https://doi.org/10.48048/tis.2022.2147

Public Key Cryptosystem Based on Singular Matrix

Maxrizal

Institut Sains Dan Bisnis Atma Luhur, Pangkalpinang, Indonesia

(Corresponding author's e-mail: maxrizal@atmaluhur.ac.id)

Received: 14 September 2020, Revised: 27 May 2021, Accepted: 14 June 2021

Abstract

The algorithms such as RSA, ElGamal and ECC work on integers. Commutative operations on integer multiplication leave these algorithms vulnerable to attack by eavesdroppers. For this reason, experts develop the concept of non-commutative algebra in the public key cryptosystem by adding non-commutative properties to groups, semirings, semiring division, matrices and matrix decomposition. However, the key generating process in some public key cryptosystems is quite complicated to carry out. Therefore, in previous research, Liu used nonsingular matrices to form a simpler public key cryptosystem. However, eavesdroppers use the inverse of nonsingular matrices to construct the private key. As a result, this public key cryptosystem is still vulnerable to attacks. Therefore, we use a singular matrix to modify and build the proposed public key cryptosystem in this study. This study indicates that the singular matrix can be used to modify the public key cryptosystem. The results also show that the key generating algorithm only uses ordinary matrix multiplication (without using matrix power operations), so it is not too complicated. Furthermore, the proposed public key cryptosystem works on a matrix over integers so that the possible brute force attack trials are endless. The proposed public key cryptosystem also cannot be attacked by matrix inversion because it uses a singular matrix.

Keywords: Singular matrix key, Public key cryptosystem, Non-commutative singular, Non-commutative algebra, Singular private key

Introduction

Nowadays, algorithms such as RSA, ElGamal, and ECC are still widely used in securing message delivery. We know that these algorithms are asymmetric cryptographic algorithms developed to increase the security of sending messages [1-3]. These algorithms work on integers whose operations are commutative. An eavesdropper can hack some of these algorithms by attacking commutative properties, finding matching integer pairs and factoring large integers. For example, in RSA and its variant with moduli of the form $N = p^r q^l$ (where $r > l \geq 2$), the researchers found several cryptoanalytic attacks, namely small secret exponent attacks and lattice-based attacks and successfully factored N [4]. These attacks occur because RSA and its variants work on integers whose operations are commutative. This condition causes some algorithms that work on integers to be unsafe enough to send messages.

For this reason, experts began to introduce and develop cryptography using the concept of non-commutative algebra. This concept is based on the non-commutative concept in groups, semirings, division Semirings [5-8]. However, the key generating process is quite complicated to work out. In addition, the experts also developed the concept of public key cryptosystems on matrix [9-12] and matrix decomposition [13,14]. In [9], the researchers developed a simple number theory concept to avoid creating complicated keys. Note that matrix multiplication operations are generally non-commutative.

In previous research, Liu *et al.* [13] improved and formed a public key cryptosystem using Polynomial Symmetrical Decomposition on a non-commutative group $GL_n\left(F_q\right)$. Notation of $GL_n\left(F_q\right)$ is an invertible matrix group with size $n \times n$ over a field F_q. In this study, Liu uses nonsingular matrices, namely matrices that have an inverse. However, this system can be attacked via direct attack, linearization equations attack, and overdefined systems of multivariate polynomial equations attack [13]. In essence, the attack occurs because eavesdroppers can find a matrix or multivariate polynomial that corresponds to the public key matrix. Next, the eavesdropper uses the inverse of the matrix to construct the private key.

Thus, the public key cryptosystem of non-commutative algebra using a matrix and its decomposition is still vulnerable to eavesdropping attacks.

Furthermore, we also examine a symmetric cryptographic system that works on nonsingular matrices, namely Hill Cipher. This system has also undergone many developments and modifications [15-17]. In the original Hill Cipher, the message sender encrypts $C = KP \bmod p$, with C a ciphertext matrix, P any plaintext matrix, and K a nonsingular matrix (private key). To reverse the message received, the recipient of the message carries a description $P = K^{-1}C \bmod p$. Note that in the Hill Cipher cryptosystem, the message recipient can retrieve the plaintext matrix P because of the inverse concept of the matrix K. If the matrix K is singular, then the matrix K^{-1} is absent, and the plaintext matrix P will not be found. This is why the singular matrix is not widely applied because it is considered unfavourable in a cryptosystem.

Based on the cryptosystem facts of Hill Cipher, we see one advantage to the equation $C = KP$. If we choose K singular, then $P = K^{-1}C$ it does not apply. Thus, the matrix P cannot be rebuilt and cannot be found. Meanwhile, based on the weakness of the public key cryptosystem developed by Liu, eavesdroppers use the inverse of a nonsingular matrix to build private keys so that the public key cryptosystem is still vulnerable to attacks. For this reason, in this study, we improve and modify the public key cryptosystem using a singular matrix [18]. The goal is to prevent attacks that take advantage of the inverse of the nonsingular matrix. For encryption and description, we use the ordinary matrix addition and the subtraction operation.

Materials and methods

This research forms a public key cryptosystem by adopting a singular matrix and applying it to research by Liu. In Hill Cipher, if the key is a singular matrix, the plaintext cannot be read back by legitimate recipients or eavesdroppers. Since matrix K^{-1} does not exist, we cannot describe $P = K^{-1}C$. Thus, we can conclude that Hill Cipher using singular matrix (non-invertible matrix) keys is very secure. But such a system is not good because even legitimate message recipients cannot read the plaintext.

Furthermore, the many possible keys of this singular matrix were applied to improve the public key cryptosystem in Liu's research. According to Liu *et al.* [13], the nonsingular matrix concept used by Liu leaves the cryptosystem vulnerable to attacks. Eavesdroppers use the inverse property of a nonsingular matrix (invertible matrix) to generate private keys. Therefore, in this study, we formed a public key cryptosystem by adopting a singular matrix to modify Liu's research. The goal is to prevent attacks that take advantage of the inverse of the nonsingular matrix.

On the proposed research, the sender and recipient of the message build a key generate protocol before doing encryption and description. In the key generate protocol, the recipient and sender of the message are proposed using the equation $Y = AX$, where X and Y are public matrix (non-confidential), X is a singular matrix, and A is a private matrix (secret). We assume the sender chooses any singular matrix X and private matrix A. The sender of the message forms $Y = AX$. Although X and Y are public, the matrix A cannot be rebuilt by unauthorized parties (eavesdroppers). If the matrix X^{-1} is absent, then the equation $A = YX^{-1}$ does not apply. The eavesdroppers are forced to carry out a brute force attack to find the matrix A. However, it is not easy to do if the matrix size is large, and the matrix entry is an integer (*Z*), which is infinite.

This research is a literature study. The primary literature in this study is Liu *et al.* with the title "Cryptanalysis of Schemes Based on Polynomial Symmetrical Decomposition" and Reddy *et al.* [17] with the title "A Modified Hill Cipher Based on Circulant Matrices". We studied the original Hill Cipher in the 1st stage and replaced the nonsingular matrix key with a singular matrix. In the 2nd stage, we studied the public key cryptosystem that Liu *et al.* worked on using a nonsingular matrix $GL_n\left(F_q\right)$. In the 3rd stage, we replaced the singular matrix to modify the public key cryptosystem from the Liu research. We used Mathematica 5.0 software to help calculate matrix operations by try and error at the final step.

Results and discussion

Public key cryptosystem scheme by Liu

In previous research, Liu used $GL_n\left(F_q\right)$, the group of invertible $n \times n$ matrices over the field F_q, and she used $M_n\left(F_q\right)$, the group of $n \times n$ matrices over the field F_q. This study also uses a polynomial

$f(x) \in F_q[x]$. In the initial stage, this research forms a non-commutative group $(M_n(F_q),.)$ and selects any $a,b \in Z$. Next, Liu selects 2 elements $P \in GL_n(F_q)$, $Q \in M_n(F_q)$, with $PQ \neq QP$ (non-commutative). Output (P,Q) as a public key pair.

1) Alice chooses a polynomial $f(x) \in F_q[x]$ randomly such that $f(P) \in GL_n(F_q)$. She calculated $y = f^a(P)Qf^b(P)$, and she sends y to Bob.

2) Bob chooses a polynomial $h(x) \in F_q[x]$ randomly such that $h(P) \in GL_n(F_q)$. He calculated $u = h^a(P)Qh^b(P)$, and she sends y to Alice.

3) After the exchange y and u, Alice and Bob have the key $K = f^a(P)uf^b(P) = h^a(P)yh^b(P)$.

Suppose Alice has plaintext P. She encrypts $C = K + P$ and sends it to Bob. Next, Bob described with $P = C - K$.

In general, the weakness of this public key cryptosystem lies in choosing $P \in GL_n(F_q)$. To obtain private key K, an eavesdropper can attack via direct attack, linearization equations attack, and overdefined systems of multivariate polynomial equations attack [13]. The eavesdropper makes use of the inverse of the matrix corresponding to $P \in GL_n(F_q)$ to construct of the private key.

Advantages of private key with singular matrix

In a previous study [18], Liu established a public key cryptosystem using a nonsingular matrix $P \in GL_n(F_q)$. The number of possible matrices P is equivalent to the order of $GL_n(F_q)$, namely $\prod_{k=0}^{n-1}(q^n - q^k) = (q^n - 1)(q^{n-1} - q)\ldots(q^n - q^{n-1})$ [8]. Therefore, the number of possible brutal force attacks on this public key cryptosystem is $\prod_{k=0}^{n-1}(q^n - q^k)$.

Furthermore, in the proposed public key cryptosystem, we are given a singular matrix X and a nonsingular matrix A. We form $Y = AX$. We define Y and X to be public and A is private. If an eavesdropper wants to steal A, he must finish $A = YX^{-1}$. But this is not possible. The matrix X^{-1} does not exist for a singular matrix, so the calculations cannot be completed. Note that the singular matrix X works on integers (*Z*). Therefore, the brutal force attack on the proposed public key cryptosystem will be infinite (the properties of an infinite integer).

Eavesdroppers only try to string together a combination of entries from different matrices until they find the correct combination in a brute-force attack. Suppose that in the research by Liu, we were given a matrix. $P = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \in GL_2(Z_{19})$. Eavesdroppers can guess the matrix P as many as $\prod_{k=0}^{3-1}(19^3 - 19^k) = 304812862560$ is possible. We use a singular matrix X over Z in the proposed public key cryptosystem. Thus, an eavesdropper must try an infinite number of possibilities due to the infinite integer (*Z*). If the public key cryptosystem by Liu works on matrices over Z_p, then the purposed public key cryptosystem works on matrices over *Z*.

Characteristics of singular matrix formation

The singular matrix used in the proposed public key cryptosystem is a matrix of size $n \times n$ with 0 determinant. A matrix X is a singular matrix if:

1) There is a 0 column or row.

2) There are 2 comparable rows or 2 columns.

3) There is a row or column that is a linear combination of other rows or columns [18].

To get more complicated singular matrices, then we need to modify the singular matrix with elementary row or column operations. Note that the matrix $X=\begin{bmatrix}1&2&3\\4&4&1\\2&4&6\end{bmatrix}$ is singular because the 3rd row is twice the 1st row. Next, we can form a more complicated singular matrix by performing elementary rows operations to make it a more randomized plaintext. We work on the following matrix X.

$$X=\begin{array}{c}b_1\\b_2\\b_3\end{array}\begin{bmatrix}1&2&3\\4&4&1\\2&4&6\end{bmatrix}=\begin{array}{c}b_1^*=4b_2-b_1\\b_2^*=b_2\\b_3^*=b_3+b_2\end{array}\begin{bmatrix}15&14&1\\4&4&1\\6&8&7\end{bmatrix}$$

Note that the matrix X does not look like a singular matrix (unless we calculate the determinant). The determinant of the matrix X is 0. This 3rd type is what we use to get better matrix randomization.

Generating a singular matrix really helps us form a large singular matrix (without having to check the determinant). As an example, we will create a singular matrix X with a size of 20×20. The easiest step we can do is to select any entry for rows $b_1,b_2,\ldots,b_{19}$. Next, we form row b_{20} by 4 times row b_1, namely $b_{20}=4b_1$. Thus the matrix $X=\begin{bmatrix}b_1\\b_2\\\vdots\\b_{20}=4b_1\end{bmatrix}$ is singular.

Key generating algorithms in the proposed public key cryptosystem

We take a singular matrix X as a public key. Alice and Bob will form a key together and agree to use a singular matrix X (which may be sent on an insecure path).

1) Alice chooses any nonsingular matrix A and keeps it secret. She calculated $Y=AX$, and she sends the matrix Y to Bob.

2) Bob chooses any nonsingular matrix B and keeps it secret. He calculated $U=XB$, and he sends the matrix U to Alice.

3) After the exchange Y and U, Alice calculates the key $K_a=AU$, and Bob calculates the key $K_b=YB$. Note that $K_a=AU=A(PB)=(AP)B=YB=K_b$.

Thus, the private key Alice and Bob have the same as $K=K_a=K_b$ without having to exchange keys. The following is a simple diagram of the key generating algorithms.

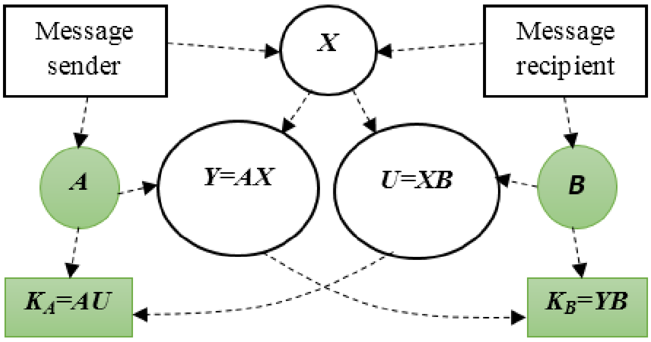

Figure 1 Key generating flow.

Note that A and B are confidential. If A or B is found, the primary key K_a or K_b is so easy to see that the system becomes insecure. The main strength of the proposed cryptosystem lies in the use of the singular matrix X.

Message encryption and description

We assume Alice has a message P (plaintext). Alice will calculate the ciphertext $C = P + K_A$ on encryption. To reverse the message P, Bob calculates $P = C - K_b$ in the description. Notice again that applies $K_a = K_b = K$. In this system, we propose simple encryption and description that we only use the addition and inverse addition (subtraction) on matrix operations. Furthermore, the encryption and description system can be agreed upon by the sender and receiver.

Comparison of the public key cryptosystem by Liu and the proposed public key cryptosystem

According to [13], Liu *et al.* used parameters $GL_n(F_q), M_n(F_q)$ and polynomial $f(x) \in F_q[x]$ to build a public key cryptosystem. For example, if the field is *F* = *Z*, then Liu research works on a matrix of *Z*$_p$. Whereas in this study, we use a singular matrix X (non-invertible matrix) over *Z*. The number of brute force attacks in Liu's research is $\prod_{k=0}^{n-1}(q^n - q^k)$, and in this study, there are countless. The key generation equation in the public key cryptosystem by Liu is quite complicated because it involves matrix multiplication and the power of the matrix. In addition, the public key cryptosystem by Liu uses a pair of nonsingular matrices (P, Q) as the public key. Whereas in this study, we only use the singular matrix X as the private key.

Table 1 Comparison of the public key cryptosystem by Liu and the proposed public key cryptosystem.

	The Public Key Cryptosystem by Liu	**The Proposed Public Key Cryptosystem**
Parameters used	Matrices $GL_n(F_q), M_n(F_q)$ and polynomials $f(x) \in F_q[x]$	Matrices *M*$_n$(*Z*) and singular matrix X
Possible Brute force attack	Finite is $\prod_{k=0}^{n-1}(p^n - p^k)$	Infinity
Public key	A pair of matrices $P \in GL_n(F_q)$ $Q \in M_n(F_q)$	A singular matrix $X \in M_n(Z)$
Key Generating Algorithms	$K = f^a(P)uf^b(P) = h^a(P)yh^b(P)$	$K = K_a = AU = YB = K_b$
Key generating algorithm properties	Quite complicated because it involves the matrix multiplication and power matrix equations	It is easier; only use matrix multiplication
Attack on matrix inverse	It can be attacked with a matrix inverse	It cannot be attacked with a matrix inverse

Examples of proposed algorithm cases

Alice and Bob will send a message. They agreed to use the singular matrix $X_{3\times3} = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 3 & 2 \\ 4 & 10 & 14 \end{bmatrix}$ to build the key. Notice that the matrix X is formed by selecting any rows $b_1 = \begin{bmatrix} 2 & 5 & 7 \end{bmatrix}$ and $b_2 = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix}$. Furthermore, the row b_3 is twice the row b_1, namely $b_3 = 2b_1 = \begin{bmatrix} 4 & 10 & 14 \end{bmatrix}$.

1) Alice chooses any nonsingular matrix $A=\begin{bmatrix}1&5&3\\2&2&8\\6&4&9\end{bmatrix}$ and keeps it secret. She calculated $Y=AX=\begin{bmatrix}19&50&59\\38&96&130\\52&132&176\end{bmatrix}$, and she sends the matrix Y to Bob.

2) Bob chooses any nonsingular matrix $B=\begin{bmatrix}9&3&13\\12&21&8\\2&14&9\end{bmatrix}$ and keeps it secret. He calculated $U=XB=\begin{bmatrix}92&209&129\\49&94&55\\184&418&258\end{bmatrix}$, and he sends the matrix U to Alice.

3) After the exchange Y and U, Alice calculates the key $K_a=AU=\begin{bmatrix}889&1933&1178\\1754&3950&2432\\2404&5392&3316\end{bmatrix}$, and Bob calculates the key $K_b=YB=\begin{bmatrix}889&1933&1178\\1754&3950&2432\\2404&5392&3316\end{bmatrix}$. It applies $K_a=K_b=K$, without primary key exchange.

We suppose Alice will send a message $P=\begin{bmatrix}22&12&11\\12&34&54\\23&11&71\end{bmatrix}$. She calculates the ciphertext $C=P+K_A=\begin{bmatrix}911&1945&1189\\1766&3984&2486\\2427&5403&3387\end{bmatrix}$ and sends it to Bob. Next, Bob receives the ciphertext C from Alice and calculates the plaintext $P=C-K_B=\begin{bmatrix}22&12&11\\12&34&54\\23&11&71\end{bmatrix}$.

Proposed cryptosystem type

Maximum planitext type

This type forms the main key matrix of the size of the plaintext matrix P. Suppose there is plaintext $P_{n\times n}$, the sizes of Y,A and X are $n\times n$. If there are q entries then the size of the matrix P is $n=\left\lceil\sqrt{q}\right\rceil=floor\left(\sqrt{q}\right)$. Suppose there are 35 entries then size $n=\left\lceil\sqrt{35}\right\rceil=6$. Thus, the size of the matrix P is 6×6. So, there is the last entry that can be used as a dummy entry.

Type plaintext blocks

This type forms plaintext into matrix blocks that are smaller than plaintext size. Suppose there is P, then it is blocked $\left(P_1\right)_{s\times s},\left(P_2\right)_{s\times s},\ldots,\left(P_k\right)_{s\times s}$. Thus, there are Y,A and X size $s\times s$. If there is a missing entry, then fill it with a dummy entry. This concept adopts plaintext blocks in ElGamal, Hill Cipher, and RSA [1-3,15,17].

Conclusions

This study indicates that the singular matrix can be used to modify the public key cryptosystems. In previous studies, the public key cryptosystem involved a pair of nonsingular matrices as the public key. It also has 3 parameters: General linear group, polynomial, and nonsingular matrix over the field. In the proposed study, we only use a singular matrix as the public key and 2 parameters, namely the set matrix of integers and the singular matrix. The results also show that the key generating algorithm only uses ordinary matrix multiplication (without using matrix power operations), so it is not too complicated. Furthermore, the proposed public key cryptosystem works on a matrix over integers so that the possible brute force attack trials are endless. The proposed public key cryptosystem also cannot be attacked by matrix inversion because it uses a singular matrix.

Acknowledgements

This research was funded by the Institut Sains Dan Bisnis Atma Luhur, Indonesia, through an internal grant for the 2020 fiscal year. For this reason, the authors would like to thank the Institut Sains Dan Bisnis Atma Luhur for holding training activities and research writing workshops to obtain this grant.

References

[1] FY Rao. On the security of a variant of ElGamal encryption scheme. *IEEE Trans. Dependable Secur. Comput.* 2015; **14**, 1-4.

[2] CR Bharathi. Improved ElGamal encryption for elliptic curve cryptography. *Int. J. Pure Appl. Math.* 2018; **118**, 341-53.

[3] M Joye. *Secure ElGamal-type cryptosystems without message encoding*. *In*: P Ryan, D Naccache and JJ Quisquater (Eds.). The new codebreakers. Lecture notes in computure science. Springer, Berlin, Heidenberg, Germany, 2016, p. 470-8.

[4] Y Lu, L Peng and S Sarkar. Cryptanalysis of an RSA variant with moduli $N = p^r q^l$. *J. Math. Cryptol.* 2017; **11**, 117-30.

[5] MR Valluri. Zero-knowledge authentication schemes using quasi-polynomials over non-commutative groups. *Open J. Inf. Secur. Appl.* 2014; **1**, 43-9.

[6] B Tsaban. Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography. *J. Cryptol.* 2015; **28**, 601-22.

[7] A Mahalanobis. A simple generalization of the ElGamal cryptosystem to non-abelian groups II. *Commun. Algebra* 2012; **40**, 3583-96.

[8] GSGN Anjaneyulu and A Sanyasirao. Distributed group key management protocol over non-commutative division semirings. *Indian J. Sci. Technol.* 2014; **7**, 871-6.

[9] ZY Karatas, E Luy and B Gonen. A public key cryptosystem based on matrices. *Int. J. Comput. Appl.* 2019; **182**, 47-50.

[10] M Zeriouh, A Chillali and A Boua. Cryptography based on the matrices. *Bol. Soc. Paran. Mat.* 2019; **37**, 75-83.

[11] M Andrecut. A matrix public key cryptosystem, Available at: https://arxiv.org/abs/1506.00277, accessed May 2020.

[12] AVN Krishna, AH Narayana and KM Vani. A novel approach with matrix based public key cryptosystems. *J. Discret. Math. Sci. Cryptogr.* 2017; **20**, 407-12.

[13] J Liu, H Zhang and J Jia. Cryptanalysis of schemes based on polynomial symmetrical decomposition. *Chinese J. Electron.* 2017; **26**, 1139-46.

[14] J Liu, H Zhang, J Jia, H Wang, S Mao and W Wu. Cryptanalysis of an asymmetric cipher protocol using a matrix decomposition problem. *Sci. China Inf. Sci.* 2016; **59**, 052109.

[15] Maxrizal. Hill cipher cryptosystem over complex numbers. *Indones. J. Math. Educ.* 2019; **2**, 9-13.

[16] Maxrizal and BDA Prayanti. Application of rectangular matrices: Affine cipher using asymmetric keys. *Cauchy* 2019; **5**, 181-5.

[17] KA Reddy, B Vishnuvardhan, Madhuviswanatham and AVN Krishna. A modified hill cipher based on circulant matrices. *Procedia Technol.* 2012; **4**, 114-8.

[18] H Anton and C Rorres. *Elementary linear algebra: Applications version.* 11th eds. John Wiley & Sons, Hoboken, NJ, 2013.