International Journal of Cryptology Research 12(1): 1 - 12 (2022)

Security Analysis of a Public Key Cryptosystem for Synchronized Chaotic Systems

Zahari Mahad[1], **Muhammad Asyraf Asbullah***[1], **Muhammad Rezal Kamel Ariffin**[1,2], and **Arif Mandangan**[3]

[1]*Laboratory of Cryptography, Analysis and Structure, Institute for Mathematical Sciences, Universiti Putra Malaysia*
[2]*Department of Mathematics & Statistics, Faculty of Science, Universiti Putra Malaysia*
[3]*Mathematics, Real-Time Graphics and Visualization Laboratory, Faculty of Sciences and Natural Resources, Universiti Malaysia Sabah*

Email: ma_asyraf@upm.edu.my
** Corresponding author*

ABSTRACT

This article introduces a novel cryptanalysis technique for the Diffie-Hellman Key Exchange (DHKE) protocol, created in 2014 by Balasubramanian and Muthukumar. The suggested approach is based on synchronised chaotic systems with linear control and Fibonacci Q matrix as a support matrix. According to their findings, the suggested cryptosystem is more secure than the standard ElGamal public key cryptosystem due to the difficulty of identifying the private keys r and s. Simultaneously, the initial value of synchronised chaotic systems is unknown as well. While in this article, we demonstrate that the suggested method may be broken by demonstrating that the private keys r and s can be effectively solved in practice even when the initial value of the synchronised chaotic systems is unknown.

Keywords: Chaotic Maps, Chaos Synchronization, Fibonacci Q matrix, Linear Control, Public-key Cryptosystem

1 INTRODUCTION

In 1976, Diffie and Hellman coined the term "public-key cryptography." They demonstrated how to construct a public-key cryptosystem using a one-way trapdoor function (PKC). Later the same year, in 1985, Taher ElGamal created the ElGamal public-key cryptosystem. ElGamal is a "minor" addition to Diffie-Hellman Key Exchange (DHKE) that enables the direct transmission of encrypted communications (Zazali & Mior, 2012). However, the encrypted message is frequently utilised as an additional key in symmetric encryption, typically employed for vast data. Due to its underlying component based on the Discrete Logarithmic Problem, ElGamal has been one of the most popular cryptosystems (Mandangan et al., 2020). ElGamal's cryptosystem comprises three distinct processes: key generation, encryption, and decryption, which are discussed in depth below.

Algorithm 1. The ElGamal Cryptosystem (*ElGamal,1985*).

Key Generation:

1. Choose a private key $1 \leq a \leq p-1$.
2. Compute $A = g^a \pmod p$.
3. Publish the public key A.

Encryption:

1. Choose plaintext m.
2. Choose random ephemeral key k.
3. Use Along's public key A compute:
   i. $c_1 = g^k \pmod p$.
   ii. $c_2 = mA \pmod p$.
4. Send ciphertext (c_1, c_2) to Along.

Decryption:

1. Compute $(c_1^a)^{-1} c_2 \pmod p = m$.

Numerous academics have studied various versions of ElGamal's original encryption technique when it comes to ElGamal-type public key cryptosystem. For instance, Ismail & Hasan (2006) present a new digital signature that eliminates the use of one-time secret key. This property will make all attacks, aiming at revealing the one-time secret key irrelevant. Zazali

& Mior (2012) introduced key exchange procedure using elliptic curve cryptography that analogous to ElGamal's encryption. Asbullah & Ariffin (2012) proposes a variant of the ElGamal which implement the intractability of the Gap Hashed Diffie-Hellman assumption and present a practical way to encrypt short messages. Sarbini et al. (2019) identified many features of Lucas sequence variations that are connected from a cryptosystem called the LUCELG by combining Lucas sequences onto the Diffie-Hellman and ElGamal cryptosystems to enhance efficiency.

On the other hand, Sarbini et al. (2019) illustrate that the security response of the Lucas-Based ElGamal Cryptosystem in the Elliptic Curve group over a finite field against the Greatest Common Divisor (GCD) attack. Using great common divisor and Dickson polynomials, the Lucas-based El-Gamal Cryptosystem in the Elliptic Curve group over the finite field was mathematically exposed to the GCD attack. Thus, the study showed that when two plaintexts deviate slightly in the group Elliptic curve over the finite field from a fixed number, this is extremely risky since the cryptanalyst may obtain the plaintext rapidly without decryption.

Chaos synchronisation has generated enormous attention globally in communication systems since it may be used to encrypt or decode data for secure conversations (Al-Saidi et al., 2020). Balasubramaniam and Muthukumar (2014) discussed synchronising chaotic systems using linear and nonlinear feedback control techniques. The primary goal is to adapt the synchronised chaotic systems' lowest synchronisation error in protected communication. Additionally, the authors studied synchronisation utilising linear and nonlinear control approaches and traditional security analysis. While chaotic systems are deterministic, in the sense that their initial condition determines their whole future course, it is difficult to anticipate their long-term behaviour (Natiq et al., 2021).

According to Balasubramaniam and Muthukumar (2014), the linear feedback control approach successfully synchronises chaotic systems depending on the cost and inaccuracy of synchronisation. As a result, the ElGamal cryptosystem uses the linear feedback control approach for synchronising chaotic systems to increase its security level. The Diffie-Hellman key exchange protocol is introduced via Diffie-Hellman key exchange using Fibonacci Q matrices. It is based on synchronised chaotic

systems and the ElGamal public-key cryptosystem. Their work modifies the Diffie-Hellman key exchange protocol in general and asserts that their suggested work is superior to the standard Diffie-Hellman key exchange protocol. The cryptosystem proposed in Balasubramaniam and Muthukumar (2014) shall be referred to as the BM-ElGamal cryptosystem from here on.

We demonstrate step-by-step how to break the BM-ElGamal using just the public key in this article. We will create a lookup table for the public keys that correlate to the private keys. Due to the parameters used in their works, the building of the lookup table is easy and practicable. Finally, the BM-ElGamal cryptosystem is utterly breakable.

The following is an overview of how the remainder of the paper. The section Materials and Methods outlines the BM-ElGamal cryptosystem. The Results and Discussion section demonstrates how to solve the discrete logarithm issue using the lookup table, thus breaking the considered cryptosystem. In the final part, we give a brief conclusion.

2 BACKGROUNDS

This section addresses Balasubramaniam and Muthukumar (2014) proposed work on the synchronisation of integer-order chaotic systems applied to the ElGamal cryptosystem and the Fibonacci Q matrix. Rather than going into depth about the concept and execution of chaotic synchronisation, we will concentrate on their cryptography efforts.

Definition 2.1. Let the Fibonacci sequences as 0, 1, 1, 2, 3, 5, 8, 13, 21 and so. Then we define $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$ and so on.

Definition 2.2 Let the Fibonacci Q matrix defined by (Silvester, 1979) as $Q = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Then for any integer $n \geq 1$, the n-th power of Q matrix has the form $Q^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$.

Example 2.1. Let $n = 2$. By referring to Definition 1 and Definition 2, we can compute the Fibonacci Q matrix, Q^2 as

$$Q^2 = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} F_3 & F_2 \\ F_2 & F_1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}.$$

3 THE DIFFIE–HELLMAN KEY EXCHANGE PROBLEM BASED ON SYNCHRONIZED CHAOTIC SYSTEMS

As with the Diffie–Hellman key agreement, two parties, Alice, and Bob, seek to collaborate to generate a shared key. This shared key may then be used to encrypt and decrypt messages between the two parties using any symmetric cryptosystem. Let $Q = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ be an initial Fibonacci Q matrix and $n = 38$ referring to the number of characters as in Table 1 below.

Table 1. Assignment of numbers with characters.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	⋯	37
Character	0	1	2	3	4	5	6	7	8	9	-	.	A	B	⋯	Z

By referring to Algorithm 2, the Diffie-Hellman Key Exchange based on Chaos Synchronization is demonstrated step-by-step until both Alice and Bob have a shared secret key.

Algorithm 2. Diffie-Hellman Key Exchange based on Chaos Synchronization

Alice		**Bob**
Alice and Bob agree on a public element Q and $n = 38$ referring to the number of characters as in Table 1 above.		
1. Chooses a secret number $t_{Alice} > t_0$. 2. Compute x at t_{Alice} using the concept of chaos synchronization.		1. Chooses a secret number $t_{Bob} > t_0$. 2. Compute x at t_{Bob} using the concept of chaos synchronization.

3. Compute a secret key $s \equiv [t_{Alice} \cdot \lvert x \rvert] \pmod{n}$ 4. Compute a public key as $A \equiv Q^s \pmod{n}$		3. Compute a secret key $r \equiv [t_{Bob} \cdot \lvert x \rvert] \pmod{n}$. 4. Compute a public key as $B \equiv Q^r \pmod{n}$
Key Exchange Procedure: Alice sends her public key A to Bob. Similarly, Bob sends his public key B to Alice.		
5. Compute a shared key as $K \equiv B^s \pmod{n}$.		5. Compute a shared key as $K \equiv A^r \pmod{n}$.
Now, Alice and Bob have a shared key for encryption and decryption procedures. $B^s \equiv K \equiv A^r \pmod{n}$. $(Q^r)^s \equiv K \equiv (Q^s)^r \pmod{n}$. $Q^{sr} \equiv K \equiv Q^{rs} \pmod{n}$.		

Algorithm 3 below shows a step-by-step for the Diffie-Hellman Key Exchange and ElGamal Cryptosystem based on Chaos Synchronization.

Algorithm 3. Diffie-Hellman Key Exchange and ElGamal Cryptosystem based on Chaos Synchronization

Public Parameters	
A trusted third party publish a public element Q and $n = 38$ referring to the number of characters as in Table 1 above.	
Key Creation as Algorithm 1	
1. Chooses a secret number $t_{Alice} > t_0$. 2. Compute x at t_{Alice} using the concept of chaos synchronization. 3. Compute a secret key $s \equiv [t_{Alice} \cdot \lvert x \rvert] \pmod{n}$. 4. Compute a public key as $A \equiv Q^s \pmod{n}$.	
Alice sends the public key $\boldsymbol{pk}$: $(\boldsymbol{Q}, \boldsymbol{n}, \boldsymbol{A})$ to Bob.	
Encryption	
	5. Chooses a secret number $t_{Bob} > t_0$. 6. Compute x at t_{Bob} using the

	concept of chaos synchronization. 7. Compute a secret key $r \equiv [t_{Bob} \cdot \lvert x \rvert] \pmod{n}$. 8. Compute the ciphertext $(c_1, c_2) = E_{pk}(m)$ of the plaintext $m \in m_{2\times 2}(\mathbb{R})$ where $c_1 = B \equiv Q^r \pmod{n}$ and $c_2 = E \equiv m \cdot A^r \pmod{n}$.
Bob sends the ciphertext (c_1, c_2) to Alice.	
Decryption	
9. Compute $K \equiv {c_1}^s \equiv (Q^r)^s \pmod{n}$ 10. Compute the plaintext $m' \equiv D_{sk}(c_2) \equiv \frac{c_2}{K} \equiv \frac{mA^r}{Q^{rs}} \equiv \frac{mQ^{rs}}{Q^{rs}} \equiv m \pmod{n}$ where $m' = m$.	

4 RESULT AND DISCUSSIONS

The BM-ElGamal cryptosystem is simply a Diffie–Hellman key agreement scheme with a modulus of $n = 38$ characters instead of a big prime integer. As opposed to the general Diffie–Hellman algorithm, the suggested Diffie–Hellman algorithm is designed to loosen the constraint of using integers to make it valid for any numbers. Meanwhile, the technique is deemed unsafe when effectively creating a lookup table for the public key that corresponds to the private key owing to the tiny modulus. If an eavesdropper observes A and B, he or she can solve the discrete logarithm problem $Q^r \pmod{38}$ to determine r or $Q^s \pmod{38}$ to determine s using the created lookup table and then compute the shared key $Q^{rs} \pmod{38}$.

This section demonstrates how to solve the discrete logarithm issue using the technique given in Balasubramaniam and Muthukumar (2014). Let the initial Fibonacci Q matrix, and modulus n are public. The attacker can then build Fibonacci sequences and a lookup table for each set of public keys

included in $\mathbb{Z}_n$. Remark that creating a lookup table for the Fibonacci Q matrix for a small modulus n is a simple polynomial-time operation (i.e., swift). We will now walk through the process of creating such a lookup table step by step.

Step 1: Given $n = 38$, the total number of characters used as in BM-ElGamal cryptosystem. An attacker can generate a list of first 39 Fibonacci numbers as in Table 2 below.

Table 2. Fibonacci sequences for $n = 38$.

Index	0	1	2	3	4	5	6	7	⋯	37	38
Fibonacci Number	0	1	1	2	3	5	8	13	⋯	24157817	39088169

From the partial information, he can simplify the underlying CVP instance to an easier instance with smaller error vector. By setting a vector $\vec{s} \in \{\sigma\}^n$, the congruence $\vec{e} + \vec{s} \equiv \vec{0} (\mathrm{mod}\ 2\sigma)$ holds. From the encryption formula, we have $\vec{c} = B\vec{m} + \vec{e}$. Then,

Step 2: Given $Q = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ and $n = 38$. Based on Table 2, an attacker can generate a list of Fibonacci Q Matrix as in Table 3. Observe that, for the same public key value, A and B would have more than one private key value representing s and r that produces a shared key used by Alice and Bob. **Green** and **Orange** colour: representing Alice's and Bob's public key and private key, respectively.

Table 3. The corresponding public keys A and B to the private keys s and r.

r or s	A or B	r or s	A or B	r or s	A or B
0	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	13	$\begin{pmatrix} 35 & 5 \\ 5 & 30 \end{pmatrix}$	26	$\begin{pmatrix} 34 & 21 \\ 21 & 13 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	14	$\begin{pmatrix} 2 & 35 \\ 35 & 5 \end{pmatrix}$	27	$\begin{pmatrix} 17 & 34 \\ 34 & 21 \end{pmatrix}$
2	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	15	$\begin{pmatrix} 37 & 2 \\ 2 & 35 \end{pmatrix}$	28	$\begin{pmatrix} 13 & 17 \\ 17 & 34 \end{pmatrix}$
3	$\begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix}$	16	$\begin{pmatrix} 1 & 37 \\ 37 & 2 \end{pmatrix}$	29	$\begin{pmatrix} 30 & 13 \\ 13 & 17 \end{pmatrix}$
4	$\begin{pmatrix} 5 & 3 \\ 3 & 2 \end{pmatrix}$	17	$\begin{pmatrix} 0 & 1 \\ 1 & 37 \end{pmatrix}$	30	$\begin{pmatrix} 5 & 30 \\ 30 & 13 \end{pmatrix}$

5	$\begin{pmatrix} 8 & 5 \\ 5 & 3 \end{pmatrix}$	18	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	31	$\begin{pmatrix} 35 & 5 \\ 5 & 30 \end{pmatrix}$
6	$\begin{pmatrix} 13 & 8 \\ 8 & 5 \end{pmatrix}$	19	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	32	$\begin{pmatrix} 2 & 35 \\ 35 & 5 \end{pmatrix}$
7	$\begin{pmatrix} 21 & 13 \\ 13 & 8 \end{pmatrix}$	20	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	33	$\begin{pmatrix} 37 & 2 \\ 2 & 35 \end{pmatrix}$
8	$\begin{pmatrix} 34 & 21 \\ 21 & 13 \end{pmatrix}$	21	$\begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix}$	34	$\begin{pmatrix} 1 & 37 \\ 37 & 2 \end{pmatrix}$
9	$\begin{pmatrix} 17 & 34 \\ 34 & 21 \end{pmatrix}$	22	$\begin{pmatrix} 5 & 3 \\ 3 & 2 \end{pmatrix}$	35	$\begin{pmatrix} 0 & 1 \\ 1 & 37 \end{pmatrix}$
10	$\begin{pmatrix} 13 & 17 \\ 17 & 34 \end{pmatrix}$	23	$\begin{pmatrix} 8 & 5 \\ 5 & 3 \end{pmatrix}$	36	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
11	$\begin{pmatrix} 30 & 13 \\ 13 & 17 \end{pmatrix}$	24	$\begin{pmatrix} 13 & 8 \\ 8 & 5 \end{pmatrix}$	37	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$
12	$\begin{pmatrix} 5 & 30 \\ 30 & 13 \end{pmatrix}$	25	$\begin{pmatrix} 21 & 13 \\ 13 & 8 \end{pmatrix}$		

Step 3: Given a public key Alice, A and a public key Bob, B. An attacker can determine the correspondent private keys r and s easily due to a small number as a modulus.

Step 4: Finally, an attacker can generate a shared key used by Alice and Bob.

By referring to Table 3, more than one of the values of the private keys r and s for the duplicate public keys A and B will produce a correct shared key used by Alice and Bob. This scenario happens when the order of $Q \pmod n$ is small due to utilizing a small and composite number as a modulus. It is feasible to generate a list of Fibonacci Q matrix as in Table 3 for case modulus $n = 256$, which is the total number of ASCII characters. Additionally, there are many valid values for the private keys r and s for the duplicate public keys A and B where the modulus $n = 38$ is not a prime number. A modulus of a prime integer can be used to solve this problem. For instance, $n = 37$ when the letter '-' at index 10 in Table 1 is omitted. See the following Table 4.

Table 4. The corresponding public keys A and B to the private keys s and r when modulus n is a prime number.

r or s	A or B	r or s	A or B	r or s	A or B
0	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	13	$\begin{pmatrix} 7 & 11 \\ 11 & 33 \end{pmatrix}$	26	$\begin{pmatrix} 22 & 33 \\ 33 & 26 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	14	$\begin{pmatrix} 18 & 7 \\ 7 & 11 \end{pmatrix}$	27	$\begin{pmatrix} 18 & 22 \\ 22 & 33 \end{pmatrix}$

2	$\begin{pmatrix}2 & 1\\1 & 1\end{pmatrix}$	15	$\begin{pmatrix}25 & 18\\18 & 7\end{pmatrix}$	28	$\begin{pmatrix}3 & 18\\18 & 22\end{pmatrix}$
3	$\begin{pmatrix}3 & 2\\2 & 1\end{pmatrix}$	16	$\begin{pmatrix}6 & 25\\25 & 18\end{pmatrix}$	29	$\begin{pmatrix}21 & 3\\3 & 18\end{pmatrix}$
4	$\begin{pmatrix}5 & 3\\3 & 2\end{pmatrix}$	17	$\begin{pmatrix}31 & 6\\6 & 25\end{pmatrix}$	30	$\begin{pmatrix}24 & 21\\21 & 3\end{pmatrix}$
5	$\begin{pmatrix}8 & 5\\5 & 3\end{pmatrix}$	18	$\begin{pmatrix}0 & 31\\31 & 6\end{pmatrix}$	31	$\begin{pmatrix}8 & 24\\24 & 21\end{pmatrix}$
6	$\begin{pmatrix}13 & 8\\8 & 5\end{pmatrix}$	19	$\begin{pmatrix}31 & 0\\0 & 31\end{pmatrix}$	32	$\begin{pmatrix}32 & 8\\8 & 24\end{pmatrix}$
7	$\begin{pmatrix}21 & 13\\13 & 8\end{pmatrix}$	20	$\begin{pmatrix}31 & 31\\31 & 0\end{pmatrix}$	33	$\begin{pmatrix}3 & 32\\32 & 8\end{pmatrix}$
8	$\begin{pmatrix}34 & 21\\21 & 13\end{pmatrix}$	21	$\begin{pmatrix}25 & 31\\31 & 31\end{pmatrix}$	34	$\begin{pmatrix}35 & 3\\3 & 32\end{pmatrix}$
9	$\begin{pmatrix}18 & 34\\34 & 21\end{pmatrix}$	22	$\begin{pmatrix}19 & 25\\25 & 31\end{pmatrix}$	35	$\begin{pmatrix}1 & 35\\35 & 3\end{pmatrix}$
10	$\begin{pmatrix}15 & 18\\18 & 34\end{pmatrix}$	23	$\begin{pmatrix}7 & 19\\19 & 25\end{pmatrix}$	36	$\begin{pmatrix}36 & 1\\1 & 35\end{pmatrix}$
11	$\begin{pmatrix}33 & 15\\15 & 18\end{pmatrix}$	24	$\begin{pmatrix}26 & 7\\7 & 19\end{pmatrix}$		
12	$\begin{pmatrix}11 & 33\\33 & 15\end{pmatrix}$	25	$\begin{pmatrix}33 & 26\\26 & 7\end{pmatrix}$		

When the modulus n is not a prime integer, the correspondence between public and private keys is 1:1, as shown in Table 3. However, the attacker can still obtain the private key since a lookup table containing a list of the public keys that correspond to the private keys can be generated.

We have shown our steps to breaking the BM-ElGamal cryptosystem. We end this section with a few facts about the excellent and secure criteria of the Diffie-Hellman key exchange protocol parameters. When the $Q \pmod n$ order is significantly large, the Diffie-Hellman key exchange is most secure. Typically, the group size is designed to be big enough that the discrete logarithm issue posed by the Diffie-Hellman algorithm cannot be solved in a reasonable time. The modulus n is nearly always a prime integer and should be at least 200 digits in length due to the difficulty of the discrete logarithm issue in such a scenario (Wagstaff Jr., 2019).

5 CONCLUSION

This paper demonstrates that the Diffie-Hellman key exchange protocol and ElGamal public-key cryptosystem namely the BM-ElGamal cryptosystems are insecure for the synchronized chaotic system. The study indicates that even if the initial value of the synchronized chaotic systems is unknown, the private keys r and s may be quickly acquired by an outsider. Exemplifying with a small number (even a small prime number) as a modulus for the Diffie-Hellman key exchange protocol demonstrates that this is a terrible idea. As a result, the cryptosystem, as mentioned earlier, cannot be considered secure in its current state. This paper illustrates how critical it is for a company to undertake security assessments to secure future technology sustainability and improved cybersecurity management.

REFERENCES

Al-Saidi, N.M., Younus, D., Natiq, H., K Ariffin, M.R., Asbullah, M.A. & Mahad, Z. 2020. A New Hyperchaotic Map for A Secure Communication Scheme with An Experimental Realization. *Symmetry*, *12*(11): 1881.

Asbullah, M.A. & Ariffin, M.R.K. 2012. A Proposed CCA-Secure Encryption on An Elgamal Variant. In *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, IEEE: 499-503.

Balasubramaniam, P., & Muthukumar, P. 2014. Synchronization Of Chaotic Systems Using Feedback Controller: An Application to Diffie–Hellman Key Exchange Protocol And Elgamal Public Key Cryptosystem. *Journal of the Egyptian Mathematical Society*, *22*(3): 365-372.

Ismail, E.S. & Hasan, Y.A. 2006. A new version of ElGamal signature scheme. *Sains Malaysiana*, *35*: 69-72.

Mandangan, A., Kamarulhaili, H. & Asbullah, M.A. 2020. A Security Upgrade on the GGH Lattice-Based Cryptosystem. *Sains Malaysiana*, *49*(6): 1471-1478.

Natiq, H., Kamel Ariffin, M.R., Asbullah, M.A., Mahad, Z. & Najah, M. 2021. Enhancing chaos complexity of a plasma model through power input with desirable random features. *Entropy*, *23*(1): 48.

Sarbini, I.N., Jin, W.T., Feng, K.L., Othman, M., Said, M.R.M. & Hung, Y.P. 2018. Garbage-man-in-the-middle (type 2) Attack on the Lucas Based El-Gamal Cryptosystem in the Elliptic Curve Group Over Finite Field. In *Cryptology and Information Security Conference 2018* (CRYPTOLOGY 2018): 35-41.

Silvester, J. R. 1979. Fibonacci Properties by Matrix Methods. *The Mathematical Gazette*, *63*(425): 188-191.

Wagstaff Jr, S. S. 2002. *Cryptanalysis of number theoretic ciphers*. CRC Press.

Zazali, H. H., & Mior, W. A. 2012. Key Exchange in Elliptic Curve Cryptography Based on the Decomposition Problem. *Sains Malaysiana*, *41*(7), 907-910.