Design Of Public-Key Cryptosystems Based On Matrices Discrete Logarithm Problem

Ali M. Sagheer
Information System Department, College of Computer, University of Al-Anbar,
Abdul Monem S. Rahama **Ahmad T. Sadiq**
Department of Computer Science, University of Technology, Iraq

Abstract

The Matrices group is a group defined over finite field that forms an Abelian group, which is a suitable choice for constructing a good problem similar to Discrete Logarithm Problem (DLP) and Elliptic Curves Discrete Logarithm Problem (ECDLP). This idea is encouraged to define a new one-way trap-door function over finite matrices group. This leads to create cipher systems based on the difficulty of solution of the presented one-way trap-door function. That is appearing a clear change in the cryptography, and opens new windows for treatment with special groups and new operations.

This paper proposes one-way trap-door function defined over Matrices group, We call it Matrices Discrete Logarithm Problem (MDLP) and introduces the first proposed cryptosystems that employ the finite matrices group in the public key cryptosystems.

The complication associated with the desined cipher system comes from the wide variety of possible group structures of the matrix element in the Matrices group, and from the fact that matrices multiplication is complicated. The security of the system depends on how difficult it is to determine the integer d, given the square matrix B and the square matrix A where $B= A^d \bmod q$, A and B are square matrices defined over finite field F_q, this is referred to as the MDLP. In addition, that appears to offer equal security for a far smallest bit size, that for two reasons. The first reason is that the operations are applied -instead of multiplication of two integer numbers- as a matrix-by-matrix multiplication, in the other hand, the complexity and intractability are increase as much as the size of base matrix is increased. The second reason is that the order of the Matrices group $|M(F_q)|$ with $n \times n$ base matrix appears at most q^n-1 or its factors, that mean the calculation is applied with q-bit size, needs q^n-1 matrix-by-matrix multiplications to solve the MDLP.

Keywords: Public-key Cryptosystems, DLP, ECDLP, DSA, Matrices

الخلاصة

ان زمرة المصفوفات هي زمرة معرفة على الحقل المنتهي والتي تشكل زمرة ابيلية، وهذا خيار مناسب لبناء مسألة جيدة مشابهة لمسألة اللوغاريتم المنفصل (DLP) ومسألة اللوغاريتم المنفصل في المنحني الإهليلجي (ECDLP). وهذه الفكرة شجعت الى تعريف دالة جديدةوهي دالة باب المصيدة احادية الاتجاه المعرفة على زمرة المصفوفات المنتهية. وهذا ادى الى امكانية انشاء انظمة تشفيرتستند على صعوبة حل دالة باب المصيدة احادية الاتجاه المقدمة. وذلك يظهر تغير واضح في الكتابة المشفرة، ويفتح لنا نوفذ جديدة للتعامل بالزمر الخاصة والعمليات الجديدة.

يقدّم هذا البحث اقتراح دالة باب المصيدة احادية الاتجاه المعرفة على زمرة المصفوفات المنتهية وهي مسألة اللوغارتم المنفصل في المصفوفات وكذلك يقدم هذه البحث الاقتراح الاول لانظمة التشفير ذو المفتاح المعلن التي توضف زمرة المصفوفات المنتهية في هذه الانظمة. وان التعقيد المرتبط بالكتابة المشفّرة بالمصفوفات (Matrices Cryptography) مشتق من الانواع المختلفة من تراكيب الزمر الممكنة من عنصر المصفوفة في زمرة المصفوفات، وكذلك من حقيقة أنّ عملية ضرب المصفوفات معقدة. ويعتمد أمن الكتابة المشفّرة بالمصفوفات على مدى صعوبة ايجاد العدد الصحيح d، مع اعطاء المصفوفة المربعة B والمصفوفة المربعة A والمعرفة على الحقل المنتهي (Finite Field F_q) حيث $B= A^d \bmod q$. وهذه تدعى باسم مسالة اللوغاريتم المنفصل في المصفوفات.

بالاضافة الى ذلك هو يظهر إعطاء أمن مساوي للانظمة الاخرى مع حجم قطعة صغير جدا، لسببين. السبب الأول هو أن العمليات تطبق –بدلا من ضرب عددين صحيحين– تطبق كضرب مصفوفة بمصفوفة، ومن ناحية اخرى فان التعقيد وصعوبة الحل تزداد كلما زاد حجم المصفوفة الأساسية. والسبب الثاني هو أن حجم زمرة المصفوفات (عدد عناصر زمرة المصفوفات) والذي يرمز له

$|M(F_q)|$ مع مصفوفة الاساس $n \times n$ يظهر على الأغلب q^n-1 أو احد عوامله وهذا يعني ان الحساب المطبق على حجم q-bit، يحتاج الى q^n-1 عملية ضرب مصفوفات لحل مسالة اللوغاريتم المنفصل في المصفوفات.

1. Introduction

This paper introduces the first proposal cryptosystems that is employed Matrices group in cryptography. Unlike previous cryptosystems, matrices work with a finite group formed by the matrices elements on an Matrices group defined over a finite field. The cryptosystems includes key distribution, encryption/decryption schemes, and Digital Signature Algorithm (DSA). The key distribution algorithm is used to share a secret key, the encryption/decryption algorithm enables confidential communication, and the DSA is used to authenticate the signer and validate the integrity of the message. They do not invent new cryptographic algorithm, but they are the first to implement existing public-key cryptosystem using Matrices group. The proposal is an analogues to the Diffie-Hellman key exchange protocol, analogues to ElGamal, Massey-Omura schemes, and DSA.

2. Matrices Group Laws

The paper proposes a definition of a new group. The new group is the Matrices group defined over finite field $M(F_q)$. In this subsection, we shall study some of theoretic properties of the Matrices group $M(F_q)$.

Theorem 1 (Matrices Group Laws): The Matrices group $(M,*)$ has the following group theoretic properties:

(1). Elements: Let M be defined over a field F_q $(M(F_q))$, then

$$M^*(F_q) = \{\forall \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, a_{i,j} \in F_q\}$$

is a group of $M(F_q)$, that is generated from square matrix.

(2). Existing of identity: $A * I = I * A = A$ for all $A \in M(F_q)$.

(3). Existing of inverse: *Let* $A \in M(F_q)$, there is a Matrix of $M(F_q)$ denoted $A^{-1} \in M(F_q)$, so that:

$$A^* A^{-1} = I.$$

(4). Associativity: Let $A, B, C \in M(F_q)$, then:

$$(A^* B) * C = A * (B^*C).$$

Definition 1: Let A be an element square matrix of the group $M(F_q)$, Then A is said to have order k if

$$A^k = \underbrace{A * A \ldots\ldots * A}_{k \text{ product}} = I$$

with $A^{k'} \neq I$ for all $1 \leq k' \leq k$ (that is, k *is* the smallest integer such that $A^k=I$). If such a k exists, then, the subgroup of $M(F_q)$ is said to have finite order k, otherwise, it has infinite order.

Definition 2: From here on, for the group operations on a Matrices group M, for $k \in Z$ and $A \in M(F_q)$, then:

$A^k = A * A * \ldots\ldots * A$, *(k times),* for $k > 0$,

$A^0 = I$, and

$A^k = (A^{-1})^{-k}$, for $k < 0$.

Definition 3: The order of a Matrices group is defined as the number of matrix element of the Matrices group and denoted by $\#M$.

If $A \in M(F_q)$ is of order k, then

$$H = \{ A^i \mid 0 \le i < k\text{-}1\},$$

is a *subgroup* of $M(F_q)$ of order k.

Definition 4: Let A be an element matrix of the group $M(F_q)$, Then A is said to generator matrix if

$$ord(A) = \#M$$

Then,

$$M(F_q) = \{ A^k \mid 0 \le k < \#M\text{-}1\},$$

Theorem 2: (Matrices SubGroup Laws): Each square matrix that is defined over finite field and have the invers (i.e. the determenant is not equal zero), it can be generate a subgroup of the Matrices group $M(F_q)$.
Such that:
The square matrix A that is defined over finite field F_q, and $|A| \neq 0$, it means the invers of A (A^{-1}) is found, then

- ***The Matrices group (M,*) is closure on the * operation such that:***

for all $A, B \in M(F_q)$, there exists $A * B \in M(F_q)$.

- ***Existing of identity***

for all $A \in M(F_q)$, there exists $A * I = I * A = A$.

- ***Existing of inverse***

Let $A \in M(F_q)$, there is a Matrix of M denoted $A^{-1} \in M(F_q)$, so that $A^* A^{-1} = I$.

- ***Associativity:***

Let $A, B, C \in M(F_q)$, then, $(A^* B) * C = A * (B^*C)$.

- ***Commutativity:***

$A * B = B * A$ for all $A, B \in M(F_q)$.
In other words, the Matrices group law makes M into an Abelian group with identity (neutral) element I.

Finally, Then when A has the inverse A^{-1}, it generates a subgroup of matrices in the Matrices group (M,*) by exponentiate the Matrix A from 0 to N, that satisfies the group conditions.

3. Matrices Discrete Logarithm Problem (Mdlp)

One of the most interesting open problem in cryptography is the realization of a trapdoor on the discrete logarithm, in which to solve the DLP is hard only if published parameters are used, while it is easy by using a secret key (trapdoor key) (Nathanson, 2000)).

The DLP can be defined on various finite groups as well as multiplicative group over a finite filed F_q (Mollin, 1989), this idea can be extended to arbitrary groups and, in particular, to Matrices groups. A typical example except the multiplicative group is the discrete logarithm problem on Matrices group over F_q , and many cryptographic schemes are constructed on the MDLP.

***Definition 5 (MDLP*)**: For a Matrices group M, let $A, B \in M(F_q)$, recall that in the MDLP, to find an integer $k \in Z$, is such that $A^k = B$.

Since a *Matrices group M* is made into Abelian group by a matrix-by-matrix multiplicative operation, "The exponential of a Matrix element on $M(F_q)$" actually refers to the repeated multplications. Therefore, $B=A^i$ is the i^{th} power of $A \in M(F_q)$ is the i^{th} multiple of A. The logarithm of B to the base A would be i (i.e. the inverse of exponentiation). The MDLP is of interest because its apparent intractability forms the basis for the security of *Matrices group* cryptographic schemes.

4. Matrices Cryptosystems (MCS)

Unlike previous cryptosystems, matrices work with a finite Abelian group formed by the square matrices elements on an Matrices group defined over a finite field $M(F_q)$. MCS include key distribution, encryption/decryption schemes, and Digital Signature Algorithm (DSA). The key distribution algorithm is used to share a secret key, the encryption/decryption algorithm enables confidential communication, and the DSA is used to authenticate the signer and validate the integrity of the message.

This section proposes cryptosystems that employs Matrices group. It does not invent new cryptographic algorithm, but it is the first to implement existing public-key cryptosystem using Matrices group. The proposal is an analogues to the Diffie-Hellman key exchange protocol, analogues to ElGamal, Massey-Omura schemes, and DSA. The matrix-by-matrix multiplication operation in MCS is the counterpart of modular multiplication in RSA and ElGamal, and exponentiation of matrix in MCS is the counterpart of the modular exponentiation. To form cryptographic system using Matrices group, we need to find a "hard problem" corresponding to the difficulty of factoring the product of two prime or taking the discrete logarithm or elliptic curve discrete logarithm.

Consider the equation $B=A^k$, where A and B are two matrices in the Matrices group and k is an integer. It is relatively easy to calculate B given A and k, but determining the integer k from a multiple of a matrix A^k, even with the knowledge of A, B and $M(F_q)$ is a very difficult problem, known as the Matrices Discrete Logarithm Problem (MDLP).

4.1 Exponentiation over Matrices

The fundamental operation in Matrices cryptographic schemes is that of matrix exponentiation of a square matrix element by an integer. If not the most confusing term, certainly the idea of multiplying matrix refers to computing $B=A^k$, where A and B are two square matrices in the Matrices group and k is an integer. This really means that we multiply A to itself k times.

Definition 6 (Exponentiation of a square matrix on an Matrices group by an integer): Given $k \in Z$, and A is a square matrix element on a Matrices group $M(F_q)$, *then*

$$A^k = A * A \ldots\ldots * A \qquad (k \text{ times}) \ldots\ldots\ldots\ldots (1)$$

And it is so called matrix exponentiation [16], and it is the dominant cost operation in matrices cryptographic scheme, and it dominates the execution time of matrices cryptographic schemes, especially the representation of MDLP.

The algorithm that can be used to compute the matrix exponentiation in the Matrices group is Repeated-Squaring and Multiplication or fast group operation Method.

4.2 Repeated-Squaring and Multiplication Method

The most fundamental computation on Matrices group is the group operation $A^k = A * A \ldots\ldots * A$ *(k times,)* where $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$ is a matrix element on a Matrices group over F_q $M(F_q)$ and k are very large positive integer, since the computation of A^k is so fundamental in all matrices related computations and applications, it is desirable that such computations are carried out as fast as possible.

Remarkably enough, the idea of repeated squaring for fast exponentiation can be used almost directly for fast group operation on matrices.
Let $e_{n-1}\ e_{n-2}\ \ldots\ e_1\ e_0$ be the binary representation of k. then for i starting from n-1 down to 0 (e_{n-1} almost 1 and used for initialization), check whether or not e_i =1. If e_i =1, then perform a squaring and a multiplication group operation; otherwise, just perform a squaring operation. For example: compute A^{43}, since 43=101011, we get the following table:

Table 1: Compute A^{43} using repeated doubling and addition

i	e_I	Value	*Operations*	Status
5	e_5	1	A	Initialization
4	e_4	0	$A^2 = A^2$	squaring
3	e_3	1	$(A^2)^2*A = A^5$	Squaring and Multiplication
2	e_2	0	$((A^2)^2*A)^2 = A^{10}$	squaring
1	e_1	1	$(((A^2)^2*A)^2)^2*A = A^{21}$	Squaring and Multiplication
0	e_0	1	$((((A^2)^2*A)^2)^2*A)^2*A = A^{21}$	Squaring and Multiplication

We have the following algorithm which implements this idea of repeated squaring and multiplication (fast group operation) for computing A^k, that is, it reduces the complexity of the computation of A^k from k to $\log k$.

Algorithm (Repeated-Squaring and Multiplication)

Input: *a matrix $A \in M(F_q)$ and positive integer k.*

Output: *$B = A^k$.*

1. *Write k in the binary expansion form $k = e_{n-1}\ e_{n-2}\ \ldots\ e_1\ e_0$ where each e_i either* 0 *or* 1*(Assume k has n bits)*
2. *Set B = I.*
3. *Compute A^k:*
   - *3.1 For i from n-*1 *down to* 0 *do*
   - *3.2 $B = B^2$.*
   - *3.3 if e_i =1, then $B = B*A$.*
4. *Output B: (now $B = A^k$).*

5 Design Of Matrices Public-Key Cryptosystems

The section introduces design of public-key cryptography that employs the Matrices group. More specifically, it'll introduce matrices cryptosystems analogues to several well known public-key cryptosystems including key exchange, encryption/decryption, and DSA schemes.

5.1 Matrices Public-key Cryptosystems

For any cryptographic system based on the DLP, there is an analogy to Matrices group. In what follows, it'll introduce matrices cryptosystems analogues to four widely used public-key cryptosystems, namely Diffie-Hellman key exchange system, the Massey-Omura, the El-Gamal public-key cryptosystems and DSA.

1. Analogy of the Diffie-Hellman Key Exchange System

This system is merely a method for exchanging keys; no messages are involved. Alice and Bob first publicly choose a finite field F_q and a Matrices group $M(F_q)$ defined over it. Then they publicly choose a matrix $B \in M(F_q)$ to serve as their "Base matrix". It is a generator of the key. To generate a key, Alice chooses random integer e between 1 and N, where N is the order number of the of Matrices group $M(F_q)$, and keeps it secret. She then computes $B^e \in M(F_q)$ and makes that public. Bob

chooses his own secret random integer d between 1 and N, and makes public $B^d \in M(F_q)$. The secret key is then $B^{ed} \in M(F_q)$. Both Alice and Bob can compute this key. For example,
Alice knows B^d (public knowledge) and her own secret e. Charlie, on the other hand, only knows B, B^e and B^d. Without solving the MDLP, (finding d knowing B and B^d), there is no way for him to compute B^{ed} only knowing B^e and B^d. The following algorithm illustrates this manner.

Algorithm (Diffie-Hellman key exchange system with MDLP)

1. ***Initialization***
   - *Alice and Bob publicly choose a finite field F_q and the Matrices group M over F_q ($M(F_q)$).*
   - *They publicly choose a random "Base matrix" $B \in M(F_q)$ such that B generates a large subgroup of $M(F_q)$.*
2. ***Key generation***
   - *Alice chooses a secret random integer e. She then computes $B^e \in M(F_q)$.*
   - *Bob chooses a secret random integer d. He then computes $B^d \in M(F_q)$.*
   - *Make B^e and B^d public and keep e and d secret.*
3. ***Calculation of the secret key matrix B^{ed}***
   - *Alice computes the secret key $B^{ed} = (B^d)^e$.*
   - *Bob computes the secret key $B^{ed} = (B^e)^d$.*

There is no known fast way to compute B^{ed} if only knows B, B^e and B^d, this is MDLP.

Example:- *Let the square matrix element generator that is defined over F_{29} is $G=\begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}$ the group of the generator G is cyclic because $11*9-12*7=15\neq 0 \bmod 29$, and give 840 element group size where $29*29-1=840$.*

Therefore the cyclic group $M(F_{29})$ on $G=\begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}$ of order 840 is

Apply the Diffie-Hellman key exchange system with MDLP

1. ***Initialization***
   - *Alice and Bob publicly choose a finite field F_{29} and the Matrices group M over F_{29} ($M(F_{29})$).*
   - *They publicly choose a random "Base matrix" $B \in M(F_{29})$ such that $B=\begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}$ generates a large subgroup of $M(F_{29})$.*
2. ***Key generation***
   - *Alice chooses a secret random integer e=367. She then computes $B^e=B^{367}=\begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}^{367}=\begin{pmatrix} 25 & 22 \\ 17 & 27 \end{pmatrix}$.*
   - *Bob chooses a secret random integer d=692. He then computes $B^d=B^{692}=\begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}^{692}=\begin{pmatrix} 6 & 14 \\ 24 & 2 \end{pmatrix}$.*
   - *Make B^e and B^d public and keep e and d secret.*

3. Calculation of the secret key matrix B^{ed}

- *Alice computes the secret key* $B^{ed} = (B^d)^e = \begin{pmatrix} 6 & 14 \\ 24 & 2 \end{pmatrix}^{367} = \begin{pmatrix} 3 & 3 \\ 1 & 27 \end{pmatrix}$.
- *Bob computes the secret key* $B^{ed} = (B^e)^d = \begin{pmatrix} 25 & 22 \\ 17 & 27 \end{pmatrix}^{692} = \begin{pmatrix} 3 & 3 \\ 1 & 27 \end{pmatrix}$.

2. Analogy of the Massey-Omura Cryptosystem

In this system the finite field F_q and the Matrices group $M(F_q)$ have been made publicly known. Alice and Bob both select a random integer e_1 and e_2 between 1 and N, where N is the order number of the of Matrices group $M(F_q)$, respectively with gcd(e_1, N)=1 and gcd(e_2, N)=1. They also compute their inverses $d_1 = e_1^{-1} \bmod N$ (ie. $d_1e_1 = 1 \bmod N$) and $d_2 = e_2^{-1} \bmod N$ (ie. $d_2e_2 = 1 \bmod N$), then, keep everything secret. If Alice wants to send the message P_m (PlainText matrix) to Bob, she first sends him the message $P_m{}^{e1}$. This means nothing to Bob, since he does not know d_1. He ever, he can exponentiate it by his e_2 and send the message $P_m{}^{e1e2}$ back to Alice. Then Alice can help unravel the message by exponentiating this new message by d_1 which sends $P_m{}^{e1e2d1} = P_m{}^{e2}$ back to Bob. Then Bob can exponentiate this message by d_2 to get the original message ($P_m{}^{e2d2} = P_m$). During this process Charlie sees $P_m{}^{e1}$, $P_m{}^{e2}$, and $P_m{}^{e1e2}$.

Without solving the MDLP –finding e_2 and then its inverse knowing $P_m{}^{e1}$ and $P_m{}^{e1e2}$ - there is no way for him to find P_m. The following algorithm illustrates this manner.

Algorithm (Massey-Omura Cryptosystem with MDLP)

1. Initialization

- *Alice and Bob publicly choose a finite field F_q and the Matrices group M over F_q ($M(F_q)$).*
- *They publicly known the order number of the of Matrices group $M(F_q)$ denoted by N.*

2. Key generation

- *Alice chooses a secret random integer e_1 between 1 and N, such that gcd(e_1 , N) =1. She then computes its inverse $d_1 = e_1^{-1} \bmod N$.*
- *Bob chooses a secret random integer e_2 between 1 and N, such that gcd(e_2 , N) =1. He then computes its inverse $d_2 = e_2^{-1} \bmod N$.*
- *Keep e_1, d_1, e_2, and d_2 secret.*

3. Transmission procedure

Alice sends the message P_m to Bob as follows

- *Alice computes $P_m{}^{e1}$, and sends it to Bob.*
- *Bob computes $P_m{}^{e1e2}$, and sends it to Alice.*
- *Alice computes $P_m{}^{e1e2d1} = P_m{}^{e2}$, and sends it to Bob.*
- *Bob computes $P_m{}^{e2d2} = P_m$).*

Example:- *Let the square matrix element generator that is defined over F_{29} is* $G = \begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}$ *the group of the generator G is cyclic because 11*9-12*7=15≠0 mod 29, and give 840 element group size where 29*29-1=840.*

Therefore the cyclic group $M(F_{29})$ on $G = \begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}$ *of order 840 is*

Apply the Massey-Omura Cryptosystem with MDLP

1. Initialization

- *Alice and Bob publicly choose a finite field F_{29} and the Matrices group M over F_{29} ($M(F_{29})$).*
- *They publicly known the order number of the of Matrices group $|M(F_{29})|$ on* $B=\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}$, *denoted by N=840.*

2. Key generation

- *Alice chooses a secret random integer $e_1=547$. She then computes its inverse $d_1=547^{-1} \bmod 840=43$.*
- *Bob chooses a secret random integer $e_2=97$. He then computes its inverse $d_2=97^{-1} \bmod 840=433$.*
- *Keep e_1, d_1, e_2, and d_2 secret.*

3. Transmission procedure

Alice sends the message $P_m=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ *to Bob as follows*

- *Alice computes* $P_m{}^{e1}=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}^{547}=\begin{pmatrix}10 & 24\\ 7 & 17\end{pmatrix}$, *and sends it t o Bob.*
- *Bob computes* $P_m{}^{e1e2}=\begin{pmatrix}10 & 24\\ 7 & 17\end{pmatrix}^{97}=\begin{pmatrix}27 & 1\\ 16 & 14\end{pmatrix}$, *and sends it to Alice.*
- *Alice computes* $P_m{}^{e1e2d1}=\begin{pmatrix}27 & 1\\ 16 & 14\end{pmatrix}^{43}=\begin{pmatrix}23 & 17\\ 11 & 5\end{pmatrix}=P_m{}^{e2}$, *and sends it to Bob.*
- *Bob computes* $P_m{}^{e2d2}=\begin{pmatrix}23 & 17\\ 11 & 5\end{pmatrix}^{433}=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}=P_m$.

3. Analogy of the ElGamal Cryptosystem

In this system the finite field F_p, the Matrices group $M(F_q)$, and the "Base matrix" $B \in M(F_q)$ are public information. Bob randomly chooses an secret integer d ($1<d<N$, where N is the order number of the Matrices group $M(F_q)$) and publishes the matrix B^d. If Alice ants to send the message P_m (PlainText matrix) to Bob, she will choose a secret random integer e ($1<e<N$) and send ($P_m * B^{ed}$, B^e) to Bob. Bob will then exponentiate the second matrix in the pair by d to get B^{ed}, the compute the inverse of the key matrix B^{ed} to get $(B^{ed})^{-1}$ and multiply by the first matrix in the pair $P_m * B^{ed}$ to find P_m. In the meantime, Charlie has only seen B^e and B^d. Without solving the MDLP (eg. finding d knowing B and B^e), there is no way for him to find P_m. The following algorithm illustrates this manner.

Algorithm (ElGamal Cryptosystem with MDLP)

1. Initialization

- *Alice and Bob publicly choose a finite field F_q and the Matrices group M over F_q ($M(F_q)$).*
- *They publicly choose a random "Base matrix" $B \in M(F_q)$ such that B generates a large subgroup of $M(F_q)$.*

2. Key generation

- *Bob chooses a secret random integer d in interval [2, N].*
- *He then computes* $Q = B^d$.
- *Make Q public and keep d secret.*

3. Encryption

Alice sends the message P_m *to Bob as follows*

- *Select random integer e in interval [2, N].*
- *Compute* B^e.
- *Compute* $K= Q^e$, *(i.e.* $K=B^{de}$*).*
- *Compute ciphertext* $C= P_m * K$.
- *Transmit the pair matrices* (C, B^e).

4. Decryption

Bob retrieves the message as follows

- *Compute* $K= (B^e)^d$, *(i.e.* $K=B^{ed}$*).*
- *Compute* K^{-1}.
- *Multiply* K^{-1} *by the ciphertext matrix C:*

$$P_m = C * K^{-1}.$$

Example:- *Let the square matrix element generator that is defined over* F_{29} *is* $G=\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}$ *the group of the generator G is cyclic because 11*9-12*7=15≠0 mod 29, and give 840 element group size where 29*29-1=840.*

Therefore the cyclic group $M(F_{29})$ *on* $G=\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}$ *of order 840 is*

Apply the ElGamal Cryptosystem with MDLP

1. Initialization

- *Alice and Bob publicly choose a finite field* F_{29} *and the Matrices group M over* F_{29} $(M(F_{29}))$.
- *They publicly choose a random "Base matrix"* $B =\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}$.

2. Key generation

- *Bob chooses a secret random integer d =218.*
- *He then computes* $Q = B^d= B^{218}=\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}^{218} =\begin{pmatrix}22 & 17\\ 25 & 13\end{pmatrix}$.
- *Make Q public and keep d secret.*

3. Encryption

Alice sends the message $P_m =\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ *to Bob as follows*

- *Select random integer e=793.*
- *Compute* $B^e= B^{793} =\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}^{793} =\begin{pmatrix}24 & 26\\ 28 & 0\end{pmatrix}$.
- *Compute* $K= Q^e =\begin{pmatrix}22 & 17\\ 25 & 13\end{pmatrix}^{793} =\begin{pmatrix}23 & 23\\ 27 & 4\end{pmatrix}$.

- *Compute ciphertext* $C = P_m * K = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} * \begin{pmatrix} 23 & 23 \\ 27 & 4 \end{pmatrix} = \begin{pmatrix} 19 & 2 \\ 3 & 27 \end{pmatrix}$.
- *Transmit the pair matrices* $(C, B^e) = \left(\begin{pmatrix} 19 & 2 \\ 3 & 27 \end{pmatrix}, \begin{pmatrix} 24 & 26 \\ 28 & 0 \end{pmatrix} \right)$.

4. Decryption

Bob retrieves the message as follows

- *Compute* $K = (B^e)^d = \begin{pmatrix} 24 & 26 \\ 28 & 0 \end{pmatrix}^{218} = \begin{pmatrix} 23 & 23 \\ 27 & 4 \end{pmatrix}$.
- *Compute* $K^{-1} \bmod 29 = \begin{pmatrix} 23 & 23 \\ 27 & 4 \end{pmatrix}^{-1} \bmod 29 = \begin{pmatrix} 16 & 24 \\ 8 & 5 \end{pmatrix}$.
- *Multiply* K^{-1} *by the ciphertext matrix C:*

$$P_m = C * K^{-1} = \begin{pmatrix} 19 & 2 \\ 3 & 27 \end{pmatrix} * \begin{pmatrix} 16 & 24 \\ 8 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}.$$

5.2 Matrices Digital Signature Algorithm(MDSA)

The MDSA is analog to the DSA in using the Matrices group. DSS are the counterpart to handwritten signatures. A digital signature is the number that depends on the secret key and is only known by the signer and depends on the contents of the message being signed. Signatures must be verifiable without access to the signer's private key. Signatures should be existentially unforgeable under chosen-message attacks. This asserts that an adversary who is able to obtain Alice's signatures for any messages of his choice cannot forge Alice signature on a single other message.

Suppose Alice wants to send a digitally signed message to Bob. They first choose a finite field F_q, and the Matrices group $M(F_q)$, defined over that field and the "Base matrix" B with order n. Alice's key pair is (d, Q), where d is her private integer key and $Q = B^d$ is her public matrix key . To sign a message P_m (PlainText matrix) Alice does the following:

1. Choose a random integer number k with k: $1 \le k \le n$ -1.
2. Compute $B^k = \begin{pmatrix} a_{11} & \vdots \\ \cdots & \ddots \end{pmatrix}$, and $r = a_{1,1} \bmod n$. If r =0 then go to 1.
3. Compute $k^{-1} \bmod n$.
4. Compute $e = \mathrm{H}(P_m)$.
5. Compute $s = k^{-1}(e + d\,r) \bmod n$. If s =0 then go to 1.
6. Alice signature for the message P_m is (r, s).

To verify Alice's signature (r, s) on the message P_m, Bob obtains an authentic copy of Alice's parameters and public key. Bob should validate the obtained parameters! Bob then does the following:

Verify that r, s are integers in the interval $[1, n-1]$.

1. Compute $e = \mathrm{H}(P_m)$.
2. Compute $w = s^{-1} \bmod n$.
3. Compute $u_1 = e\,w \bmod n$, and $u_2 = r\,w \bmod n$.
4. Compute $\begin{pmatrix} a_{11} & \vdots \\ \cdots & \ddots \end{pmatrix} = B^{u_1} * Q^{u_2}$. If $\begin{pmatrix} a_{11} & \vdots \\ \cdots & \ddots \end{pmatrix} = I$, then reject the signature.
5. Otherwise, Compute $v = a_{1,1} \bmod n$.
6. Accept the signature if and only if $v = r$.

If the signature (r, s) on the message P_m was indeed generated by Alice, the $s = k^{-1}(e + d\,r) \bmod n$. With this information we have

$$k = s^{-1}(e + d\,r) \bmod n = s^{-1} e + s^{-1} r\, d = w\, e + w\, d\, r$$
$$= u_1 + u_2\, d \quad \bmod n.$$

Thus

$$B^{u_1} * Q^{u_2} = B^{u_1 + u_2 d} = B^k.$$

and so

$$v = r \quad \text{as required.}$$

The following algorithm describes the above mentioned steps.

Algorithm (MDSA)

1. ***Initialization***
   - *Alice and Bob publicly choose a finite field F_q and the Matrices group M over F_q ($M(F_q)$).*
   - *They publicly choose a random base point $B \in M(F_q)$ with order n, such that B generates a large subgroup of $M(F_q)$.*
2. ***Key generation***
   - *Choose a secret random integer d in interval [2, N].*
   - *He then computes $Q = B^d$.*
   - *Make Q public and keep d secret.*
3. ***Signature generation***

Alice sends the digitally signed message P_m to Bob as follows

   - *Select random integer k in interval [2, N].*
   - *Compute* $\begin{pmatrix} a_{11} & \vdots \\ \cdots & \ddots \end{pmatrix} = B^k$.
   - *Compute* $r = a_{1,1} \bmod n$.
   - *Compute* $e = H(P_m)$.
   - *Compute* $s = k^{-1}(e + d\,r) \bmod n$.
   - *The signature for P_m is (r, s).*
4. ***Signature verification***

Bob verifies Alice's signature (r, s) on message P_m as follows

   - *Compute* $e = H(P_m)$.
   - *Compute* $w = s^{-1} \bmod n$.
   - *Compute* $u_1 = e\, w \bmod n$.
   - *Compute* $u_2 = r\, w \bmod n$.
   - *Compute* $\begin{pmatrix} a_{11} & \vdots \\ \cdots & \ddots \end{pmatrix} = B^{u_1} * Q^{u_2}$. If $\begin{pmatrix} a_{11} & \vdots \\ \cdots & \ddots \end{pmatrix} = I$, then reject the signature.
   - *Otherwise, Compute* $v = a_{1,1} \bmod n$
   - *Accept the signature if and only if v =r.*

Example:- *Let the square matrix element generator that is defined over F_{29} is* $G = \begin{pmatrix} 11 & 7 \\ 12 & 9 \end{pmatrix}$ *the group of the generator G is cyclic because 11*9-12*7=15≠0 mod 29, and give 840 element group size where 29*29-1=840.*

Therefore the cyclic group $M(F_{29})$ on $G=\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}$ of order 840 is

Apply the MDSA

1. ***Initialization***
   - *Alice and Bob publicly choose a finite field F_{29} and the Matrices group M over F_{29} ($M(F_{29})$).*
   - *They publicly choose a random base point $B=\begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}$ with order N=840.*
2. ***Key generation***
   - *Choose a secret random integer d =719.*
   - *He then computes $Q = B^d = \begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}^{719} = \begin{pmatrix}27 & 8\\ 22 & 4\end{pmatrix}$.*
   - *Make Q public and keep d secret.*
3. ***Signature generation***

Alice sends the digitally signed message P_m to Bob as follows

   - *Select random integer k = 242.*
   - *Compute* $B^k = \begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}^{242} = \begin{pmatrix}21 & 16\\ 15 & 4\end{pmatrix}$.
   - *Compute* $r= a_{1,1} \bmod n=21 \bmod 840=21$.
   - *Compute* $e= H(P_m)$: *Let* $e=20$.
   - *Compute* $s= k^{-1}(e + d\,r) \bmod n=481(20+719*21) \bmod 840=359$.
   - *The signature for P_m is (r, s)= (21, 359)*
4. ***Signature verification***

Bob verifies Alice's signature (r, s) on message P_m as follows

   - *Compute* $e= H(P_m)$: *Let* $e=20$.
   - *Compute* $w = s^{-1} \bmod n=599$.
   - *Compute* $u_1 = e\,w \bmod n=20*599 \bmod 29=220$.
   - *Compute* $u_2 = r\,w \bmod n=21*599 \bmod 29=819$.
   - *Compute* $\begin{pmatrix}a_{11} & \vdots\\ \cdots & \ddots\end{pmatrix} = B^{u_1} * Q^{u_2} = \begin{pmatrix}11 & 7\\ 12 & 9\end{pmatrix}^{220} * \begin{pmatrix}27 & 8\\ 22 & 4\end{pmatrix}^{819}$
     $= \begin{pmatrix}6 & 27\\ 9 & 19\end{pmatrix} * \begin{pmatrix}25 & 5\\ 21 & 7\end{pmatrix} = \begin{pmatrix}21 & 16\\ 15 & 4\end{pmatrix}$,
   - *Compute* $v = a_{1,1} \bmod n=21 \bmod 840=21$
   - *Accept the signature where, v =r=21.*

6. The Computational Complexity

The Computational Complexity of the DLP computing is compared to proposed problem MDLP of encryption and decryption function as follows:

1. **DLP:**
   - Let the size of the input message unit be *n.*
   - The complexity of the computing $b = a^x \bmod p$ is:
     $T(b) = T(a^x) = O(\log n)$ arithmetic (multiplication) operation, using Fast Exponential Algorithm (Yan, 2000.)
     Then,

$T(a^x) = O(log^3\ n)$ bit operation.

2. MDLP:

- Let the size of the input message unit be *n*.
- Let the size of the *Base Square Matrix* be *m*.
- The complexity of the computing $B = A^x \bmod p$ is:
  $T(B) = T(A^x) = O(log\ n)$ group (matrix-by-matrix multiplication) operation, using Repeated-Squaring and Multiplication Algorithm.
  Then,
  $T(A^x) = O(m^2\ log\ n)$ arithmetic (multiplication) operation.
  $= O(m^2\ log^3\ n)$ bit operation.

7. The Running Time Comparison

The proposed system is programmed by Delphi7 programming language on P4 PC computer with CPU of 3.2 G.B and RAM of 512 M.B. Then the methods is applied on different size messages, which takes plaintext of K Bytes then encrypts it and computes the running time of its operation, then decrypts its and computes the running time of the decryption. Next, 10 K Bytes, 20 K Bytes, 30 K Bytes, 40 K Bytes, 50 K Bytes and M Byte and computes the running time of the encryption and decryption of each messages.

Table 2 shows the running time of the El-Gamal method with DLP over F_{97} and the base number is 23, the public key is 58 and the secret key is 43. The order of the multiplicative group over F_q is q-1, then, the order of the multiplicative group over F_{97} is 96. The DLP over F_{97} is solved by 0 msec.

The running time of the El-Gamal method with MDLP over F_{97} and the base 2×2 matrix is $\begin{pmatrix} 17 & 93 \\ 43 & 39 \end{pmatrix}$, the public key 2×2 matrix is $\begin{pmatrix} 67 & 9 \\ 73 & 66 \end{pmatrix}$ and the secret key is 8394. The order of the Matrices group of base matrix 2×2 over F_{97} is 9408, 9408 equal 97*97-1, then, The order of the Matrices group of base matrix 2×2 over F_q *is* q^2-1. The MDLP over F_{97} with base matrix 2×2 is solved by 16 msec.

The running time of the El-Gamal method with MDLP over F_{97} and the base 3×3 matrix is $\begin{pmatrix} 11 & 29 & 39 \\ 43 & 53 & 61 \\ 79 & 83 & 91 \end{pmatrix}$, the public key 3×3 matrix is $\begin{pmatrix} 93 & 26 & 52 \\ 56 & 26 & 71 \\ 65 & 89 & 1 \end{pmatrix}$ and the secret key is 100000. The order of the Matrices group of base matrix 3×3 over F_{97} is 912672, 912672 equal 97*97*97-1, then, The order of the Matrices group of base matrix 3×3 over F_q *is* q^3-1.The MDLP over F_{97} with base matrix 3×3 is solved by 188 msec.

The running time of the El-Gamal method with MDLP over F_{97} and the base 4×4 matrix is $\begin{pmatrix} 11 & 29 & 39 & 43 \\ 53 & 61 & 79 & 83 \\ 91 & 31 & 7 & 23 \\ 17 & 57 & 73 & 57 \end{pmatrix}$, the public key 4×4 matrix is $\begin{pmatrix} 72 & 88 & 38 & 9 \\ 22 & 70 & 4 & 54 \\ 23 & 85 & 47 & 73 \\ 62 & 5 & 40 & 0 \end{pmatrix}$ and the secret key is 1000000. The order of the Matrices group of base matrix ٤×٤ over F_{97} is ٨٨٥٢٩٢٧٩, ٨٨٥٢٩٢٧٩ equal 97*97*97*97-1, then, The order of the Matrices group of base matrix 4×4 over F_q *is* q^4-1.The MDLP over F_{97} with base matrix 4×4 is solved by 35375 msec.

Therefore, we conclude the order of the Matrices group of base matrix $n \times n$ over F_q is q^n-1.
The following table explain the running time of the both encryption and decryption of each method:

Table2: Total Running Time of encryption/decryption methods in msec

Message Size	El-Gamal with DLP	El-Gamal with MDLP 2×2	El-Gamal with MDLP 3×3	El-Gamal with MDLP 4×4
K Bytes	32	32	32	32
10 K Bytes	219	234	250	250
20 K Bytes	437	468	515	515
30 K Bytes	656	703	781	781
40 K Bytes	875	938	1046	1046
50 K Bytes	1094	1172	1296	1296
M Byte	2187	2343	2594	2594

Table 3: shows the Running Time of solving and analysis of DLP and MDLP over F_{97} with 2×2, 3×3 and 4×4 base matrices.

Table 3: Running Time of solving DLP and MDLP over F_{97}

Problem Type	DLP	MDLP 2×2	MDLP 3×3	MDLP 4×4
Running Time in msec	0	16	188	35375

There is a clear growth of the time execution when use the Matrices group and increase as long as the matrix size is increased. This increasing with small numbers, what is happen when a large number is applied, such as 100 digit number, 200 digit number or more, the complexity is increased rapidly, show Figure 1.

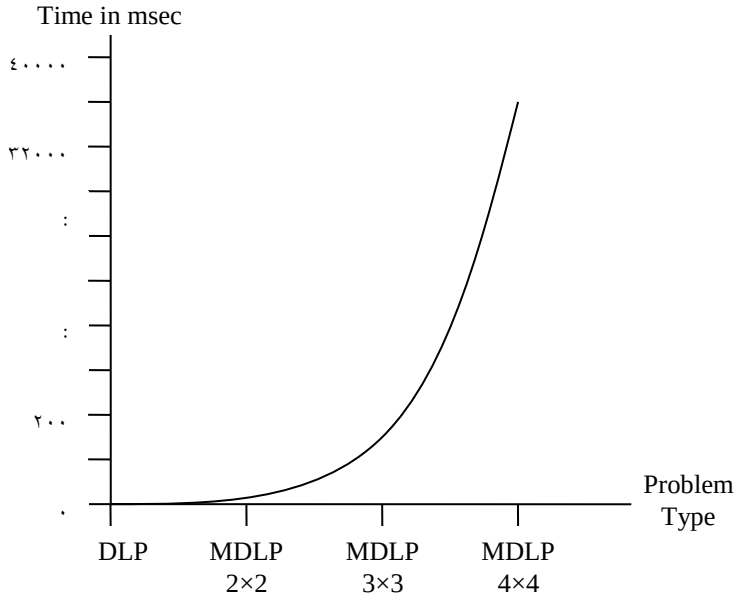

Figure 1 The complexity of DLP and MDLPs

8. Security Of Matrices Cryptography(Mc)

The complication associated with MC comes from the wide variety of possible group structures of the matrix element in the Matrices group and from the fact that matrix multiplication is somewhat complicated.

The security of MC depends on how difficult it is to determine the integer d, given the square matrix B and the square matrix A^d where $B= A^d \bmod q$. This is referred to as the MDLP. Also that it appears to offer equal security for a far smallest bit size.

The group structure of the Matrices group has a complex operation such that, multiplying a matrix-by-matrix, therefore, the group structure of the matrices increases complexity as long as the matrix size is increased. This gives more complicated operation than group structure of the ECC.

Also it appears to offer equal security for the smallest bit size, for two reasons. The first reason is that the operations are applied -instead of multiplication of two integer numbers- as matrix-by-matrix multiplications, in the other hand, the complexity and intractability are increased as much as the size of base matrix is increased. The second reason is that the size (order) of the Matrices group $M(F_q)$ of matrix of order n appears at most q^n -1 or its factors that means the calculation is applied with q-bit size, while to solve the MDLP needs q^n -1 matrix multiplications.

9. Conclusion

The project defined the Matrices group that proved as an Abelian group to use it in the proposed cryptosystems. Then, discover that the Matrices group has a one way function similar to DLP and ECDLP, which MDLP. The construction of cipher system is based on the difficulty of solution of the MDLP that is a clear change in the cryptography, and opens new windows for treatment with special group and new operations. There is a computational advantage in using the matrices cryptography with the shortest key length that reduces the overall calculations with secure system. The structures of the square matrices consist of many numbers that provide the ability to encipher large blocks of plaintext. Each matrix consists of four, nine, sixteen, and so on, that makes the cryptogram may encipher efficiently with the shortest key size. The MDLP appears more complex than ECDLP, because the matrices operations increase the complexity as long as the matrix size is increased.

The MDLP over F_q is more intractable than the DLP in F_q and ECDLP in $E(F_q)$. It is this feature that makes cryptographic system based on the MDLP even more secure than that based on the DLP and ECDLP, because the $M(F_q)$ gives a large group over small field size. Since the group $M(F_q)$ with $n \times n$ base matrix may give group of order q^n-1, therefore, some of the strongest algorithms for solving DLP or ECDLP cannot be adaptive to the MDLP.

References

Anton, H.; I. Bivens and S. Davis, *Calculus*, John Wiley & Sons Inc., 2002.

Denning, D. E. R." Cryptography and Data Security ", Addison Wesley Pub., 1982.

Fibikova L. & J. Vyskoc, Practical cryptography - the Key Size Problem: PGP after years, 2001.

Finney R. L.and G. B. Thomas, *Calculus*, Addison Wesley Pub., 1990.

Fraleigh, J. B.*; A First Course in Abstract Algebra*, Addison Wesley Pub., 1982.

IEEE P1363, Standard Specifications for Public-Key Cryptography, draft, 1997.

Kizza, J. M.Computer Network Security, Springer Inc., 2005.

Koblitz, N. *A Course in Number Theory and Cryptography*, Springer-Verlag, 1987.

Koblitz, N. A. Menezes, and S. Vanstone, The State of Elliptic Curve Cryptography, Designs, Codes and Cryptography, 19, 173–193 (2000), 2000.
Lubbe, J. C. A. V. D.*Basic Methods for Cryptography*, Cambridge ISPN 1998.
Mollin, R. A. *Number Theory and Applications*, NATOASI series 1989.
Nathanson, M. B. (2000). *Elementary Methods in Number Theory*, Graduate Text in Mathematics 195, Springer-Verlag, 2000.
Okamoto T. & S. Uchiyama, Security of an Identity-Based Cryptosystem and the Related Reductions, Advance in Cryptology, EuroCrypt 98, LNCS Vol. 1403, pp. 576-560, Springer-Verlag,1998.
Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, New York, 2nd edition, 1996.
Stallings, W. *Cryptography and Network Security, Principle and Practice*, Addison Wesley, 1999.
Stamp, M. *Information Security Principles and practice*, JohnWiley & Sons, Inc., 2006.
Williams, G. *Linear Algebra with Applications*, 4th edition, Jones and Bartlett Publishers, Inc., 2001.
Yan, S.Y.*Number Theory for Computing* , Springer-Verlag, 2000.