TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 6, December 2020, pp. 2948~2955
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i6.17278

Authentication system for e-certificate by using RSA's digital signature

Kritsanapong Somsuk[1], Mongkhon Thakong[2]
[1]Department of Computer and Communication Engineering, Faculty of Technology,
Udon Thani Rajabhat University, Thailand
[2]Office of General Education, Udon Thani Rajabhat University, Thailand

Article Info

Article history:

Received Jun 4, 2020
Revised Jun 19, 2020
Accepted Jul 20, 2020

Keywords:

Checking application
Digital signature
E-certificate
RSA
Signing application

ABSTRACT

Online learning and teaching become the popular channel for all participants, because they can access the courses everywhere with the high-speed internet. E-certificate is being prepared for everyone who has participated or passed the requirements of the courses. Because of many benefits from e-certificate, it may become the demand for intruders to counterfeit the certificate. In this paper, Rivest-Shamir-Adleman (RSA)'s digital signature is chosen to sign e-certificate in order to avoid being counterfeited by intruders. There are two applications to manage e-certificate. The first application is the signing application to sign the sub image including only participant's name in e-certificate. In general, the file of digital signature is divided from e-certificate. That means, both of them must be selected to compare each other in checking application. In fact, the solution will be approved when each pixel of participant's name is equal to each part from the decrypted message at the same position. In experimental session, 40 e-certificates are chosen for the implementation. The results reveal that the accuracy is 100% and both of signing and checking processes are completed rapidly fast, especially when signing application is applied with Chinese remainder theorem (CRT) or the special technique of CRT. Therefore, the proposed method is one of the best solutions to protect e-certificate from the forgery by intruders.

Corresponding Author:

Kritsanapong Somsuk,
Department of Computer and Communication Engineering,
Udon Thani Rajabhat University,
Udon Thani, 41000, Thailand.
Email: kritsanapong@udru.ac.th

1. INTRODUCTION

In Thailand, schools and educational institutions enter a new world of online learning and teaching which consist of regular and short courses. In general, e-certificate is provided to all participants who have participated or passed the requirements of the courses. In fact, e-certificate provides benefits for the trainees such as job application and getting a promotion. With many benefits, it may become the demand for intruders to counterfeit the certificate.

Cryptography [1] is a technique to protect the information through communication channels which are considered as the insecure channel such as internet. Encryption process is chosen to transform the plaintext as unreadable message which is called ciphertext before sending this value to receivers. The receiver, after arriving of ciphertext, can recover the original plaintext by using the decryption process. In addition, there are two types of cryptography. The first technique is called symmetric key cryptography. It used the same key, is called secret key, for both of encryption and decryption processes. Speed and time [2, 3] to finish the processes in both sides are

the prominent point of this technique. However, the obstacle is about the way to find the secure channel to share the secret key. In 1976, the other type of cryptography which is called asymmetric key cryptography [4] or public key cryptography was proposed. The different keys for encryption and decryption processes are required. Public key is the disclosed key to everyone in group but private key is kept secretly by owner. Nevertheless, exchanging the secret key is only one capability of this method. In 1978, Rivest-Shamir-Adleman (RSA) [5] was proposed by R. Rivest, A. Shamir and L. Adleman. It's beneficial for various tasks such as data encryption, digital signature and key exchanging. Moreover, multimedia can be implemented by using RSA such as text, image and voice [6-8]. However, RSA is based on integer factorization problem (IFP) [9-12]. If modulus is factored as prime numbers, RSA is broken. Therefore, the modulus should be assigned at least 1024 bits [13-15] to avoid an attack by intruders.

In this paper, to ensure that e-certificate is not counterfeited, the RSA's digital signature is chosen to sign the image. After the signing process, the other file which belongs to the digital signature of e-certificate is generated. Then, to validate the e-certificate, it should be compared with the decrypted digital signature. In addition, e-certificate will be approved when all pixels of the sub image are equipvalent to the outcome of the decrypted file at the same position.

2. RELATED WORKS

2.1. RSA

RSA is the best well-known public key cryptography. The word of RSA stands for Rivest, Shamir and Adleman who were three inventors of this algorithm. In fact, there is not any algorithm which can break RSA in polynomial time when the modulus is assigned at least 1024 bits and all parameters are strong. Therefore, it is the reason that RSA is still widely used at present. For data encryption, there are three main processes as follows:

Process 1 (key generation process): it is the process to generate a pair of keys to encrypt and decrypt the processes. It is divided as five steps as follows:

Step 1: Generate two large and strong prime numbers, p and q, randomly. In addition, the result of the multiplication between p and q should not be less than 1024 bits.

Step 2: Compute the modulus, n, from $n = p*q$

Step 3: Compute Euler' s totient function, $\Phi(n)$, from $\Phi(n) = (p-1)(q-1)$

Step 4: Choose public key, e, with the following condition: $1 < e < \Phi(n)$ and the greatest common divisor between e and $\Phi(n)$ must be equal to 1 ($\gcd(e, \Phi(n)) = 1$).

Step 5: Compute private key, d, from the following equation: $ed \equiv 1 \bmod \Phi(n)$. In fact, the key to find the result in this step is to use extended Euclidean algorithm [16-18].

Process 2 (Encryption Process): it is the process to transform plaintext, m, where $0 < m < n$, as unreadable message or ciphertext, c, before sending to receiver by using the following equation:

$$c = m^e \bmod n \quad (1)$$

Process 3 (Decryption Process): the process to recover m by using the decryption equation:

$$m = c^d \bmod n \quad (2)$$

In the (2), it implies that m is always recovered after finishing the decryption process.

However, d should be always assigned larger than e. The reason is that the (2) takes very high computation cost in comparison to the (1). Therefore, to reduce time in decryption process, many techniques were proposed such as Chinese remainder theorem (CRT) [19, 20], private key with low hamming weight [21] and low private key (may become the drawback of the system) [22-24].

2.2. RSA's digital signature

Besides data encryption, RSA can be also chosen as digital signature [25]. In fact, this method is selected for authentication. For digital signature, d is selected as the exponent instead of e in order to sign the signature. Therefore, all users can verify the signed text by using the exponent, e, for decryption process. In general, if the signed text is the authentic information, it is always recovered after finishing the decryption process.

2.3 Speed up RSA using CRT

ue to the fact that d is always assigned larger than e, then CRT was proposed to speed up the process by dividing d as two small sub exponents, d_p and d_q. In fact, both of them are calculated from the following equations:

$$d_p = d \bmod p - 1 \quad (3)$$

$$d_q = d \bmod q - 1 \quad (4)$$

In addition, they are the exponents of two equations as follows:

$$m_p = c^{d_p} \bmod p \quad (5)$$

$$m_q = c^{d_q} \bmod q \quad (6)$$

And the equation to recover m is as follows:

$$m = (m_p y_q q + m_q y_p p) \bmod n \quad (7)$$

where $y_p = p^{-1} \bmod q$ and $y_q = q^{-1} \bmod p$.

However, this technique may consume the high computation cost when both of d_p and d_q are still large. Therefore, the special technique [26] to speed up CRT was proposed in 2018. In fact, this technique can finish the process faster than original CRT when d_p and d_q are larger than $\frac{p}{2}$ and $\frac{q}{2}$, respectively. On the other hand, it becomes slower than original CRT. Assuming d_p and d_q are found and gcd (d_p, $p - 1$) = gcd (d_q, $q - 1$) = 1, x_p and x_q must be computed at first.

$$x_p = p - d_p \quad (8)$$

$$x_q = q - d_q \quad (9)$$

In fact, they are represented as the new exponents for the below equations:

$$m_p = (c^{-1})^{x_p - 1} \bmod p \quad (10)$$

$$m_q = (c^{-1})^{x_q - 1} \bmod q \quad (11)$$

Then, m is recovered by using the (7) after m_p and m_q are calculated.

3. THE PROPOSED METHOD

The aim of this paper is to propose the system to authenticate e-certificate which is the red green blue (RGB) image. The advantage is to protect the masquerade from intruders. RSA's digital signature is chosen as a tool to sign the certificate. Two algorithms are presented to solve the problem. The first algorithm is for signing the digital signature of e-certificate. This system must be kept secretly because d is the key of the process. The other algorithm is for checking the signature. The key process is to select both of original image and its digital signature to verify each other.

In addition, only sub image including the participant' s name is chosen for both of signing and checking processes. The reason is that the other parts of the original image are similar to the other images at the same position. Furthermore, the plaintext of each pixel of sub image is from the following equation:

$$t(i, j) = (r(i, j) + g(i, j) + b(i, j))/3 \quad (12)$$

where, i, j are the position of row and column of the sub image
r is the red component
g is the green component
b is the blue component
t is plaintext which is the pixel in row i and column j

Figure 1 is shown the system of Signing algorithm that is divided as seven steps as follows:
Step 1: choosing the original image
Step 2: selecting sub image which is chosen as plaintext for signing process
Step 3: computing the average value of each pixel
Step 4: managing all plaintexts of sub image, m_1, m_2, ⋯, m_i, ⋯, m_j, the examples of managing m_1 and m_2 are as follows:

$$m_1 = t_{(1,1)} t_{(1,2)} t_{(1,3)} \cdots t_{(1,s)}$$
$$m_2 = t_{(1,s+1)} t_{(1,s+2)} t_{(1,s+3)} \cdots t_{(1,2s)}$$

Step 5: computing ciphertexts, c_1, c_2, ⋯, c_i, ⋯, c_j, as follows:

$$c_1 = m_1^d \bmod n,\ c_2 = m_2^d \bmod n,\ \cdots,\ c_i = m_i^d \bmod n,\ \cdots, c_j = m_j^d \bmod n$$

Step 6: $c = c_1$+"A"+c_2+"A"+⋯+c_i + "R"+⋯+c_j
Step 7: Writing c is in the format of text file

In general, there is only the participant's name in e-certificate that is different from the other images at the same position. Then, it is selected for signing the signature. However, total pixels of this part may be larger than the maximum value of original plaintext. That mean, this part must be divided repeatly and all parts of sub image will be encrypted by using d as the exponent. Furthermore, for the simple implementation, t should be expanded as 3 digits when it is less than 3. For example, if $t = 26$, it should be changed as 026. The letter "R" is chosen to inform receiver about the scope of total pixels in one row. On the other hand, the letter "A" is selected to inform receiver about the length of one ciphertext. Because RGB image has three components, the average is the representation of each pixel. In fact, Algorithm 1 is shown the sequential steps to sign e-certificate thoroughly.

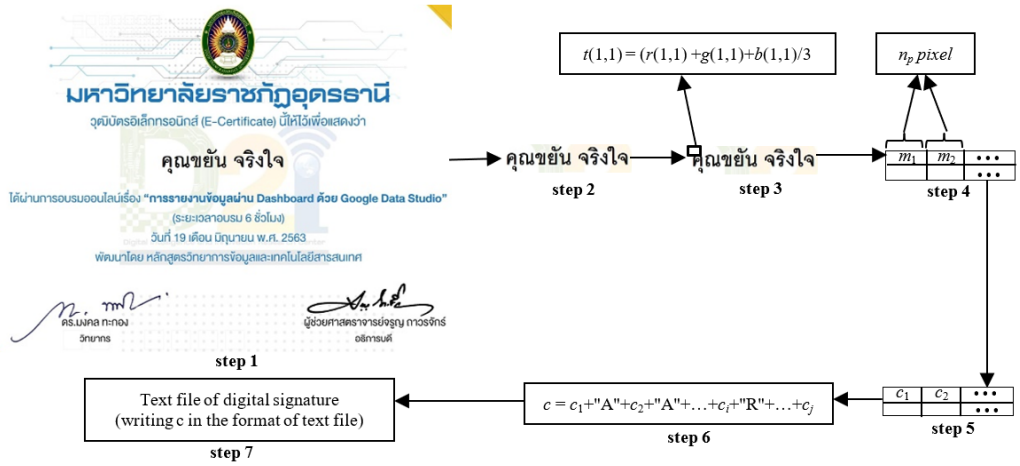


Figure 1. System of signing algorithm

Algorithm 1: Signing e-certificate by using RSA's digital signature

Input: n, d

1. m ← Read image (type RGB) of e-certificate
2. m ← Choose only sub image of m that is different from the other images (width = w and height = h)
3. n_p ←Numbers of pixel
4. $count$ ← 0, p ← "", c ← ""
5. For i = 0 to h − 1
6. c ← 0
7. p ←0
8. For j = 0 to w − 1
9. r ←Red component of image in position Row i and Column j
10. g ← Green component of image in position Row i and Column j
11. b ← Blue component of image in position Row i and Column j
12. t ← (r + g + b)/3
13. IF j = w − 1 then
14. p ← p + ""+t
15. End IF
16. c ← c + "" + String(p^d mod n) + "R"
17. p ←""
18. IF $count$ = n_p then
19. $count$ ← 0
20. c ←c + ""+String(p^d mod n) + "A"
21. End IF
22. End For
23. End For
24. s ←c (writing in text file)

output: s (digital signature of original image)

Example 1: Assuming n = 1528125953 (43313 x 35281), e = 617, d = 334337753 and the average of each pixel of sub image is as follows:

187	74	3	254	18
145	126	47	123	148

Finding the signd text by using Algorithm 1 with n_p = 3

Sol: Because $n_p = 3$, then plaintext is generated by every three pixels and the remainder of each row.

1st row: $p_1 = 187074003$, $p_2 = 254018$, 2nd row: $p_3 = 145126047$, $p_4 = 123148$

Then, the signature of each plaintext is as follows:

1st row: $c_1 = 187074003^{334337753} \bmod 1528125953 = 632263766$

$c_2 = 254018^{334337753} \bmod 1528125953 = 694599143$

2nd row: $c_3 = 145126047^{334337753} \bmod 1528125953 = 553190888$

$c_4 = 123148^{334337753} \bmod 1528125953 = 860993807$

Therefore, the signed text is 632263766A694599143R553190888A860993807

In fact, Algorithm 2 is selected to decrypt the digital signature to compare each pixel of the result with sub image. The signature is approved when all pixels from both of them are equipvalent to each other. On the other hand, the image is rejected whenever only one pixel is different.

Algorithm 2: Checking e-certificate by using RSA's digital signature

```
Input: n, e
       1.  m ← Read image (type RGB) of e-certificate
       2.  m ← Choose only sub image of m that is different from the other images (width =
           w and height = h)
       3.  t ← Read digital signature of m that is the result from Algorithm 1
       4.  Array of l ← Split m by "R"
       5.  For r = 0 to (length of l) - 1
       6.      Array of k ← Split l[r] by "A"
       7.      y ← ""
       8.      For c = 0 to (length of k) -1
       9.          y ← y + String(k[c]^e mod n)
       10.     End For
       11.     For i = 0 to w-1
       12.         out[r][i] ← cut y from 3i to 3i+3 // each pixel of output which has 3
           digits to compare with original image
       13.     End For
       14. End For
       15. count ← 0
       16. For i = 0 to h -1
       17.     For j = 0 to w - 1
       18.         r ←Red component of m in position Row i and Column j
       19.         g ← Green component of m in position Row i and Column j
       20.         b ← Blue component of m in position Row i and Column j
       21.         t ← (r + g + b)/3
       22.         IF out[i][j] = t then
       23.             count++
       24.         End IF
       25.     End For
       26. End For
       27. IF count == w*h then
       28.     The signature is approved
       29. Else
       30.     The signature is rejected
       31. End IF
output: output of signature
```

Example 2: Using Algorithm 2 to check the relation between sub image and the signed text in example 1

Sol: From example 1, the signed text is 632263766A694599143R553190888A860993807. It imples that there are two rows, because the letter 'R' is happened only one position. The signed text is divided for each row as follows: 1st row: 632263766A694599143, 2nd row: 553190888A860993807

Moreover, the letter 'A' is happened only one position in each row, then it is divided as two ciphertext:1st row: $c_1 = 632263766$ and $c_2 = 694599143$, 2nd row: $c_3 = 553190888$ and $c_4 = 860993807$

Then, the decrypted message of each cipher is as follows:

1st row: $p_1 = 632263766^{617} \bmod 1528125953 = 187074003$, $p_2 = 694599143^{617} \bmod 1528125953 = 254018$

2nd row: $p_3 = 553190888^{617} \bmod 1528125953 = 145126047$, $p_4 = 632263766^{617} \bmod 1528125953 = 123148$

After that, each plaintext is divided as 3 digits per group, the result is as follows;

1st row: group1: 187, group2: 74, group3: 3, group4: 254, group5: 18, 2nd row: group1: 145, group2: 126, group3: 47, group4: 123, group5: 148.

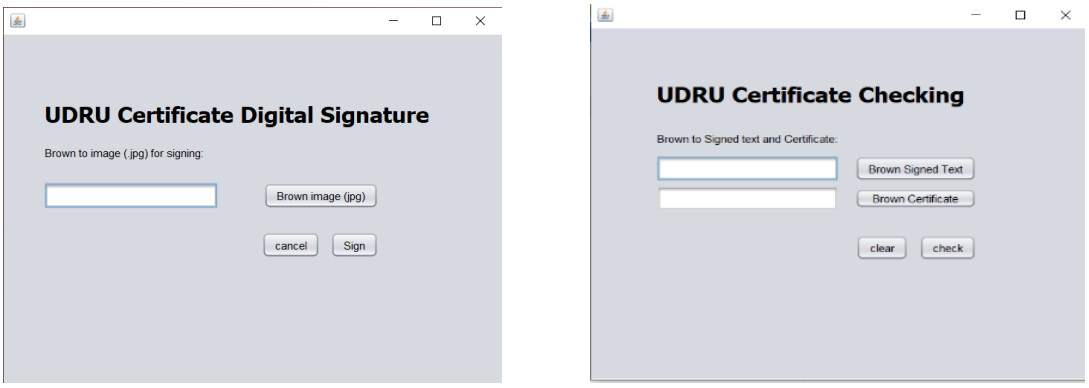
Because each group is matched with each pixel of sub image at the same position. Therefore, this signature is approved. From both of example 1 and example 2, the multiplication between 3 (3 digits per 1 pixel) and n_p must be less than total digits of n. The reason is that the plaintext must be always less than n. In fact, if it is larger than n, then the original plaintext can not be recovered. Furthermore, time to finish the process can be reduced by using one of CRT or the special techniaue improved from CRT [26]. In fact,

original CRT is suitable for small d_p and d_q, $d_p < \frac{p}{2}$ and $d_q < \frac{q}{2}$. On the other hand, the method in [26] is chosen instead of original CRT.

4. RESULTS

In this section, it shows the experimental results which are divided into two parts. Part 1 shows the signing application and the checking application. Part 2 shows time consuming in both applications by using different algorithms. In addition, big integer class which is one of the built-in classes in java programming language is chosen for the implementation, because it can be implemented as the unlimited size of integer. Furthermore, all experiments were conducted on 1.80 GHz an Intel(R) Core™ i5 with 8 GB memory.

Figure 2 (a) shows the signing application. Assuming e-certificate must be signed. It must be selected from "Brown image (jpg)" button. After e-certificate is selected, the location of this file will be displayed in text field. In fact, the image will be signed when the collect location is occurred and "sign" button is chosen. Figure 2 (b), Checking Application is the program to compare between all pixels of sub image and the decrypted value of signed text. In fact, the result is verified when all comparisons of pixels are matched to each other. On the other hand, if only one pixel at the same position is different, e-certificated is rejected. Figure 3, it is the example of input and output from signing application. Figure 3 (a) is the original image for the process. After pressing "sign" button, only sub image in Figure 3 (b) will be selected for signing the signature. Figure 3 (c) is the digital signature of sub image in Figure 3 (b).



(a) (b)

Figure 2. (a) Signing application, and (b) checking application

The next experiment is about time-consuming in both of signing application and checking application. All E-Certificates in the short course (40 participants) are chosen as the original images in both processes. In addition, the width and height of sub image are assigned as 300 pixels and 70 pixels, respectively. In fact, d is assigned very large in comparison to e in order to avoid an attack by using some methods which have very high performance when d is small such as wiener's attack [22] and some other improvements [23, 24]. Moreover, three methods are chosen for signing application to consider the best solutions.

In Figure 4, the condition 1 is $d_p < \frac{p}{2}$ and $d_q < \frac{q}{2}$, the condition 2 is $d_p \approx \frac{p}{2}$ and $d_q \approx \frac{q}{2}$and the condition 3 is $d_p > \frac{p}{2}$ and $d_q > \frac{q}{2}$. The experimental results show that checking application is the fastest application, because e is usually smaller than d. However, signing application with CRT is certainly faster than the method in [26] in condition 1 but it becomes slower than this method in condition 3. The reason is that the exponents for the special method become small when d_p and d_q are large. Moreover, in condition 2 original CRT is faster than the special method. Although the lenght of exponents are same, the special method has to compute modular inverse of ciphertext. Therefore, the best solution to choose decryption algorithm is based on the characteristic of d. Nevertheless, the accuracy for all 40 images are 100%. In fact, assuming intruders intend to sign the fake signature, they have to find d which is kept secretly by admins. Moreover, if all parameters are strong and n is assigned at least 1024 bits, there is no method that can break RSA within polynomial time. Therefore, the proposed method is one of the best solutions to protect the forgery of e-certificate.

Although, elliptic curve cryptography (ECC) [27-28] which is a type of public key cryptography can be chosen to process digital signature as RSA, this algorithm is not suitable to apply with the proposed method. The reason is that string concatenation during many pixels of sub image is required. However, the plaintext

must be the point on curve. In fact, if ECC is chosen instead of RSA, the plaintext from the combination of many pixels must be converted as the point on curve before including in encryption process.

(a) (b)

(c)

Figure 3. Example of input and output from signing application;
(a) original, (b) sub image, and (c) digital

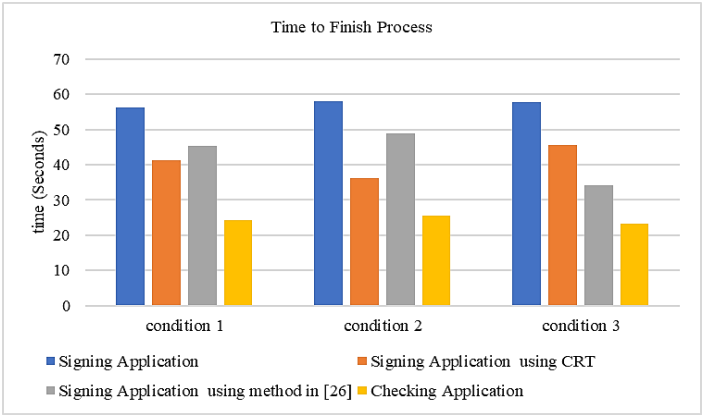


Figure 4. Time to finish process with different techniques

5. CONCLUSION

In this paper, RSA's digital signature is proposed to apply which e-certificate which is the RGB image to avoid counterfeiting the certificate. Only participant's name in e-certificate is selected to the signing process, because this is only part which is different from the other images at the same position. Additionally, the main process to check the signed text is the selection from both of e-certificate, only participant's name, and decrypted signed text to compare each other. In fact, e-certificate is approved when each pixel of the sub image is matched with the decrypted message. However, for the real situation, private key (d) should be assigned larger than public key (e) to avoid an attack by some methods. Therefore, CRT or its improvement shoud be

selected to speed up the process. The experimental results show that the proposed application can be chosen in order to apply with e-certificate to protect the forgery from intruders.

REFERENCES

[1] C. A. Sari, et al., "An improved security and message capacity using AES and Huffman coding on image steganography," *TELKOMNIKA Telecommunication, Computing, Electronics and Control,* vol. 17, no. 5, pp. 2400 - 2409, 2019.
[2] M. Sohal, S. Sharma, "BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing," *Journal of King Saud University - Computer and Information Sciences*, vol. 1, pp. 1-9, 2018.
[3] R. Jyothi , N. G. Cholli, "An efficient approach for secured communication in wireless sensor networks," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 1641 - 1647, 2020.
[4] W. Diffie, M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, 1976.
[5] R. L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of ACM*, vol. 21, no. 2, pp. 120-126, 1978.
[6] N. Anane, M. Anane, H. Bessalah, M. Issad, K. Messaoudi, "RSA Based Encryption Decryption of Medical Images," *Proceedings of International Multi-Conference on Systems, Signals and Devices*, pp. 1-4, June 27-30, 2010.
[7] G. Zhao, X. Yang, B. Zhou, W. Wei, "RSA-Based Digital Image Encryption Algorithm in Wireless Sensor Networks," *Proceedings of International Conference on Signal Processing Systems*, pp. 640-643, 5-7, 2010.
[8] L. Gong, K. Qiu, C. Deng, N. Zhou, "An optical image compression and encryption scheme based on compressive sensing and RSA algorithm," *Optics and Lasers in Engineering*, vol. 121, pp. 169-180, 2019.
[9] M. E.Wu, R. Tso, H. M. Sun, "On the improvement of Fermat factorization using a continued fraction technique," *Future Generation Computer Systems*, vol. 30, no. 1, pp.162-168, 2014.
[10] K. Omar, L. Szalay, "Sufficient conditions for factoring a class of large integers," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 13, no. 1, pp. 95-103, 2010.
[11] S. Vynnychuk, Y. Maksymenko, V. Romanenko, "Application of the basic module's foundation for factorization of big numbers by the fermat method," *EasternEuropean Journal of Enterprise Technologies*, vol. 6, no. 4, pp. 14-23, 2018.
[12] Q. Wang, X. Fan, H. Zang, Y. Wang, "The Space Complexity Analysis in the General Number Field Sieve Integer Factorization," *Theoretical Computer Science*, vol. 360, pp. 76-94, 2016.
[13] K. Gyu-Chol, L. Su-Chol, H. Hak-Chol, "Fast rebalanced RSA signature scheme with typical prime generation," *Theoretical Computer Science*, vol. 380-381, pp. 1-19, 2020.
[14] W. Susilo, J. Tonien, G. Yang, "A generalised bound for the Wiener attack on RSA," *Journal of Information Security and Applications*, vol. 53, pp. 1-4, 2020.
[15] S. Vollala, V. V. Varadhan, K. Geetha, N. Ramasubramanian, "Design of RSA processor for concurrent cryptographic transformations," *Microelectronics Journal*, vol. 63, pp. 112-122, 2017.
[16] J. Zhou, J. Hu, P. Chen, "Extended Euclid algorithm and its application in RSA", *Proceedings of International Conference on Information Science and Engineering*, pp. 2079-2081, December 4-6, 2010.
[17] M. M. Asad, L. Marouf, Q. A. Al-Haija, A. Alshuaibi, "Performance Analysis of 128-bit Modular Inverse Based Extended Euclidean Using Altera FPGA Kit," *Procedia Computer Science*, vol. 160, pp. 543-548, 2019.
[18] Q. Zhou, C. Tian, H. Zhang, J. Yu, F. Li, "How to securely outsource the extended euclidean algorithm for large-scale polynomials over finite fields," *Information Sciences*, vol. 512, pp. 641-660, 2020.
[19] L. Peng, A. Takayasu, "Generalized cryptanalysis of small CRT-exponent RSA," *Theoretical Computer Science*, vol. 759, pp. 432-458, 2019.
[20] H. M. Sun, M. E. Wu, M. J. Hinek, C. T. Yang, V. S. Tseng, "Trading decryption for speeding encryption in Rebalanced-RSA," *Journal of Systems and Software*, vol. 82, no. 9, pp. 1503-1512, 2009.
[21] K. Somsuk, "The improving decryption process of RSA by choosing new private key", *Proceedings of International Conference on Information Technology and Electrical Engineering*, pp. 1-4, October 5-6, 2016.
[22] M. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Transactions on Information Theory*, vol. 36, no. 3, pp. 553-558, 1990.
[23] D. Boneh, G. Durfee, "Cryptanalysis of RSA with Private Key d less than No. 292," *IEEE Transactions on Information Theory,* vol. 46, no. 4, pp. 1339-1349, 2000.
[24] M. E. Wu, C. M. Chen, Y. H. Lin, H. M. Sun, "On the Improvement of Wiener Attack on RSA with Small Private Exponent," *The Scientific World Journal*, vol. 2014, pp. 1-9, 2014.
[25] J. HongSeo, "Efficient digital signatures from RSA without random oracles," *Information Sciences*, vol. 512, pp. 471-480, 2020.
[26] K. Somsuk, T. Chiawchanwattana, C. Sanemueang, "Speed up RSA's Decryption Process with Large sub Exponents using Improved CRT," *Proceedings of International Conference on Information Technology*, pp. 1-5, October 24-26, 2018.
[27] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203-209, 1987.
[28] V. S. Miller, "Uses of Elliptic Curves in Cryptography," *Conference: Lecture notes in computer sciences*, vol. 218, pp. 417-426, 1987.