# A Hybrid Encryption Algorithm Based on RSA Small-e and Efficient-RSA for Cloud Computing Environments

Faraz Fatemi Moghaddam, Maen T. Alrashdan, and Omidreza Karimi

***Abstract*—Cloud computing is an emerging technology that is still unclear to many security problems. Ensuring the security ofstored data in cloud servers is one of the most challenging issues in such environments. Accordingly, this schema presents a Hybrid Encryption algorithm that is based on RSA Small-e and Efficient RSA (HE-RSA) for improving the reliability in cloud computing environments. The main aim of this project is to use the cryptography concepts in cloud computing communications and to increase the security of encrypted data in cloud servers with the least consumption of time and cost at the both of encryption and decryption processes.In the proposed model, the number of key generation exponents has been increased in comparison to the original RSA. Moreover,a dual encryption process has been applied in this algorithm to prevent common attacks against RSA algorithm. In addition, this schema also presents the comparison between HE-RSA and original RSA in terms of security and performance. For this purpose, Key Generation, Encryption and Decryption Time in Original RSA and HE-RSA have been compared according to the different size of exponents. Moreover, some of the common attacks against RSA algorithm have been analysed to detect the resistance of the proposed algorithm against possible attacks. Finally, the effects of using HE-RSA on cloud computing environments have been reviewed in more depth and details.**

***Index Terms*—Cloud computing, security, cryptography, RSA, sharing.**

## I. Introduction

Cloud computing is a newfound technology that uses the concepts of storage, virtualization, connectivity and processing power to store resources such as networks, servers, storage, applications, as well as services and share them between computers and other devices via the internet and the central remote servers [1]. Ensuring the security of stored data is the most challenging issue in the cloud environments. This security has been divided to several parts and one of the most important parts is maintaining security in the servers of cloud computing service providers [2]. Therefore, applying a cryptographic method for authorized user is the most popular existing solutions for solving security issues and increasing the reliability of the cloud environments. According to importance of sharing concepts in cloud computing, Symmetric-key encryption algorithms may not be appropriate in these environments because the security of encrypted data could and in most cases would be affected considerably due to the secret key sharing in user's communications. According to this, Asymmetric-key (public key) cryptography algorithms have been suggested by some researchers [3]-[5] for encrypting data in cloud servers and RSA scheme is the most practical and popular Asymmetric-key encryption method when compared with the other schemes that can be utilized for both encryption and digital signature schemes [6].

Menuscript received February 25, 2013; revised April 28, 2013. This work was supported in part by the Asia Pacific University of Technology and Innovation (A.P.U.), Technology Park Malaysia, Bukit Jalil, 57000 Kuala Lumpur, Malaysia.

Faraz Fatemi Moghaddam and Omidreza Karimi are with the Faculty of Computer Science, Staffordshire University and Asia Pacific University of Technology and Innovation (A.P.U.), Technology Park Malaysia (T.P.M.), Bukit Jalil, 57000, Kuala Lumpur, Malaysia (e-mail: f.fatemi@hotmail.com, omidreza@me.com).

Maen T. Alrashdan is with the Faculty of Research Coordinator, Asia Pacific University of Technology and Innovation (A.P.U.), Technology Park Malaysia (T.P.M.), Bukit Jalil, 57000, Kuala Lumpur, Malaysia (Email: dr.maen@apu.edu.my).

## II. RSA Cryptosystem

The original RSA algorithm was publicly described in 1977 [7] and after that many similar algorithms were designed based of original RSA in order to rectify the weaknesses of the basic algorithm. The Original RSA scheme is as follows:

### A. Key Generation Algorithm

1) Randomly and secretly choose two large primes: $p, q$and compute$n = p.q$
2) Compute $\varphi(n) = (p-1)(q-1)$.
3) Select Random Integer: $e$ such as $1 < e < n$ and $\gcd(e, \varphi) = 1$.
4) Compute $d$ such as $e.d \equiv 1 \, mod \, \varphi(n)$ and $1 < d < \varphi(n)$.
5) Public Key: $(e, n)$
6) Private Key: $(d, n)$.

### B. Encryption Process

1) Suppose entity $A$ needs to send message $m$to entity $B$ (represent $m$ as an integer in the range of$0 < m < n$).
2) Entity $B$ should send his public key to entity $A$.
3) Entity $A$ will encrypt $m$ as $c = m^e mod \, n$ and will send $c$ to entity $B$.

### C. Decryption Process

1) Entity $B$ will decrypt the received message as $m = c^d mod \, n$.

The most important advantage of RSA is ensuring about

the privacy of the private key because this key will not be transmitted or revealed to another user. However, this algorithm has some considerable weaknesses. The main computational costs of the RSA are the modular exponentiations found during the key generation, encryption and decryption process [8]. Moreover, this algorithm has some weaknesses against certain attacks (i.e., Brute force, Mathematical attacks, Timing attacks and Chosen Cipher-text attacks) [9]. To reduce these problems, many algorithms have been designed and introduced based on original RSA. RSA Small-e and Efficient RSA are only two of the most popular algorithms identified for improving the main algorithm.

## III. RSA SMALL-$e$

RSA Small-e is an algorithm based on original RSA except that the public exponent is much smaller than$\varphi(n)$. Using the above mentioned algorithm, the encryption cost is significantly decreased and it also reduces the process to only a few modular multiplications [8]. RSA Small-e is considered to be safe for small public components in special conditions such as using a proper padding scheme and the prevention of Side-channel attacks on the private key [10]. In this algorithm, it is expected with high probability that the private exponent will be the same size as $\varphi(n)$. For more information see Boneh and Durfee [11].

## IV. EFFICIENT RSA

Efficient RSA was introduced as a scheme that employs the general linear group of order $h$ with values that was intentionally selected randomly from the ring of integer $mod\ n$ [6]. The key range of efficient RSA is considerably significant and can actually be used with Hill cipher method to obtain more intractable system. The difference of original RSA and Efficient RSA is on how to calculate $\varphi(n)$ in key generation process. In efficient RSA, $\varphi(n)$ was defined as follows:

$$\varphi(n,h) = (p^h - p^0)(p^h - p^1)\dots(p^h - p^{h-1}) + (q^h - q^0)(q^h - q^1)\dots(q^h - q^{h-1})$$

The other parts of key generation, encryption and decryption process is exactly the same as the original RSA. For more information see Aboud *et al.* [6].

## V. THE PROPOSED SCHEMA

The proposed schema will be introduced in this paper as Hybrid Encryption based on the strengths of Efficient RSA and RSA Small-e (HE-RSA). In this algorithm, the number of exponents will be increased to 3. The steps of HE-RSA are as follows:

### A. Key Generation Algorithm

1) Randomly and secretly choose two large primes: $p, q$ and compute $n = p.q$
2) Compute $\varphi(n) = (p-1)(q-1)$.
3) Compute
$$\gamma(n,h) = (p^h - p^0)(p^h - p^1)\dots(p^h - p^{h-1}) + (q^h - q^0)(q^h - q^1)\dots(q^h - q^{h-1}).$$
4) Select Random Integer: $r$ such as $1 < r < n$ and $\gcd(r,\varphi) = 1$ and $\gcd(r,\gamma) = 1$ ($r$ should be a small integer).
5) Compute $e$ such as $r.e \equiv 1\ mod\ \varphi(n)$ and $1 < e < \varphi(n)$.
6) Compute $d$ such as $d.e \equiv 1\ mod\ \gamma(n)$ and $1 < d < \gamma(n)$.
7) Public Key: $(e, n)$.
8) Private Key: $(r, d, n)$.

### B. Encryption Process

1) Suppose entity $A$ needs to send message $m$to entity $B$ (represent $m$ as an integer in the range of $0 < M < n$).
2) Entity $B$ should send his public key to entity $A$.
3) Entity $A$ will encrypt $m$ as :
$$c = ((m^e\ mod\ n)^e\ mod\ n)$$
After that Entity $A$ will send $c$ to entity $B$.

### C. Decryption Process

1) Entity $B$ will decrypt the received message as: $m = ((c^r\ mod\ n)^d\ mod\ n)$.

HE-RSA has used an additional exponent for increasing the security of the original RSA. In this algorithm, $r$ (as third exponent) has been defined according to the RSA small-e. This means, $r$ in HE-RSA has the same role of $e$ in RSA Small-e and has been defined as a small integer. As it was described earlier, it is expected that the exponent $e$ in HE-RSA has been the same size as $\varphi(n)$. Furthermore, exponent $d$ has been computed according to the Efficient RSA algorithm and the value of $\gamma(n,h)$. nevertheless, in the proposed schema, the encryption process has been carried out by two consecutive steps but with the very same exponents while the exponents for decrypting the cipher text have been different. The security of proposed algorithm has been affected due to this difference.

## VI. ANALYSIS OF PROPOSED ALGORITHM BY COMPARISON OF RSA AND HE-RSA

The simulation result of the proposed algorithm (HE-RSA) and the original RSA was implemented in C# and according to a 2.40 GHz Intel® Core ™ i5 CPU and 4.00 GB RAM. Moreover, the message that was used during the analysis procedure had 1000 characters. The comparison between original RSA and HE-RSA were carried out according to the

different size of exponents (256, 512, 1024 and 2048 bits). The simulation results between Original RSA and HE-RSA has been shownbelow in Table I and II:

TABLE I: EFFECT OF CHANGING THE EXPONENT SIZES FROM 256 (BITS) TO 2048 (BITS) ON KEY GENERATION, ENCRYPTION AND DECRYPTION TIME IN ORIGINAL RSA ALGORITHM

| Size of Exponents (bits) | Key Generation Time (ms) | Encryption Time (ms) | Decryption Time (ms) | Total Execution Time (ms) |
|---|---|---|---|---|
| **256** | 97 | 79 | 73 | 249 |
| **512** | 112 | 98 | 102 | 312 |
| **1024** | 296 | 191 | 189 | 676 |
| **2048** | 1017 | 756 | 723 | 2496 |

TABLE II: EFFECT OF CHANGING THE EXPONENT SIZES FROM 256 (BITS) TO 2048 (BITS) ON KEY GENERATION, ENCRYPTION AND DECRYPTION TIME IN THE PROPOSED ALGORITHM (HE-RSA)

| Size of Exponents (bits) | Key Generation Time (ms) | Encryption Time (ms) | Decryption Time (ms) | Total Execution Time (ms) |
|---|---|---|---|---|
| **256** | 146 | 91 | 92 | 329 |
| **512** | 209 | 112 | 107 | 428 |
| **1024** | 411 | 201 | 193 | 805 |
| **2048** | 1973 | 1001 | 967 | 3941 |

According to the results, the encryption process time and decryption process time were approximately the same in original RSA and HE-RSA when the size of exponents are 256, 512 or 1024 bits but when the size of exponents were 2048 bits, encryption and decryption time in HE-RSA were increased about 35%. Furthermore, the comparison ofthe original RSA and HE-RSA shows that the key generation time has been increased between 40 to 90 percent. The key generation time comparison between the two has been shown in Fig. 1.

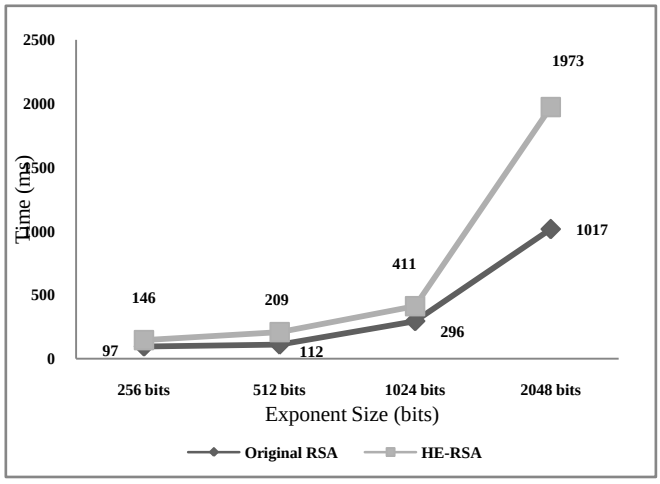



Fig. 1. Effect of changing the exponent sizes from 256 (bits) to 2048 (bits) on key generation time in original RSA and HE-RSA

The total execution time comparison of HE-RSA and original RSA shows that the total time in HE-RSA was increased approximately 30 percent when the key size had been 256, 512 or 1024 bits while this increase was about 55 percent when the key size had been 2048 bits. This comparison has been shown in Fig. 2.

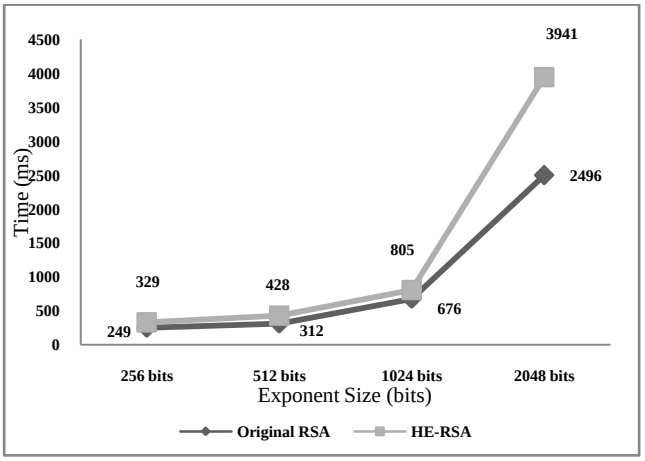



Fig. 2. Effect of changing the exponent sizes from 256 (bits) to 2048 (bits) on total execution time in original RSA and HE-RSA

As it was described earlier, the encryption process has been carried out by two consecutive steps but with the same exponents.The second encryption stage has increased the time of encryption process in comparison of the original RSA and subsequently in the decryption process and the second step of decryption has been carriedout with different exponents and has increased the decryption time in the comparison of the original RSA. This increased seems to be reasonable and justifiable according to security improvement of the proposed algorithm.

## VII. SECURITY ANALYSIS

The security analysis of original RSA and HE-RSA has been investigated according to three attack approaches: the Brute Force, Mathematical Attacks, and Timing attacks.

### *A. Brute Force Attack*

All possible combinations to guess the private key have been tried by the attacker during the brute force attack. In the original RSA, the probability of failure against this attack will be decreased considerably by choosing exponents larger than 2048 bits [9] butthe combination of HE-RSA makes this algorithm has significant resistance brute force attack even with 1024 bits exponents.

### *B. Timing Attack*

Timing attack is a side channel attack, in which the attacker determines private exponent by calculating the time with exploiting the timing variation of the modular exponentiation [12].Timing attack in original RSA may be prevented by including a random delay to the exponentiation algorithm or multiplying the cipher-text with a random number [9] while dual encryption in HE-RSA will protect the transferred message from the timing attack and it is not necessary for multiplying the cipher-text.

### *C. Mathematical Attack*

This attack will take place by determining $p, q$ or $\varphi(n)$ and could be prevented by using 2048 bits exponents in original RSA while in the proposed model, by increasing the value of $h$ the chance of successful mathematical attack would be decreased considerably [6].

Using 2 different keys for decryption process, has increased the security of the private exponents significantly and even after losing one private key by an attack the data has still been encrypted and unreachable. This is done while the algorithm has used same basic exponents in the generation process of these different keys.

## VIII. HE-RSA and Cloud Computing

The proposed algorithm were suggested for using in cloud computing environments and increasing the reliability of this new technology, but the most challenging issue in using HE-RSA encryption algorithm in cloud servers is time and memory limitations during the encryption and decryption process in servers according to the sharing performances. It is suggested to encrypt the stored data with a symmetric-key algorithm such as AES in cloud servers and after that encrypt the secret key with HE-RSA for sharing actions. This means, only the secret key would be encrypted with this algorithm and the encryption and decryption process will be more efficient and needless to sayit would be less time consuming and memory deficient.

## IX. Conclusion

In this paper, a hybrid asymmetric-key encryption algorithm has been suggested based on RSA Small-e and Efficient RSA according to the security issues in cloud computing environments. In the proposed algorithm, the number of exponents has been increased to three and a dual encryption process has been applied to raise the security level of the algorithm in comparison of original RSA. According the simulation results, the total execution time in HE-RSA was increased up to approximately 50 percent less than the original RSA and this increase may be reasonable and acceptable according to the security level and the efficiency of HE-RSA.

## Acknowledgment

We acknowledge the assistance and logistical support provided by Asia Pacific University of Technology and Innovation (A.P.U.) and Mr.ZailanArabee Bin Abdul Salam.

## References

[1] M. Malathi, "Cloud Computing Concepts," in *Proc. 3rd International Conf. on Electronics Computer Technology (ICECT)*, Kanyakumari, 2011, pp. 236-239.

[2] M.A. Alzain, E. Pardede, B. Soh, and J.A. Thom, "Cloud Computing Security: From Single to Multi-Clouds," in *Proc. 45th Hawaii International Conf. on System Science (HICSS)*, Hawaii, 2012, pp. 5490-5499.

[3] U. Somani, K. Lakhani, and M. Mundra, "Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing," in *Proc. 1st International Conf. on Parallel Distributed and Grid Computing (PDGC)*, Solan, 2010, pp. 211-216.

[4] S. Alshehri, S. P. Radziszowski, and R. K. Raj, "Secure Access for Healthcare Data in the Cloud Using Ciphertext-Policy Attribute-Based Encryption," in *Proc. 28th International Conf. of Data Engineering Workshops (ICDEW)*, Virginia, 2012, pp. 143-146.

[5] L. H. Ying, S. S. Tzuo, T. W. Guey, and B. S. P. Lin, "Toward Data Confidentiality via Integrating Hybrid Encryption Schemes and Hadoop Distributed File System," in *Proc. 26th International Conf. on Advanced Information Networking and Applications (AINA)*, Fukuoka, 2012, pp. 740-747.

[6] S. J. Aboud, M. A. Alfayoumi, M. Alfayoumi, and H. Jabbar, "An Efficient RSA Public Key Encryption Scheme," in *Proc. 5th International Conf. Information Technology: New Generations (ITNG)*, Las Vegas, 2008, pp. 127-130.

[7] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *ACM Trans. OnCommunications*, vol. 21, pp. 120-126, 1978

[8] H. M. Sun, M. E. Wu, W. C. Ting, and M. J. Hinek, "Dual RSA and Its Security Analysis," *IEEE Trans. on Information Theory*, vol. 53, no. 8, pp. 2922-2933, August 2007.

[9] A. Alhasib and A. M. Haque, "A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography," in *Proc. 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT)*, Busan, 2008, pp. 505-510.

[10] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-exponent RSA with Related Message," in *Proc.15th Annual International Conf. on Theory and Application of Cryptographic Techniques (EUROCRYPT'96)*, Berlin, 1996, pp. 1–9.

[11] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1339-1349, July 2000.

[12] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Proc. 16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'96)*, London, 1996, pp. 104-113.

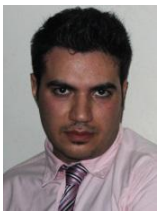
**Faraz Fatemi Moghaddam** was born at Tehran, Iran in 1985. He was graduated insoftware engineering, B. Sc from Azad university of Tehran at 2006 and also post graduated in this course of study, software engineering M. Sc from Staffordshire university at 2013.

He worked at "Djavan Daru co." for about four years from 2006 to 2010 as Information Technology manager and had a partial presence in "Journal of Persian Developers" as member of author council between 2007 to 2010. Mr. Fatemi Moghaddam has been a member of Microsoft Certified Professional (MCP) since 2009 with MCP ID: 861837 and was awarded 7 MCTS (Microsoft Certified Technology Specialist) certificates in .net frameworks subjects.

**Maen T. Alrashdan** was born at Jordan in 1978. He was graduated inelectronic engineering, B. Sc from Yarmoukuniversity of Jordan at 2006 and post graduated in computer and communication engineering M. Scfrom UKM University of Malaysia at 2004 and Ph.D in network management (computer and communication engineering) from UKM university of Malaysia at 2011.

He worked at "National Paper Company (NAPCO)" in Saudi Arabia from 2001 to 2003 as production engineer and worked at "IM Resources" in Malaysia from 2005 to 2010 as consultant, tutor and trainer. From 2011 until now he has worked at "Asia Pacific University of Technology and Innovation (A.P.U.)" as senior lecturer of research coordinator faculty.

Dr Maen has been a member of the Jordan engineers association, from 2002 until now and published five papers in the fields of computer networks, mobile communication, wireless networks, technology applications and technology management.



**Omidreza Karimi**was born at Tehran/Iranin 1983. He was graduated insoftware engineering, B. Sc from Azad university at 2005 and was graduated inmultimedia, B. Sc from Asia Pacific Institute of Information Technology (APIIT) at 2011 and also post graduated in software engineering M. Sc from Staffordshire university at 2013.

He worked at "Shafabco." for about two years from 2002 to 2004 as Information Technology manager. Mr. Karimi is a specialist in the fields of computer networks, cloud computing and green computing.