Jurnal Teknologi

Full Paper

INTEGRATION OF CFEA-COMPRESSION TECHNIQUE INTO ASYMMETRIC KEY CRYPTOSYSTEMS

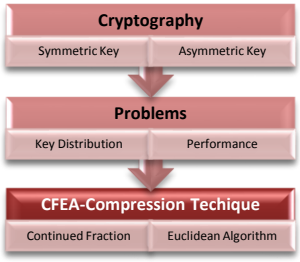
Arif Mandangan[a*], Chang Ee Hung[a], Lee Souk Yin[a], Che Haziqah Che Hussin[b]

[a]Fakulti Sains dan Sumber Alam, Universiti Malaysia Sabah, Malaysia
[b]Center of Preparatory of Science and Technology, Universiti Malaysia Sabah, Malaysia

Article history
Received
15 June 2015
Received in revised form
1 October 2015
Accepted
13 October 2015

*Corresponding author
arifman@ums.edu.my

Graphical abstract



Abstract

In order to provide good level of security, modern cryptosystems need to implement large numbers and complicated mathematical operations. As a consequence, efficiency becomes a new major issue in cryptography. By using proper parameters, some of established asymmetric cryptosystems are believed to be able to provide a good level of security. Since that, aim to develop a mechanism to accelerate encryption and decryption processes of asymmetric cryptosystem without altering their original encryption and decryption algorithms become a big consideration. The aim of this paper is to propose the integration of a compression technique that named as CFEA-Compression technique into some established asymmetric key cryptosystem such as RSA, El-Gamal and Elliptic Curve cryptosystems. CFEA-technique is a combination of Continued Fraction and Euclidean Algorithm (CFEA) which is able to reduce the number of plaintext and ciphertext prior the encryption and decryption procedures.

Keywords: Compression Techique, RSA Cryptosystem, ElGamal Cryptosystem, Elliptic Curve Cryptosystem, Continued Fraction, Euclidean Algorithm

Abstrak

Demi membekalkan tahap keselamatan yang baik, sistemkripto moden perlu menggunakan nombor-nombor besar dan dilaksanakan menggunakan operasi-operasi matematik yang rumit. Natijahnya, kecekapan menjadi isu besar dalam kriptografi. Dengan menggunakan parameter-parameter yang bersesuaian, beberapa sistemkripto tak simetri adalah dipercayai mampu untuk membekalkan suatu tahap keselamatan yang baik. Sejak itu, matlamat untuk membangunkan suatu mekanise bagi melajukan proses-proses enkripsi-dekripsi sistemkripto tak simetrik tanpa mengubah algoritma-algoritma enkripsi dan dekripsi yang asal menjadi suatu pertimbangan yang besar. Matlamat kertas kerja ini adalah untuk mencadangkan integrasi suatu teknik mampatan yang dinamakan sebagai teknik Mampatan-CFEA ke dalam beberapa sistemkripto tak simetri yang terkenal seperti sistemkripto RSA, ElGamal dan Lengkung Eliptik. Teknik Mampatan-CFEA merupakan gabungan Pecahan Berterusan dan Lagoritma Euklid yang mampu mengurangkan bilangan teks biasa dan teks sifer sebelum prosedur enkripsi dan dekripsi.

Kata kunci: Teknik Mampatan, Sistemkripto RSA, Sistemkripto Elgamal, Sistemkripto Lengkung Eliptik, Pecahan Berterusan, Algoritma Euklid

1.0 INTRODUCTION

We are now living in digital age. Almost all people in the world are now connected via variety of communication devices. We perform a lot of tasks and businesses by simply clicking a mouse or touching a screen. That is why we may easily exposed to various types of threats and network security problems. Cryptography is one of the most important part in network security. Basically, cryptography is a science of secret writing which is able to provide network security purposes such as confidentiality, authentication, data integrity and non-repudiation [1]. To provide confidentiality, cryptography scrambles the original and readable message becomes unreadable message. The original message is called plaintext and the scrambled message is called ciphertext. Through encryption process, the plaintext will be scrambled becomes ciphertext. The inverse of encryption is called decryption which is able to recover the plaintext. Both processes need parameters called key. We may classify cryptography into two major classes based on the usage of keys. Symmetric cryptography involves single key call secret key to perform encryption and decryption. On the other hand, asymmetric cryptography involves two different keys called public and private keys [2]. Mathematically, we may represents both encryption and decryption processes as follows [3]:

$$E(m)_k = C \quad (1)$$

$$D(C)_{k'} = m \quad (2)$$

where E is an encryption function, m is a plaintext, k is an encryption key, C is a ciphertext, D is an decryption key and k' is an decryption key. Three main characters in cryptography are Alice (authorized sender), Bob (authorized recipient) and Eve (unauthorized third party). To explain how cryptography provides confidentiality, we consider a scenario where Alice wants to send a secret message to Bob and Eve is interested to intercept the transmitted message between Alice and Bob and read to content of the message.

Recently, asymmetric cryptosystem is more preferred due to its practicality. The asymmetric cryptosystem has solved the key distribution problem in symmetric cryptosystem. But in order to provide a good level of security, the asymmetric cryptosystem need to use large numbers and implement more complicated mathematical operations. As a consequence, asymmetric cryptosystem becomes slower compared to symmetric cryptosystems especially when involving large amount of data. Simple example, let say we want to encrypt 100 messages $m_1, m_2, m_3, \dots, m_{100}$. We need to repeat the encryption processes 100 times to produce 100 ciphertext $C_1, C_2, C_3, \dots, C_{100}$. Symbolically, the processes are as follows:

$$\begin{aligned} E(m_1)_k &= C_1 \\ &\vdots \\ E(m_{100})_k &= C_{100} \end{aligned} \quad (3)$$

To recover the original message, we need to decrypt the 100 ciphertext 100 times as follows:

$$\begin{aligned} D(C_1)_k &= m_1 \\ &\vdots \\ D(C_{100})_k &= m_{100} \end{aligned} \quad (4)$$

The whole encryption-decryption processes consume large amount of time. Larger number of message will consume larger amount of time.

2.0 CFEA-COMPRESSION TECHNIQUE

The emergence of asymmetric key cryptography has solved the biggest problem in symmetric key cryptography, which is the key distribution problem. By using a pair of key consists of a public key and a private key, Bob is able to communicate securely with Alice without any exchange of secret information through secure channel. Every communication can be done through open channel like the Internet. Only Bob as the owner of the private key can decipher the ciphertext which have been encrypted by using his corresponding public key. As a tradeoff, computational cost becomes a new major problem in modern cryptography. Due to calculation complexity and implementation of large numbers, it is found that the asymmetric key cryptosystems perform slower than symmetric key cryptosystems.

To address this problem, some cryptologists lead the improvement of the RSA cryptosystem by proposing four improved RSA cryptosystems which are Batch RSA, Multi Prime RSA, Multi Power RSA [4] and Rebalanced RSA [5]. One of the similarities between these improvements is the major modification on the cryptosystem itself especially on the encryption and decryption algorithms. This modification will alter the security level of the original cryptosystem which probably makes the cryptosystem easier to attack [6, 7]. Thus, any improvement on the performance of cryptosystem must avoid major modification on the encryption and decryption algorithms to store its security level.

In 2013, Chang and Mandangan proposed a Compression-RSA cryptosystem which is the integration of a compression technique into the RSA cryptosystem [8]. The compression technique is a combination of simple Continued Fraction and Euclidean Algorithm. By using this compression technique, the number of plaintext can be reduced from k-plaintext where $k \in \mathbb{Z}^+$ and $k > 2$, becomes only 2-plaintext. That means, only two plaintext will undergo the encryption process and produce 2-ciphertext. To verify the efficiency of the proposed Compression-RSA technique, Loh and Mandangan run some experiments in 2013 [9]. Finding from the experiments show that the Compression-RSA cryptosystem with integrated perform better in encrypting and decrypting large number of plaintext and ciphertext compared to the original RSA cryptosystem. Since the compression technique has potential to be integrated into other asymmetric key cryptosystems, it has been rebranded into CFEA-Compression technique where the acronym CFEA is

comes from Continued Fraction and Euclidean Algorithm [10].

Let the set of original plaintext as $\{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$ where $k \in \mathbb{Z}^+$and $k > 2$. By using the CFEA- Compression technique, these k plaintext can be compressed to a pair of 2-plaintext, denoted as $\{M_1, M_2\}$. No matter how big the value k is, the plaintext will be reduced to only 2 plaintext M_1 and M_2 [8]. The CFEA- Compression technique is basically designed by combining two methods namely Continuous Fraction and Extended Euclidean Algorithm and consists of two main procedures namely compression and decompression procedures. The algorithms of these procedures are shown below:

A. Algorithm: (Compression procedure)

Let the set of original plaintext as $\{\boldsymbol{m_1, m_2, m_3, \cdots, m_{k-1}, m_k}\}$ where $\boldsymbol{k \in \mathbb{Z}^+}$and $\boldsymbol{k > 2}$. Compress the plaintext set by using Continued Fraction method as follows [11]:

$$m_1 + \cfrac{1}{m_2 + \cfrac{1}{m_3 + \cfrac{1}{\vdots \\ m_{k-1} + \cfrac{1}{m_k}}}} = \frac{M_1}{M_2}$$

The number of plaintext has been reduced from k-plaintext becomes only 2 plaintext, M_1 and M_2. The encryption process will involve only these two plaintext instead of k-plaintext. To recover the original set of plaintext, we need to invert the compression procedure. The inverse of compression procedure is called decompression procedure.

B. Algorithm: (Decompression procedure)

By using Euclidean algorithm, compute the following [12]:

$$\begin{aligned} M_1 &= M_2(q_1) + r_1 \\ M_2 &= r_1(q_2) + r_2 \\ r_1 &= r_2(q_3) + r_3 \\ &\vdots \\ r_{k-3} &= r_{k-2}(q_{k-1}) + r_{k-1} \\ r_{k-2} &= r_{k-1}(q_k) + r_k \end{aligned}$$

Where M_1, M_2 are the compressed plaintext, q_i is quotient and r_i is remainder for $i = 1,2, \ldots, k$. Finally, we have $\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\} = \{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$ which is the set of original plaintext.

3.0 ASYMMETRIC KEY CRYPTOSYSTEMS WITH INTEGRATED CFEA-COMPRESSIO TECHNIQUE

Rivest-Shamir-Adleman (RSA) Cryptosystem [13], ElGamal Cryptosystem [14] and Elliptic Curve Cryptography [15] are the most established asymmetric cryptosystem. By implementing proper parameters, these cryptosystems are able to provide a good level of security. The CFEA-Compression technique can be integrated into these cryptosystems without major alteration on the key generation, encryption and decryption algorithms of these cryptosystems. The aim is to accelerate the encryption and decryption procedures and maintain the security level at the same time. Generally, the CFEA-Compression Technique can be integrated into the asymmetric cryptosystems as shown in Figure 1 [10]:

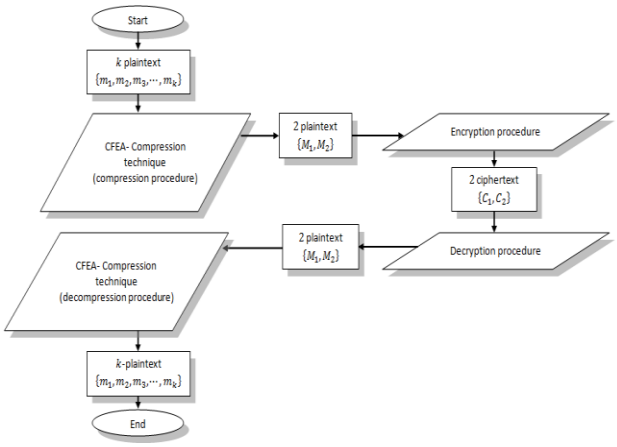


Figure 1 Integration of CFEA-Compression in asymmetric cryptosystem

To show how the CFEA-Compression technique integrated into RSA Cryptosystem, ElGamal Cryptosystem and Elliptic Curve Cryptography, we consider a scenario where Alice wants to send a set of k-plaintext $\{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$ to Bob.

3.1 RSA Cryptosystem

The integration of CFEA-Compression technique in RSA Cryptosystem is shown in the algorithm below:

a) Key generation procedure: done by Bob
   Step 1:Choose two large, random prime numbers p and q
   Step 2:Compute $N = pq$
   Step 3:Compute $\varphi(N) = (p-1)(q-1)$
   Step 4:Choose a random encryption exponent e such that $1 < e < \varphi(N)$ and $gcd\,(e, \varphi(N)) = 1$
   Step 5:Compute the decryption exponent d such that $ed \equiv 1 (\text{mod}\ \varphi(N))$
   Step 6:Send the public key set $\{e, N\}$ to Alice and keep the private key $\{d, p, q\}$ secretly.

b) Compression procedure: done by Alice
   Upon receiving the public key set from Bob, Alice compress the k-plaintext set $\{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$ to only two plaintext M_1 and M_2 as follows:

$$m_1 + \cfrac{1}{m_2 + \cfrac{1}{m_3 + \cfrac{1}{\vdots}}} = \frac{M_1}{M_2}$$
$$m_{k-1} + \frac{1}{m_k}$$

c) Encryption procedure: done by Alice
Step 1: Encrypt the new plaintext M_1 and M_2 as follows
$$C_1 = {M_1}^e (\text{mod } N)$$
$$C_2 = {M_2}^e (\text{mod } N)$$
Step 2: Submit the ciphertext set $\{C_1, C_2\}$ to Bob.

d) Decryption procedure: done by Bob
Upon receiving the ciphertext set $\{C_1, C_2\}$ from Bob, decrypt the ciphertext C_1 and C_2 as follows:
$$M_1 = {C_1}^d (\text{mod } N)$$
$$M_2 = {C_2}^d (\text{mod } N)$$

e) Decompression procedure: done by Bob
Step 1: Decompress the plaintext set $\{M_1, M_2\}$ as follows :
$$M_1 = M_2(q_1) + r_1$$
$$M_2 = r_1(q_2) + r_2$$
$$r_1 = r_2(q_3) + r_3$$
$$\vdots$$
$$r_{k-3} = r_{k-2}(q_{k-1}) + r_{k-1}$$
$$r_{k-2} = r_{k-1}(q_k) + r_k$$

Where M_1, M_2 are the compressed plaintext, q_i is quotient and r_i is remainder for $i = 1,2,\dots,k$.
Step 2: From Step 1 above, Bob has a set of quotient where
$\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\} = \{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$
which is the set of original plaintext.

3.2 ElGamal Cryptosystem

The integration of CFEA-Compression technique in ElGamal Cryptosystem is shown in the algorithm below:

a) Compression: done by Alice
Step 1: Compress the set of k-plaintext $\{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$ as follows
$$m_1 + \cfrac{1}{m_2 + \cfrac{1}{m_3 + \cfrac{1}{\vdots}}} = \frac{M_1}{M_2}$$
$$m_{k-1} + \frac{1}{m_k}$$
Step 2: Choose a large prime p such that $M_1, M_2 < p$.
Step 3: Choose an integer $b \in [0, p-2]$ as her private key
Step 4: Choose an integer g where g is a generator of finite field $\mathbb{F}_p^*$.
Step 5: Send the chosen p and g to Bob.

b) Key generation: done by Bob
Step 1: Compute $e_B = g^b \, mod \, p$ as his public key
Step 2: Send his public key e_B to Alice and keeps his private key b secretly

c) Encryption procedure: done by Alice
Step 1: Choose two random integers $u, v \in \mathbb{F}_p^*$ as her ephemeral keys
Step 2: Encrypt the new plaintext pair M_1 and M_2 to get the ciphertext pairs
$$C_{1,1} = g^u \, mod \, p$$
$$C_{2,1} = g^v \, mod \, p$$
and
$$C_{1,2} = M_1 (e_B)^u \, mod \, p$$
$$C_{2,2} = M_2 (e_B)^v \, mod \, p$$
Step 3: Send the ciphertext pairs $(C_{1,1}, C_{1,2})$ and $(C_{2,1}, C_{2,2})$ to Bob.

d) Decryption procedure: done by Bob
Step 1: Upon receiving ciphertext pairs $(C_{1,1}, C_{1,2})$ and $(C_{2,1}, C_{2,2})$ from Alice, Bob computes
$$s = \left(C_{1,1}\right)^b \, mod \, p$$
$$t = \left(C_{2,1}\right)^b \, mod \, p$$
Step 2: Compute the multiplicative inverse of s and t modulo p, denotes as s^{-1} and t^{-1} such that
$$s(s^{-1}) \equiv 1 \, (mod \, p)$$
$$t(t^{-1}) \equiv 1 \, (mod \, p)$$
Step 3: Recover the original message m as follow
$$M_1 = s^{-1}\left(C_{1,2}\right) mod \, p$$
$$M_2 = t^{-1}\left(C_{2,2}\right) mod \, p$$

e) Decompression procedure: done by Bob
Decompress the plaintext set $\{M_1, M_2\}$ as follows:
$$M_1 = M_2(q_1) + r_1$$
$$M_2 = r_1(q_2) + r_2$$
$$r_1 = r_2(q_3) + r_3$$
$$\vdots$$
$$r_{k-3} = r_{k-2}(q_{k-1}) + r_{k-1}$$
$$r_{k-2} = r_{k-1}(q_k) + r_k$$
where M_1, M_2 are the compressed plaintext, q_i is quotient and r_i is remainder for $i = 1,2,\dots,k$.
From Step 1 above, Bob has a set of quotient where
$\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\} = \{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$
which is the set of original plaintext.

3.3 Elliptic Curve Cryptosystem

The integration of CFEA-Compression in Elliptic Curve Cryptography (ECC) is slightly different with the previous two cryptosystems. This is because plaintext in ECC are in point form (a, b) where $a, b \in \mathbb{Z}$. The point (a, b) is lies on the elliptic curve $E: y^2 = x^3 + Ax + B$.

a) Agreement: done by Alice and Bob
Step 1: Agree to use an elliptic curve
$$E: y^2 = x^3 + Ax + B$$
and a large prime p

Step 2: Form the finite field $\mathbb{F}_p$
Step 3: Pick a point P that lies on E.

b) Key generation: done by Bob
Step 1: Choose a random integer x
Step 2: Compute $Q = xP$
Step 3: Send Q to Alice and keep x secretly.

c) Compression: done by Alice
Step 1: Let
$\{(a_1, b_1), (a_2, b_2), (a_3, b_3), \cdots, (a_{k-1}, b_{k-1}), (a_k,$
be the set of plaintext points. Denote plaintext points as follows:

$$a_1 = m_{11}, b_1 = m_{12}$$
$$a_2 = m_{21}, b_2 = m_{22}$$
$$a_3 = m_{31}, b_3 = m_{32}$$
$$\vdots$$
$$a_{k-1} = m_{(k-1)1}, b_{k-1} = m_{(k-1)2}$$
$$a_k = m_{k1}, b_k = m_{k2}$$

Step 2: The new set of plaintext is

$$\{m_{11}, m_{12}, m_{21}, m_{22}, m_{31}, m_{32}, \cdots, m_{(k-1)1}, m_{(k-1)2}, m_{k1}, m_{k2}\}$$

d) Encryption: done by Alice
Step 1: Choose a random integer y
Step 2: Compute $R = yP$
Step 3: For each $M_1, M_2 \in E(\mathbb{F}_p)$, calculate

$$C_1 = M_1 + yQ$$
$$C_2 = M_2 + yQ$$

Step 4: Send the ciphertext pair $\{C_1, C_2\}$ to Bob.

e) Decryption: done by Bob
Step 1: Upon receiving ciphertext pair $\{C_1, C_2\}$ from Bob, compute

$$M_1 = C_1 - xR$$
$$M_2 = C_2 - xR$$

Step 2: From Step 1 above, Bob has a set of quotient $\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\}$ where $\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\} = \{m_1, m_2, m_3, \cdots, m_{k-1}, m_k\}$ which is the set of original plaintext.

f) Decompression procedure: done by Bob
Step 1: Decompress the plaintext set $\{M_1, M_2\}$:

$$M_1 = M_2(q_1) + r_1$$
$$M_2 = r_1(q_2) + r_2$$
$$r_1 = r_2(q_3) + r_3$$
$$\vdots$$
$$r_{k-3} = r_{k-2}(q_{k-1}) + r_{k-1}$$
$$r_{k-2} = r_{k-1}(q_k) + r_k$$

where M_1, M_2 are the compressed plaintext, q_i is quotient and r_i is remainder for $i = 1,2, \ldots, k$.
Step 2: From Step 1 above, Bob has a set of quotient $\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\}$ where

$$\{q_1, q_2, q_3, \cdots, q_{k-1}, q_k\} = \{m_{11}, m_{12}, m_{21}, m_{22}, m_{31}, m_{32}, \cdots, m_{(k-1)1}, m_{(k-1)2}, m_{k1}, m_{k2}\}$$

Step 3: Recover back the set of plaintext points as follows:

$$m_{11} = a_1, m_{12} = b_1$$
$$m_{21} = a_2, m_{22} = b_2$$
$$m_{31} = a_3, m_{32} = b_3$$
$$\vdots$$
$$m_{(k-1)1} = a_{k-1}, m_{(k-1)2} = b_{k-1}$$
$$m_{k1} = a_k, m_{k2} = b_k$$

Finally Bob recovers the original set of plaintext points
$\{(a_1, b_1), (a_2, b_2), (a_3, b_3), \cdots, (a_{k-1}, b_{k-1}), (a_k, b_k)\}$

4.0 CONCLUSION

This paper has shown how the CFEA-Compression technique can be integrated into three most established asymmetric cryptosystems namely the RSA Cryptosystem, ElGamal Cryptosystem and Elliptic Curve Cryptography. These cryptosystems are the most preferred cryptosystems in real life applications. By integrating this compression technique, the number of plaintext to be encrypted and also the number of the ciphertext to be decrypted can be reduced from any number to become only a pair of plaintext and a pair of ciphertext. The compression and decompression procedures only involve simple and low cost mathematical operations. For future work, thorough analysis on the efficiency of the RSA Cryptosystem, ElGamal Cryptosystem and Elliptic Curve Cryptography with integrated CFEA-Compression technique will be conducted to verify whether the RSA, ElGamal and Elliptic Curve cryptosystems have better performance than their original version or not. As a conclusion, CFEA-Compression technique is a compression technique that can be easily integrated into asymmetric cryptosystems to reduce the number of plaintext and ciphertext before undergo to the encryption and decryption procedures.

Acknowledgement

We would like to thank Universiti Malaysia Sabah and the Ministry of Higher Education for supporting this project under *Research Acculturation Grant Scheme* (*RAGS*) RAG0001-SG-2012 (Malaysia).

References

[1] Hoffstein, J., Pipher, J. and Silverman, J. H. 2008. An Introduction to Mathematical Cryptography. New York: Springer Science+Bussiness Media. 37-39.
[2] Diffie, W. and Hellman, M. 1976. New Direction in Cryptography. *IEEE Transaction on Information Technology*. 22(6): 644-654.
[3] Stallings, W. 2011. *Cryptography and Network Security Principles and Practice*. 5th Edition. New York: Prentice Hall: 35-36.
[4] Verma, S. and Garg, D. 2011. Improvement in RSA Cryptosystem. *Journal of Advanced in Information Technology*. 2(3): 146-151
[5] Boneh, D. and Shacham, H. 2002. Fast Variants of RSA. *CryptoBytes*. 5(1): 1-9
[6] Boneh, D. and Durfee, G. 2000. Cryptanalysis of RSA with Private Key d Less Than $N^{0.292}$. *IEEE Transaction Information Theory*. 46(4): 1339-1349.
[7] R. S. Kumar, R. S., Narasimam, C. and Setty, S. P. 2012. Generalization of Boneh-Durfee's Attack for Arbitrary

Public Exponent RSA. *International Journal of Computer Applications.* 49(19): 37-40.

[8] Chang Ee Hung and Arif Mandangan.2013. Compression-RSA: New Approach of Encryption and Decryption Method. *AIP Conference Proceeding 1522.* American Institute of Physics. 50-54.

[9] Arif Mandangan, Loh Chai Mei, Chang Ee Hung and Che Haziqah Che Hussin. 2013. Comression-RSA Technique: A More Efficient Ecryption-Decryption Procedure, AIP Conference *Proceeding 1602.* American Institute of Physics. 50-55.

[10] Arif Mandangan, Loh Chai Mei, Chang Ee Hung and Che Haziqah Che Hussin. 2015. CFEA-Techique: Smaller Size of the Compressed Plaintext. *Int. Journal of Cryptology Research.* 5(1): 1-10.

[11] Moore and Charles, D. 1964. *An Introduction to Continued Fractions.* Washington, D.C.: The National Council of Teachers of Mathematics.

[12] Farouzan, B. A. 2008. *Introduction to Cryptography and Network Security*. New York: McGraw-Hill Companies: 301-311

[13] Rivest, R. L., Shamir, A. and Adleman, L. 1978. A Method for Obtaining Digital Signature and Public Key Cryptosystem. *Communication ACM*. 21: 120-126.

[14] ElGamal T. 1985. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithm. *IEEE Transactions on Information Theory*. 31(4): 469-472.

[15] Koblitz, N. 1987. Elliptic Curve Cryptosystem. *Mathematics of Computation.* 48(177): 203-209.