Undirected Complete Graph to Design New Public Key Cryptosystem

Karrar Taher R. Aljamaly\textsuperscript{1} and Ruma Kareem K. Ajeena\textsuperscript{1}
\textsuperscript{1}Mathematics Department, Education College for Pure Sciences, University of Babylon, Iraq

E-mail: karrar.taher1987@gmail.com & pure.ruma.k@uobabylon.edu.iq

Abstract. In this work, the alternative version of the ElGamal public key cryptosystem (EPKC) first is proposed. The revised EPKC depended on an extended discrete logarithm problem over matrices. More secure communications with the revised EPKC in compare to the original one are obtained. Also, a new public key cryptosystem has been designed based on undirected complete graph (UCG). The plaintext data is selected as a subset from the finite fields which is corresponded to undirected complete graph $G(V,E)$, where $V$ and $E$ are finite sets of the vertices and edges respectively. The plaintext is encrypted using the matrix power function (MPF) and the minimum spanning tree (MST) problem of undirected complete graph (UCG). The adjacent matrix representation of MST is used to speed up the calculations. More secure on the proposed cryptosystem has been determined in compare to other asymmetric public key cryptosystems. New computational results of the revised EPKC and the UCG-based public key cryptosystem are discussed.

1. Introduction

Several mathematical concepts from algebra and number theory, especially the discrete logarithm problem (DLP) is employed to solve some modern problems in mathematics, computer sciences and others. In 1976, W. Diffie and M. Hellman [1] introduced new directions in cryptography depended on the computations of the DLP and exchange the results of these computations between two entities. In 1985, Taher Elgamal [2] presented another version of the cryptosystems which are public-key cryptosystems and their digital signatures that are also based on the DLP. In 2013, Delaram Kahrobaei, et al [3], proposed the public key exchange using the matrices over group rings. In the same year, Yamuna, M., et al [4], introduced the encryption of a binary string using the music notes and graph theory. They proposed two phases of an encryption algorithm for transforming a plaintext musical notes by using the graph theory.

In 2014, Wael Al Eta'awi [5], proposed a symmetric encryption algorithm to encrypt and decrypt the data through employing the graph theory concepts. In 2015, Shubham Agarwal and Anand Singh Uniyal [6], presented a prime weighted graph (PWG) as a new graph and they proposed an encryption scheme based on the (PWG) with more secure communication. In 2016, Linkeova, Romana, and Pavel Prihoda [7], presented the cryptanalysis that is based on the theory of symmetric group representations. They introduced the modified Diffie-Hellman protocol using the matrix power. In 2018, Amudha et al [8], proposed an encryption technique, where each character of the data has been encrypted into an Euler Graph. In 2020, Karar Aljamaly and Ruma Ajeena [9] proposed two studies, first one is a new study used a matrix power function (MPF) to revise the discrete logarithm problem over matrices.
encryption schemes. They proposed alternative versions of the Diffie-Hellman key exchange and ElGamal public key cryptosystem which are based on the MPF-DLP. Also in their work, a hybrid algorithm is proposed based on the MPF and the graph theory. And in [10], they proposed a new graph to design a new version of an asymmetric encryption scheme. This graph is formed based on the scalar multiplication operation on elliptic curve defined over a prime field, which is called an elliptic scalar multiplication (ESM) graph.

In this work, first the alternative ElGamal public key cryptosystem is proposed based on the matrix power function (MPF) problem for more secure communication. Another point that this work focusses on it is to propose a new public key cryptosystem which is designed based on the undirected complete graph (UCG). The computations of the ciphertext is done on a subset of a prime field through employing the MPF and the minimum spanning tree (MST) of UCG.

The outline of this work includes Section 2, which shows the mathematical background related to some basic facts of the graph theory. In Section 3, an alternative version of the ElGamal public key cryptosystem based on the MPF. In Section 4, the UCG-based public key cryptosystem is proposed to encrypt a subset of the prime field. Section 5, discusses a study case on the UCG public key cryptosystem. In section 6, the computational results of the proposed public key cryptosystems are discussed and shown in Appendix (A). In Section 7, the security considerations on the proposed public key cryptosystems are determined. Finally, Section 8, draws the conclusions.

2. Basic Facts of the Graph Theory

The graph theory is another main part in mathematics which considers as a fundamental tool that can employ with algebra and number theory in many applications. The graph consists of two finite sets. These sets are called vertices and edges sets, $V$ and $E$ respectively. There are different kinds of graphs such as a simple graph, a multigraph, a complete graph, a planer graph, the weighted graph and others [10]. The adjacency matrix representations of the graphs and the minimum spanning tree [10] of the graphs possess a critical role in several mathematical applications. For more details of the graph theory, one can see [11,12].

3. The PM-ElGamal Public Key Cryptosystem

The original ElGamal public key (EPKC) encryption algorithm is closely related to Diffie-Hellman key exchange. It depends on the DLP. The entities agree on a public large prime $p$ and a primitive root $g$. One of them chooses a private key $a$ and computes a public key by $A = g^a \pmod p$. Another (second) entity selects a plaintext $m$ and a random ephemeral key $k$. He/she uses a public key $A$ to compute $C_1 = g^k \pmod p$ and $C_2 = mA^k \pmod p$ and sends ciphertext $(C_1, C_2)$ to first entity. The decryption process can be done using a private key to computes $(C_1^a)^{-1} \times C_2 \pmod p$ The last quantity is equal to $m^2$ [2].

The EPKC can take alternative version based on the power of matrix. The public parameters are a prime $p$ and a public matrix $D \in GL_n(F_p)$, where $GL_n(F_p)$ is a set of the matrices that have inverses which their elements belong to a prime field $F_p$. This set form a group with the multiplication operation [14]. First entity chooses randomly a private key as a number $a$ such that $a \in \{2,3,...,p-1\}$ and computes a public key $A$ by $A = D^a \pmod p$.

Second entity employs a public key $A$ to encrypt a message which is a matrix $M_{x \times n}$. He/she chooses an ephemeral key as a number $b$ such that $b \in \{2,3,...,p-1\}$. The ciphertext $C$ which is a pair $(C_1, C_2)$ of two matrices that are computed by $C_1 = D^b \pmod p$ and $C_2 = A^b \times M \pmod p$.

The decryption process to recover a message can be done first to compute $(C_1^a)^{-1}$ using a secret key $a$. The multiplication matrix $(C_1^a)^{-1} \times C_2$ is computed to give the original message $M$. Algorithms (3.1),(3.2) and (3.3) are used for obtaining the several numerical results of the revised EPKC.
3.1. Algorithm. The MP-ElGamal Public Key Cryptosystem: Keys Generation Process.
Input: A prime $p$ and a matrix $D \in GL_n(F_p)$.
Output: The public key $A$, where $A \in GL_n(F_p)$.
- Alice chooses a number $a$ as her private key, where $a \in \{2,3,\ldots, p-1\}$.
- She computes her public key $A = D^a \pmod{p}$.
- Alice keys are $(a,A)$.

3.2. Algorithm. The MP-ElGamal Public Key Cryptosystem: Encryption Process.
Input: A prime $p$ and a public key $A$.
Output: The ciphertext $C = (C_1, C_2)$, where $C_1$ and $C_2$ are two matrices.
- Bob chooses an ephemeral secret key $b$, where $b \in \{2,3,\ldots, p-1\}$.
- He chooses his message $M$.
- He computes the ciphertext through the computations of two matrices $C_1 = A^b \pmod{p}$ and $C_2 = A^b \times M$.
- Bob sends the ciphertext pair $(C_1, C_2)$ to Alice.

3.3. Algorithm. The MP-ElGamal Public Key Cryptosystem: Decryption Process.
Input: A prime $p$ and a secret key $a$.
Output: The message $M$, where $M \in \mathbb{F}_p^{m \times n}$ is a matrix.
- Alice first uses her secret key $a$ to compute $C_1^\prime$.
- She computes the matrix inverse $(C_1^\prime)^{-1}$.
- She computes the multiplication matrix $(C_1^\prime)^{-1} \times C_2 = M_{n \times m}$.

4. The UCG-based Public Key Cryptosystem
In this section, a new public key cryptosystem has been designed. The undirected complete graph (UCG) is employed for this design. Two entities agreed to choose the public parameters. These parameters are a prime $p$ and a square matrix $D \in GL_n(F_p)$. One of the entities called Alice generates the keys, public and private keys. She selects a number $a$ in a secret way, where $a \in \{2,3,\ldots, p-1\}$ which is a private key for her. Depending on a private key $a$, Alice computes her public key $A = D^a$. So, Alice's keys are given with a pair $(a,A)$. Algorithm (4.1) is used for obtaining the several numerical results to generate the keys.

4.1. Algorithm. The UCG-based Public Key Cryptosystem: Keys Generation Process.
Input: A prime $p$ and a matrix $D \in GL_n(F_p)$.
Output: The public key $A$, where $A \in GL_n(F_p)$.
1. Alice chooses a number $a$ as her a private key, where $a \in \{2,3,\ldots, p-1\}$.
2. She computes her public key $A = D^a \pmod{p}$.
3. Alice keys are $(a,A)$.

Another entity, called Bob wants to communicate with Alice for sending the important information which is represented by a plaintext $m$ that is a subset of a prime field $F_p$. Bob creates undirected complete graph $G(V,E)$ of a message $m = \{m_1,m_2,\ldots,m_j\}$, where $V = \{v_1,v_2,\ldots,v_j\}$ and $E = \{e_1,e_2,\ldots,e_j\}$. This graph convert into a weighted graph through computing the distance between any two vertices. He determines a minimum spanning tree (MST) graph $G'$ of a graph $G$. Representing the MST graph by a matrix $M$. A matrix $M$ is converted into $M'$ by adding the elements of $m$ to diagonal of a matrix $M$ respectively. Bob chooses an ephemeral secret key $b$, where
\(b \in \{2,3,\ldots,p-1\}\) and compute his ciphertext. The ciphertext \(C\) is computed through the computations of two square matrices \(C_1 = D^b\) and \(C_2 = A^b \times M^{'}.\) Bob sends the ciphertext pair \(C = (C_1, C_2)\) to Alice. Several numerical results to compute the ciphertext are got using Algorithm (4.2).

4.2. Algorithm. The UCG-based Public Key Cryptosystem: Encryption Process.

Input: A prime \(p\) and a public key \(A.\)

Output: The ciphertext \((C_1, C_2)\) where \(C_1\) and \(C_2\) are two matrices.

1. Bob selects his message \(m = \{m_1, m_2, \ldots, m_l\}\) which is a subset of \(F_p.\)
2. He forms a complete graph \(G(V,E)\) of a message \(m,\) where \(V = \{v_1, v_2, \ldots, v_l\}\) and \(E = \{e_1, e_2, \ldots, e_s\}\) such that every vertices of \(V\) has a code number of \(m\) respectively.
3. He converts a complete graph \(G\) into a weighted graph through computing the weights \(w_{ij}\) for \(i = 2,3,\ldots,s\) for all edges, which equal to the distance \(|\text{Code}(v_i) - \text{Code}(v_j)|.\)
4. He determines a minimum spanning tree (MST) \(G^{'}\) of a graph \(G.\)
5. He represents \(G^{'\} by a corresponding matrix \(M.\)
6. He converts a matrix \(M\) to \(M^{'\} by adding the elements of \(m\) to diagonal of a matrix \(M\) respectively.
7. Bob chooses an ephemeral secret key \(b,\) where \(b \in \{2,3,\ldots,p-1\}.\)
8. He computes the ciphertext through the computations of two square matrices \(C_1 = D^b\) and \(C_2 = A^b \times M^{'}.\)
9. Bob sends the ciphertext pair \((C_1, C_2)\) to Alice.

Upon receiving Alice the ciphertext \((C_1, C_2)\), she uses her private key to recover the plaintext. She first computes a matrix \(C_1^{'\). Then, an inverse matrix \((C_1^{'\})^{-1}\) is calculated. After that, the computation of the multiplication matrix \((C_1^{'\})^{-1} \times C_2 = M^{'\) is done. The diagonal elements, in \(M^{'\), are selected as a list which they form a set of the original message elements \(m.\)

4.3. Proposition. The decryption process is computed by \((C_1^{'\})^{-1} \times C_2 = M^{'\).

Proof.
\[
(C_1^{'\})^{-1} \times C_2 = ((D^b)^{-1}) \times (A^b \times M^{'},\) since \(C_1 = D^b\)
\[
= (D^b)^{-1} \times (A^b \times M^{'\), since \(C_2 = A^b \times M^{'\).
\[
= (D^b)^{-1} \times ((D^b)^{'} \times M^{'\), since \(A = D^a\)
\[
= (D^b)^{-1} \times D^{ab} \times M^{'\), since the multiplication matrix is associative
\[
= I \times M^{'\)
\[
= M^{'\.

The implemented results can be got using Algorithm (4.4).

4.4. Algorithm. The UCG-based Public Key Cryptosystem: Decryption Process.

Input: A prime \(p\) and a ciphertext \((C_1, C_2).\)

Output: The message \(m,\) where \(m\) is a subset of \(F_p.\)

1. Alice first uses her secret key \(a\) to compute \(C_1^{'\).
2. She computes the matrix inverse \((C_1^{'\})^{-1}.\)
3. She computes the multiplication matrix \((C_1^{'\})^{-1} \times C_2 = M^{'\).
4. The diagonal elements of \(M^{'\) are selected as a list which they form a set of the original message elements \(m.\)

5. The Study Case on the UCG Public Key Cryptosystem

Let \(p=941\) be a prime number. Suppose a matrix \(D\) of size \(12 \times 12\) is
where \( D \in GL(12,F_{941}) \). The matrix \( A = D^{16} \text{ (mod 941)} \) is computed as follows:

\[
\begin{pmatrix}
8 & 9 & 74 & 715 & 547 & 539 & 703 & 755 & 535 & 117 & 53 & 327 \\
302 & 331 & 620 & 541 & 290 & 640 & 314 & 393 & 324 & 239 & 824 & 664 \\
150 & 404 & 258 & 280 & 173 & 255 & 51 & 353 & 586 & 733 & 363 & 386 \\
442 & 452 & 337 & 899 & 727 & 449 & 748 & 161 & 703 & 918 & 476 & 125 \\
246 & 759 & 127 & 505 & 521 & 291 & 99 & 95 & 726 & 206 & 593 & 409 \\
178 & 274 & 237 & 286 & 720 & 751 & 16 & 243 & 363 & 262 & 876 & 551 \\
564 & 781 & 504 & 566 & 73 & 101 & 191 & 921 & 258 & 102 & 912 & 21 \\
250 & 642 & 546 & 325 & 55 & 645 & 661 & 822 & 747 & 411 & 410 & 740 \\
388 & 815 & 365 & 277 & 578 & 806 & 682 & 696 & 192 & 62 & 244 & 660 \\
55 & 621 & 483 & 305 & 478 & 607 & 217 & 582 & 916 & 786 & 687 & 424 \\
445 & 34 & 243 & 44 & 307 & 681 & 914 & 322 & 215 & 326 & 938 & 850 \\
755 & 415 & 272 & 322 & 535 & 280 & 863 & 888 & 339 & 83 & 187 & 710
\end{pmatrix}
\]

Bob chooses his message \( m = \{41,121,250,361,399,455,556,690,780,821,900,921\} \) as a subset of a finite field \( F_{941} \). The message \( m \) is corresponding to the UCG as shown in Figure (1).
Figure 1. The UCG that corresponds the message \( m \).

The weights of all edges on the UCG is computed and shown in Table (1)

| \( W_{i,j} \) | \( W_{i,j} \) value | \( W_{i,j} \) | \( W_{i,j} \) value | \( W_{i,j} \) | \( W_{i,j} \) value | \( W_{i,j} \) | \( W_{i,j} \) value |
|----------|-----------------|----------|-----------------|----------|-----------------|----------|-----------------|
| \( W_{1,2} \) | 80               | \( W_{2,9} \) | 659             | \( W_{4,9} \) | 419             | \( W_{7,8} \) | 134             |
| \( W_{1,3} \) | 209              | \( W_{2,10} \) | 700             | \( W_{4,10} \) | 460             | \( W_{7,9} \) | 224             |
| \( W_{1,4} \) | 320              | \( W_{2,11} \) | 779             | \( W_{4,11} \) | 539             | \( W_{7,10} \) | 265             |
| \( W_{1,5} \) | 358              | \( W_{2,12} \) | 800             | \( W_{4,12} \) | 560             | \( W_{7,11} \) | 344             |
| \( W_{1,6} \) | 414              | \( W_{3,4} \) | 111             | \( W_{5,6} \) | 56              | \( W_{7,12} \) | 365             |
| \( W_{1,7} \) | 515              | \( W_{3,5} \) | 149             | \( W_{5,7} \) | 157             | \( W_{8,9} \) | 90              |
| \( W_{1,8} \) | 649              | \( W_{3,6} \) | 205             | \( W_{5,8} \) | 291             | \( W_{8,10} \) | 131             |
| \( W_{1,9} \) | 739              | \( W_{3,7} \) | 306             | \( W_{5,9} \) | 381             | \( W_{8,11} \) | 210             |
| \( W_{1,10} \) | 780              | \( W_{3,8} \) | 440             | \( W_{5,10} \) | 422             | \( W_{8,12} \) | 231             |
| \( W_{1,11} \) | 859              | \( W_{3,9} \) | 530             | \( W_{5,11} \) | 501             | \( W_{9,10} \) | 41              |
| \( W_{1,12} \) | 880              | \( W_{3,10} \) | 571             | \( W_{5,12} \) | 522             | \( W_{9,11} \) | 120             |
| \( W_{2,3} \) | 129              | \( W_{3,11} \) | 650             | \( W_{6,7} \) | 101             | \( W_{9,12} \) | 141             |
| \( W_{2,4} \) | 240              | \( W_{3,12} \) | 671             | \( W_{6,8} \) | 235             | \( W_{10,11} \) | 79              |
| \( W_{2,5} \) | 278              | \( W_{4,5} \) | 38              | \( W_{6,9} \) | 325             | \( W_{10,12} \) | 100             |
| \( W_{2,6} \) | 334              | \( W_{4,6} \) | 94              | \( W_{6,10} \) | 366             | \( W_{11,12} \) | 21              |
| \( W_{2,7} \) | 435              | \( W_{4,7} \) | 195             | \( W_{6,11} \) | 445             |                 |                 |
| \( W_{2,8} \) | 569              | \( W_{4,8} \) | 329             | \( W_{6,12} \) | 466             |                 |                 |

The MST sub-graph is shown in Figure (2).
The MST sub-graph can be represented by the following matrix

\[
M = \begin{pmatrix}
0 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
80 & 0 & 129 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 129 & 0 & 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 111 & 0 & 38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 38 & 0 & 56 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 56 & 0 & 101 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 101 & 0 & 134 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 134 & 0 & 90 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 90 & 0 & 41 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 0 & 79 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 79 & 0 & 21 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 & 0
\end{pmatrix}
\]

The modified matrix $M'$ of $M$ by adding the elements of message at the diagonal. So a matrix $M'$ is given by
\[ M' = \begin{bmatrix} 41 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 80 & 121 & 129 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 129 & 250 & 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 111 & 361 & 38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 38 & 399 & 56 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 56 & 455 & 101 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 101 & 556 & 134 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 134 & 690 & 90 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 780 & 41 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 821 & 79 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 79 & 900 & 21 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 921 \end{bmatrix} \]

The ciphertext \((C_1, C_2)\) is computed as follows.

\[ C_1 = \begin{bmatrix} 771 & 744 & 771 & 409 & 112 & 841 & 855 & 727 & 80 & 384 & 351 & 709 \\ 669 & 479 & 506 & 731 & 54 & 815 & 695 & 370 & 689 & 122 & 665 & 382 \\ 717 & 763 & 816 & 46 & 4 & 471 & 134 & 884 & 620 & 177 & 58 & 719 \\ 267 & 447 & 66 & 758 & 510 & 235 & 125 & 68 & 138 & 792 & 590 & 845 \\ 928 & 319 & 738 & 200 & 801 & 771 & 371 & 182 & 857 & 134 & 523 & 797 \\ 937 & 304 & 833 & 718 & 638 & 513 & 167 & 626 & 251 & 590 & 897 & 747 \\ 222 & 557 & 810 & 213 & 480 & 669 & 533 & 682 & 735 & 506 & 323 & 502 \\ 556 & 425 & 223 & 862 & 104 & 679 & 294 & 534 & 888 & 73 & 390 & 554 \\ 517 & 281 & 31 & 436 & 797 & 821 & 410 & 523 & 883 & 922 & 112 & 708 \\ 507 & 930 & 880 & 359 & 902 & 795 & 281 & 247 & 440 & 395 & 869 & 892 \\ 453 & 640 & 143 & 781 & 607 & 328 & 833 & 932 & 577 & 781 & 687 & 829 \\ 390 & 656 & 661 & 238 & 940 & 519 & 900 & 676 & 934 & 801 & 858 & 514 \end{bmatrix} \]

Where \(C_1 = D^4 \pmod{941}\). Now, the matrix \(A^4\) is computed by
Now, for decryption process, it requires computing the following matrix

$$C_1 = A^4 \times M \pmod{941}$$

Where $C_1 = A^4 \times M \pmod{941}$. Now, for decryption process, it requires computing the following matrix.
The matrix \( M' = (C_i^{16})^{-1} \times C_2 \) is
\[ M' = \begin{pmatrix}
41 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
80 & 121 & 129 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 129 & 250 & 111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 111 & 361 & 38 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 38 & 399 & 56 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 56 & 455 & 101 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 101 & 556 & 134 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 134 & 690 & 90 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 90 & 780 & 41 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 821 & 79 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 79 & 900 & 21 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 921 \\
\end{pmatrix} \pmod{941}, \]

Then, the elements of diagonal in a matrix \( M' \) is the original message

\[ m = \{41, 121, 250, 361, 399, 455, 556, 690, 780, 821, 900, 921\} \]

6. The Computational Results on the Proposed Public Key Cryptosystems
This section discusses the simple experimental samples of the proposed public key cryptosystems as follows.

6.1. The Computational Results on the MP-ElGamal Public Key Cryptosystem
Some simple computations of the MP-ElGamal public key cryptosystem have been done. The experimental samples with different values of a prime \( p \) are chosen. The computational results to generate the keys, encryption and decryption processes are shown in Appendix (A) by Tables (2), (3) and (4) respectively.

6.2. The Computational Results on the UCG Public Key Cryptosystem
The computation on the UCG public key cryptosystem have been done with several numerical results. Some experimental samples with different values of a prime \( p \) are chosen. The computational results to generate the keys, encryption and decryption processes are given in Appendix (A) by Tables (5), (6), (7) and (8) respectively.

7. The Security Considerations on Proposed Public Key Cryptosystems
The security of the revised EPKC determines through the difficulty of finding the DLP that solving on the matrices. Whereas, the security on the proposed UCG-based public key cryptosystem can be determined first by the hardness solving the DLP that computing on the matrices and the difficulty to determine the correct MST graph among all other possible cases of the MST graphs that can be created from undirected complete subgraph. As well as, another point focuses on a good choice of the domain parameters of the revised EPKC and proposed UCG-based public key cryptosystem. Specifically, with a large prime \( p \) that gives us the possibility to choose and generate the big size matrices over a set \( GL_n(F_p) \) which help us to implement the revised EPKC and the proposed UCG based public key cryptosystem with more secure in compare with other public key algorithms.
8. Conclusions
This work first presents the revised version of the EPKC which is done based on the DLP that is computed over the matrices. And also, it proposed a new public key cryptosystem based on the UCG. A plaintext on the proposed UCG-based public key cryptosystem is selected as a subset over a prime field. It encrypted using the MP and the MST of the UCG. Some computational results are implemented and discussed with different samples of the primes $p$ and big matrices sizes using the Sage Codes for more secure. The security considerations on the revised version of the EPKC and the proposed UCG-based public key cryptosystem are determined. The UCG-based public key cryptosystem is more security than the MP-ElGamal public key cryptosystem because it depends on the difficulty to determine the MST of the weighted complete graph. And also, it is a faster than the PM-ElGamal public key cryptosystem, since a message is considered as a reduced matrix most of its elements are zeros.

References
[1] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE transactions on Information Theory, 22(6), 644-654.
[2] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on Information Theory, 31(4), 469-472.
[3] Kahrobaei, D., Koupparis, C., & Shpilrain, V. (2013). Public key exchange using matrices over group rings. Groups Complexity Cryptology, 5(1), 97-115.
[4] Yamuna, M., Sankar, A., Ravichandran, S., & Harish, V. (2013). Encryption of a Binary String using music notes and graph theory. International Journal of Engineering and Technology, 5(3), 2920-2925.
[5] Al Etaiwi, W. M. (2014). Encryption algorithm using graph theory, Journal of Scientific Research and Reports, 2519-2527.
[6] Agarwal, Shubham, and Anand Singh Uniyal. Prime Weighted Graph in cryptographic System For Secure Communication. International Journal of Pure and Applied Mathematics, 105(3),(2015), 325-338.
[7] Linkeova, R., & Prihoda, P. (2016). Cryptanalysis based on the theory of symmetric group representations. Infcommunication Journal, 8(1), 17-23.
[8] Amudha, P., Sagayaraj, A. C., & Sheela, A. S. (2018). An Application of Graph Theory in Cryptography. International Journal of Pure and Applied Mathematics, 119(13), 375-383.
[9] Aljamaly, K. T. R., & Ajeena, R. K. K. (2020). A matrix power function to revise the discrete logarithm encryption schemes. First International Conference on Mathematics (FICM2020), University of Sulaimani, Sulaimani City, Kurdistan Region, Iraq
[10] Aljamaly, K. T. R., & Ajeena, R. K. K. (2020). The scalar multiplication graph of elliptic curve groups and apply it in cryptography, Journal of Discrete Mathematical Sciences & Cryptography.(submitted).
[11] Wilson, R. J. (1979). Introduction to graph theory. Pearson Education India.
[12] Ray, Santanu Saha. Graph theory with algorithms and its applications: in applied science and technology. Springer Science & Business Media, 2012.
[13] Gross, Jonathan L., Jay Yellen, and Ping Zhang, eds. Handbook of graph theory. CRC press, 2013.
[14] Green, James A. The characters of the finite general linear groups, Transactions of the American Mathematical Society, 80(2), (1955), 402-447.
Appendix A

Table 2. The experimental results of the MP-ElGamal public key cryptosystem: key generation process.

| p   | Shear key D | A private key $a$ | A public key $A = D^a \pmod{p}$ |
|-----|-------------|-------------------|---------------------------------|
| 73  | 6  6  60 72 | 5                 | 44  1  49  44                   |
|     | 23  46  38 39 |                   | 17  27  50  65                  |
|     | 60  14  22 54 |                   | 48  65  44  11                  |
|     | 32  11  45 64 |                   | 71  33  53  18                  |
| 89  | 81  77  19 25 | 6                 | 52  47  66  1                   |
|     | 14  6  87 11 |                   | 77  50  63  64                  |
|     | 3  61  18 8 |                   | 69  65  83  54                  |
|     | 22  9  15 37 |                   | 41  27  33  39                  |
| 127 | 113  13  17 55 | 5                 | 14  109  10  42                 |
|     | 3  75  100 60 |                   | 17  36  7  35                   |
|     | 55  85  32 19 |                   | 12  20  122  48                |
|     | 10  90  29 120 |                  | 2  76  2  17                   |
| 167 | 3  77  10 14 50 |                   | 48  164  143  142  52         |
|     | 17  150 33 1 19 |                  | 66  74  150  79  54           |
|     | 10  15  9 125 40 |                 | 142  77  8  105  111        |
|     | 100  7  60 4 23 |                  | 104  7  65  70  31           |
|     | 17  5  85 15 71 |                  | 120  122  143  30  53        |
| 179 | 175  10  2 15 75 |                   | 162  52  73  117  73         |
|     | 1  145  31 99 80 |                  | 12  38  113  119  74         |
|     | 15  23  91 42 71 |                 | 113  80  18  39  149        |
|     | 101  66  89 48 6 |                  | 85  102  144  85  91        |
|     | 174  74  3 14 92 |                  | 70  130  125  124  59       |
Table 3. The computational results of the MP-ErGama public key cryptosystem: encryption process.

| b  | $C_1 = D^b \pmod{p}$ | A message $M$ | $C_2 = A^c \times M$ |
|----|------------------------|---------------|---------------------|
| 7  | 65 66 41 35            | 7 33 50 17 66 | 56 11 59 3 51       |
|    | 26 45 58 65            | 22 71 58 3 60 | 65 65 8 12 12       |
|    | 38 68 44 0             | 2 61 35 44 61 | 10 46 50 15 30      |
|    | 39 69 43 61            | 55 17 25 13 14| 14 15 61 42 47      |
|    | 66 65 34 37            | 85 11 64 72   | 68 11 77 65         |
|    | 39 69 22 80            | 10 37 45 19   | 61 81 88 15         |
|    | 87 8 39 81             | 25 40 54 59   | 47 13 74 16         |
|    | 47 7 87 35             | 68 12 2 29    | 1 29 7 53           |
|    | 85 33 103 94           | 12 4 125 111  | 126 116 111 14      |
|    | 97 62 48 62            | 89 92 17 55   | 110 36 6 100        |
|    | 97 90 7 32             | 75 78 101 51  | 125 3 39 65         |
|    | 99 60 63 17            | 66 3 99 121   | 37 59 30 23         |
| 4  | 11 86 11 28 117        | 13 165 2 7 14 | 115 62 94 82 153    |
|    | 75 50 49 24 85         | 125 15 87 85 99 | 162 22 153 134 142 |
|    | 30 61 166 160 141     | 33 78 55 34 1 | 1 155 77 92 150     |
|    | 131 159 154 37 115    | 12 135 119 8 119 | 27 6 6 70 145      |
|    | 116 165 141 2 110     | 1 11 145 5 100 | 144 47 3 14 131    |
|    | 71 115 54 151 125     | 101 5 176 152 15 | 46 34 51 85 163    |
|    | 80 135 173 41 122     | 14 6 87 164 85 | 125 19 41 113 115  |
|    | 17 53 51 47 69        | 11 3 91 22 18  | 80 124 115 170 168 |
|    | 63 3 72 51 136        | 18 8 132 36 75 | 106 159 48 31 86   |
|    | 81 42 46 83 78        | 22 66 9 111 102 | 20 48 56 114 74    |
Table 4. The computational results of the MP-ElGamal public key cryptosystem: decryption process.

| $C_1^* \pmod{p}$ | $(C_1^*)^{-1} \pmod{p}$ | $(C_1^*)^{-1} \times C_2 \pmod{p} = M$ |
|-------------------|-------------------------|-----------------------------------|
| 64 53 38 34       | 43 1 65 59              | 7 33 50 17 66                     |
| 34 48 49 21       | 38 40 18 16             | 22 71 58 3 60                     |
| 32 1 72 13        | 43 67 60 53             | 2 61 35 44 61                     |
| 19 4 53 62        | 63 67 23 71             | 55 17 25 13 14                    |
| 34 81 76 60       | 7 46 37 11              | 85 11 64 72                       |
| 35 57 32 26       | 74 33 68 4              | 10 37 45 19                       |
| 34 17 62 52       | 23 12 85 53             | 25 40 54 59                       |
| 84 12 35 61       | 2 71 8 31               | 68 12 2 29                        |
| 70 83 5 64        | 16 57 13 86             | 12 4 125 111                      |
| 35 79 3 47        | 16 124 18 126           | 89 92 17 55                       |
| 28 72 41 34       | 114 118 42 41           | 75 78 101 51                      |
| 122 100 47 98     | 62 101 106 35           | 66 3 99 121                       |
| 87 50 74 45 106   | 91 40 15 70 151         | 13 165 2 7 14                     |
| 52 5 87 156 130   | 11 16 13 78 160         | 125 15 87 85 99                   |
| 66 17 0 34 117    | 99 68 59 146 159        | 33 78 55 34 1                     |
| 87 99 20 120 119  | 18 49 157 114 67        | 12 135 119 8 119                  |
| 107 151 37 86 3   | 86 145 151 143 159      | 1 11 145 5 100                    |
| 54 172 149 160 17 | 90 66 72 129 177        | 101 5 176 152 15                  |
| 65 170 4 167 30   | 86 172 124 7 128        | 14 6 87 164 85                    |
| 73 18 12 177 149  | 70 129 41 81 51         | 11 3 91 22 18                     |
| 78 42 76 50 13    | 153 103 133 154 172     | 18 8 132 36 75                    |
| 151 60 178 112 171| 2 46 96 24 57           | 22 66 9 111 102                   |
Table 5. The experimental results of the UCG public key cryptosystem: key generation process.

| $p$  | Shear key $D$ | A private key $a$ | A public key $A = D^a \pmod{p}$ |
|------|--------------|------------------|-------------------------------|
| 41   | $\begin{bmatrix} 10 & 5 & 33 & 1 \\ 25 & 17 & 8 & 12 \\ 15 & 18 & 23 & 5 \\ 2 & 39 & 13 & 3 \end{bmatrix}$ | 5 | $\begin{bmatrix} 19 & 1 & 8 & 8 \end{bmatrix}$ |
|      | $\begin{bmatrix} 1 & 0 & 42 & 12 & 5 \\ 3 & 10 & 1 & 26 & 9 \end{bmatrix}$ | 7 | $\begin{bmatrix} 5 & 27 & 0 & 16 & 47 \end{bmatrix}$ |
| 59   | $\begin{bmatrix} 41 & 0 & 33 & 1 & 58 \\ 3 & 7 & 5 & 21 & 1 \\ 1 & 4 & 0 & 1 & 30 \end{bmatrix}$ | 4 | $\begin{bmatrix} 55 & 53 & 47 & 17 & 36 \end{bmatrix}$ |
|      | $\begin{bmatrix} 60 & 0 & 1 & 1 & 59 \\ 33 & 7 & 5 & 21 & 1 \end{bmatrix}$ | 6 | $\begin{bmatrix} 49 & 35 & 6 & 28 & 48 \end{bmatrix}$ |
| 61   | $\begin{bmatrix} 88 & 91 & 0 & 1 & 2 \\ 1 & 0 & 1 & 77 & 9 \end{bmatrix}$ | 6 | $\begin{bmatrix} 10 & 9 & 59 & 31 & 22 \end{bmatrix}$ |
|      | $\begin{bmatrix} 60 & 2 & 1 & 93 & 3 \\ 12 & 23 & 5 & 21 & 1 \end{bmatrix}$ | 4 | $\begin{bmatrix} 54 & 29 & 41 & 0 & 77 \end{bmatrix}$ |
| 97   | $\begin{bmatrix} 185 & 123 & 2 & 99 & 14 & 3 \\ 77 & 15 & 1 & 190 & 5 & 45 \end{bmatrix}$ | 5 | $\begin{bmatrix} 48 & 57 & 0 & 165 & 29 \end{bmatrix}$ |
|      | $\begin{bmatrix} 3 & 18 & 13 & 85 & 91 & 10 \\ 10 & 6 & 7 & 14 & 11 & 2 \end{bmatrix}$ | 5 | $\begin{bmatrix} 28 & 33 & 57 & 53 & 3 \end{bmatrix}$ |
| 191  | $\begin{bmatrix} 185 & 123 & 2 & 99 & 14 & 3 \\ 77 & 15 & 1 & 190 & 5 & 45 \end{bmatrix}$ | 5 | $\begin{bmatrix} 163 & 147 & 172 & 75 & 68 & 66 \end{bmatrix}$ |
|      | $\begin{bmatrix} 3 & 18 & 13 & 85 & 91 & 10 \\ 10 & 6 & 7 & 14 & 11 & 2 \end{bmatrix}$ | 5 | $\begin{bmatrix} 164 & 58 & 35 & 164 & 165 & 8 \end{bmatrix}$ |
|      | $\begin{bmatrix} 87 & 25 & 12 & 85 & 152 & 52 \\ 45 & 32 & 9 & 135 & 23 & 13 \end{bmatrix}$ | 5 | $\begin{bmatrix} 39 & 48 & 185 & 77 & 170 & 20 \end{bmatrix}$ |
Figure 3. The UCGs $G_a, G_b, G_c, G_d, G_e$ and their MSTs $G'_a, G'_b, G'_c, G'_d, G'_e$ that correspond to Table (6).
Table 6. The computational results of the representation a message by a matrix : encryption process.

| A message m | UCG | MST |
|-------------|-----|-----|
|              | \( G_a \) | \( G_a' \) | \( G_a \) | \( G_a' \) | \( G_a \) | \( G_a' \) | \( G_a \) | \( G_a' \) | \( G_a \) | \( G_a' \) |
| \{39,19,23,5\} | 0 0 16 0 | 0 0 4 14 | 0 0 4 14 | 0 0 16 0 | 0 0 4 14 | 0 0 4 14 |
| \{7,15,19,34,56\} | 0 8 0 0 0 | 0 8 15 4 0 0 | 0 15 34 22 | 0 8 15 4 0 0 | 0 8 15 4 0 0 | 0 15 34 22 |
| \{23,5,60,43\} | 0 0 14 0 20 | 0 0 4 0 0 | 0 0 4 0 0 | 0 0 14 0 20 | 0 0 4 0 0 | 0 0 4 0 0 |
| \{4,27,33,75,91\} | 0 23 0 0 0 | 0 23 0 0 0 | 0 23 0 0 0 | 0 23 0 0 0 | 0 23 0 0 0 | 0 23 0 0 0 |
| \{7,13,37,97,149,179\} | 0 6 0 0 0 | 0 6 0 24 0 0 | 0 24 0 60 0 0 | 0 6 0 24 0 0 | 0 24 0 60 0 0 | 0 6 0 24 0 0 |

\[ M = \begin{pmatrix} 0 & 0 & 16 & 0 \\ 0 & 0 & 4 & 14 \\ 16 & 4 & 0 & 0 \\ 0 & 0 & 14 & 0 \\ 0 & 14 & 0 & 0 \end{pmatrix} \]
\[ M' = \begin{pmatrix} 39 & 0 & 16 & 0 \\ 0 & 19 & 4 & 14 \\ 16 & 4 & 23 & 0 \\ 0 & 14 & 0 & 5 \\ 0 & 14 & 0 & 5 \end{pmatrix} \]
Table 7. The computational results of the UCG public key cryptosystem.

| b | $C_1 = D^x \pmod{p}$ | $C_2 = A^y \times M \pmod{p}$ |
|---|---------------------|---------------------|
| 6 | \[
\begin{bmatrix}
23 & 35 & 21 & 7 \\
8 & 23 & 6 & 19 \\
2 & 30 & 32 & 30 \\
32 & 29 & 30 & 34
\end{bmatrix}
\] | \[
\begin{bmatrix}
4 & 5 & 36 & 32 \\
16 & 10 & 36 & 35 \\
28 & 18 & 36 & 38 \\
0 & 19 & 0 & 12
\end{bmatrix}
\] |
| 4 | \[
\begin{bmatrix}
54 & 22 & 32 & 26 & 16 \\
1 & 33 & 18 & 0 & 48
\end{bmatrix}
\] | \[
\begin{bmatrix}
56 & 40 & 24 & 57 & 33 \\
8 & 33 & 10 & 47 & 50
\end{bmatrix}
\] |
| 7 | \[
\begin{bmatrix}
31 & 17 & 31 & 9 & 10 \\
30 & 59 & 42 & 4 & 27
\end{bmatrix}
\] | \[
\begin{bmatrix}
19 & 51 & 1 & 34 & 37 \\
44 & 25 & 9 & 1 & 10
\end{bmatrix}
\] |
| 5 | \[
\begin{bmatrix}
51 & 68 & 85 & 35 & 66 \\
58 & 10 & 4 & 60 & 72
\end{bmatrix}
\] | \[
\begin{bmatrix}
62 & 76 & 39 & 91 & 84 \\
38 & 31 & 43 & 73 & 24
\end{bmatrix}
\] |
| 3 | \[
\begin{bmatrix}
109 & 54 & 32 & 134 & 22 & 185 \\
165 & 7 & 57 & 106 & 143 & 83
\end{bmatrix}
\] | \[
\begin{bmatrix}
26 & 108 & 115 & 45 & 132 & 30 \\
159 & 132 & 49 & 125 & 36 & 129
\end{bmatrix}
\] |
Table 8. The computational results of the UCG public key cryptosystem: decryption process.

| \((C_i^*)^{-1} \pmod{p}\) | \((C_i^*)^{-1} \times C_2 = M^{-1}\) | A message \(m\) |
|---------------------------|--------------------------------|------------------|
| \(\begin{bmatrix} 29 & 29 & 29 & 23 \\ 8 & 4 & 20 & 21 \\ 40 & 20 & 1 & 30 \\ 10 & 21 & 10 & 38 \end{bmatrix}\) | \(\begin{bmatrix} 39 & 0 & 16 & 0 \\ 0 & 19 & 4 & 14 \\ 16 & 4 & 23 & 0 \\ 0 & 14 & 0 & 5 \end{bmatrix}\) | \{39,19,23,5\} |
| \(\begin{bmatrix} 3 & 9 & 41 & 33 & 3 \\ 51 & 10 & 5 & 26 & 43 \\ 57 & 4 & 36 & 43 & 37 \\ 53 & 23 & 44 & 19 & 9 \\ 51 & 35 & 56 & 24 & 33 \end{bmatrix}\) | \(\begin{bmatrix} 7 & 8 & 0 & 0 & 0 \\ 8 & 15 & 4 & 0 & 0 \\ 0 & 4 & 19 & 15 & 0 \\ 0 & 0 & 15 & 34 & 22 \\ 0 & 0 & 0 & 22 & 56 \end{bmatrix}\) | \{7,15,19,34,56\} |
| \(\begin{bmatrix} 0 & 14 & 54 & 7 & 37 \\ 9 & 25 & 51 & 49 & 37 \\ 4 & 44 & 52 & 15 & 47 \\ 34 & 46 & 59 & 43 & 8 \\ 11 & 7 & 16 & 46 & 25 \end{bmatrix}\) | \(\begin{bmatrix} 23 & 0 & 14 & 0 & 20 \\ 0 & 5 & 4 & 0 & 0 \\ 14 & 4 & 9 & 0 & 0 \\ 0 & 0 & 60 & 17 & 0 \\ 20 & 0 & 0 & 17 & 43 \end{bmatrix}\) | \{23,5,9,60,43\} |
| \(\begin{bmatrix} 43 & 62 & 81 & 52 & 54 \\ 59 & 21 & 12 & 67 & 68 \\ 15 & 10 & 31 & 45 & 7 \\ 62 & 16 & 50 & 57 & 47 \\ 36 & 49 & 76 & 28 & 42 \end{bmatrix}\) | \(\begin{bmatrix} 4 & 23 & 0 & 0 & 0 \\ 23 & 27 & 6 & 0 & 0 \\ 0 & 6 & 33 & 42 & 0 \\ 0 & 0 & 42 & 75 & 16 \\ 0 & 0 & 0 & 16 & 91 \end{bmatrix}\) | \{4,27,33,75,91\} |
| \(\begin{bmatrix} 189 & 119 & 76 & 11 & 60 & 68 \\ 167 & 123 & 155 & 15 & 124 & 142 \\ 166 & 92 & 143 & 49 & 38 & 85 \\ 101 & 5 & 118 & 124 & 6 & 106 \\ 110 & 35 & 67 & 57 & 44 & 27 \\ 89 & 14 & 137 & 29 & 3 & 60 \end{bmatrix}\) | \(\begin{bmatrix} 7 & 6 & 0 & 0 & 0 & 0 \\ 6 & 13 & 24 & 0 & 0 & 0 \\ 0 & 24 & 37 & 60 & 0 & 0 \\ 0 & 0 & 60 & 97 & 52 & 0 \\ 0 & 0 & 0 & 52 & 149 & 30 \\ 0 & 0 & 0 & 0 & 30 & 179 \end{bmatrix}\) | \{7,13,37,97,149,179\} |