A Secure Key Exchange Protocol and a Public Key Cryptosystem for Healthcare Systems

Yuvasri R, Manimaran A

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, India
E-mail: manimaran.a@vit.ac.in

Received: 16 November 2023; Revised: 10 April 2024; Accepted: 20 May 2024

Abstract: Diffie-Hellman (DH) is the reason for the existence of a solution to the key distribution problem, where two parties can mutually set up a shared secret key over an insecure channel without any previous communication. In this study, a novel key exchange protocol based on the difficulty of the Discrete Logarithm Problem with Factor Problem (DLPFP) utilizing matrices in non-commutative semigroup over semiring is proposed. The security and complexity analyses of the protocol are discussed. Also, an ElGamal cryptosystem based on a group of invertible matrices and DLPFP over a semiring to encrypt the Sensitive Health Information (SHI) in healthcare systems is suggested.

Keywords: public key cryptography, key exchange protocol, Diffie-Hellman, ElGamal cryptosystem, semiring

MSC: 11T71, 94A60, 06E20

1. Introduction

Cryptography is the study of private and confidential communication mechanisms that restrict the sender and intended recipient’s access to a message’s particulars. Recent years have seen the creation of a high volume of data from numerous sources, necessitating secure processing and storage. Personal information about the user and also ubiquitous data collected from many “everyday” gadgets that belong to the cyber world are included in this data. In the healthcare system, due to the ease of combining a variety of health information sources to build a centralized patient record that is quickly accessible, Sensitive Health Information (SHI) has gained popularity among users as a patient-centric model of health information sharing. A patient can create, maintain, and have control over all of her personal health information through a SHI service, which has improved the efficiency of data storage, retrieval, and sharing. The data owner has full control over the SHI. They can also make their health information available to a variety of consumers. A number of privacy and security issues arise during the sharing process, potentially preventing its widespread adoption. The privacy of users is in jeopardy if SHI is kept on a server operated by an unreliable third party. Therefore, encrypting the data is vital. Figure 1 shows the scenario for the healthcare system.

Copyright ©2024 Manimaran A, et al.
DOI: https://doi.org/10.37256/cm.5220243942
This is an open-access article distributed under a CC BY license (Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/
In the communication between the sender and the receiver, the key exchange algorithm is used to generate the secret key. The safe and secure transfer of the key between the entities is the most difficult task. While exchanging the keys, unauthorized users should not be able to access the information. Whitfield Diffie, Hellman, and Martin Hellman developed the Diffie-Hellman (DH) algorithm in 1976, which facilitated the exchange of private keys between the parties. This algorithm employs the platform group $F_p^*$, which is the group of integers modulo $p$ under multiplication, where $p$ is prime. It has been proposed that the Discrete Logarithm Problem (DLP) and subsequent integer factorization problems are key exchange protocols with essential structures that are quite similar to the DH technique previously outlined.

The intractability of problems related to number theory forms the foundation for all of these cryptosystems. Despite their widespread use, these public key techniques suffer from two major flaws. The first is quantum computing’s rapid advancement. The creation of a reasonably powerful quantum computer will render certain frequently used public key algorithms, like the Rivest Shamir and Adleman (RSA) algorithm, ElGamal, and elliptic curve cryptosystems, no longer secure. The second issue arises from the incompatibility of these number theory-based approaches with small computer hardware, like card readers with low processing speeds.

Despite the fact that quantum computers are still many generations away from being a reality, the cryptography community will soon be on guard against them. As a result, developing faster and more secure non-numerical theory-based cryptographic systems and key exchange methods is critical. In [1], a more effective new type of elliptic curve public key cryptosystem featuring quicker key generation calculations is presented. The Conjugacy Search Problem (CSP) [2] over a set of invertible matrices for groups and semigroups is one instance of a challenge that has inspired multiple attempts to build different public key cryptosystems.

In [3], a novel hill cipher is suggested based on vector spaces over non-singular (invertible) matrices. It resolves the known plaintext attack issue. A message exchange protocol based on finite associative-commutative rings with unity characteristics is described in [4]. For the digital signature system described in [5, 6], the general linear group over $Z_n$ and the non-commutative ring are the building blocks. A public key symmetric cipher approach that uses semiring action is proposed in [7]. However, the authors do not explicitly state the security of this scheme. Stickel’s key exchange protocol in tropical algebra was updated by the authors [8] utilizing commuting matrices, and they also proposed additional protocols.
The basis for our study is a non-commutative group of invertible matrices over semirings. As a result of including all the highlights, our suggested scheme’s security has improved.

Based on the premise of a strong computational DH without sacrificing security, [9] offers a tight security key exchange technique. A new modified and enhanced DH non-commutative key exchange mechanism is proposed in [10]. It protects against linear algebra attacks as well as both linear and non-linear decomposition attacks. For increased data privacy and integrity, a revised encryption technique built on the DH approach is suggested in [11]. It protects against man-in-the-middle attacks and discrete logarithm attacks. Only a few known attacks explain the security of these schemes, as they fail to overcome many others.

In [12, 13], a modified algorithm based on the ElGamal and Cramer-Shoups cryptosystems is presented. In [14], a simple, lightweight, and efficient encryption scheme is given. An updatable ElGamal encryption scheme that achieves forward and backward security for cloud storage is proposed in [15]. In [16–18], an ElGamal scheme and key exchange protocols for the Internet of Things environment are designed. In [19, 20], cryptosystems using blockchain are proposed. Some encryption schemes for enhancing the security of healthcare systems are given in [21–24].

Our protocol is based on a set of invertible matrices over a finite field. A non-invertible key exchange protocol over a ring is suggested by the author in [25] and affirms that it takes less time than DH, ElGamal, and RSA. In the view of [2, 26], the DLP on \( GL_n(F_r) \) is converted into a basic factoring issue or a DLP over finite fields [26] due to attacks like determinant attack, eigenvalue attack, and the Cayley-Hamilton attack. A platform is suggested to get around this conversion problem. It is made up of the semigroup of matrices over group ring \( M_n \times \times (F_r[S_s]) \) under normal matrix multiplication [27] and the group of invertible matrices over group ring \( GL_n(F_r[S_s]) \) [28]. [29] claims that the “semi-simple algebra” structure of group rings can decipher the key exchange protocols based on DLP. [30] suggests a Factorization Discrete Logarithm Problem (FDLP)-based key exchange protocol and analyses the security and complexity. Also, an ElGamal cryptosystem is proposed. In [31], this protocol’s security is examined thoroughly. A modified RSA algorithm to encrypt the SHI is proposed in [32]. Our approach is based on a group of invertible matrices over semiring, which is a more feasible generalization of the group ring. Our approach fixes the flaws in the previously discussed systems.

Recently, several attempts were made to secure the SHI by using cryptographic protocols. However, to our best knowledge, a few of them have provided these protocols based on ElGamal cryptosystems and mathematical hard problems. In [30], a protocol based on the group of invertible matrices over group rings is presented. Semiring is a generalized concept of group rings. This motivates us to design a new problem, Discrete Logarithm Problem with Factor Problem (DLPFP), upon which we construct a key exchange protocol and an ElGamal cryptosystem using matrices in the non-commutative semigroup over semiring to secure the SHI. The proposed protocol is more secure but has the same time complexity. Also, we have enhanced our scheme by concealing the subsemigroup.

In this study, we attach a hidden parameter to the ideas of the DLP and Factor problem (FP) over a non-commutative semigroup \( M_n \times \times (R_r) \), which becomes the DLPFP problem. Using the above-given attacks does not reveal the secret key, making the suggested protocol more efficient than previously published protocols. Then, a key exchange protocol is proposed based on this DLPFP. We have presented the protocol’s complexity and security measures. Also, based on DLPFP over a group of invertible matrices, an ElGamal cryptosystem for encrypting the data in the healthcare system is proposed.

The order of the paper’s sections is as follows: The terms DLP, FP, DLPFP, centralizer, and semiring are defined, and the model, structure, flow, and security goals of the system and the list of notations used are given in Section 2. In Section 3, a key exchange protocol based on DLPFP is proposed along with a theorem. In Section 4, the proposed protocol’s security and complexity are evaluated. Using the group of invertible matrices over a semiring, an ElGamal cryptosystem for SHI is proposed, with an example in Section 5. Lastly, in Section 6, the paper is concluded.

2. Preliminaries

In this section, DLP, FP, DLPFP, Centralizer, and Semiring are defined [30, 33, 34], and the model, structure, flow, and security goals of the system are given.
2.1 Basic definitions

**Definition 1** (DLP [30]) Let \( p \) be a prime and given an element \( \beta \in F_p^* \) where \( F_p^* \) is a cyclic group of order \( p - 1 \) generated by \( \alpha \), find an integer \( 0 \leq e \leq p - 1 \) such that \( \alpha^e \equiv \beta \mod p \).

**Definition 2** (FP [30]) Given an element \( g \) from non-abelian semigroup \( G \) and two subsemigroup of \( G \) namely \( H_1 \), and \( H_2 \), find two elements \( h_1 \in H_1 \) and \( h_2 \in H_2 \) such that \( g = h_1 \cdot h_2 \).

**Definition 3** (DLPFP) Given an element \( g \) from non-commutative semigroup \( G \) and two subsemigroup of \( G \) namely \( H_1 \), and \( H_2 \), find three elements \( h_2 \in H_2 \) and \( t, s \in \mathbb{Z}^* \) where \( r \) is an integer such that \( g = h_1^t \cdot h_2^s \).

**Example** Let \( GL_3(\mathbb{Z}) \) be a non-abelian semigroup and the subsemigroup of \( GL_3(\mathbb{Z}) \) be the set of all \( 3 \times 3 \) diagonal matrices. Also, let \( \mathbb{Z}^* \) be \( \mathbb{Z}^*_{11} \). Consider the equation \( g = h_1^t \cdot h_2^s \) where the given values are \( t = 2 \), \( s = 3 \),

\[
g = \begin{bmatrix} 733 & 1,253 & 1,238 \\ 420 & 700 & 704 \\ 207 & 355 & 350 \end{bmatrix} \in GL_3(\mathbb{Z}),
\]

\[
h_1 = \begin{bmatrix} 3 & 4 & 1 \\ 1 & 4 & 0 \\ 1 & 1 & 0 \end{bmatrix} \in GL_3(\mathbb{Z}),
\]

\[
h_2 = \begin{bmatrix} 2 & 5 & 3 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \in GL_3(\mathbb{Z}).
\]

Now, the problem is to find the values of \( h_2 \), \( t \), and \( s \) from the above equation.

**Definition 4** [33] For an element \( g \in G \), let \( C(g) \) be the set of elements that commute with \( g \), i.e., \( C(g) = \{a \in G : ga = ag\} \). \( C(g) \) is called the centralizer of \( g \) in \( G \).

**Definition 5** [34] A non-empty set \( R \), is called a semiring if the operations of addition and multiplication are defined in the following ways:

(i) \((R, +)\) is a semigroup;

(ii) \((R, \cdot)\) is a semigroup;

(iii) \( a(b + c) = ab + ac \) and \( (b + c)a = ba + ca \) for every \( a, b, c \in R \).

2.2 System model

The proposed system model consists of the data owner, i.e., the Patient (\( P_a \)), Personal Domain (\( P_eD \)), Public Domain (\( P_uD \)), Administrator (\( A_d \)), and Big Data Server (\( S \)). The categorizations of the model are given in Figure 2 and are described as follows:

\( P_a \): The details of \( P_a \) like name, date of birth, gender, address, contact number, disease, etc., are registered in the system and are stored in \( S \).

\( P_eD \): The domain includes family members and friends who are very close to \( P_a \). They are granted access rights by \( P_a \) in accordance with the required conditions.

\( P_uD \): The domain includes parties like insurance providers, additional hospitals, etc. The parties cannot access the SHI until \( P_a \) gives permission.

\( A_d \): First, when \( P_a \) enters the hospital for treatment or consultation, he or she is asked to provide the details for their profile and also register. The registration ID and private key (updated each time) are provided to \( P_a \) by \( A_d \).
The SHI of \( P_a \) is stored here. \( S \) makes storing, processing, and analyzing enormous amounts of data simple. In this paper, the server is considered semi-honest.

![System Model](image)

**Figure 2.** The system model

### 2.3 Structure and flow of the proposed protocol

Three roles are specified for handling \( P_a \) health data: a doctor, a nurse, and a staff member. A doctor has the ability to add, edit, or remove the treatment records of \( P_a \). He or she is granted access to the entire medical records (SHI) of the \( P_a \)s that are allocated to them. A nurse also has access to and control over SHI, but with fewer privileges this time. In particular, a nurse is able to add information to the SHI. However, their access to the visit histories of \( P_a \) is limited, preventing them from viewing specific medical records of doctors assigned to them. The third role owner, i.e., the staff member, can view the basic information, including his identity, visit date, time, and cost. The staff allocates the \( P_a \) to a suitable and accessible healthcare professional and keeps track of how long they wait. The \( A_d \) is in charge of creating roles, assigning users to these jobs, and handling role granting and revocation, for example, by recruiting or firing doctors, nurses, and other staff members. The workflow of the proposed protocol is described as follows:

During the first visit, \( P_a \) provides \( A_d \) with profile details. \( A_d \) uploads it in \( S \). When registering, \( P_a \) specifies the members of \( P_aD \). The access privileges of \( P_aD \) members are specified by \( P_a \). \( P_a \) also clearly knows about the individuals who have access to \( P_aD \). If someone wants to access SHI, then a secret key request is sent to \( P_a \), and he or she approves or declines the request. The SHI is encrypted under a domain-based access policy and stored in \( S \). It is decrypted by \( P_a \), or the authorized user with the private key. The users are those who are specified in the public or private domain with access rights. The individuals from \( P_aD \) are close friends and family members. Therefore, \( P_a \) allows them to access the data with respect to their relationship.
2.4 Security goals

The main aim of the patient-centered scheme is that the $P_a$ should have control over his or her SHI. In order to achieve this, the system needs to maintain data confidentiality, integrity, adaptability, scalability, and efficiency. They are described as follows:

1. Confidentiality: The decryption of the SHI file is done only by authorized individuals, as mentioned by $P_a$. Unauthorized individuals lacking the necessary access privileges cannot decrypt the SHI.
2. Integrity: Preventing unauthorized individuals from gaining access to modify the data.
3. Adaptability: It is necessary for the access policy to be flexible when it is characterized by constant change.
4. Scalability: The system must encourage the $P_a D$ and $P_a D$ users. Even if the size of the $P_a D$ increases, the system must support it. Also, high adaptability and applicability are required.
5. Efficiency: It is necessary to minimize the effort required to handle the clients and keys.

3. A key exchange protocol based on DLPFP

In this section, a DLPFP-based key exchange protocol that employs matrices over semiring is proposed, and a theorem is established.

The Protocol: Assume that under multiplication, $A = M_n \times n(R_i)$ is a finite non-commutative semigroup. Let $B$ be a commutative subsemigroup of $A$, and $F^* = \{1, 2, 3, \ldots, r - 1\}$, where $r$ is a big positive integer. Suppose $a \in A$ and $C_A(a)$ is the centralizer of $a$ in $A$. The public parameters are $A$, $a$, and $r$. Let the parties involved in the key exchange protocol be Alice and Bob, who know the public information. The following is the proposed protocol:

1. Alice computes $C_1 = d^i y^j$, where the secret parameters $i$, $j$, and $y$ are chosen from $F^*$ and $B \setminus C_A(a)$, respectively. $B \setminus C_A(a)$ contains all the elements in $B$ except the ones that commute with $a$. Alice sends $C_1$ to Bob.

2. Bob computes $C_2 = d^k z^l$ where the secret parameters $k$, $l$, and $z$ are chosen from $F^*$ and $B \setminus C_A(a)$, respectively. Bob sends $C_2$ to Alice.

3. Alice calculates $K_1 = a^i d^i y^j$.

4. Bob calculates $K_2 = a^k d^k z^l$. The exchanged secret key ($K$) is $K_1 = a^i d^i y^j = a^k y^j d^k = a^i C_1 y^j = K_2 = K$ because of the commutative property of $y$ and $z$, that is, $yz = zy$ in $B$.

In the healthcare system scenario, the key exchange takes place between the $P_a$ or the users from $P_a D$ or $P_a D$ (whoever requests the SHI of $P_a$) and the $A_d$. The $P_a$ or the users from $P_a D$ perform Alice’s part during the key exchange. Bob’s part is done by the $A_d$. In order to create a different key each time, the matrix $y$ is calculated as follows: The alphabets in the $P_a$ and the consulting doctor’s password are converted into numerical values, and the corresponding binary value is calculated. Now, the first binary value of $P_a$’s password and the doctor’s password are added using the logical operator $\oplus$. Then, the resulting binary value is again converted into a numerical value, which is the first entry of the matrix $y$. The same method is followed to calculate each value. According to the size of the matrix $y$, entries are added or deleted randomly.

Remark The constraint that $y$ and $z$ do not commute with $a$ cannot be modified since if $ay = ya$ or $az = za$, then the secret key is easily found by the adversary because

$$K_1 = d^i C_2 y^j = d^i a^k d^k y^j = d^i y^j a^k = C_1 C_2$$

or the other way, that is,

$$K_2 = d^k C_1 z^l = d^k a^i d^i y^j = d^k z^l a^i = C_2 C_1.$$
This is because the values of $C_1$ and $C_2$ are sent through an insecure channel.

**Remark** Let us assume that $za = az$ and $y, z \in B \setminus C_A(a)$. $C_A(a)$ is the centralizer of $a$ in $A$, and it contains all the elements in $A$ that commute with $a$. Here, $B \setminus C_A(a)$ contains all the elements in $B$ except the ones that commute with $a$. This contradicts our assumption. Therefore, $za \neq az$. Also, $y$ and $z$ belong to $B \setminus C_A(a)$, where $B$ is a commutative subsemigroup. Therefore, $yz = zy$.

**Remark** In the above protocol, another way of choosing $y$ is from $A \setminus C_A(a)$ and proceeding with another modification, i.e., by selecting the secret polynomials $F_i[x]$ such that $f(x)$ and $g(x)$ commute with each other and the key is also calculated in the same way, that is

$$C_1 = a^i f(y)^j \text{ and } C_2 = a^j g(y)^i.$$

**Theorem 1** Let $A = M_n \times n(R_s)$ be a finite non-commutative semigroup under multiplication over semiring $R_s$ and let $F_r = \{1, 2, 3, \ldots, r - 1\}$, where $r$ is a big positive integer. If the key exchange protocol is based on $DLPFP$, then the keys are exchanged securely through the protocol when $B$ is a commutative subsemigroup of $A$.

**Proof.** Given that $A = M_n \times n(R_s)$ is a finite non-commutative semigroup over semiring, under usual matrix multiplication, and $B$ is a subsemigroup of $A$, where $B = \{\{n \times n(R_s)\} \cap F_r = \{1, 2, 3, \ldots, r - 1\}$ where $r$ is a big positive integer and $a \in A$, and $y, z \in B \setminus C_A(a)$.

To prove: The key exchange is securely done based on $DLPFP$ when $B$ is chosen as a commutative subsemigroup of $A$ and $a \in A$, and $y, z \in B \setminus C_A(a)$.

According to the key exchange protocol, let $a \in A$ and $C_A(a)$ be the centralizer of $a$ in $A$. Now, the parameters in the public domain are $A, a$, and $r$. Assume that the people transferring keys are Alice and Bob.

The key exchange is as given in the above Key Exchange Protocol based on $DLPFP$. The exchanged secret key $K$ is $K_1 = a^i C_2 y^j = a^{i+k} z^j y^i$, and $K_2 = a^j C_1 z^i = a^k y^i z^j$. Therefore, $K_1 = K_2 = K$ (since $B$ is commutative). The key exchange is successful.

Let us assume that the key exchange is insecure, i.e., with knowledge of the exchanged values ($C_1, C_2$) and the public parameters ($A, a$, and $r$), the adversary can deduce the secret keys. To compute the secret key $K_1 = a^i a^j a^k y^i z^j$ with the known values, the adversary calculates $C_1 C_2 = a^i y^i a^k z^j$. To equate the values of $C_1 C_2$ and $K_1$, $y$ must commute with $a$. This, however, contradicts our assumption that $ya \neq ay$. So even if the adversary intercepts the exchanged values and knows the public parameters, deducing the secret keys remains infeasible due to non-commutative property of the elements involved. Therefore, the key exchange is secure. 

\[\square\]

### 4. The security analysis and complexity analysis

In this section, the security analysis of the $DLPFP$-based key exchange protocol is examined, and the number of operations required to run this protocol is analyzed along with the time complexity graph.

#### 4.1 The security analysis

The security and time complexity of the suggested $DLPFP$ key exchange protocols are examined in this section.

Assuming that the above problem is computationally difficult, our Section 3 protocol is secure: It is difficult to calculate the shared key $K$ given the public data $a$, $C_1$, and $C_2$. This is the protocol’s computational assumption. The tougher decisional version of this condition is: It is challenging to tell the shared key $K$ from an arbitrary element of the type $a^m$ given $a$, $C_1$, and $C_2$. The security evaluation of our protocol against many forms of attacks, some of which are published in [2, 26, 29, 35], is provided below:

(i) Attacks by using matrices’ characteristics: FDLP is resistant to the Cayley-Hamilton attack [26], the determinant attack [2], and the eigenvalue attack. The eigenvalue attack is irrelevant since $a$’s eigenvalues are identified but not those
of $a^i$. The non-commutative determinant prevents the determinant attack from working [2]. In contrast to situations where DLP is the root cause of the issue, det($a'y^i$) and det($a$) are known, but currently, there are three unknown variables ($i$, $j$, and $y$). Since $y$ is unknown, this approach is not suitable to determine $i$ or $j$. To use the Cayley-Hamilton attack, you must know $a, y \in M_n \times R$ such that $a'y^i = g(a)y$, such that $g(x), h(x) \in R$, and

$$g(x) = \sum_{i=0}^{r-1} b_i a^i, \quad h(y) = \sum_{i=0}^{r-1} c_i y^i.$$  

Since $y$ and $j$ are not known, it is unattainable to detect $h(y)$. Therefore, the Cayley-Hamilton attack cannot solve this.

(ii) Attacks over DLP: To solve the DLP, numerous algorithms were introduced. Our underlying problem is DLP, and the algorithms cannot be used in our protocol since in DLP, $a$ is known and $a'$ is unknown, so it was broken through the algorithms. But, here, “r” is hidden using two parameters, $y$ and $j$, which are confidential such that $C_1 = a'y^i$ and the subsemigroup $B$ is hidden as given in [33], which makes it more difficult for the adversary to attack.

(iii) Brute force attack: For given $a_1, a_2$, and $a_3 \in A$ and $t, s \in F_+$, $a_1 = a_2^t a_3^s$ and in $F_*$, $r$ is a big positive integer, where $A = \{d_1, d_2, \ldots, d_k\}$ is a non-commutative semigroup of $\alpha$ elements. The elements of $A$ are expressed as $a_1 = d_1, a_2 = d_2$, and $a_3 = d_3$ for some $i, j, l \in \{1, 2, \ldots, \alpha\}$. $\alpha$ is the number of options for $a_3$, and $r$ is the number of options for $a_1$ and $a_2$. By brute force attack, to solve the DLPFP, the steps involved are $O(\alpha r) = \alpha r$ bits, which is exponential. $\alpha r = ^{\log_2 \alpha r} = ^{\log_2 2 \log_2 \alpha r} = ^{f \log_2 \alpha}$ and $f = \log_2 2$. This grows reasonably fast as the values of $\alpha$ and $r$ increase. Therefore, brute force attack is not possible.

Algorithm 1 Exhaustive search algorithm

**Input:** Let $a \in A$ such that $C_1 = a'y^i$, and $C_2 = a'h^j$  

**Output:** Secret parameters: $B \subset A, y, z \in B \backslash c_A(a), i, j, k, l \in F_+, K_1$, and $K_2$  

for $i \leftarrow 1$ to $r - 1$ do  

for $j \leftarrow 1$ to $r - 1$ do  

$K_1 \leftarrow a'y^i$;  

for $k \leftarrow 1$ to $r - 1$ do  

for $l \leftarrow 1$ to $r - 1$ do  

$K_2 \leftarrow a'h^j$;  

compare $K_1 = K_2$;  

if $K_1 = K_2$ then  

return $K_1 = K_2$ & exit;  

else  

go to next step;  

end if  

end for  

end for  

end for  

end for  

end for  

For the key exchange protocol based on DLPFP, an exhaustive search method is provided in Algorithm 1.

(iv) Linear algebra attack: Only in the following way, our strategy is compatible with the linear algebra attack described in [36]:

1. Find a matrix $u$ with the property $ua = au$.
2. Find a matrix $v$ with the condition $vz = vz$. Because it is abelian, such a matrix $v$ is chosen from the hidden subsemigroup $B$.
3. \( C_1, u, \) and \( v \) should satisfy the equation \( C_1 = uv. \)

Then, the secret key can be evaluated as:

\[
u C_2 v = u a^l v = a^l uvz^j = a^l C_1 z^j.
\]

This value matches the secret key. At least one solution exists for this method, i.e., \( u = a^l, v = y^j. \) The solution for the group of matrices over semirings is given below.

1. If \( u \) such that \( ua = au; \) Let us consider the set of all \( 2 \times 2 \) matrices. Suppose the adversary knows the matrix \( a = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \) and has to find the unknown matrix \( u = \begin{bmatrix} u_1 & u_2 \\ u_3 & u_4 \end{bmatrix} \) so that it satisfies \( ua = au. \) Now, we have four equations while solving \( ua = au, \) i.e., \( a_1 u_1 + a_2 u_3 = u_1 a_1 + u_2 a_3, a_1 u_2 + a_2 u_4 = u_1 a_2 + u_2 a_4, a_3 u_1 + a_4 u_3 = u_3 a_1 + u_4 a_3, \) and \( a_3 u_2 + a_4 u_4 = u_3 a_2 + u_4 a_2. \) If the elements are taken from a commutative group, they are linear equations with variables that are unknown, i.e., \( a_2 u_3 = u_2 a_3, a_1 u_2 + (u_4 - u_1) a_2 - u_2 a_4 = 0, a_3 u_1 + (a_4 - a_1) u_3 - u_4 a_3 = 0, \) and \( a_3 u_2 - u_3 a_2 = 0. \) This reduces to \( q = 0 \) for some matrix \( q \) and \( t = [u_1 u_2 \ldots u_6]^T \) where \( u_1, u_2, \ldots, u_6 \in u. \) If the elements are taken from a non-commutative group, then it is not possible for the equations to be linear since \( u a \neq au. \) As given in (iii) above, a brute force attack is not applicable here, even if the equations are written in linear form, because by brute force attack, the number of steps required is \( O(\alpha r) \) bits (exponential), and it grows reasonably fast as the size of the platform increases. Also, invertibility is not mandatory for the matrix \( q. \) The restrictions we have made disable the attacks on finding \( u. \)

2. Also, the basic platform where \( y, z \) is taken from \( (B) \) is hidden. Therefore, the adversary remains unaware of the origin of these matrices.

3. To find \( v \) such that \( vz = zv; \) To find \( v \) from the hidden subsemigroup \( B \) is not possible as discussed in \([30] \) and (1) above. It will also not work out, as in the case of Remark 2, since the commutative subsemigroup \( B \) is hidden.

4. Solving the equation \( C_1 = uv; \) The known matrix is \( C_1. \) The matrices \( u \) and \( v \) are unknown, where \( u = a^l \) and \( v = y^j \) (since \( C_1 = x^l y^j). \) The secret parameters are \( y, i, \) and \( j. \) It is considered a hard problem.

A system of quadratic equations is produced by solving the matrices, as shown in \([30] \). The only way to find the resolution is by using a brute force attack. However, convincing evidence has demonstrated that the time complexity of solving quadratic equations remains safe from this attack.

5. Ciphertext only attack: The value of \( C_2 = a^l z^j \) is used as the ciphertext \( C_2 \) in the ElGamal Public Key Cryptosystem \( 5. \) Therefore, the ciphertext only attack can be used on \( C_2 \) here. In the ciphertext only attack, the adversary only knows \( C_1 \) or \( C_2. \) An adversary knows only the matrices \( C_1, C_2, \) and \( a, \) which are public parameters. To find the plaintext \( o, \) he must first find the unknown matrices \( y, z, K_1, \) and \( K_2. \) He assumes random matrices \( y = y_1 \) and \( z = z_1. \)

Therefore, \( C_1 = d^l y^j_1. \) In the same way, \( C_2 = a^l z^j_1. \) Here, he has a large system of nonlinear equations for the ciphertext corresponding to each plaintext. He finds the corresponding solution, \( o = o^j. \) Thus, for each fixed \( y = y_1, \) the adversary gets a number of \((C'_1, C'_2, z_1, o^j). \) This process generates a large system of equations with a large number of unknowns. No matter how an adversary rearranges these equations, the problem of having a product of two unknown matrices persists. This solution becomes infeasible.

6. Known plaintext attack: Here, the adversary knows the ciphertext \( C = (C_1, C_2) \) corresponding to the plaintext \( o_i \) for \((i = 2, 3, 4, \ldots, j). \) Let the plaintext-ciphertext pair be \((o, C). \) He wants to find the next plaintext, \( o_{i+1}, \) that corresponds to the ciphertext, \( C_{i+1}, \) from this plaintext-ciphertext pair. In the proposed public key cryptosystem, these types of attacks are impossible because different keys are used to encrypt every new plaintext. Hence, it does not provide any information about the next unknown plaintext-ciphertext pair. As a result, we have demonstrated that our proposed cryptosystem is secure against known plaintext attack. As discussed earlier, brute force attack is also impossible.

7. Man in the middle attack: In \([33] \), the man in the middle attack is beaten by two enrichments. We conceal the subsemigroup \( B \) and discreetly select the matrix \( y \) from \( B \) and the matrix \( z \) from \( B, \) respectively, to enhance the security of our proposed system. The location where \( y \) and \( z \) are selected from is not reflected here, making it challenging for the foe to locate \( B. \) So, as previously described, it is a secure protocol.
4.2 The complexity analysis

The time taken to perform the DLPFP algorithm is evaluated in this subsection. The computational time to calculate the shared key, i.e., the number of operations essential in bits, is determined as follows:

1. To multiply two matrices of rank \( n \), \( O(n^3) \) bit operations are required.
2. To calculate the matrix \( a \) raised to the power of \( i \), it takes \( n^3 \log r \) number of bit operations.
3. In order to determine \( C_1 = a^i y^j \), it is necessary to do \( n^3 \log r + n^3 \log r + 3 = 2n^3 \log r + n^3 \) bit operations.
4. To evaluate \( a^k C_1 z^l \), the number of bit operations needed is \( n^3 \log r + 2n^3 \log r + n^3 + n^3 \log r + 2n^3 = 4n^3 \log r + 3n^3 \).
5. To calculate the secret key, the total number of bit operations required is \( 2n^3 \log r + n^3 + 4n^3 \log r + 2n^3 = 6n^3 \log r + 3n^3 \). \( 6n^3 \log r + 3n^3 \) is proportionally equal to \( O(n^3 \log r) \).

The time complexity to calculate the key \( K \) in the DH process requires \( O((\log_2 r)^3) \) bit operations where the elements are taken from the commutative group \( F_r \). Here, the time complexity is \( O(n^3 \log r) \) because the entries of the key are taken from the semiring, and to evaluate the shared key, four matrices are to be multiplied. The semigroup \( M_{n \times n}(R_r) \) has \( \eta \) elements, and the matrices are simply raised to the powers of the \( F_r \) elements. So, the time complexity for an exhaustive search is \( O(\eta r) \). The protocol is strong enough to withstand attacks using eigenvalues and determinants [2] because it employs the semigroup \( M_{n \times n}(R_r) \).

The time complexity of the key exchange protocol based on DLPFP and FDLP [30] is compared in Figure 3. The \( x \)-axis represents the order of the matrix (\( n \)). The \( y \)-axis represents the order of time complexity of the key exchange protocol based on DLPFP \( (O(6n^3 \log r + 3n^3)) \) and FDLP \( (O(2n^3 \log r + 3n^3)) \). Here, \( r = 13 \). The red and blue curves represent the time complexity of the key exchange protocol based on DLPFP and FDLP, respectively. As the order of the matrix increases, the time complexity also increases. There is a minuscule difference in the slope of the curves, and it is because of the coefficient present in the time complexity of both protocols. But the time complexity of both protocols is proportional to \( O(n^3 \log r) \).

The advantage of the proposed protocol is that the total operations needed to run the proposed program is the same as the FDLP, and the security of the proposed key exchange protocol is more promising because of the inclusion of the secret parameters \( i, j, y, z \) and the hidden commutative subsemigroup. Additionally, our protocol overcomes some attacks, which makes it impossible to attack using the known methods.
5. ElGamal public key cryptosystem

This section introduces an ElGamal public key cryptosystem that is based on the suggested key exchange mechanism. Also, a theorem is given stating that solving the DLP and this ElGamal cryptosystem are the same.

Let \( A = M_n \times n(R_s) \) be a finite non-commutative semigroup under multiplication. Let \( B \) be a commutative subsemigroup of \( A \) and \( F^*_r = \{1, 2, 3, \ldots, r - 1\} \), where \( r \) is a big positive integer. To encrypt and decrypt, the following ElGamal cryptosystem is used. A flowchart for the proposed cryptosystem is given in Figure 4.

![Flowchart for the proposed ElGamal cryptosystem](image)

**Setup** Alice and Bob are the two classical entities taking part in the communication, and suppose a message is being sent by Bob to Alice. First, Alice does the following to compute her public key: Alice chooses \( a \in A \), such that \( a^q = I \), where \( q \) is the order of \( a \), and selects \( y \in B \setminus C_A(a) \) and \( i, j \in F^*_r \) and calculates \( p = a^iy^j \). Alice’s secret key is \( S_k = (i, j, y) \), and her public key is \( R_k = (p, a) \).

**Encryption** Bob performs the following computations to encrypt the plaintext:
1. Bob chooses \( z \in B \setminus C_A(a) \) and \( k, l \in F^*_r \) and keeps it secret. He computes \( C_1 = a^kz^l \).
2. The original plaintext is given in the form of a matrix \( o \in M_n \times n(R_s) \).
3. \( C_2 = a^k p z^l \) is calculated by Bob, and the ciphertexts \( (C_1, C_2) \) are sent to Alice.

**Decryption** Alice decrypts the original plaintext message \( o \) by calculating

\[
o = (a^iC_1y^j)^{-1}C_2
\]

since
\[(a'C_1y^j)^{-1}C_2 = (a'd^{k}z^j)^{-1}a'p'z'o
\]
\[= (a'^{d'k}z^j)^{-1}a'p'z'o
\]
\[= (a'p'z')^{-1}a'p'z'o
\]
\[= o.
\]

The matrices \(a\), \(y\), and \(z\) are chosen in such a way that they are invertible. Therefore, \(C_1\) and \(a'C_1y^j\) are also invertible. Hence, the original plaintext can be decrypted by using these invertible matrices.

**Remark** Clearly, to encrypt and decrypt, the matrix \(o\) need not be invertible.

In the case of healthcare systems, the encryption and decryption of SHI are as follows: The \(P_a\) gives the details, and his or her profile is encrypted and uploaded in \(S\) by \(A_d\). When the \(P_a\) gives the details, he/she also does the part of Alice as given in Section 5. If the authorized users of \(P_aD\) or the users of \(P_a\)'s SHI data, then they will do the part of Alice. That is, the authorized users who want to decrypt the data do the work of Alice in Section 5, and \(A_d\) does the work of Bob. Then, the \(P_a\), or the authorized users, get to decrypt the SHI file.

**Theorem 2** The Diffie-Hellman Problem (DHP) can be broken if the above mentioned ElGamal public key cryptosystem can be cracked.

**proof.** We have to prove

(i) Solving the DHP is all that is required to crack the ElGamal public key cryptosystem.

(ii) To crack the DHP, it is enough to solve the ElGamal public key cryptosystem.

(i) Suppose, Eve is the attacker who can approach the DHP oracle, represented as \(DHP_o\).

Here, the values of \(p = a'y^j\) and \(C_1 = a'^{d'}z^j\) are appealed to the \(DHP_o\) by Eve. The oracle replies back with the value \(K = a'^{d'k}z^jy^j\), that is,

\[DHP_o(a'y^j, a'^{d'}z^j) = a'^{d'k}z^jy^j.
\]

Eve can decrypt and find the original plaintext message \((o)\). The ElGamal cryptosystem can be broken, as the attacker can calculate \(o = K^{-1}C_2\) by computing \(K^{-1}\).

Therefore, it is sufficient to crack the DHP in order to crack the ElGamal cryptosystem.

(ii) In order to prove (ii), we assume that an attacker, Eve, who can access an ElGamal oracle exists. The oracle reveals the value of \(o\) by taking the values of \(p, C_1,\) and \(C_2\), where \(p = a'y^j, C_1 = a'^{d'}z^j,\) and \(C_2 = a'^{d'}p'z'o\). Since

\[C_2 = pC_1o,
\]
\[o = C_2(pM_1)^{-1}.
\]

As, \(K = a'^{d'k}z^jy^j = M_2o^{-1}\).

The attacker computes \(K\), as \(o\) is known [37].

Therefore, solving the DHP just requires breaking the ElGamal cryptosystem. 

\(\square\)
5.1 A toy example

A toy example to illustrate our proposed public key cryptosystem is given in this subsection. Let the semiring \( R_s = \mathbb{Z} \),

\[
A = M_4 \times a(\mathbb{Z}) = \left\{ \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} : a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p \in \mathbb{Z} \right\},
\]

\[
B = \left\{ \begin{bmatrix} q & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & t \end{bmatrix} : q, r, s, t \in \mathbb{Z} \right\} \quad \text{and} \quad F_{59}^* = \{1, 2, \ldots, 58\}.
\]

**Setup**

1. Alice chooses

\[
a = \begin{bmatrix} 5 & 7 & 2 & 4 \\ 3 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 4 \end{bmatrix} \in A, \quad y = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in B \setminus C_A(a),
\]

and \( i = 5, j = 4 \in F_{59}^* \).

2. Alice calculates

\[
p = a^i y^j = \begin{bmatrix} 701,360 & 45,643 & 1,665,279 & 43,184 \\ 290,208 & 18,833 & 688,095 & 17,812 \\ 173,648 & 11,306 & 412,290 & 10,691 \\ 208,000 & 13,533 & 498,474 & 13,073 \end{bmatrix}.
\]

**Encryption**

1. Bob chooses

\[
z = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in B \setminus C_A(a), \quad k = 2,
\]

and \( l = 3 \in F_{59}^* \).

2. Bob calculates
\[ C_1 = a^k p^l = \begin{bmatrix} 6,500 & 50 & 621 & 45 \\ 2,500 & 25 & 243 & 18 \\ 1,625 & 12 & 162 & 11 \\ 1,625 & 14 & 216 & 22 \end{bmatrix}. \]

3. Suppose, Bob wants to send the message “BANK DETAILS OF A”, he converts it into the matrix

\[ o = \begin{bmatrix} 1 & 0 & 13 & 10 \\ 3 & 4 & 19 & 0 \\ 8 & 11 & 18 & 14 \\ 5 & 0 & 0 & 0 \end{bmatrix} \in A \]

by assigning the values 0 to 25 to the alphabets A to Z.

4. Bob calculates \[ C_2 = a^k p^l o. \]

\[ C_2 = \begin{bmatrix} 41,103,545,785 & 45,431,974,062 & 178,939,755,459 & 138,219,991,524 \\ 16,986,320,616 & 18,774,671,941 & 73,951,226,769 & 57,122,774,666 \\ 10,177,629,323 & 11,249,372,353 & 44,307,321,432 & 34,224,688,538 \\ 12,240,965,335 & 13,532,352,165 & 53,271,243,335 & 4,114,887,449 \end{bmatrix}. \]

5. Bob sends the ciphertext \((C_1, C_2)\) to Alice.

**Decryption**

1. Alice retains the original message matrix \(o\) by calculating \[ o = (a^k C_1 y^l)^{-1} C_2. \]

\[ o = \begin{bmatrix} 8,041,878,000 & 4,184,109 & 4,128,657,966 & 3,970,346 \\ 3,323,654,000 & 1,729,033 & 1,706,159,619 & 1,640,513 \\ 1,991,258,000 & 1,036,054 & 1,022,293,467 & 983,085 \\ 2,393,222,000 & 1,245,195 & 1,229,761,035 & 1,183,894 \end{bmatrix} \times C_2 \]

\[ o = \begin{bmatrix} 1 & 0 & 13 & 10 \\ 3 & 4 & 19 & 0 \\ 8 & 11 & 18 & 14 \\ 5 & 0 & 0 & 0 \end{bmatrix}. \]

2. Alice converts the matrix \(o\) into the message.

If \(B \setminus C_A(a)\) were empty, it would mean every element in \(B\) commutes with \(a\) (i.e., \(B \subseteq C_A(a)\)). If all elements in \(B\) commute with \(a\), then the chosen secret elements \((y\) and \(z)\) also commute with \(a\). This compromises the security of the protocol. The adversary could potentially calculate the secret key \(K\) from the intercepted messages \((C_1, C_2)\) by exploiting commutativity.

Consider \(B\) as the set of all diagonal matrices. Due to matrix multiplication principles, the elements of \(B\) do not commute with matrices in \(A\) that contain non-zero off-diagonal elements. Therefore, \(B \setminus C_A(a)\) is nonempty.

Let
\[ A = M_4 \times 4(Z) = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} : a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p \in Z, \]

and

\[ B = \begin{bmatrix} q & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & t \end{bmatrix} : q, r, s, t \in Z. \]

The centralizer of \( a (C_A(a)) \) consists of all elements in \( A \) that commute with \( a \) (i.e., \( a' a = aa' \) for all \( a' \) in \( C_A(a) \)). Since elements in \( B \) have a zero in the bottom left corner, they cannot commute with \( a \). Let

\[ a = \begin{bmatrix} 5 & 7 & 2 & 4 \\ 3 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 4 \end{bmatrix} \in A, \]

\[ y = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in B \setminus C_A(a) \]

where \( ay \neq ya \). We can find other such matrices like

\[ \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \]

\[ \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \]

and so on from \( B \setminus C_A(a) \). This ensures that \( B \setminus C_A(a) \) is non-empty.

6. Conclusion

In this paper, a key exchange protocol based on DLPFP under matrix multiplication is proposed. Additionally, the protocol’s complexity and security are examined. Then, based on DLPFP, we have also suggested an ElGamal cryptosystem for SHI in healthcare systems with improved security.

Acknowledgment

The authors are thankful to the reviewers and editorial members for their valuable suggestions and constructive comments.
Conflict of interest

The authors declare that there is no conflict of interest.

References


