International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 3 944 - 949

Analytical Study of Modified RSA Algorithms for Digital Signature

Sangita A. Jaju
Department of Computer Science
Dayanand Science College
Latur, (M.S.), India
jaju.sangita@gmail.com

Santosh S. Chowhan
School of Computational Science
S. R. T. M. University
Nanded, (M.S.), India
drschowhan@gmail.com

Abstract- Digital signature has been providing security services to secure electronic transaction. Rivest Shamir Adleman (RSA) algorithm was most widely used to provide security technique for many applications, such as e-mails, electronic funds transfer, electronic data interchange, software distribution, data storage, electronic commerce and secure internet access. In order to include RSA cryptosystem proficiently in many protocols, it is desired to formulate faster encryption and decryption operations. This paper describes a systematic analysis of RSA and its variation schemes for Digital Signature.

Keyword :- *Cryptosystem; Digital Signature; PKC; Private Key; Public key;*

*****

I. INTRODUCTION

In a cryptosystem, RSA algorithm plays an important role [1]. Number of researchers had studied RSA algorithm and implemented it on various applications which are related to security, confidentiality and integrity of information, for document authentication in digital media. Digital signatures plays vital role. Digital signatures are generated by using DSA and RSA algorithms [1, 2]. To enhance speed of encryption, decryption process, to increase level of security and information to be encrypted or decrypted, modified version of RSA algorithm was developed by many researchers. In Digital Signatures cost, security and time are three important factors for feasible solution to any problem. In order to reduce cost of encryption and decryption it is important to increase speed of conversion and at the same time security level should be increased [3].

II. RELATED WORKS

A variety of articles can be found, which proposes different approaches on RSA.

For hardware implementation of RSA, Milan Markovic et al. had proposed Optimizing RSA Algorithm Implementation on Signal Processor for Asymmetric Private key length. The proposed work was influenced on optimization of RSA algorithm which was especially for assembler of Texas instrument family of signal processor TM532054X. Optimization techniques for modular evaluation such as Standard Dividing Procedure with remainder, Reciprocal value method and Montgomery's Procedure, Chinese Remainder Theorem (CRT) for RSA private key operation were used [4]. The improvement was based on modified Karasuba – Offman's multiplication algorithm which was 10% for RSA with modulus '*n*' and length was more than 1024 bits. By CRT method the private key accelerated more than three times, best results are obtained by Montgomery's approach in which private key accelerated about 5 times by applying complete set of considered optimization procedure. From the experimental results states that T1TM5430C54X signal processor was suitable for hardware implementation of RSA.

E-commerce is one of the applications of RSA digital signature algorithm. Complexity of large integer operation is main factor that affects the efficiency of RSA system. Ying Yu Cao and Chong Fu had proposed a approach based on '*n*' carry array large integer to speed up the calculation in RSA key generation and data encryption/ decryption process, so as to improve the efficiency of RSA system. By using this random RSA public key pair, arbitrary length can be generated effectively. A 1024 bits RSA key can be generated within 2 minutes on common PC platform while the encryption and decryption operation on data less than 1024 bit can be done in 2 seconds. In this way the efficiency of RSA system was greatly improved which provides guarantees for implementation high security RSA algorithm with long keys on PC platform [5].

For speed up in decryption and signature process of RSA algorithm authors developed a variant RSA algorithm called as EAMRSA - Encrypt Assistant Multi prime RSA [6]. The performance of RSA decryption and digital signature has direct relationship with efficiency of modular exponentiation implementation, reeducation of modulus and private exponentiation for speedup decryption and signature. Experimental result uses various key lengths and shows the new variant in RSA algorithm which not only speedup RSA decryption but also guarantees the security of RSA cryptosystem for 1795 to 3072 bit. In the proposed algorithm the work was divided between server and client to reduce load from weak client. Such as encryption is carried out at server side by sending parameter '*n*', '*d* ' and 'x'and decryption was carried out at weak client. The purpose is to reduce computational burden from decryption to encryption and hence the name EAMRSA. However the new variant RSA reduces the computation load at the decryption and introduces certain computational cost to the encryption.

Sonal Sharma et al. had resolved the problem in Subset Sum Cryptosystem., in which each element in the set is greater than the sum of all the numbers before it. In which, polynomial time with simple greedy algorithm approach was used. So the new algorithm is secure against Mathematical and Brute force attack on RSA. The proposed workwas alsocompared between Modified Subset Sum over RSA Public Key Cryptography (MSSRPKC) and RSA cryptosystem with respect to security and performance. As the size of number increases, the possibility for factoring number decreases and hence security increases [7].

Ayush Chabra et al. had proposed modified version of RSA algorithm. The modification was based on elimination of 'n'which is product of two random big prime numbers.A big number wasused in public key, which was quite difficult for intruder to guess the factor of 'n'and hence the encrypted message remains safe from hackers. This approach uses public key exponent which is known by 'p'and 'q'which are factors of,'n' hence encrypted message is secured and confidential [8].

Xin Zhou et al. had implemented RSA, encryption, decryption solution based on RSA public key, by adopting probabilistic algorithms to generate large prime numbers, where 'p' and 'q'are large prime numbers, with factorization methods. The proposed work secures these keys from tampering, forgery and counterfeiting, and for that the procedure for encryption and decryption had been stated [9].

Xuewen Ton et al. had proposed improved version of RSA for speeding up decryption method. Load transferring technique and multi prime technique in batch algorithm had been implemented. RSA has very high computational cost in compare to its speed. Parallel computing providesgood potential for speeding up [10].The key to $< n, d >$ parallel program is exploitable concurrency. The task requires decomposition of data and how it can be decomposed into distinct chunks. In the BSIPRSA (Batch-RSA-SI Multi Power RSA) data is public key, private key and cipher text. In compare to normal RSA, the public key of BSIPRSA becomes $< N, d_1, \ldots d_k >$ private key becomes $< N, e_i, f_1, \ldots f_k, e\_inv\_p, p^2\_inv\_q >$, that is public key and private key are decomposed into a matrix which are just $< n, d >$ and $< n, e >$ in normal RSA. Every unit in matrix is pair of public and private keys which are operated independently and hence it is possible to parallelize the application by associating each unit with task.

Sami A Nagar et al. had implemented of RSA algorithm during data transmission between different communications networks and Internet. Four level of security has been provided such as low, medium, medium-high and high whose key length (bits) are 512, 1024, 2048 and 4096 respectively also indexes exchange scheme *Nid*, *Eid* and *Did* have been proposed to provide security over attacks [11].

Dhakar et al. had proposed the Modified version of RSA Encryption Algorithm (MREA). As Compare to RSA, MREA is secured one, which is based on the factoring problem it has an additive homomorphism cryptosystem, the encryption is computed using $m_1 + m_2$. The homomorphism properties: "the product of two cipher text will decrypt to the sum of their corresponding plain texts". MREA is also asymmetric key cryptosystem like RSA which uses public key & private key.But, unlike RSA, it is one way, thus it is unusable for authentication by cryptographic signing. The Comparison has been made between RSA and MERAwith respect to security and performance [12].

Neetesh Saxena et al. had proposed Secure Encryption with Digital Signature Approach for SMS. Comparative study has been illustrated of RSA algorithms for digital signature on various applications. Various related work in encryption technique for transmitted messagestudy, they apply digital signature to SMS for security purpose [13].

To improve the security of information using RSA Algorithm, Norihidaya Mohammad et al. had proposed Loop Based Key Generation Algorithm using String Identityuses email-id of user as public key in their key generation process. The proposed algorithm which was improved version of previous work. Now the result of proposed algorithm reaches to 66.6% as compared to previous algorithm result which was 46.67%. The result was improved by using looping process in key generation, the selection of value of p, q which was processed until $k = 1$ and email-id can be key. In this process of algorithm the first step is to generate any two random prime numbers p and q, then n is calculated by using $n = p \times q$. The $\phi(n) = (p-1) \times (q-1)$ is calculated in further step. Then the user provides the email-id as parameter 'e' and gets decimal value of string email-id using CRC hash function. Then test either $GCD(h(id), \phi(n)) = 1 or \neq 1$ If $GCD = 1$ then that email-id can be used as public key. For valid identity user inputs different values of p and q through loop and calculate $\phi(n)$ until k = 1 i.e. $GCD(h(id), \phi(n)) = k = 1$. Then that email–id is considered as public key.For speed up the implementation of RSA algorithm during data transmission across the network, modified version of RSA algorithm was proposed. Since RSAalgorithm is the only powerful algorithm for secure and confidential transfer of data over network. The authors redesigned RSA algorithm in which they used third prime number in order to make a modulus 'n' which made challenging for intruders to decompose 'n'. A database system is used to store the key parameters of RSA algorithm before it start algorithm implementation. Database maintains two separate tables to store keys. One table contains the values of 'p', 'q', 'N1', 'ϕ' and second table contain the values of 'e', 'd', 'r', '$E1$' and '$D1$' . This is offline storage. Because of third prime number it is challenging for intruders to guess the value of modulus 'n' since $n = p \times q \times r$, similarly $E1$ and $D1$ are indexes of public and private key 'e' and 'd' respectively [15].

Rohit Minni et al. had proposed a modified version of RSA algorithm. They eliminated transmission value of 'n'which is product of random prime numbers. In RSA algorithm random numbers are found from large number 'n', then it is easy to find private and public key value and then encryption and decryption of message is possible. The author was proposed a mathematical transformation over 'n' to get 'x'which was replacement of 'n' using which one can't trace back to the factor of 'n' that are 'p' and 'q', this increases security of RSA algorithm by greater extent. The value of public key 'e' is also taken from the range between square root of 'n' and $\phi(n)$ [16]. Author Suli Wang et al. stated that it is possible to use RSA algorithm for file encryption. This developed algorithm encapsulated with 32-bit windows platform so that any window encryption

operation can use RSA algorithm for encryption [17]. Hemant Kumar and Ajit Singh developed a new SRNN algorithm based on RSA algorithm with some modification and included more security. This algorithm overcomes shortcomings of RSA system and achieves high security for digital signature [18].

III. COMPARISON OF DS ALGORITHMS

The concept of public-key cryptography (PKC) which is heart of digital signatures was publicly introduced by Diffie and Hellman [1]. The first practically applicable signature scheme RSA was proposed by Rivest, Shamir and Adleman in 1978 [2]. The RSA scheme since that time stands ahead as most commonly used approach for public key encryption. The security of the RSA algorithm is based totally on the hardness of the factorization of products of two large primes. In 1984, ElGamal announced a public key scheme based on the discrete logarithm problem which was used to propose a new type of digital signature scheme based on it [3]. The ElGamal signature scheme is the first digital-signature scheme which is probabilistic in nature. The Digital Signature Algorithm (DSA) was a variant of the ElGamal digital signature scheme, proposed as a standard by the National Institute of Standards and Technology (NIST) in 1991 [19, 25]. In recent years the key length has increased for secure RSA use which cause heavier processing load on application. A competing system challenges RSA is Elliptic Curve Cryptography (ECC). In 2001, the Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed by Johnson et al.and was again accepted as a digital-signature standard [20]. The attractive point of ECC in compare to RSA is that it offers equal security for a far small key size there by reducing processing cost. ECC is fundamentally more difficult than other. Bernstein et al. in 2011, proposed the Edwards curve digital signature algorithm (EdDSA) [21]. ECDSA and EdDSA derive their security from the apparent intractability of the discrete logarithm problem in elliptic and Edwards curves defined over finite fields. To verify an ElGamal-like signature, one requires two finite-field exponentiations (for DSA) or two scalar multiplications in the underlying curve (for ECDSA and Ed-DSA). Such modular exponentiation or scalar multiplication is considerably more time-consuming than the other finite-field operations. EdDSA verification additionally involves a square-root computation in the finite field. This overhead addresses the need of easy batch verification but incurs significant overhead even during individual verification. We can nevertheless adapt the ECDSA verification algorithm to EdDSA, thereby avoiding the costly square-root computation.

Signature verification is somewhat slower in ElGamal-like signature schemes, than the other signing procedure. Many applications (often real-time) need to verify multiple signatures in batches. In 1994, Fiat A.et al. introduced a method to handle signature batches [22]. In the proposed technique of batch verification where the verifier simultaneously verifies a batch in time less than the total time associated with the individual verification of the signatures. Various interactive batch-verification procedures such as DSA, RSA were introduced by many researchers. In this scheme multiple signatures signed by the same private key can be verified simultaneously. Harn's scheme uses only one exponentiation for batches of any size [23]. However, its drawback was that it does not adapt the case of signatures from multiple signers. The key sizes of ECDSA signatures are much smaller than the key sizes of RSA and DSA signatures at the same security level [24].

With all this overview we have studied various RSA algorithms i.e. improved and modified versions of it and we have portrayed analytical study on improved and modified version of RSA.

IV. FLOW OF RSA

RSA algorithm is divided in three parts: Key generation, Encryption by Public key, and Decryption by Private key. Following are steps for how RSA Algorithm executes.

A. Key Generation

1. Select two prime numbers,
   $p = 17$ and $q = 11$
   Calculate
   $n = p \times q$
   $n = 17 \times 11$
   $n = 187$
2. Calculate
   $\phi(n) = (p-1) \times (q-1)$
   $\phi(n) = (17-1) \times (11-1)$
   $\phi(n) = 16 \times 10 = 160$
3. Select e such that e is relatively prime to $\phi(n)$ and less than $\phi(n)$ i.e. $GCD(e, \phi(n)) = 1$
   We choose $e = 7$
4. Determine d such that
   $d \equiv 1 (\mathrm{mod}\, n)$ and $d < n$
   $d \equiv 1 (\mathrm{mod}\, 160)$ and $d < 160$

d is generated by calculation using extended Eculid's algorithm and satisfying above condition, here $d = 23$
The resulting keys are

Public Key = [7,187]
Private Key = [23,187]
Consider the plain text $M = 88$
Encryption of plain text to cipher text

1. Plain text is $M < n$
2. Cipher text
   $C = M^e \bmod n$
   $C = 88^7 \bmod 187$
   $C = 11$

Now, the generation of plain text is performed by decrypting the cipher text as followed in part B.

 ISSN: 2321-8169
944 - 949

B. Decryption of Cipher to plain text

1. Cipher text is C
2. Plain text

$$M = C^d \bmod n$$
$$M = 11^{23} \bmod 187$$
$$M = 88$$

We get the plain text M

Figure 1. Illustrates the sequence of events for key generation, encryption and decryption [26].

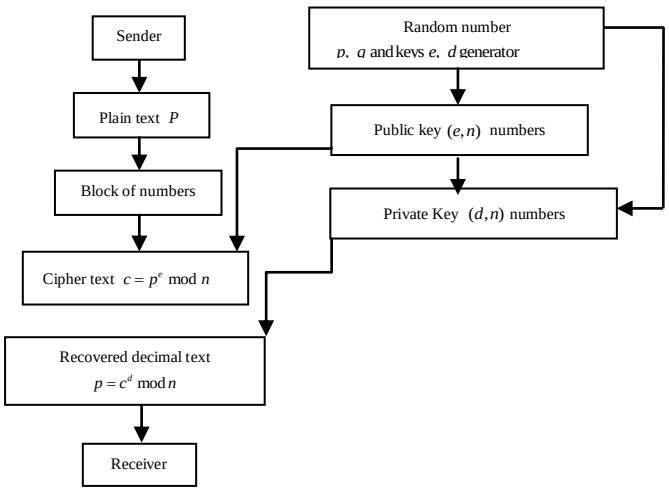

Figure 1. Block diagram of RSA Algorithm

Algorithm of Modified RSA

A. Key Generation

1. Select two prime numbers,
   $p = 11, q = 13$ and r=19
2. Calculate
   $n = p \times q \times r$
   $n = 11 \times 13 \times 19 = 2717$
3. Calculate
   $\phi(n) = (p-1) \times (q-1) \times (r-1)$
   $\phi(n) = (11-1) \times (13-1) \times (19-1)$
   $\phi(n) = 10 \times 12 \times 18 = 2160$
4. Select e such that e is relatively prime to $\phi(n)$ and less than
   $\phi(n)$ i.e. $GCD(e, \phi(n)) = 1$ We choose $e = 53$
5. Determine d such that
   $d \equiv 1 (\bmod n)$ and $d < n$
   $d \equiv 1 (\bmod 2717)$ and $d < 2717$

d is generated by calculation using extended Euclid's algorithm and satisfying above condition, here $d = 1997$

The resulting keys are

Public Key = $[53, 2717]$

Private Key = $[1997, 2717]$

Consider the plain text $M = 'ab'$

B. Encryption of plain text to cipher text

1. Plain text is M < n
2. Cipher text
   $C = M^e \bmod n$
   $C = 1227\ 673$

C. Decryption of Cipher to plain text

1. Cipher text is C
2. Plain text
   $M = C^d \bmod n$
   $M = ab$

We get the plain text = M

Figure 2. Illustrates the sequence of events for key generation, encryption and decryption in modified RSA.

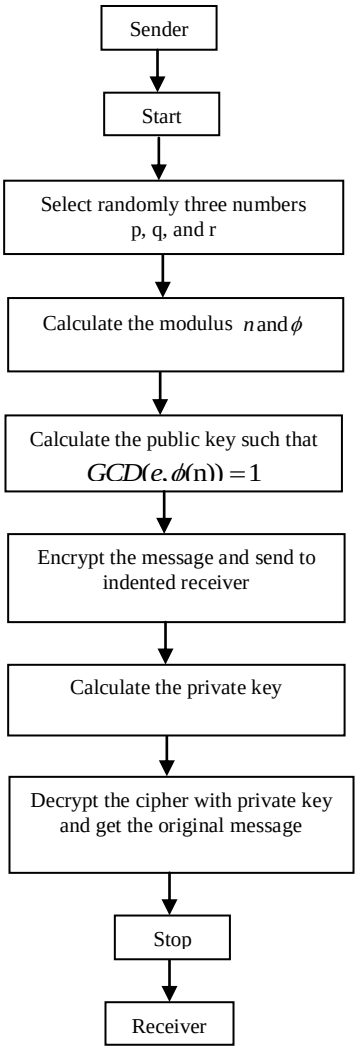

Figure 2.Block diagram of Modified RSA Algorithm

TABLE I. COMPARISION OF RSA AND MODIFIED RSA

Sr No.	Authors	Factors of Modified RSA	Forte / Flaw of RSA
1	RohitMinni et al.	Mathematical transformation over 'n' to get replacement of 'n'.	Trivial rise in time complexity.
2	RituPatidar et al.	Security & Speed is enhanced through offline storage $n = p*q*r$ with three prime numbers.	Require more memory for offline storage.
3	Norhidayah Muhammad et al.	i-RSA has been proposed by using Email-Id & public key generation is about 66.6% as compared to previous one is 46.7%.	Adequate security.
4	Neetesh Saxena, et al.	Variant of ECDSA had been proposed over SMS.	Security level enhanced as compare to ECDSA.
5	Ravi Shankar et al.	Additive homomorphic properties are used over Modified RSA Encryption Algorithm with four random prime numbers i:e (p, q, r and s).	Security level enhanced as compare to RSA.
6	SamiA. Nagar et al.	Four security levels have been proposed i:e Low, Medium, Medium-High, High level.	Key exchange method is used to increase security levels. Where *Nid*, *Eid* and *Did* indexes are used.
7	Xuewen Tan et al.	New Variant of RSA is introduced which is called BS1PRSA is proposed.	Higher decryption as compared to Batch RSA.
8	Aayush Chhabra et al.	Eliminates the need of transfer of 'n', the product of two random numbers, but transfer big prime numbers.	Challenging for intruders to guess factors of n.
9	Sonal Sharma et al.	Modified Subset-Sum over RSA Public Key cryptosystem (MSSRPKC) is introduced against brute-force attack.	Hybrid cryptographic algorithm more secure for mathematical attacks.
10	Yunfei Li et al.	Proposed Variant of RSA cryptosystem (EAMRSA)	Improved the performance by assessing a large number of exp with reduction modules and private exponents $d_{i,j}$, for $1 \le i \le b$ and $1 \le j \le k$
		$Z_{i,j} = C^{e_{i,j}} \bmod N$ cipher text message in vector Z and decryption is done CRT (Chinese Remainder Theorem).	
11	Ying-yu Cao et al.	a, n carry array based for large integer is proposed to swiftness large integer calculation.	A 1024 bits RSA key can be generated within 2 minutes. Encryption & decryption operation can be done within 2 sec.
12	Xin Zhou et al.		

V. CONCLUSION

A digital signature ensures that confidentiality, authenticity, data integrity, and undeniable of information over electronic transaction. For which RSA, Elgamal, ECDSA are promptly used to generate digital signature. This paper provides an analytical study over RSA and Modified RSA algorithms. Where, ECDSA and its variation provide higher security over brute force attacks parallel BS1PRSA also provides higher decryption as compared to Batch RSA.

VI. REFERENCES

[1] Diffie, W., Hellman, M.E.,*New directions in cryptography*, IEEE Transactions on InformationTheory 22 (1976), 644–654.

[2] Rivest, R., Shamir, A., Adleman, L., *A method for obtaining digital signatures and public key cryptosystems*, Communications of the ACM 21 (1978), 120–126.

[3] ElGamal, T, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory 31 (1985), 469–472.

[4] Milan Markovic, Goran d ordevic, TomislavUnkasevic, *On Optimizing RSA Algorithm Implementation on Signal Processor Regarding Asymmetric Private Key Length*, IEEE, April (2003) 73-78.

[5] Ying Yu Cao, Chong Fu, *An efficient Implementation of RSA Digital Signature Algorithm*, 2008 International Conference on Intelligent Computation Technology and Automation, IEEE, May (2008) 100-105.

[6] Yunfei Li, Qing Liu Tong Li, *Design and Implementaion of An Improved RSA Algorithm*, 2010 International Conference on E-Health Networking, Digital Ecosystems and Technologies, IEEE, October (2010) 390-395.

[7] Sonal Sharma, Prashant Sharma, Ravi Shankar Dhakar, *RSA Algorithm Using Modified Subset Sum Cryptosystem*, ICCCT, IEEE, (2011) 457-461.

[8] AyushChabra, SrushtiMathur, *Modified RSA Algorithm: A Secure Approch*, 2011 International Conference on Computational Intelligence and Communication Systems, IEEE, (2011) 545-550.

[9] Xin Zhou, Xiaofei Tang, *Research and Implementation of RSA Algorithm for Encryption and Decryption*, 6th

International Forum on Strategic Technology, IEEE, July (2011) 1118-1121.

[10] Xuewen Tan, Yunfei Li, *Parallel Analysis of an Improved RSA Algorithm*, 2012 International Conference on Computer Science and Electronics Engineering, IEEE, June(2012) 318-320.

[11] Sami A. Nagar, SaadAlshamma, *High Speed Implementation of RSA Algorithm with Modified Keys Exchange*, 6th International Conference on Science of Electronics, Technologies of Information and Telecommunications(SETIT), (2012).

[12] Ravi Shankar Dhakar, Amit Kumar Gupta, Prashant Sharma, *Modified RSA Encryption Algorithm (MREA)*, 2012 Second International Conference on Advanced Computing & Communication Technologies, IEEE, (2012) 426-429.

[13] NeeteshSaxena, Narendra S. Chaudhari, *Secure Encryption with Digital Signature Approch for Short Message Services (SMS)*, IEEE, Augest(2012) 803-806.

[14] Norhidayah Muhammad, Jasni Mohammad Zani, MdYazidMohdSaman, *Loop Based RSA Key generation using String Identity*, 13 th International on Control, Automationand System (ICCAS), ICROS, (2013)255-258.

[15] RituPatidar, RupaliBhartiya, *Modified RSA Cryptosystem Based on Offline Storage and Prime Number*, IEEE, February (2013).

[16] RohitMinni, KaushalSultania, Saurabh Mishra, *An Algorithm to Enhance Security in RSA*, 4th ICCCNT 2013, IEEE, July (2013).

[17] Suli Wang, Ganlai Liu, *File Encryption and Decryption System based on RSA Algorithm*, 2011 International Conference on Computational and Information Sciences, IEEE, January (2011).

[18] Hemant Kumar, Dr.Ajit Singh, *An Efficient Implementation of Digital Signature Algorithm with SRNN Public KeyCryptography*, IJRREST, June (2012).

[19] NIST, The *digital signature standard*, Communications of the ACM 35(7) (1992), 36–40.

[20] Johnson D., Menezes A., Vanstone S.A, *The elliptic curve digital signature algorithm (ECDSA)*, Int. J. Inf. Sec. 1(1) (2001), 36–63.

[21] Bernstein D.J., Duif N., Lange T., Schwabe P., Yang B.Y,: *High-speed high-security signatures*, Journal of Cryptographic Engineering 2(2) (2012), 77–89.

[22] Fiat A, *Batch RSA*, Journal of Cryptology 10 (1997), 75–88.

[23] Harn, L.: Batch verifying multiple RSA digital signatures. Electronics Letters 34(12) (1998), 1219–1220.

[24] Hwang M.S., Lin I.C., Hwang K.F.: *Cryptanalysis of the batch verifying multiple RSA digital signatures*. Informatica 11(1) (2000), 15–19.

[25] Naccache D., M'Raihi D., Rapheali D., Vaudenay S, *Can DSA be improved: Complexity trade-offs with the Digital Signature Standard. In*, Eurocrypt '94, Spinger (1994), 85–94.

[26] Willam Stallings, *Cryptography and Network Security*, fifth edition, (2012).