_1 \equiv 11^{\max k=-2+2x_1+x_2+S_1} \quad mod\ 23$$

Setting the objective function to zero

$$C_1 \equiv 11^{\max k+2x_1+x_2-S_1=-2} \quad mod\ 23$$

Bob will form a prime table for its own objective function and shared secret constraints

Table 4.55: An initial value of new objective function and constraints for Bob for Example 4.2.4.1

	↓ x_1	x_2	S$_1$	S$_2$	S$_3$	R$_1$	R.H.S
← R_1	$\boxed{2}$	1	-1	0	0	1	2
S$_2$	1	3	0	1	0	0	2
S$_3$	0	1	0	0	1	0	4
r	-2	-1	0	1	0	0	-2

Then he extracts the input variable x_1 that corresponds to the largest negative in the objective function,

then he extracts the output variable R_1, which will be obtained by dividing the right-hand side by the coefficients of the input variable (the result of the least quotient will be output).

The common element is (2) dividing R_1 by (2) to get inside variable x_1, to get new S_2 - row = old S_2-row + (-1)(new x_1- row)

$$\begin{array}{rccccccc} old\ S_2 & (1 & 3 & 0 & 1 & 0 & 0 & 2) \\ +(-1)(1 & \tfrac{1}{2} & -\tfrac{1}{2} & 0 & 0 & \tfrac{1}{2} & 1) & \\ \hline \text{new} S_2 & (0 & \tfrac{5}{2} & \tfrac{1}{2} & 1 & 0 & -\tfrac{1}{2} & 1) \end{array}$$

to get new S_3 - row = old S_3 -row + (0) (new x_1- row)

factor is zero, so, the value of old S_3 will not change

to get new r - row = old r -row + (2) (new x_1- row)

$$\begin{array}{rccccccc} old\ r & (-2 & -1 & 0 & 1 & 0 & 0 & -2) \\ +(2)(1 & \tfrac{1}{2} & -\tfrac{1}{2} & 0 & 0 & \tfrac{1}{2} & 1) & \\ \hline \text{new} r & (0 & 0 & 0 & 0 & 0 & 1 & 0) \end{array}$$

Table 4.56: New values of objective function and constraint for Bob for Example 4.2.4.1

	x_1	x_2	S_1	S_2	S_3	R_1	R.H.S
x_1	1	1/2	-1/2	0	0	1/2	1
S_2	0	5/2	1/2	1	0	-1/2	1
S_3	0	1	0	0	1	0	4
r	0	0	0	0	0	1	0

He notes that the value of the $r = 0$, so he moves on to the next stage.

The second stage begins by deleting the artificial variables from the table of the first stage. Then they complete the solution by substituting the value of the variables into the original objective function and then finding them

Phase II

Table 4.57: Original objective function for Bob for Example 4.2.4.1

	x_1	x_2	S_1	S_2	S_3	R.H.S
x_1	1	1/2	-1/2	0	0	1
S_2	0	5/2	1/2	1	0	1
S_3	0	1	0	0	1	4
k	-5	4	0	0	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss Gordon method to get the new table.

*new k - row=old k – row + (5 * x_1 – row)*

New k - row = (-5 4 0 0 0 0)

$$\begin{array}{l} +(\ 5\ \ 5/2\ \ -5/2\ \ 0\ \ 0\ \ 5) \\ \hline \ \ 0\ \ 13/2\ \ -5/2\ \ 0\ \ 0\ \ 5 \end{array}$$

Table 4.58: New values of objective function and constraint for Bob for Example 4.2.4.1

	x_1	x_2	$\downarrow S_1$	S_2	S_3	R.H.S
x_1	1	1/2	-1/2	0	0	1
$\leftarrow S_2$	0	5/2	$\boxed{1/2}$	1	0	1
S_3	0	1	0	0	1	4
k	0	13/2	-5/2	0	0	3

Table 4.59: New values of objective function and constraint for Bob for Example 4.2.4.1

	x_1	x_2	S_1	S₂	S₃	R.H.S
x_1	1	3	0	1	0	2
S_1	0	5	1	2	0	2
S₂	0	1	0	0	1	4
k	0	23/2	0	5	0	10

Now, note that every value in previous table is positive and zero then we get

$$\therefore\ x_1 = 2\,, x_2 = 0 \longrightarrow \max k = 10$$

$$\begin{aligned} C_1 &\equiv 11^{10} && (mod\ 23) \\ &\equiv (5)^2 && (mod\ 23) \\ &\equiv 2 && (mod\ 23) \end{aligned}$$

and

$$\begin{aligned} C_2 &\equiv MA^{\max k} && (mod\ 23) \\ &\equiv 16 \cdot (9)^{10} && (mod\ 23) \\ &\equiv 16 \cdot 18 && (\ mod\ 23) \\ &\equiv 12 && (mod\ 23) \end{aligned}$$

Thus, the ciphertext (C_1, C_2) is (2,12)

Alice recovers the ciphertext (C_1, C_2) and recovers the original plaintext M by

$$\begin{aligned} M &\equiv (C_1^{maxa})^{-1} \cdot C_2 && (mod\ 23) \\ &\equiv (2^6)^{-1} \cdot 12 && (mod\ 23) \\ &\equiv (18)^{-1} \cdot 12 && (mod\ 23) \\ &\equiv (9) \cdot 12 && (mod\ 23) \\ &\equiv 16 && (\ mod\ 23) \end{aligned}$$

Example 4.2.4.2 Alice and Bob agree to use the prime $p\ = 29$ and the generator element $g = 5$ and Alice chooses her secret key $a = 4x_1 + x_2$ and Bob selects his plaintext $M = 25$ and he chooses secret ephemeral key $k = 3x_1 + x_2$ and uses his private share secret constraints

$$\begin{gathered} 3x_1 + \ \ x_2 = 3 \\ 4x_1 + 3x_2 \geq 6 \\ x_1 + 2x_2 \leq 3 \\ x_1\ , x_2\ \geq 0 \end{gathered}$$

Alice first computes her public key *A* by

$$A \equiv 5^{\min a = 4x_1 + x_2}\ (mod\ 29)$$

s.t

$$\begin{gathered} 3x_1 + \ \ x_2 = 3 \\ 4x_1 + 3x_2 \geq 6 \\ x_1 + 2x_2 \leq 3 \\ x_1\ , x_2\ \geq 0 \end{gathered}$$

Phase I

Alice begins by adding slackness variables. And she is adding artificial variables

She defines a new objective function *r* that depends on only the artificial variables.

$$A \equiv 5^{\min r = R_1 + R_2}\ \ (mod\ 29)$$

s.t

$$\begin{aligned} &3x_1 + x_2 + R_1 = 3 &&\longrightarrow R_1 = 3 - 3x_1 - x_2 \\ &4x_1 + 3x_2 - S_1 + R_2 = 6 &&\longrightarrow R_2 = 6 - 4x_1 - 3x_2 + S_1 \\ &\ x_1 + 2x_2 + S_2 = 3 && \\ &\ x_1\ , x_2\ , S_1\ , S_2\ , R_1\ , R_1 \geq 0 && \end{aligned}$$

With more details

$$A \equiv 5^{\min a=3-3x_1-x_2+6-4x_1-3x_2+S_1} \quad (mod\ 29)$$

$$\equiv 5^{\min a=-7x_1-4x_2+9+S_1} \quad (mod\ 29)$$

Setting the objective function to zero

$$A \equiv 5^{\min a+7x_1+4x_2-S_1=9} \quad (mod\ 29)$$

Alice will form a prime table for it is own objective function and share secret constraint

Table 4.60: An initial value of new objective function and constraints for Alice for Example 4.2.4.2

	↓ x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
← R_1	$\boxed{3}$	1	0	0	1	0	3
R_2	4	3	-1	0	0	1	6
S_3	1	2	0	1	0	0	3
r	7	4	-1	0	0	0	9

Then she extracts the input variable x_1 then she extracts the output variable R_1,

The common element is (3) dividing R_1 by (3) to get inside variable x_1

Table 4.61: New values of objective function and constraint for Alice for Example 4.2.4.2

	x_1	↓ x_2	S_1	S_2	R_1	R_2	R.H.S
x_1	1	1/3	0	1/3	0	0	1
← R_2	0	$\boxed{5/3}$	-1	-4/3	1	0	2
S_2	0	5/3	0	-1/3	0	1	2
r	0	5/3	-1	-7/3	0	0	2

Then she extracts the input variable x_2 then she extracts the output variable R_2, The common element is (5/3) dividing R_2 by (5/3) to get inside variable x_2

Table 4.62: New values of objective function and constraint for Alice for Example 4.2.4.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_1	1	0	1/5	3/5	-1/5	0	3/5
x_2	0	1	-3/5	-4/5	3/5	0	6/5
S_3	0	0	1	1	-1	1	0
r	0	0	1	-1	-1	0	0

She notes that the value of the $r = 0$, so he moves on to the next phase.

Phase II

Table 4.63: Original objective function for Alice for Example 4.2.4.2

	x_1	x_2	S_1	S_2	R.H.S
x_1	1	0	1/5	3/5	3/5
x_2	0	1	-3/5	-4/5	6/5
S_3	0	0	1	1	0
A	-4	-1	0	0	0

Once again, the coefficients of the exponential variables in the target function line must be zeros, so some initial transformations are done on the array by applying the Gauss Gordon method to get the new table

New a – row = old a – $\text{row} + (4 * \mathrm{x}_1 - \text{row}) + (\mathrm{x}_2 - \text{row})$

New a – row = (-4 -1 0 0 0)

+(4 0 4/5 12/5 12/5)

+(0 1 -3/5 -4/5 6/5)

(0 0 1/5 8/5 18/5)

Table 4.64: New values of objective function and constraint for Alice for Example 4.2.4.2

	x_1	x_2	S₁	S₂	R.H.S
x_1	1	0	1/5	3/5	3/5
x_2	0	1	-3/5	-4/5	6/5
S_3	0	0	1	1	0
A	0	0	1/5	8/5	18/5

Now, note that every value of previous table is positive and zero then we get

$$\therefore\ x_1 = \frac{3}{5}\ ,\ x_2 = \frac{6}{5}$$

$$\min a = \frac{18}{5}$$

$$A \equiv 5^{\frac{18}{5}} \quad (mod\ 29) \qquad \longrightarrow A \equiv 5^{18\cdot 5^{-1}} \quad (mod\ 29)$$

$$\begin{aligned} &\equiv 5^{18\cdot 6} && (mod\ 29) \\ &\equiv 5^{108} && (mod\ 29) \\ &\equiv (5^4)^{25}\cdot 5^8 && (mod\ 29) \\ &\equiv (16^5)^5\cdot 5^8 && (mod\ 29) \\ &\equiv 25\cdot 24 && (mod\ 29) \\ &\equiv 20 && (mod\ 29) \end{aligned}$$

Bob encrypts his message *M* through computing the ciphertext there (C_1, C_2) by

$$C_1 \equiv 5^{\min k = 3x_1 + x_2} \ \ (mod\ 29)$$

s.to

$$\begin{aligned} 3x_1 + \ x_2 &= 3 \\ 4x_1 + 3x_2 &\geq 6 \\ x_1 + 2x_2 &\leq 3 \\ x_1\, , x_2 &\geq 0 \end{aligned}$$

Phase I

Bob begins by adding slackness variables. And he is adding artificial variables

He defines a new objective function r that depends on only the artificial variables.

$$C_1 \equiv 5^{\min r = R_1 + R_2} \ (mod\ 29)$$

s.t

$$3x_1 + x_2 + R_1 = 3 \qquad \longrightarrow R_1 = 3 - 3x_1 - x_2$$

$$4x_1 + 3x_2 - S_1 + R_2 = 6 \qquad \longrightarrow R_2 = 6 - 4x_1 - 3x_2 + S_1$$

$$x_1 + 2x_2 + S_2 = 3$$

$$x_1\,, x_2\,, S_1\,, S_2\,, R_1\,, R_1 \geq 0$$

With more details compute C$_1$

$$C_1 \equiv 5^{\min k = 3 - 3x_1 - x_2 + 6 - 4x_1 - 3x_2 + S_1} \ (mod\ 29)$$

$$C_1 \equiv 5^{\min k = -7x_1 - 4x_2 + 9 + S_1} \qquad (mod\ 29)$$

Setting the objective function to zero

$$C_1 \equiv 5^{\min k + 7x_1 + 4x_2 - S_1 = 9} \qquad (mod\ 29)$$

Bob will form a prime table for it is own objective function and share secret constraint

Table 4.65: An initial value of new objective function and constraints for Bob for Example 4.2.4.2

	$\downarrow x_1$	x_2	S$_1$	S$_2$	R$_1$	R$_2$	R.H.S
$\leftarrow R_1$	$\boxed{3}$	1	0	0	1	0	3
R$_2$	4	3	-1	0	0	1	6
S$_2$	1	2	0	1	0	0	3
R	7	4	-1	0	0	0	9

Then he extracts the input variable x_1 then he extracts the output variable R_1,

The common element is (3) dividing R_1 by (3) to get inside variable x_1

Table 4.66: New values of objective function and constraint for Bob for Example 4.2.4.2

	x_1	$\downarrow x_2$	S_1	S_2	R_1	R_2	R.H.S
x_1	1	1/3	0	1/3	0	0	1
$\leftarrow R_2$	0	$\boxed{5/3}$	-1	-4/3	1	0	2
S_2	0	5/3	0	-1/3	0	1	2
R	0	5/3	-1	-7/3	0	0	2

Then he extracts the input variable x_2 then he extracts the output variable R_2,

The common element is (5/3) dividing R_1 by (5/3) to get inside variable

Table 4.67: New values of objective function and constraint for Bob for Example 4.2.4.2

	x_1	x_2	S_1	S_2	R_1	R_2	R.H.S
x_1	1	0	1/5	3/5	-1/5	0	3/5
x_2	0	1	-3/5	-4/5	3/5	0	6/5
S_2	0	0	1	1	-1	1	0
R	0	0	0	-1	-1	0	0

Phase II

Table 4.68: Origin objective function for Bob for Example 4.2.4.2

	x_1	x_2	S₁	S₂	R.H.S
x_1	1	0	1/5	3/5	3/5
x_2	0	1	-3/5	-4/5	6/5
S₂	0	0	1	1	0
k	-3	-1	0	0	0

New k – row = old k – row $+ \ (3 * \mathrm{x}_1 - \mathrm{row}) + (\mathrm{x}_2 - \mathrm{row})$

New k – row = (-3 -1 0 0 0)

+(3 0 3/5 9/5 9/5)

+(0 1 -3/5 -4/5 6/5)

0 0 0 1 3

Table 4.69: New values of objective function and constraint for Bob for Example 4.2.4.2

	x_1	x_2	S₁	S₂	R.H.S
x_1	1	0	1/5	3/5	3/5
x_2	0	1	-3/5	-4/5	6/5
S₂	0	0	1	1	0
K	0	0	0	1	3

Now, note that every value in previous table is positive and zero then we get

$$\therefore \ x_1 = \frac{3}{5} \quad , \quad x_2 = \frac{6}{5}$$
$$\min k = 3$$

$$\begin{aligned} C_1 &\equiv 5^{\min k} && (mod\ 29) \\ &\equiv 5^3 && (\ mod\ 29) \\ &\equiv 9 && (\ mod\ 29) \end{aligned}$$

and

$$\begin{aligned} C_2 &\equiv M(A)^{\min k} && (mod\ 29) \\ &\equiv 25 \cdot (20)^3 && (mod\ 29) \\ &\equiv 25 \cdot 25 && (mod\ 29) \\ &\equiv 16 && (mod\ 29) \end{aligned}$$

Thus, the ciphertext (C_1, C_2) is (9,16)

Alice recovers the ciphertext (C_1, C_2) and recover the original plaintext M by

$$\begin{aligned} M &\equiv \left(C_1^{\min a}\right)^{-1} \cdot C_2 && (mod\ 29) \\ &\equiv (9^{108})^{-1} \cdot 16 && (mod\ 29) \\ &\equiv ((9^4)^{25} \cdot (9)^8)^{-1} && (mod\ 29) \\ &\equiv ((7^5)^5 \cdot 9^8)^{-1} \cdot 16 && (mod\ 29) \\ &\equiv ((16^5) \cdot 9^8)^{-1} \cdot 16 && (mod\ 29) \\ &\equiv (23 \cdot 20)^{-1} \cdot 16 && (mod\ 29) \\ &\equiv (25)^{-1} \cdot 16 && (mod\ 29) \\ &\equiv 7 \cdot 16 && (mod\ 29) \\ &\equiv 25 && (mod\ 29). \end{aligned}$$

4.3. The Security Considerations on the ODHKE and OEPKC

The security considerations of new proposed ODHKE and OEPKC are determined based on the hard computation of the ODLFs. In comparison with original DHKE and EPKC that is depended on the DLP, the proposed version is more secure since it is depended on the possibilities to compute the exponent a1x1+a2x2 which represents the ODLF. There are many cases need to guess to compute correct a1x1+a2x2. In other words, the probability to choose a1 is 1/(p-1) and in similar way for choosing a2, x1 and x2, since these parameters are chosen from the range [1, p-1], where p is a large prime number. So, it is more difficult to determine the value of ODLF.

4.5 Comparison on the Optimized DL – Cryptosystems

In this section, the comparison between the proposed optimized DL-cryptosystems that is designed based on the optimization problems and the original DL-cryptosystems is discussed as follows.

4.5.1 Comparison between the DHKE and ODHKE

1. In comparison to the original DHKE which is depended on the DLP, one can find the proposed ODHKE is depended on the proposed ODLFs that is considered as hardest mathematical problem in comparing with the DLP, since it depends on more than one variable. So, to find the value of ODLF needs more and more possible cases to determine the correct one

2. ODHKE uses the linear optimization problems to compute the shared secret key SSK complicatedly, while a SSK in the DHKE is computed easily.

3. the computation of ciphertext on ODHKE is done complicatedly. The proposed ODHKE considers as a more secure protocol for communication schemes in compared to the original DHKE.

4. ODHKE is more secure compared to the origin DHKE.

5. The time required to decode in ODHKE is more than the time required to decode in DHKE.

4.5.2 Comparison between the EPKC and OEPKC

1. The original EPKC depends on the DLP while the proposed OEPKC is depended on the proposed ODLFs that is considered as hardest mathematical problem in comparing with the DLP. The ODLFs uses more than one variable.

So, finding the value of ODLF needs more and more possible cases to determine the correct one.

2. OEPKC uses the linear optimization problems to compute the public key PK complicatedly in compared to computing the PK in the original EPKC.

3. The computation of ciphertext on OEPKC is done complicatedly using different optimization methods. The proposed OEPKC considers as a more secure public key cryptosystem for communication schemes in compared to the original EPKC.

4. OEPKC is more secure compared to the origin EPKC.

5. The time required to decode in OEPKC is more than the time required to decode in EPKC.

Chapter Five

Conclusions and Future works

Chapter Five

Conclusions and Future works

5.1 Conclusions

A new computation of the hardness problem known as ODLF is used in this thiesis to propose an alternative version of the DHKE and EPKC algorithms. In the proposed ODHKE and OEPKC, SSK is computed based on the linear optimization problem using the graphical method and the algebraic method. Also, the simplified method as the Big-M method and the two-stage method as well are used to give other versions of ODHKE and OEPKC. The newly computed results are analyzed through some study cases. The security considerations on the proposed ODHKE and OEPKC are determined. The comparison between the proposed algorithms and original ones are discussed.

5.2 Future works

Some suggestions can be considered for future works:

1- New encryption schemes can also be suggested using other nonlinear optimization problem to solving strategies.

2- Other encryption schemes such that can also be used for procedures that are used in linear optimization issues.

REFERENCES

[1] Zaidan, B. B., Zaidan, A. A., Al-Frajat, A. K., & Jalab, H. A. (2010). On the differences between hiding information and cryptography techniques: An overview. Journal of Applied Sciences, 10(15), 1650-1655.

[2] Hassan, Soukaena., (2012). Data Security: University of Technology Computer Science Department.

[3] Harn, L., Mehta, M., & Hsin, W. J. (2004). Integrating Diffie-Hellman key exchange into the digital signature algorithm (DSA). IEEE communications letters, 8(3), 198-200.

[4] Kumar, C., & Vincent, P. D. R. (2017, November). Enhanced diffie-hellman algorithm for reliable key exchange. In IOP conference series: materials science and engineering (Vol. 263, No. 4, p. 042015). IOP Publishing.

[5] Khaldi, A. (2018). Diffie-Hellman key exchange through Steganographied images. Law, State and Telecommunications Review, 10(1), 147-160.

[6] Hussein, H. I., & Abduallah, W. M. (2021). An efficient ElGamal cryptosystem scheme. International Journal of Computers and Applications, 43(10), 1088-1094.

[7] Ahmed, J. M., & Ali, Z. M. (2011, July). The enhancement of computation technique by combining RSA and El-Gamal Cryptosystems. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics (pp. 1-5). IEEE.

[8] Hwang, M. S., Chang, C. C., & Hwang, K. F. (2002). An ElGamal-like cryptosystem for enciphering large messages. IEEE Transactions on Knowledge and Data Engineering, 14(2), 445-446.

[9] Bresson, E., Chevassut, O., & Pointcheval, D. (2001, December). Provably authenticated group Diffie-Hellman key exchange—the dynamic case. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 290-309). Springer, Berlin, Heidelberg.

[10] Bresson, E., Chevassut, O., & Pointcheval, D. (2002, April). Dynamic group Diffie-Hellman key exchange under standard assumptions. In International conference on the theory and applications of cryptographic techniques (pp. 321-336). Springer, Berlin, Heidelberg.

[11] Yang, C. C., Chang, T. Y., Li, J. W., & Hwang, M. S. (2003). Simple Generalized Group-Oriented Cryptosystems Using ElGamal Cryptosystem. Informatica, 14(1), 111-120.

[12] Bresson, E., Chevassut, O., & Pointcheval, D. (2007). Provably secure authenticated group Diffie-Hellman key exchange. ACM Transactions on Information and System Security (TISSEC), 10(3), 10-es.

[13] Li, N. (2010, April). Research on Diffie-Hellman key exchange protocol. In 2010 2nd International Conference on Computer Engineering and Technology (Vol. 4, pp. V4-634). IEEE.

[14] Sharma, P., Sharma, S., & Dhakar, R. S. (2011, September). Modified elgamal cryptosystem algorithm (MECA). In 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011) (pp. 439-443). IEEE.

[15] Mahalanobis, A. (2012). A simple generalization of the ElGamal cryptosystem to non-abelian groups II. Communications in Algebra, 40(9), 3583-3596.

[16] Rewagad, P., & Pawar, Y. (2013, April). Use of digital signature with diffie hellman key exchange and AES encryption algorithm to enhance data security in

cloud computing. In 2013 International Conference on Communication Systems and Network Technologies (pp. 437-439). IEEE.

[17] Ara, A., Al-Rodhaan, M., Tian, Y., & Al-Dhelaan, A. (2017). A secure privacy-preserving data aggregation scheme based on bilinear ElGamal cryptosystem for remote health monitoring systems. IEEE Access, 5, 12601-12617.

[18] Partala, J. (2018). Algebraic generalization of Diffie–Hellman key exchange. Journal of Mathematical Cryptology, 12(1), 1-21.

[19] Arboleda, E. R. (2019). Secure and fast chaotic el gamal cryptosystem. Int. J. Eng. Adv. Technol, 8(5), 1693-1699.

[20] Salem, F. M., & Amin, R. (2020). A privacy-preserving RFID authentication protocol based on El-Gamal cryptosystem for secure TMIS. Information sciences, 527, 382-393.

[21] Merkle, R. C. (1978). Secure communications over insecure channels. Communications of the ACM, 21(4), 294-299.

[22] Lidl, Rudolf, Introduction to finite fields and their applications. University of Tasmania, Launceston, Australia, 2000.

[23] Leslie, Martin, Elliptic Curve Cryptography, Advanced Combinatorics, 2006.

[24] Ling, San and Xing, Chaoping, Coding Theory. A First Course, Cambridge.

[25] Jeffery, Hoffstein, Jill, Pipher and Joseph, silverman H, An Introduction to Mathematical cryptography. Volume. 1. New York: springer, 2000.

[26] Kakish, Malek Jakob, Authenticated and secure El-Gamal cryptosystem over elliptic curve. International Journal of Recent Research and Applied Studies (IJRRAS) 10 (2), 2012.

[27] Kerl, John, Computation in finite fields. Arizona State University and Lockheed Martin Corporation, 2004.

[28] Idrees, Zunera, Elliptic Curves Cryptography. Master thesis. Linnaeus university, 2012.

[29] Burton, David M., Elementary number theory. University of New Hampshire, seventh edition, 2010.

[30] Hankerson, Darrel, Alfred J Menezes, and Scott Vanstone, Guide to elliptic curve cryptography. Springer-Verlag Professional Computing Series, 2004.

[31] Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). Guide to elliptic curve cryptography. Springer Science & Business Media.

[32] Merkle, R. C. (2019). Protocols for public key cryptosystems. In Secure communications and asymmetric cryptosystems (pp. 73-104). Routledge.

[33] Toradmalle, D. K., Muthukuru, J., & Sathyanarayana, B. (2018). Cryptanalysis of an Improved ECDSA. International Journal of Engineering Research and Technology, 11(4), 615-619.

[34] Agrawal, M., & Mishra, P. (2012). A comparative survey on symmetric key encryption techniques. International Journal on Computer Science and Engineering, 4(5), 877.

[35] Salman, A. K., & Al-Jilawi, A. S. (2021, March). On the Bridge Between Exterior and Interior Penalty Method. In Journal of Physics: Conference Series (Vol. 1818, No. 1, p. 012146). IOP Publishing.

[36] Mahto, D. G. (2014). Essentials of Operations Research-Chapter 3: Network Techniques (PERT & CPM). Available at SSRN 2887249.

[37] Taha, H. A. (2011). Operations research: an introduction (Vol. 790). Upper Saddle River, NJ, USA: Pearson/Prentice Hall.

[38] Hoffstein, J., Pipher, J., Silverman, J. H., & Silverman, J. H. (2008). An introduction to mathematical cryptography (Vol. 1). New York: springer.

[39] ElGamal, T. (1985). A subexponential-time algorithm for computing discrete logarithms over GF (p^ 2). *IEEE transactions on information theory*, *31*(4), 473-481.

الملخص

إن طريقة تبادل مفاتيح Diffie Hellman (DHKE) وطريقة تشفير المفتاح العام EL-Gamal (EPKC)هي خوارزميات تشفير مفتاح غير متماثل، والتي سنقترح مخططات جديدة لها بهدف زيادة أمان النظامين لحماية المعلومات السرية من الاختراق من خلال صعوبة الوصول إلى المفتاح السري والرقم الأولي. ومن خلال دراسة مشاكل التحسين العددي وجد الباحث صلة بين هاتين الطريقتين من خلال اقتراح مخططات تشفير غير متماثلة تعتمد على طرق حل مشاكل التحسين العددي التي تهدف إلى رفع مستوى الأمان لهذين النظامين. اعتمد الإصدار المقترح على وظائف اللوغاريتم المنفصلة المحسنة (ODLFs)التي تم تقديمها كتعريف جديد في هذا العمل. تعد الطريقة الرسومية، والطريقة المبسطة، وطريقةBig-M ، والطريقة ذات المرحلتين أدوات أساسية لحساب مفتاح سري مشترك (SSK) في طريقة ديفي هلمان لتبادل المفاتيح المحسنة ونظام تشفير المفتاح العام للجمال المحسنOEPKC .ان OEPKCو ODHKE هما بروتوكولات أكثر أمانًا لحساب SSK مقارنةً ببروتوكولات DHKE و EPKCالأصلية. يتم تحديد الاعتبارات الأمنية الخاصة بـ OEPKC و ODHKEالجديدتين المقترحتين على أساس الحساب الصعب لـ .ODLFs مقارنة بـEPKC و DHKE الأصلية.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل

كلية التربية للعلوم الصرفة

قسم الرياضيات

أنظمة التشفير غير المتماثلة المعتمدة على مسائل أمثلية عددية

رسالة

مقدمة الى مجلس كلية التربية للعلوم الصرفة في جامعة بابل كجزء من متطلبات نيل درجة الماجستير في التربية / الرياضيات

من قبل

هدى كاظم محمود كاظم

(بكالوريوس كلية التربية للعلوم الصرفة / قسم الرياضيات / جامعة بابل / ٢٠١١)

بأشراف

أ.م. د. رومى كريم خضر عجينة

١٤٤٤هـ ٢٠٢٢ م