Enhancing RSA Security Capability Using Public Key Modification

Sarjiyus, O.

Department of Computer Science, Adamawa State University, Mubi, Nigeria

***Abstract*: This research, Enhancing RSA Security Capability Using Public Key Modification is basically a modification of the public key parameter in the existing RSA in order to enhance the security and performance of the algorithm, all in an effort to secure the confidentiality, authenticity and integrity of sensitive data during internet transactions. The purpose is to design an RSA technique that accords faster key generation and raises security to a more secure, higher level ahead of the existing RSA technique by the 'e' to 'f' transformation. Factorization attacks are very common and have a devastating effect on RSA by compromising the privacy of data. These have been overcome in the improved RSA algorithm by choosing two prime numbers, p and q so large enough to make it hardly possible for attackers to factor out the components of the modulus n, and freely, get the exact value of the totient function, Φ (n) for which they can possibly, easily use to deduce the value of the private key'd'. For the performance parameter, inputs were captured primarily from the prime numbers p and q used to define the modulus n, the public key value, e; while the outputs range from designated data sizes in (kb), encryption and decryption times (in Secs) for the two algorithms. The result show that in the context of security, the improved RSA technique based on public key transformation produced more complex ciphers than the existing RSA technique during encryption process thereby enhancing the security of the modified RSA. For performance, the improved RSA shows a slight increase in time complexity in the encryption process, but not in decryption. Hence, the new RSA technique is most suitably used for systems desiring high security but less speed of execution.**

***Keywords:* Key generation, Private Key, Public Key, Security, Sensitive Data, Technique.**

I. INTRODUCTION

Globally, the emphasis is on the use of cryptographic techniques to secure the transmission of vital information through insecure channels. It has been revealed that cryptography is one technique that is used to accomplish such purpose by ensuring data secrecy/confidentiality, authenticity and integrity.

However, the data transferred through network channels faces some threats all in a bid to uncover the secret information being transmitted. Developing more information technology platforms often results in raising the level of Cyber threats and vulnerabilities of the data in insecure channels[1]. To curb these problems, researchers are developing various techniques to discover unconventional alternatives to improve the security of transmitted data by ensuring data gets to the receiver side untempered, unaltered. At present various improved algorithms have been simulated and are known to provide much better security; but such sophisticated algorithms could be very expensive and known to consume much computational resources. As it is known, cryptography is the procedure used to protect sensitive data over a conventional network channel, which may not be necessarily secure being a public channel, and the other side receives the message as it is, unaltered. Nowadays, the subject of data confidentiality becomes a very crucial issue in information security. Easy administration of the Internet today and data globally led to the importance of data security even though, its emergence has created new dangers for users who want their data to remain safe. As it is, hackers are using a diversity of techniques to penetrate and break into information transferred through the channel and steal the information or alter the original data content [2],[3].

Nowadays, cryptography algorithms afford a high level of confidentiality by concealing private data of any individual or group. Many of the ongoing researches aim at finding out the new cryptographic algorithms that are more efficient based on security and complexity [2]. Again, one secret-key cryptography algorithm could be generated and used for both approaches which is known as asymmetric key. This technique proved that it has lesser computational efforts but unfortunately has many drawbacks such as key management issues, which can lead to the tendency of the private key becoming compromised [4],[5].

Asymmetric cryptography algorithm uses a public-key to achieve encryption pattern and then use a private key to decrypt the ciphered information. However, asymmetric cryptography implements two different mathematical approaches, like a public key and the other, a private key. Dissimilar symmetric algorithms applied only one key to both the encryption/decryption approach [6], [7]. Moreover, the public key is stored openly and can be used to encrypt data by anyone. On the other hand, the private key is kept secret and implemented by the receiver side to decrypt the received encrypted data.

Rivest, Shamir and Adleman (RSA) in 1977 were the first to describe the algorithm that implements the public key [2]. RSA algorithm alied different keys as public/private keys but are related to a large scale of alications. As a result, reliably secured results and better security transfer of data are big prime integer numbers chosen for both the public and private keys [6], [7]. RSA alied extensively for encryption/decryption

problems leading to a protected transmission of the data. RSA technique tends to have more enhanced protection when the value of the key is big enough and it becomes much more difficult to figure out its common factors. An asymmetric key means that it works on two different keys as public key and a private key. The public key should be known by everyone but the private key is not and it is kept privately to the receiver only. The objective of RSA is founded on the fact that it should be hard to factorize a key because a large prime integer is chosen. This research therefore, focuses on designing an improved RSA technique which has the ability to fasten key generation while at the same time raises security to a second higher step ahead of the existing RSA algorithm as a result of the transformation of the public key exponent e, to f= ((e*2) +1) to achieve better, higher and stronger security and the use of very large prime numbers to battle factorization attacks.

II. REVIEW OF RELATED LITERATURE

Cryptography can be viewed as the science of devising methods, techniques and practices that allow information to be sent in a secure form, through a communications channel in such a way that the only person that is able to retrieve this information is the intended recipient. It is the practice and study of hiding information. In modern times, cryptography is considered a branch of both mathematics and computer science, and also closely affiliated with information theory, Computer security and engineering [8]. Cryptography plays a vital role in most applications used in technologically advanced societies; examples include the security of ATM cards, computer passwords, and electronic commerce which all depend on cryptography [8]. In other words, Cryptography is the process of encoding messages in order to make them non-readable for the purpose of achieving security [9].

[10] Presents an insight into the basics of cryptography and different types of cryptography. It describes the RSA technique and explores its intricacies and characteristics. It goes further to illustrate in general all the variants of the RSA algorithm and presents various comparisons among them on the basis of complexity, security and certain other parameters. It portrays that much work could be carried out on RSA algorithm. More specifically, the work discussed in the thesis focuses on the implementation of multi-prime and multi-power RSA on 2048-bits. The work proposed the implementation of a combination of these two algorithms as the proper way of securing data in the cloud. The snag of this work has to do with flows in security, where the keys generated are weak which makes the algorithm vulnerable to certain threats and the algorithms implemented separately.

[11] Proposed Multi power RSA with $N = P^m * Q$ using the Chinese Remainder Theorem to make the decryption process faster and in a more secure way. The proposed Multi power RSACRT with $N = P^m * Q$ is described as taking less execution time compared to most other RSA algorithms. The proposed algorithm also gives a better performance at the cost of a small decrease in decryption time. Beside all it provides the semantic security to the system which is not provided by Multi prime RSA algorithm. The algorithm was only proposed but not implemented. The main problem of this algorithm is that the exponent can be predicted by a hacker and as such, can easily find the prime numbers used in key generation.

[12] Proposed the use of Fermat's little theorem with RSA algorithm. The fermat's little theorem is used during key generation in order to increase the speed. The main problem of the conventional RSA algorithm is that it is very slow when it comes to key generation; as large key size numbers are selected, it increases key generation time. This problem can be solved by alying Fermat's little theorem during key generation process. This method helps to trust users in cloud computing environment.

[13] Proposed a Multi-prime RSA algorithm which was implemented in the middle layer before the data is stored in the cloud. When an authorized user requests the data, the data is decrypted and provided to the user. In client phase, the client sends the query to the server. Depends on the query the server responds to the client with the corresponding file. Before this process, the client authorization step is involved. In the server side, it checks the client name and password for security process. If satisfied, the queries are received from the client and the corresponding files are searched in the database. Finally, the corresponding file is retrieved and sent to the client. The shortcoming of this algorithm has to do with its vulnerability to brute force attack [9]. Proposed the design of hybrid cryptographic algorithm that enhances data security in cloud computing. This algorithm was designed using a combination of two modern cryptography algorithm; multi-prime RSA and MD5 algorithms. In the proposed algorithm, multi-prime RSA is used for encrypting data while the MD5 is used to generate a message digest and attach a digital signature concurrently. The proposed algorithm was found to be efficient in terms of speed, and required less memory space compared to most existing algorithms. It was also found to have the ability to resist some threats surrounding cryptographic algorithms. The major snag of this algorithm is that it can only solve problems of data security in the internet and cannot be applied to solve other digital communication problems.

In the study presented by Sahu J. et al. [14], a better version of RSA algorithm with enhanced security was revealed. The study focused on the elimination of n from the original RSA algorithm and the introduction of a new number f in place of n. The replacement of n that is, f is used in both private and public key generation. The RSA algorithm is prone to mathematical factorization attacks. Since n has been replaced with f, it is very hard to factorize it and get the original prime numbers p and q. This makes the algorithm more secure but with a slight increase in time complexity.

The research of Ivy P., Mandiwa P. and Kunar M [15], illustrated the use of 'n' prime numbers which provided the security over the networks from where the quality that makes

it easier for the cryptography to have a good use of 'n' prime numbers. The 'n' prime numbers play an important role in the RSA cryptosystem to develop the RSA algorithm for 'n' prime numbers and also used four prime numbers. In this paper no qualitative measurements were carried out; it only shows the process involved. The study of [16], revealed a method for implementing a public-key cryptosystem (RSA) using two public keys and some mathematical relation. The two public keys are sent separately making the attacker not to get much knowledge about the key and so, make the attacker unable to decrypt the message. The only constraint here is that the proposed RSA is only used in systems where high security with less speed is required [16].

The research conducted by [17] is a study of number theory and public key cryptosystems and based on this, improving the RSA cryptosystem that is more reliable in curbing brute force attack.RSA cryptosystem produces one public key to encrypt the message. Though it is hard to find out the component factors of n being p and q, two large prime numbers, therefore brute force attack is more difficult in the proposed algorithm as the encryption keys are sent separately, not at once. However, the proposed RSA is restricted for use only in systems that need high security but with less speed [17].

III. METHODOLOGY

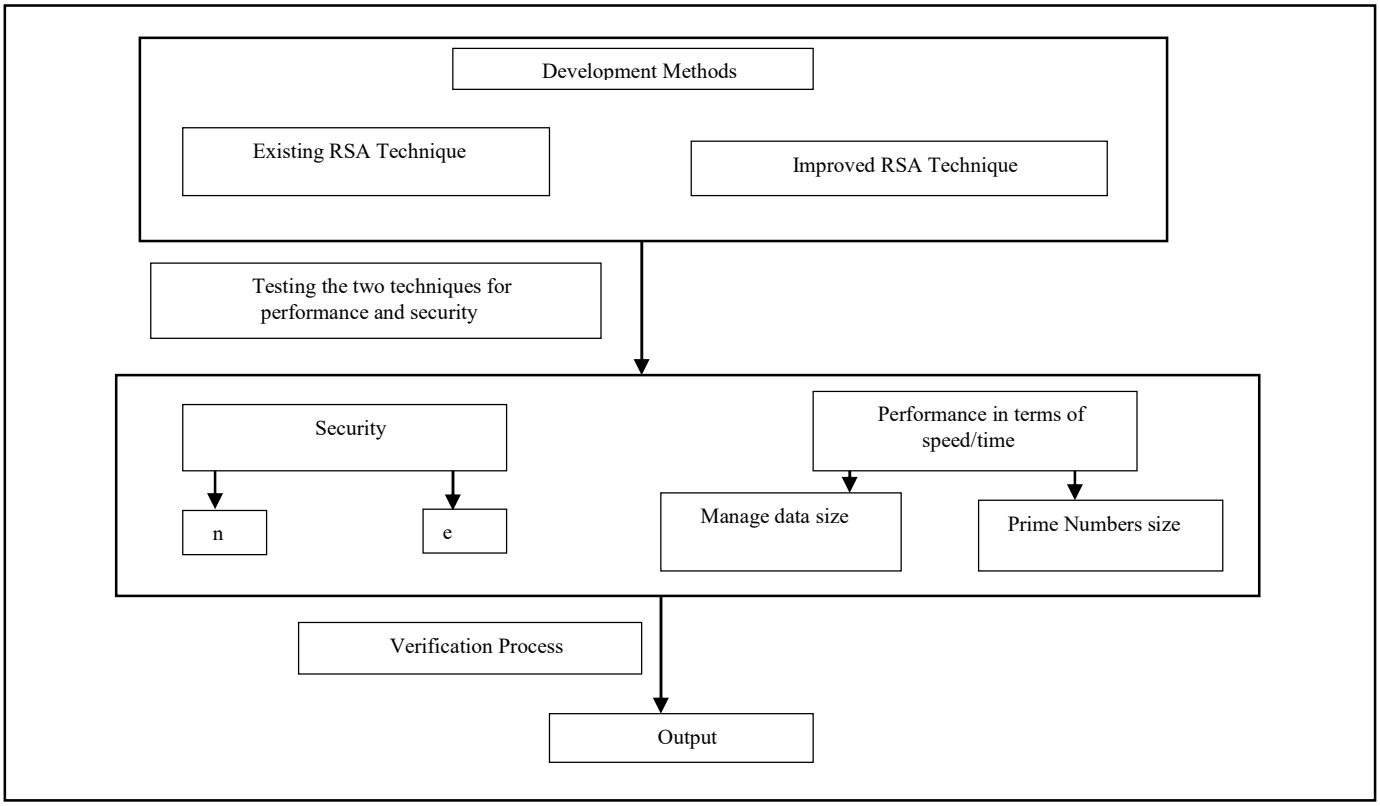

Figure 1: Methodology for the developmental process

The exisiting RSA technique and the Improved RSA technique were comparatively developed using the JAVA platform. An improved key generation for the RSA technique was targetted at improving the security of the technique.

The methodology for modifying the public key equation was been used during the encryption process while also extending to the decryption process. The eperiment was carried out using small prime numbersfor simplicity and clarity in the computation process. Testing the two techniques makes for a variation of results of the modulus value, *n* and the pulic value *e* . the two techniques are tested using the same hardware and software environment and specifications. Considering the hardware specification, AMD processor with 1.5 GHz and RAM 4GB was used while the software specification used was 64-bit windows 10 Operaing System. Figure 1 is an illustration of the stepwise procedure of the techniques to implimentation stage.

3.1 Existing RSA algorithm

A. Key Generation

1. Select two large prime numbers p and q.
2. Compute n =p*q.
3. Compute the totient function Φ(n) =(p-1)*(q-1)
4. Collect e with the following condition { p> e >n, coprime n} i.e (e, n) =1 and [e,(Φ(n)] = 1
5. Select random 'e' from list

6. Compute d from the relation {de mod Φ(n) = 1}
7. Send Public key (e, n)
8. Send Private key (d, n)

B. Encryption

Encrypting a plaintext, M using the public key functionality e,

$$C = M^e \bmod(n)$$

C. Decryption

The encrypted text is decrypted using the private key d as,

$$D = C^d \bmod(n)$$

3.2 RSA modification algorithm based on public key

A. Key Generation

1. Select large prime numbers p and q.
2. Compute n =p*q.
3. Compute n =(p-1)*(q-1)
4. Collect e with the following condition { p> e >n, coprime n} i.e (e, n) =1 and [e,(Φ(n)] = 1
5. Select random 'e' from list
6. Calculate f=(e*2)+1.
7. Select d with the following condition {de mod n = 1}
8. Send Public key (f, n)

The 'f' serve as a new public key which will hide the original e value.

9. Send Private key (d, n)

B. Encryption

$$C = M^{((e*2)+1)} \bmod (n)$$

C. Decryption

$$D = C^d \bmod (n)$$

3.3 Illustration:

1. Get two prime numbers p and q
   For the prime number p = 11, and the prime number q = 7
2. Compute the modulus n = p*q, which implies n = 11*7 = 77
3. Compute the totient function Φ(n) = (p-1)*(q-1), Φ(n) = (11-1)*(7-1)
   = 10*6 = 60
4. Select e such that, {p>e>Φ(n), coprime} and gcd(e, Φ(n)) = 1; e=23
5. Compute f = ((e*2)+1), f = ((23*2)+1) = 46+1 = 47
6. Compute d using the relation {de mod Φ(n) = 1}
   d*e ≡ 1(mod Φ(n)) ……..(eqni)
   d*23 = 1(mod 60)
   hence, using the extended Euclidean algorithm,
   d = 47
7. Send the public key (f, n), (23, 77)
8. Send the private key (d, n), (47, 77)

The process of encryption and decryption can be readily performed by sending a message, 'Banking Security' from the sender to the receiver by encrypting it using the public key e and on getting to the receiver end, the message can then be decrypted using the private key, d.

A sample implementation is given below:

Banking Security

ASCII representation of the message is given as

66	97	110	107	105	110	103	32
83	101	99	117	114	105	116	121

The message is encrypted as

74	41	20	12	33	20	45	65
67	76	7	31	1	33	30	64

At the receiving end, it is decrypted back to

66	97	110	107	105	110	103	32
83	101	99	117	114	105	116	121

Giving back the decrypted message as Banking Security

IV. RESULTS

4.1 Results for the public key encryption: The tests shown in table 1 and 2, was conducted to generate several values of private key *d* for varying, corresponding public key funtionalities *e* and message, *m*.

The result of encryption of the existing RSA technique and that of the improved RSA technique is based on the relationship;

$C = M^e \bmod n$ and *e* transformation to *f* as

$C = M^{(e*2)+1} \bmod n$.

Table 1: Cipher Text result of text with varying values of *e* for P = 5, Q = 11 for the existing RSA

			cipher	
e	d	m=4	m=5	m=6
7	23	49	25	41
13	37	9	15	51
17	33	49	25	41
19	19	14	20	46
21	21	4	5	6

Table 2: Cipher Text Result of text with varying values of Public key *e* for P = 5 and q = 11 obtained from existing RSA and improved RSA techniques

Cipher				
e(existing RSA)	e(Improved RSA	d	existing RSA	Improved RSA
7	15	23	49	34
13	27	37	9	49
17	35	33	49	34
19	39	19	14	14
21	43	21	4	9

4.2 Result for comparative Performance in terms of speed/time for the two algorithms:

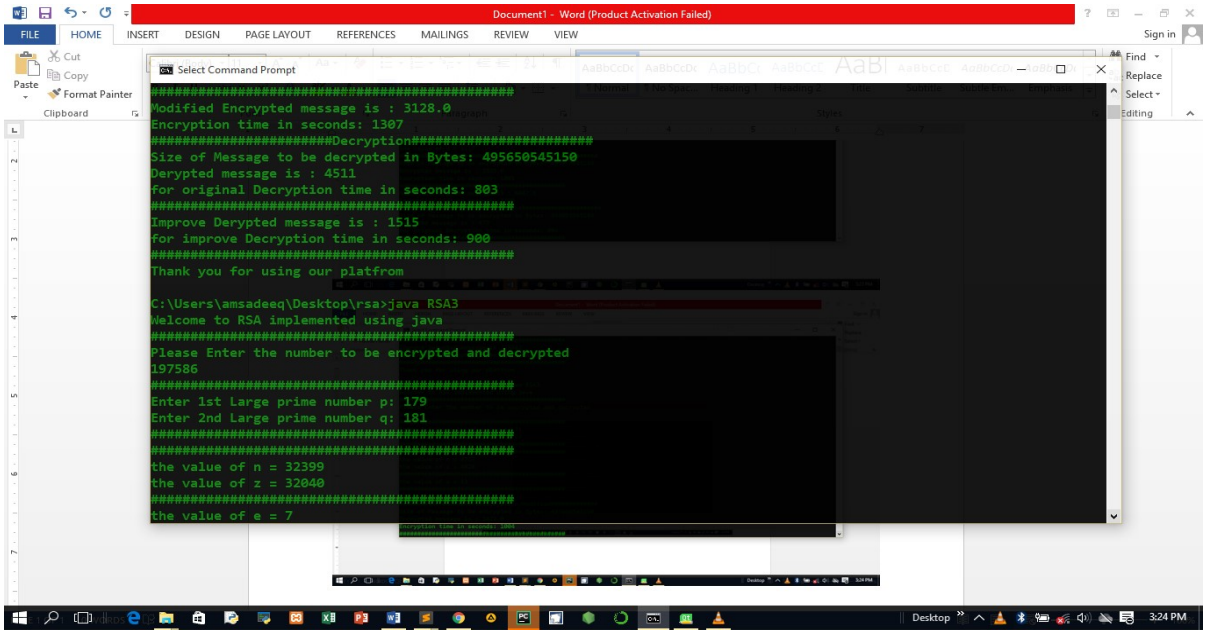

Figure 2: Interface showing the existing and the improved RSA in comparative terms with p = 179, q = 181

The interface is for the various input parameters and their corresponding results. The input range from prime numbers *p, q* use to define the modulus *n*, the public key value *e* for the existing RSA technique and f = (e*2)+1 for the improved RSA technique with *m* representing the plain text or message to be encrypted. The expected output range from designated data sizes in kilobytes (kb), encryption and decryption time in seconds (sec.) for the exsiting and improved RSA technique respectively.

Table 3 and table 4 are a display of the performance (in terms of time) of the exisiting RSA technique and the improved RSA technique all implimented on AMD processor, 1.5 GHz, 4 GB and with 64-bit windows 10 OS.

Table 3: Encryption time (Existing VS Improved RSA)

	Encryption	
	Time in Second/s	
Size (KB)	Existing	Improved
51.31	8.06	13.03
483.67	8.72	13.51
512.88	10.50	14.00
4835.41	11.81	14.75
5036.62	13.90	15.84
48335.61	14.42	17.23
50327.43	16.51	19.23
581528.72	20.83	25.02

Table 4: Decryption time (Existing VS Improved RSA)

	Decryption	
	Time in Second/s	
Size (KB)	Existing	Improved
51.31	8.09	8.09
483.67	8.74	8.74
512.88	9.21	9.21
4835.41	11.84	11.84
5036.62	12.01	13.03
48335.61	13.35	15.54
50327.43	16.01	16.93
581528.72	17.06	18.22

V. DISCUSSION

For this study, the public key *e* functionalities of the existing RSA technique has been tempered with deliberately, transforming *e* to f = (e*2)+1 so as to improve the security of the existing RSA technique by adding a second layer to it. It was however revealed that the improved RSA technique yielded more complex cipher values for each public key *e* as shown in table 1 and table 2.

Furthermore, the existing RSA and the improved RSA techniques were tested against many parameters. The two techniques were assessed in terms of various file sizes of messages and recorded in terms of the efficiency and recorded

in terms of encryption and decryption times in seconds. It was revealed that the improved RSA technique has a higher complexity in terms of encryption time than the existing RSA technique as shown in table 3 and table 4.

VI. CONCLUSION

The research focuses on the improvement of the existing RSA algorithm for better security and performance in terms the speed of key generation, encryption process among others. After testing, it was revealed that the improved RSA algorithm tends to be more reliable in terms of security of data against the problems of confidentiality, integrity and authenticity leakages, but with a slightly increased time complexity in execution of the encryption process when compared to the existing RSA technique. In the future, other parameters such as memory space utilization of the two algorithms could be looked into in order to determine the most efficient in that aspect.

REFERENCES

[1] Obaid, T. A. S. (2020). "Study a Public Key in RSA Algorithim". European Journal of Engineering Research and Science. 5(4), 396-397.

[2] Jamgekar R. S., Joshi G.S., (2013), "File Encryption and Decryption Using Secure RSA," *International Journal of Emerging Science and Engineering (IJESE)*. 1(4) ISSN:2319–6378.

[3] Khyoon, A. I. (2005) "Modification on the Algorithm of RSA Cryptography System," *Al-Fatih Journal.* 1(24), 80-89. ISSN: 87521996.

[4] Obaid T. A. S. Khami M. and Shehab L. G. (2017), "Hiding Secured key in digital media", *Int. Jo. Eng. Res. A.* www.ijera.com 7(9), 58-63.

[5] Nisha S., and Farik M., (2017). "RSA Public Key Cryptography Algorithm – A Review", *International Journal of Scientific & Technology Research*. 6(7). ISSN 2277-8616.

[6] Al-Lehiebe A., (2015), "Ciphered Text Hiding in an Image using RSA algorithm", *Journal of College of Education for Women.* 26(3).

[7] Cid C. (2019), "Cryptanalysis of RSA: A Survey", *SANS Institute. International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, DOI: 10.1109/ICCUBEA .8463720, Publsher:IEEE.

[8] Gupta S. M. D and Ritu G. (2017) "Mobile Cloud Computing: A Scientometric Assessment of Global Publications Output during 2007-16". *Journal of Scientometric Res.* 6(3):186-194.

[9] Ismail A. and Rashid H. (2017) "Performance Analysis of Multi-Level Algorithm For Data Storage Security In Cloud Computing". *World Journal of Engineering Research and Technology WJERT* 3(5), 480-487. ISSN 2454-695X

[10] Zareen (2011) "*Enhancement on Implementation of Multi-prime and Multi-power RSA Algorithm*" An M.Sc Thesis Submitted to the Department of Computer Science and Engineering Department, Thapar University, Patiala.

[11] Padmavathama M. and Sreedevi (2017) "New Variant Digital Signature Schemes based on Jk-RSA Cryptosystem" *International Journal of Artificial Intelligence and Computational Research* 2009 1(2) .95- 100 ISSN.0973-6794 0.559

[12] Mohsen B., Sharifah M, Ramlan M., Zurina M. (2014) "Comparison of ECC and RSA Algorithm in Resource Constrained Devices" Department of Computer Science Faculty of Computer Science and Information Technology Universiti Putra Malaysia.

[13] Ranganathan N. K. (2014). An Implementation of Multi-Prime RSA Algorithm in Data Cloud using Cloud Sql. In NCDMA – 2014, IJERT Conference Processding, Volume 2, Issue 15.

[14] Sahu J., Singh V., Sahu V. and Chopra A. (2017) "An Enhanced Version of RSA to Increase the Security". *Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org*. 7(4), 1-2.

[15] Ivy P. and Mandiwa P. & Kumar M. (2012) "A Modified RSA Cryptosystem Based on 'n' Prime Numbers". *IJECT.* 1(2), 63-66.

[16] Ayele A. and Sreenivasarao V. (2013) "A Modified RSA Encryption Technique Based on Multiple public keys". *International Journal of Innovative Research in Computer and Communications Engineering* 1(4), 859-900

[17] Jahan I, Asif M. and Rozario L. J. (2015). "Improved RSA cryptosystem based on the study of number theory and public key cryptosystems". *American Journal of Engineering Research (AJER)*. 4(1),143-149. e-ISSN: 2320-0847 p-ISSN: 2320-0936.