Reliability and Performance Assessment of Multifarious Hybrid Cryptosystems

IJAZ ALI SHOUKAT[1,2], ABDULLAH AL-DHELAAN[1], MZNAH AL-RODHAAN[1], KAMALRULNIZAM ABU BAKAR[2], SUBARIAH IBRAHIM[2]

[1] Computer Science Department, College of Computer and Information Sciences
King Saud University, P.O. Box. 51178, Riyadh 11543, Saudi Arabia
ishoukat@ksu.edu.sa , dhelaan@ksu.edu.sa, rodhaan@ksu.edu.sa

[2] Department of Computer Science, Faculty of Computing
Universiti Teknologi Malaysia
81510 Johor Bahru Malaysia
kamarul@fsksm.utm.my, subariah@fsksm.utm.my

Abstract: - Performance efficiency of symmetric and reliable key exchange of a-symmetric cryptosystems are being trendier towards the use of hybrid cryptosystems. Still, the question of an ideal and optimal selection of hybrid cryptosystem requires further analysis. Earlier studies have just evaluated the performance of few hybrid cryptosystems with smaller input dataset, without having the analysis of hybrid crypto-models in terms of two-way origin authenticity, false modification, feasibility, memory, power, forgery and password guessing attacks. In this article, firstly, we have reviewed several existing hybrid crypto-models through which symmetric and asymmetric algorithms can be modeled as an ideal hybrid cryptosystem. Secondly, we have practically applied the selected hybrid crypto-model to combine different symmetric and asymmetric algorithms in order to assure not only the accuracy but also the confidentiality, false modification and origin authentication of both parties. Finally, we have evaluated the performance of several hybrid cryptosystems (IDEA-RSA, AES-ECC, TDES-RSA, AES-RSA and RC2-RSA) through practical experimentations (encryption, decryption, certificate generation and verification) in order to analyze which hybrid cryptosystem is feasible for encrypting large input dataset with low electric-power. Results show that, joint AES-RSA scheme is significantly outperformed in data encryption and takes less electric power to guarantee optimal privacy goals.

Key-Words: - Hybrid encryption, AES-RSA, AES-ECC, RC2-RSA, IDEA-RSA, TDES-RSA

1 Introduction

Efficient, secure and reliable exchange of personal secrets is being more desirable in remote communications. Symmetric cryptosystems are efficient in performance but un-reliable in key exchange. Asymmetric (public key) algorithms are deficient in performance but more reliable in key exchange, thus, their distinctive characteristics are likely triggering the use hybrid cryptosystems rather to utilize any standalone conventional encryption schemes (symmetric or asymmetric). Literature also reflects that asymmetric encryption techniques associate feeble natured performance issues such as computational processing, huge memory, massive energy consumptions and implementation limits on bulky data sets but these techniques are relatively secure and reliable in key exchange over public networks [1]. Moreover, asymmetric cryptosystems use modular exponentiation and nontrivial mathematical functions that are the actual causes of consuming more memory and processing power as compared to any equivalent input sample ciphered with symmetric encryption algorithm. On the other hand, symmetric cryptosystems (i.e. AES) are 100 times efficient relatively to asymmetric encryption algorithm (i.e. RSA) in encryption phase and 2000 times faster in decryption phase [2] but symmetric cryptosystems are not fully reliable in key exchanging mechanism due to unfulfilling of required set of security goals. Secure and reliable exchange of confidential information (key, data) requires to achieve some standard security objectives such that confidentiality, integrity (authenticity, non-repudiation) and availability [3]. Confidentiality assures secrecy and privacy of

transaction which means transaction (message) can only be viewed by the concerned person. Integrity concerns with two requirements: authenticity – *verification of senders identity on receiving of message* and Non-repudiation – *verification of fake or false modifications in the original message*. While availability means, information (message, key, certificate verification) and medium (Certification Authority Server, online services) should be timely available upon need. Consequently, symmetric schemes are efficient in processing of large dataset because these schemes require less memory and less CPU cycles to save battery power as compared to asymmetric schemes but alone symmetric cryptosystem are lacked to detect *non-repudiation in message, false modifications* in *secret key* and *origin authentication* of sender and receiver. However, Asymmetric schemes are relatively reliable in exchanging of encryption-key by fulfilling of complete set of security goals as discussed earlier. These prominent distinctive factors of both symmetric and asymmetric schemes have birthed to hybrid cryptosystems that work with mutual committee of symmetric and asymmetric algorithms in order to combine the benefits of both schemes. Variety of hybrid cryptosystems has been reported under various infrastructural designs [4, 5, 6, 7, 8]. These hybrid schemes differ in functional design, feasibility, computational efficiency and security threats.

The reliable hybrid-crypto-model: Generic Hybrid System (GHES) [1] is feasible to combine symmetric and asymmetric techniques in order to achieve complete set of standard security objectives rather to the other hybrid encryption schemes as discussed in Table 1. The analysis of Table 1. Shows that, the discrepancy associated with *Lamport's* scheme is the requirement of additional password table in order to verify the user credentials. Later on *Wu* modified *Lamport*scheme but *Wu's* hybrid schemes remained unsuccessful due to the applicability of forgery and password guessing (session key recovery) attacks. *Ramaraj's* hybrid encryption scheme [7] is based on *Wu's* scheme therefore, it is also fully vulnerable against these two discussed attacks. How these two attacks are applicable on *Wu's* hybrid scheme it has been discussed by the authors of study [1]. The hybrid schemes proposed by *Dubal* in 2011 [4] and *Subasree* in 2010 [5] have capability to detect non-repudiation and origin authenticity but these schemes are computationally feeble and vulnerable against forgery and password guessing (session key recovery) attacks.

Several other authors discussed hybrid encryption framework with RSA public key algorithm. RSA is robust public key algorithm and its history goes back to 1977 [9]. In 2011, a basic hybrid approach to assure the confidentiality of electronic payment systems was discussed in study [10] and in the mid of 2013 the similar hybrid framework (AES-RSA) discussed by [11]. These both basic hybrid encryption frameworks did not deal to generate digital signature and hash function due to which these schemes cannot detect origin authenticity and non-repudiation against cipher-text. Moreover, in previous years, RSA algorithm had been combined with AES, DES and 3DES to check time and memory consumption in which RSA-DES had shown optimal in execution time and memory consumption [10]. The International Data Encryption Algorithm (IDEA) was combined with RSA in 2005 in which digital signature scheme of RSA was implemented with SHA-256 [12] because, Digital Signature Algorithm (DSA) is 10% to 40% slower in signature verification process as compare to RSA [13]. The generation of signature and verification in case of RSA is outperformed than Elliptic Curve Cryptography (ECC) however ECC takes smaller key, shorter sized signature with less key generation time better option for cloud environment in which many users connects with small time spans [17, 18]. The Generic Hybrid Encryption System (Figure 1) facilitates optimal security as compared prior hybrid encryption techniques because GHES can detect non-repudiation, origin authenticity and it is also secure against forgery and password guessing (session key recovery) attacks. Further details about GHES can be consulted in study [1]. On the base of Table 1, we have selected the Generic Hybrid Encryption System (GHES) as a framework to jumble the AES and RSA due to its extra-ordinary features as compared to the other hybrid encryption frameworks (Table 1). The selection of GHES fulfills the first objective of this article. The second objective of this article is to apply the selected model (GHES) on existing symmetric and asymmetric encryption schemes in order to assure accuracy, confidentiality, false modification and origin authentication of both sender and receiver. Under the third objective, as a first step, we have practically evaluated the working accuracy of selected hybrid schemes (e.g. AES-RSA, AES-ECC etc.) to verify the generated certificates and non-repudiation in cipher-text and encrypted private key. In second step, we have evaluated and compared the performance of prominent hybrid cryptosystems such as AES-RSA, AES-ECC, IDEA-RSA, RC2-

RSA and TDES-RSA through practical experimentations (encryption, decryption and certificate generation) in order to analyze which hybrid cryptosystem is feasible for encrypting large input dataset by fulfilling all required security goals.

2 Methodology and Implementation

The adopted methodology includes two systematic approaches:- first one is the selection of hybrid encryption framework (crypto-model) and second one is the experimental performance evaluation method. We implements the Generic Hybrid Encryption System (GHES) as a framework to combine symmetric and asymmetric cryptographic schemes (e.g. AES-RSA). The selected GHES model was implemented to combine selected hybrid cryptosystems using a tool written in C++ language. The utilized key size for AES, TDES and RC2 was 128 bit and the key size for RSA was 1024 bit as well as the key size for ECC was 160 bit. The session key was generated randomly and public key certificate was generated for an ordinary user A (secret code: 1234). The User A was consider as a sender and another ordinary User B was selected as receiver. For purpose of evaluating the accuracy and performance of selected hybrid cryptosystems (AES-RSA and AES-ECC etc.), we have selected large sized plaintext (8.18 MB) written in notepad file equal to more than 3500 pages of any Microsoft word document. All possible data types such as numeric, alphabet, special character and space characters was the part of selected plaintext file. The step-by-step implementation of selected hybrid cryptosystems (e.g. AES-RSA etc.) in encryption and decryption phases has been provided in this section in order to test the working accuracy and fulfillment of security goals.

2.1 Encryption Phase by User A

Step-1: Selection of Plaintext *(P)*

Step-2: We randomly generated the symmetric secret key as a session key (S_k). S_k= *C1 D3 20 C3 14 58 48 18 50 42 BB 72 A5 FD 50 32*

Step-3: We practically encrypted the *P* with $S_k$by using AES algorithm that returns cipher of plaintext *(C*)*

Step-4: Selection of asymmetric RSA public key *(B-P_bK)* for *user B.* We practically generated *B-P_bK* by using selected tool. Now user A has S_k, *P* and *B-P_bK* and *C**.

Step-5: Encryption of S_k with *B-PbK* are conducted practically which is referred as ΔS_k.

Step-6: Calculate the hash (digest) ofS_k and Cipher Text *(C*)* through *MD5* which are referred as:-Hash of secret Key= $h(S_k)$ and Hash of cipher text = *h(C*)*

Step-7: Encrypt the $h(S_k)$ *and h(C*)* with *A-P_rK*by using RSA algorithm which returns the *digital signature* referred as $D= A\text{-}P_rK(hS_k + hC^*)$

Step-8: Send the final encrypted message *(ΔS_k + C* + D)* to *User B.*

2.2 Decryption and CIA verification Phase by User B

Step- 9: On receiving *(ΔS_k + D+ C*), User B* applies the following functions

(a) Decryption of $\Delta S_k$by applying his private key *(B-P_rK)* in order to get original session key *(S_k).*

(b) Decryption of "*D*" by applying the public key (A-P_bK) of user A through RSA. After that *User B* can compare the old hash value of both S_k and *C** as sent by *User A* with the newly created hash values of both S_k and *C**. If the hash values are the same it means no false modifications in both S_k and *C**. The successful implementation of *A's* public key ensure the origin authentication of *user A.* the successful verification of hash values also authenticate the originality of both S_k and *C**.

Step-10: After that *user B* can decrypt the *C** by applying original S_k and AES algorithm in order to generate the plaintext *(P).*

All discussed steps show that the required security goals (confidentiality, origin authenticity, non-repudiation) can be fulfilled by any hybrid cryptosystem like AES-RSA scheme. By applying defined methodology (10 steps), we practically tested the selected hybrid cryptosystems with a C++ based tool against all phases including encryption, certificate generation, verifications and decryption. After ensuring its practical implementation, we have input the selected large sized plaintext to each hybrid cryptosystems such as AES-RSA, AES-ECC, IDEA-RSA, TDES-RSA and RC2-RSA to examine performance reliability. The subsequent section summarizes practical results and provides the comprehensive analysis on measuring the performance of selected hybrid cryptosystems.

3 Results and Analysis

Our pervious evaluation was limited to only two hybrid cryptosystems (AES-RSA, AES-ECC) but here, we have extended our experimental evaluation with 3 more hybrid cryptosystems TDES-RSA, RC2-RSA and IDEA-RSA. We not only evaluated the performance but we have also analyzed the reliability of each hybrid cryptosystem in terms of certificate generation and verification for fulfilling of required security goals such as confidentiality, origin authenticity and non-repudiation have been examined mathematically in methodology and implementation section. However, this section provides the experimental results for testing the accuracy and performance of several hybrid cryptosystems including RC2-RSA, Triple-DES-RSA, IDEA-RSA, AES-RSA and AES-ECC. The reason of selecting asymmetric algorithm (RSA) each time with the symmetric natured algorithms in the form of hybrid cryptosystem is due it less power consumption in signature verification and encryption process [19, 21,22] as shown in Table 2. Moreover, RSA takes lowest memory and possesses higher encryption speed as compared to asymmetric (Public-Key) algorithms such as ElGAMAL and Paillier [20]. All reported results were recorded against a large sample of plaintext (8.18 MB notepad file = more than 3500 pages of doc file) which was individually encrypted and decrypted using five selected hybrid crypto-schemes. Both encryption and decryption phases were executed and as a result all five hybrid cryptosystems have shown with great accuracy. The selected large sized data set was encrypted using symmetric class algorithm and the symmetric secret key was encrypted and using RSA encryption algorithm. The methodology of encryption, decryption and key exchanging has been modeled in Figure 1. During the experimentations, the RSA, AES-ECC, RC2- RSA, IDEA-RSA, AES-RSA and Triple-DES-RSA encryption schemes have been tested and compared using the same input sample (plaintext) to record encryption and decryption times of each encryption scheme.

AES, RC2, Triple-DES (TDES) belong to symmetric class of block ciphers which are considered as fastest algorithms. On the other hand, the RSA and ECC belong to asymmetric (public key cryptography) classes which have been considered as greatly slower than the symmetric ciphers in encryption or decryption phases but they have an advantage of ensuring false modifications and origin authentication over symmetric ciphers. Alone asymmetric typed encryption algorithm such as RSA is 2000 times slower in encryption phase as compared to symmetric algorithm such as AES because RSA uses complex computations through non-retrieval mathematics with huge wastage of memory and electric power too [2]. Symmetric schemes are 100 times faster as compared to asymmetric schemes and have no employment limitations upon large datasets (video, audio, image).

Therefore, the decision of hybrid encryption is more suitable in which plaintext is ciphered with symmetric algorithm and only the symmetric secret key is encrypted by utilizing asymmetric algorithm. In this way, speed, memory, electric power and employment issues against large dataset can be fully overcame. The other privacy related issues such as origin authentication and false modification require the use of signatures. The best implementation of signature lies under public key cryptography. In case of public key algorithms, the signature generation and verification times always matter for purpose of ensuring false modification and origin authentication.

3.1 RSA and ECC

The public key algorithm RSA and ECC have individually been compared in terms of measuring the time for key generation, signature generation and verification as shown in Figure 3. In signature generation and verification RSA takes about 13 ms but ECC takes 157 ms for generating signature and 235 ms for verifying the signature. The encryption time of RSA was found 7.225 seconds as shown in Figure 2. The signature generation and verification times of RSA are significantly smaller than the signature generation and verification times of ECC but in case of generating key, RSA is slower as it has shown 170 ms as compared to the key generation time of ECC which was just 80 ms as shown in Figure 3.

3.2 AES-RSA

While encrypting the larger dataset (8.18 MB), the RSA itself has shown average encryption time (7.225 Sec) but in case of hybrid method (AES-RSA), the encryption was done using AES and RSA was used to exchange and encrypt symmetric key (session key) which have significantly resulted the reduction in encryption time. The improved encryption time of AES-RSA was recorded just (0.994 Sec) which is 6 times better to single public key encryption algorithm (e.g. RSA) as shown in Figure 2. Similarly, the decryption time of RSA is

more than 96 seconds but in case of decryption AES-RSA the decryption time was just 1.145 seconds as shown in Figure 4. Therefore, mutually the AES-RSA becomes outperformed to RSA in encryption and decryption of data.

3.3 AES-ECC

In case of encrypting the same plaintext the AES-ECC have shown average encryption time (164.42 Sec) as shown in Figure 2. Similarly, the average decryption time of AES-ECC was recorded to (1.088 sec) as depicted in Figure 4. Therefore, AES-ECC takes less time in decryption comparatively to its own encryption but the decryption time of AES-ECC is almost equivalent to AES-RSA decryption time but due to the huge difference in the encryption time of AES-ECC and AES-RSA, overall the AES-RSA is exposed to be outperformed. However, comparatively, to single RSA, the encryption time of AES-RSA was 164.42 seconds which is larger to RSA but the decryption time (1.088 sec) has been shown many times smaller to decryption time of RSA as depicted in Figure 2 and Figure 4.

3.4 IDEA-RSA

The combination of IDEA-RSA was examined to record encryption and decryption time using the similar sized plaintext dataset which was utilized in the evaluation of AES-RSA. During the examination, the average encryption of IDEA-RSA was recorded to (2.112 sec) and the average decryption time was found to be 3.187 seconds which is greatly higher to the average encryption and decryption of AES-RSA hybrid scheme as shown in Figure 5. Thus, the results show that AES-RSA is efficient to IDEA-RSA hybrid scheme.

3.5 TDES-RSA

When, Triple-DES (TDES) was conjectured with RSA to perform experimental examination of encryption and decryption times, the results were against shown to be in support of AES-RSA. The average encryption time of TDES-RSA was recorded to (2.52 sec) and the average decryption time of TDES-RSA was 3.52 seconds as shown in Figure 6. Thus, the encryption and decryption performance of TDES-RSA is almost 2 times least efficient rather to the encryption and decryption of AES-RSA upon the same plaintext dataset.

3.6 RC2-RSA

To compare the performance of AES-RSA, with RC2-RSA, the experimentation was conducted again to test the encryption and decryption performance of RC2-RSA as hybrid cryptosystem. The same sized large plaintext was used in the experimentation which was used for other hybrid cryptosystems. During the experimental examination, the average encryption time of RC2-RSA was recorded to (1.91 sec) and the average decryption time was found to be 2.17 seconds which is shown to be 2 times higher to the encryption and decryption time of AES-RSA hybrid scheme as depicted in Figure 7. Thus, the RC2-RSA was also exposed to be lacked in encryption and decryption performance comparatively to AES-RSA performance. On the other hand, the average decryption time of AES-ECC was recorded only (1.08 Sec) which is slightly smaller to AES-RSA but overall AES-ECC has been shown deficient to AES-RSA due to the huge encryption time (164.42 sec) as shown in Figure 2.

Comparatively to RSA, the ECC takes less memory and less electric power to generation signature but RSA consumes less electric power (energy) in encryption process and signature verification rather to ECC as shown in Table 2. The average encryption time of AES-RSA is 0.994 seconds which is greatly improved as compared to the encryption time of RSA (7.225 Sec) and AES-ECC (164.42 Sec) using the same dataset as depicted in Figure 2, but decryption time of both AES-RSA and AES-ECC are almost nearest to each other. However, the electric power consumption in signature verification and encryption process declares that RSA is better to ECC as shown in Table 2 and more significantly due to least encryption time, AES-RSA is outperformed to AES-ECC. Similarly, all other hybrid cryptosystems such as IDEA-RSA, TDES-RSA and RC2-RSA were shown to be non-efficient in encryption and decryption performance as compared to the AES-RSA hybrid crypto-scheme as depicted in Figure 8. Thus, hybrid encryption scheme (AES-RSA) is most ideal decision to merge the benefits of both symmetric and asymmetric cryptosystems followed by RC2-RSA.

4 Conclusion

Hybrid Cryptosystems are optimal tools to evolve the benefits of symmetric and asymmetric cryptosystems. In this article, the asymmetric algorithms (RSA, ECC) were firstly combined with AES and then RSA was combined with other symmetric algorithms such as IDEA, Triple-DES and RC2 to conduct practical examination including all steps such as encryption, decryption, certificate generation and verification. As a result, the AES-

RSA scheme has been shown efficient in case of encryption, decryption, and signature verification times as compared to AES-ECC. However, with respect to memory requirement AES-ECC is shown better to AES-RSA. Moreover, comparatively to any single public key cryptosystem (e.g. RSA), the hybrid cryptosystem (e.g. AES-RSA) is significantly outperformed for enciphering of large sized data. The GHES framework has been shown good to combine the symmetric and asymmetric cryptosystems because it provides optimum privacy without having the applicability of forgery and password (session key) guessing attacks. Thus, the joint AES-RSA cryptosystem is not only robust to fulfill most important security goals (confidentiality, false modification and origin authentication etc.) but it is also outperformed to all selected hybrid cryptosystems. AES-RSA hybrid cryptosystem is more than *100 times* faster to AES-ECC in data encryption but decryption time of both schemes is almost same. Moreover, the AES-RSA scheme is almost *2 times* outperformed than the other hybrid cryptosystems such as RC2-RSA, IDEA-RSA and TDES-RSA.

5 Future Directions

With hybrid encryption methods although the privacy and speed related issues can be resolved but actually the utilized symmetric algorithm requires the same security implications as it would be expected for any symmetric cryptographic algorithm. The merging of symmetric and asymmetric algorithms did not mean that the need of randomness and dynamicity is no longer required. The security of any cryptosystem (singular or hybrid) significantly relies on randomness properties. The need of randomness and dynamic data blocking mechanism is a desirable feature in future cryptosystems [15]. The most serious situation with current symmetric algorithms is their static design that contains the fixed data blocking and static substitution. Even the AES utilizes fixed data blocking mechanism (non-dynamic) as agreed by the authors of study [16]. Both opinions discussed in [14][15] reflect that, future encryption methods must be evaluated under the characteristic of dynamic data block partitioning because dynamic data block partitioning is an adequate way of creating randomness for the cracker. Similarly, as compare to the fixed substitution (static s-boxes), the use of randomized substitution (dynamic s-boxes) is a needy trait for upcoming encryption algorithms to enhance their randomness and dynamicity.

Acknowledgement

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-264.

References

[1] I.A. Shoukat, K.A. Bakar, S. Ibrahim, "A Generic Hybrid Encryption System (HES)", *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 05, No. 09, 2013, pp. 2692-2700.

[2] C. Fontaine, F. Galand, "A Survey of Homomorphic Encryption for Non-specialists", *EURASIP Journal on Information Security*, Volume 2007, 2007, pp. 01-10.

[3] S. O. Melia, A. J. Elbirt, "Enhancing the Performance of Symmetric-Key Cryptography via Instruction Set Extensions", *Journal of IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 18, Issue 11, 2010, pp. 1505-1518.

[4] M. J. Dubal, T. R. Mahesh, P.A. Ghosh, "Design of new security algorithm using hybrid cryptography architecture", *3rd International Conference on Electronics Computer Technology (ICECT)*, Vol. 5, 2011, pp. 99-101.

[5] S Subasree, N. K. Sakthivel, "Design of a New Security Protocol Using Hybrid Cryptography Algorithms", *International Journal of Research and Reviews in Applied Sciences*, Vol. 2, Issu,2, 2010, pp. 95-103.

[6] L. Lamport, "Password authentication with insecure communication", *Communication of ACM*, Vol. 24, 1981, pp. 770-772.

[7] S.T. Wu, B. C. Chieu, "A user friendly remote authentication scheme with smart cards", *Computers and Security Elsevier*, Vol. 22, No.06, 2003, pp.547-550.

[8] E. Ramaraj, S. Karthikeyan, M. Hemalatha, "A Design of Security Protocol using Hybrid Encryption Technique (AES-Rijndael and RSA)" , *Int. J. comp. Intern. Mgnt*, Vol. 17, No.01, (2009, pp. 78-86.

[9] R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *ACM*

Transaction Communications, vol. 21, 1978, pp.120-126.

[10] M. Irum, A. Khan, M. S. H. Khiyal, "Confidentiality of Messages in A Cradles Electronic Payment System", *International Journal of Reviews in Computing*, Vol. 6, 2011, pp. 29-32.

[11] K.Rege, G. Goenka, P.Bhatada, S. Mane, "Bluetooth Communication using Hybrid Encryption Algorithm based on AES and RSA", *International Journal of Computer Applications*, Vol. 71, No.22, (2013, pp. 10-13.

[12] M. A. Khan, Y. P. Singh, "On the security of Joint Signature and Hybrid Encryption", *13th Int. Conf. on Communication*, Vol. 01, Nov. (16-18), 2005, Malaysia, pp. 109-112.

[13] G. Olias, "Security and auto-configuration of Location Servers for IP Telephony", *Mater Thesis, Helsinki University of Technology (HUT)*, 2002.

[14] I.A. Shoukat, K. A. Bakar, "Effective Evaluation Metrics for the Assessment of Cryptographic Algorithms and Key Exchange Tactics", *Information-Tokyo*. Vol. 16, No 05, (2013, pp.2801-2814.

[15] I.A. Shoukat, K. A. Bakar, M Iftikhar, "A Survey about the latest trends and research issues of cryptographic elements", *Int. J. of Comp. Sc. Issues (IJCSI)*, Vol. 8, No. 02, Issue 3, 2011, pp. 1694-0814.

[16] I.A. Shoukat, K. A. Bakar, S. Ibrahim S, "A Novel Dynamic Data Blocking Mechanism for Symmetric Cryptosystems", *Research Journal of Applied Sciences, Engineering and Technology* 7(21): 4476-4489. 2014

[17] B. Bhavani-Bai and Devi N. R. (2014). Ensuring Security at Data Level in Cloud using Multi Cloud Architecture. The International Journal of Science & Technoledge. 2(6): 254-263.

[18] X. Yang, X. Li, J. Li, "Application Study on Public Key Cryptography in mobile payment", 5th *WSEAS* Int. Conference on Information Security and Privacy, Venice, Italy, November (20-22):97-102, 2006

[19] R. Kayalvizhi, M. Vijayalakshmi, V. Vaidehi, "Energy Analysis of RSA and ELGAMAL Algorithms for wireless Sensor Networks", 8th *WSEAS* International Conference on Applied Electromagnetic, Wireless And Optical Communications, 2010: PP. 20-24

[20] S. Farah, Y. Javed, A. Z. Shamim, T. Navaz, "An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms". *Recent Advances in Information Science (WSEAS)*. ISBN: 978-1-61804-140-1 , PP. (121-124), 2012

[21] Luo, X., Zheng, K., Pan, Y., & Wu, Z. (2004, October). Encryption algorithms comparisons for wireless networked sensors. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Vol. 2, pp. 1142-1146). IEEE.

[22] K. N. Bidkar, "Energy Analysis of Algorithms in Public Key Cryptography of WSN", International Journal of Advance Research in Computer Science and Management Studies, 3(3): 190-197, 2015

Appendix

The Appendices section includes the Tables and Figures used in this article.

Table 1: Comparison of Hybrid Crypto-models

Evaluation Parameters	**Prior Hybrid Encryption Schemes**			
	(Dubal and Mahesh et. al. 2011) [4]	(Subasree and Sakthivel, 2010) [5]	(Ramaraj and Karthikeyan, 2009) [7]	Generic Hybrid Encryption System (GHES) [1]
Based on Symmetric and Asymmetric techniques	×	×	√	√

Computationally Proficient	×	×	√	√
Optimal feasibility for bulky data sets (Audio, Video)	×	×	√	√
Non-repudiation and Fake Modifications	√	√	√	√
Origin authenticity of sender and receiver	√	√	√	√
Applicability of Forgery Attack	×	×	√	×
Applicability Password (session key) Guessing Attack	×	×	√	×
Customer Satisfaction	×	×	×	√
Access of Third party over	Keys Digest (Hash values)	Keys Digest (Hash values)	Keys Digest (Hash values)	Keys Digest (Hash Values)
Memory Requirement	High	High	low	Low
Electric Power (energy) consumption	High	High	Low	Low

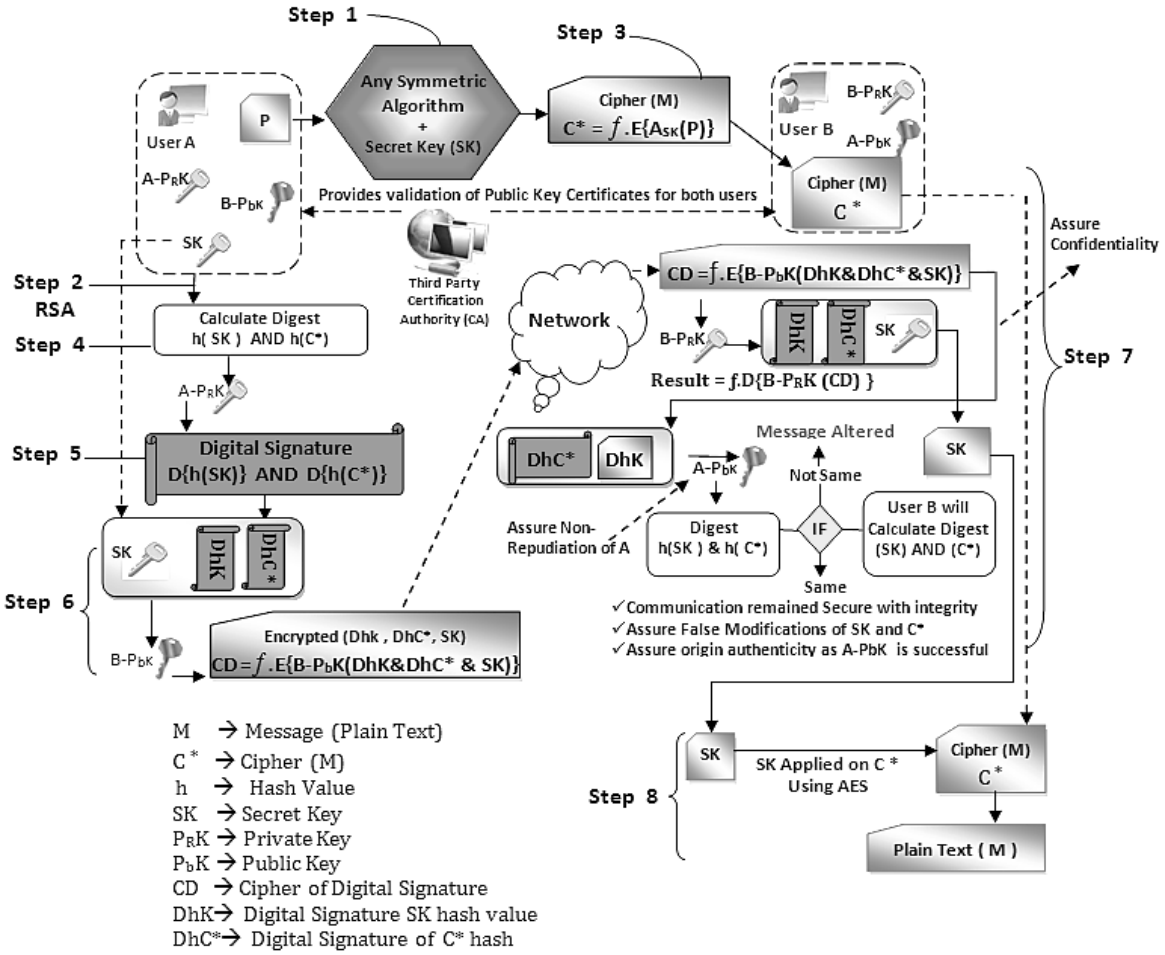

Figure 1: Generic Hybrid Encryption System (GHES) Framework

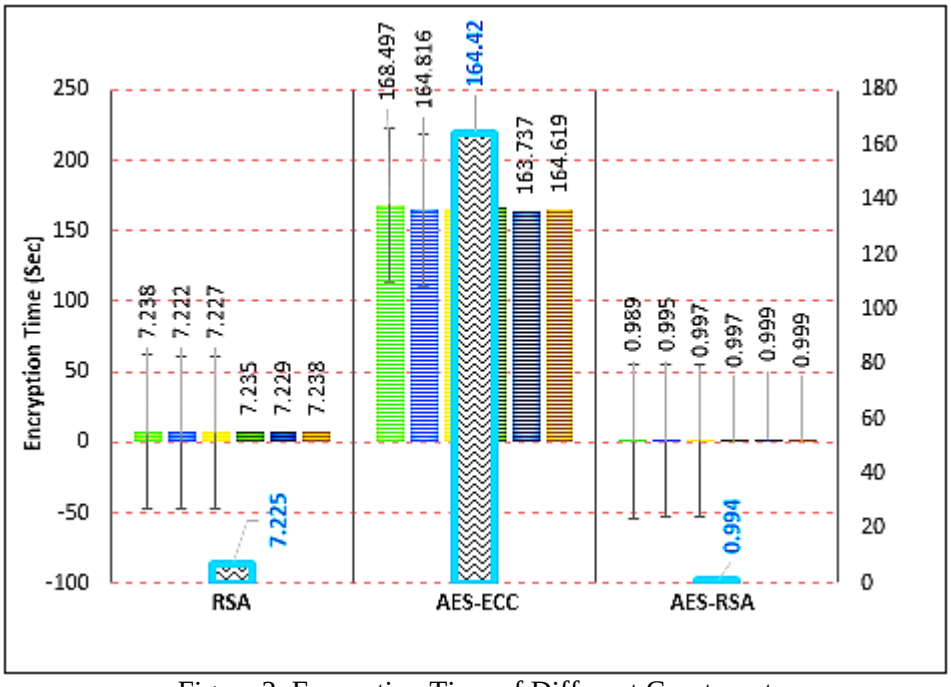

Figure 2: Encryption Time of Different Cryptosystems

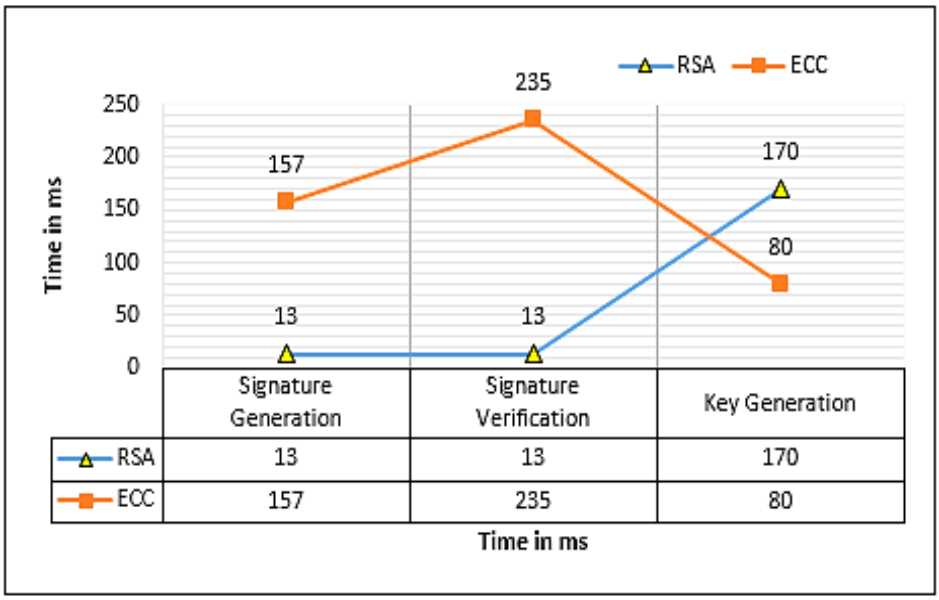

Figure 3: Key Generation, Signature Generation and verification Time

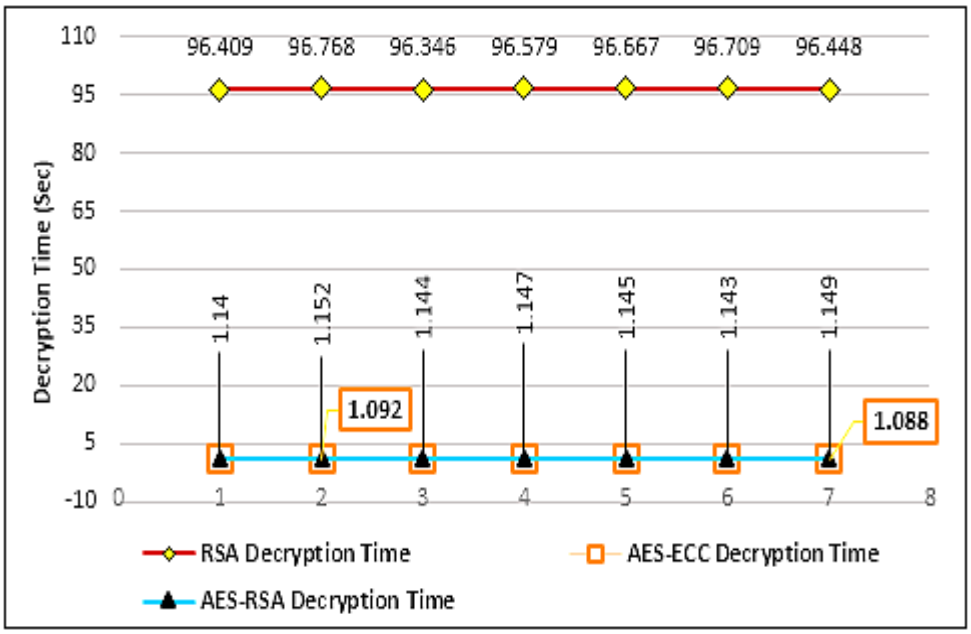

Figure 4: Decryption Time of Different Cryptosystems

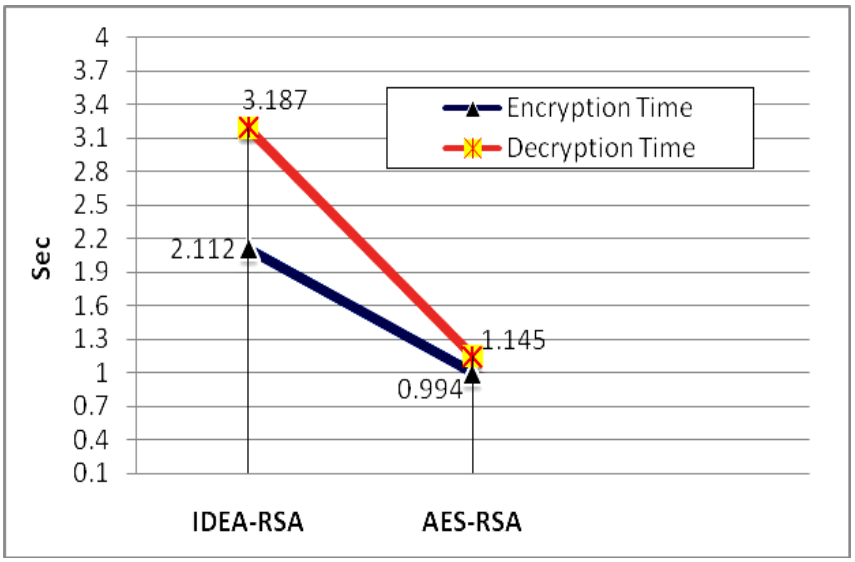

Figure 5: Encryptoin and Decryption time of IDEA-RSA

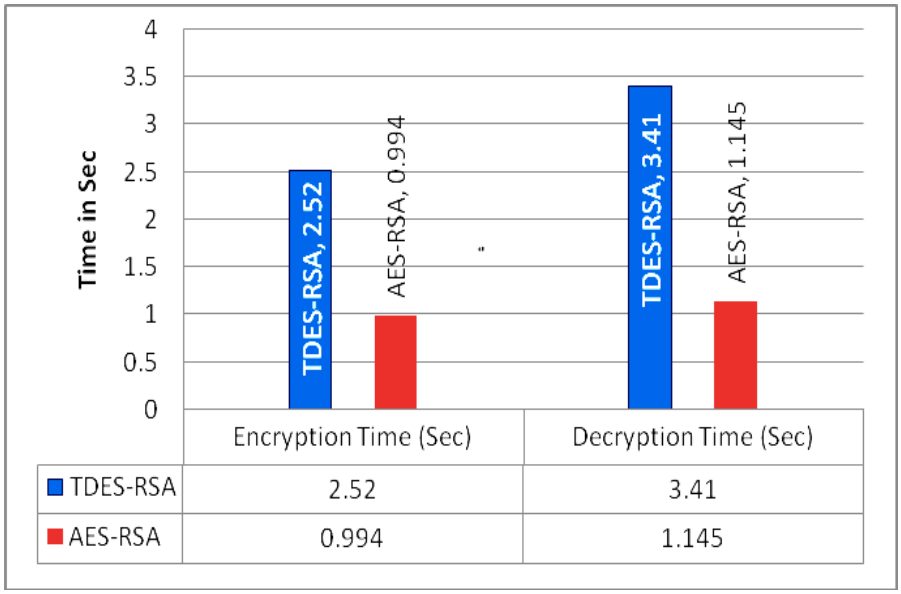

Figure 6: Encryption and Decryption time of TDES-RSA

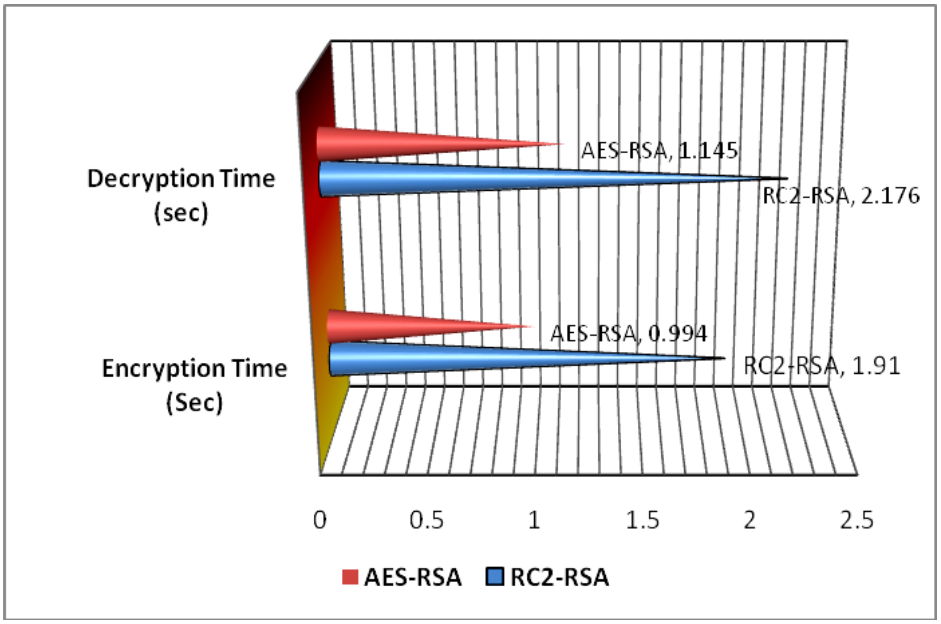

Figure 7: Encryption and Decryption time of RC2-RSA

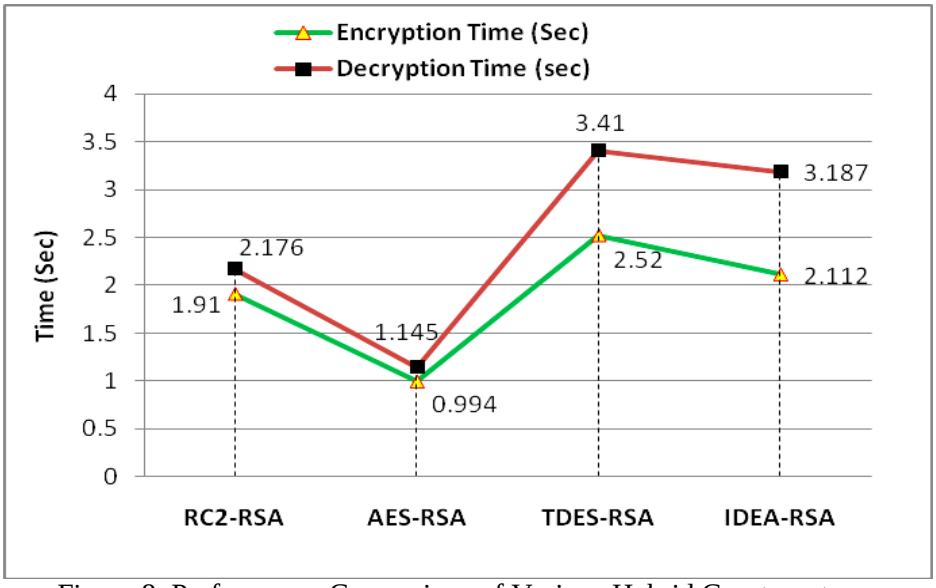

Figure 8: Performance Comparison of Various Hybrid Cryptosystems

Table 2: Comparison of RSA and ECC

Comparison Parameters	RSA-1024	ECC-160
Power Consumption in Encryption process [21][22]	$12.5x10^{-3}J$	$24.5x10^{-3}J$
Power Consumption Signature Verification [22]	21%	45%
Power Consumption in Signature Generation [22]	302 mJ	22 mJ ECDSA-160
Memory Usage	More	less