Practical Evaluation of Hybrid Cryptosystems

IJAZ ALI SHOUKAT1,2, ABDULLAH AL-DHELAAN1, MZNAH AL-RODHAAN1, KAMALRULNIZAM ABU BAKAR2, SUBARIAH IBRAHIM2

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: - Cryptosystems play imperative role in securing and ensuring of confidential information related to the defensive activities of any country or an individual. This short article deals with hybrid cryptosystems to target three objectives. As a first objective it reviews several existing hybrid crypto-models through which symmetric and asymmetric algorithms can be modeled as an ideal hybrid cryptosystem. The second objective of this article is to apply 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, as a third objective, the two most trendier hybrid cryptosystems (AES-RSA and ECC-AES) have been evaluated and compared through practical experimentations (encryption, decryption and certificate generation) in order to analyze which hybrid cryptosystem is feasible for encrypting large input dataset. The work conducted in this article differs with prior studies because; earlier efforts are limited in such sort of experimental and comparative effort for hybrid cryptosystems as we have conducted in this article. Our experimental results show that, AES-RSA guarantees optimal privacy to fulfill all security goals and it is significantly outperformed to AES-ECC in encryption, decryption, signature generation and verification.

Key-Words: - Hybrid encryption, asymmetric encryption, symmetric encryption, security, privacy

1 Introduction
Security can never be compromised in swapping of confidential transaction over insecure communication channels. Hybrid encryption schemes are being trendier in encryption matters rather to utilize standalone conventional encryption schemes (symmetric or asymmetric) due to several reasons. 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.

An optimal hybrid scheme such as Generic Hybrid System (GHES) [1] which is feasible to combine symmetric and asymmetric techniques in order to achieve complete set of standard security objectives rather to other hybrid encryption schemes as discussed in Table 1. 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’s 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 with less key generation time better option for cloud environment in which many users connects with small time spans [17]. 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 AES-RSA and AES-ECC 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 both AES-RSA and AES-ECC schemes to find out which hybrid encryption method is most feasible for encrypting large dataset by fulfilling all required security goals.
2 Methodology and Implementation

The concerned methodology includes two systematic approaches: first one is the selection of hybrid encryption framework and second one is the experimental performance evaluation method. We followed 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 AES with RSA and ECC using a tool written in C++ language. The utilized key sizes were 128 bit for AES, 1024 bit for RSA and 160 bit for ECC. 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 both hybrid cryptosystems (AES-RSA and AES-ECC), 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 both encryption and decryption phases against hybrid cryptosystem (e.g. AES-RSA) has been provided in this section in order to test the working accuracy and fulfillment of security goals of AES-RSA Scheme.

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-PbK) for user B. We practically generated B-PbK by using selected tool. Now user A has S_k, P and B-PbK and C*.
Step-5: Encryption of S_k with B-PbK are conducted practically which is referred as \( \Delta S_k \).
Step-6: Calculate the hash (digest) of S_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-PbK by using RSA algorithm which returns the digital signature referred as \( D = A-PbK(h(S_k) + h(C*)) \)
Step-8: Send the final encrypted message (\( \Delta S_k + D + C* + D \)) to User B.

2.2 Decryption and CIA verification Phase by User B

Step-9: On receiving (\( \Delta S_k + D + C* \)), User B applies the following functions
(a) Decryption of \( \Delta S_k \) by applying his private key \( (B-PbK) \) in order to get original session key \( (S_k) \).
(b) Decryption of “D” by applying the public key \( (A-PbK) \) 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).

These steps discuss how standard security goals (confidentiality, origin authenticity, non-repudiation) can be fulfilled by any hybrid cryptosystem like AES-RSA scheme. By applying defined methodology (ten 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 (AES-RSA and AES-ECC) to examine several performance related experimental results. The subsequent section summarizes practical results and provides the comprehensive analysis on measuring the performance of selected hybrid cryptosystems.

3 Results and Analysis

The 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 both hybrid cryptosystems (AES-RSA and AES-ECC). All reported findings 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 selected hybrid cryptosystem. Both encryption and decryption phases were executed and as a result both AES-RSA and AES-ECC have shown full accuracy. During the experimentations, the RSA, AES-ECC and AES-RSA encryption schemes have been tested and compared against the same input sample of plaintext to record encryption time in seconds as shown in Figure 2. The average encryption time for RSA was recorded (7.225 Sec) and mutally AES-ECC have shown average encryption time (164.42 Sec) however, the average encryption time for AES-RSA was shown (0.994 Sec). In encryption phase the AES-RSA is outperformed to AES-ECC even from singular RSA as depicted in Figure 2.

AES belongs 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 overcome. 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. Therefore, both public key algorithms (RSA and ECC) have 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 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.

In average, RSA has showed (96.5 Sec) decryption time and AES-RSA scheme has showed decryption time (1.14 Sec) as depicted in Figure 4. However, the average decryption time of AES-ECC was recorded only (1.08 Sec) which is slightly smaller to AES-RSA and highly smaller to RSA as shown in Figure 4. Thus, hybrid encryption scheme (AES-RSA) is most ideal decision to merge the benefits of both symmetric and asymmetric cryptosystems. With hybrid encryption methods although the privacy and speed related issues can be fixed 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.
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 combined with symmetric AES 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, signature generation and signature verification times as compared to AES-ECC. 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 AES-ECC hybrid encryption scheme. 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. However, the decryption time of both AES-RSA and AES-ECC are almost nearest to each other which is also improved as compared to RSA. The signature generation and verification of RSA is better than ECC that makes the AES-RSA scheme as an outperformed over AES-ECC.

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


Appendix

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

Table 1: Comparison of Hybrid Crypto-models

<table>
<thead>
<tr>
<th>Evaluation Parameters</th>
<th>Prior Hybrid Encryption Schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on Symmetric and Asymmetric techniques</td>
<td>×</td>
</tr>
<tr>
<td>Computationally Proficient</td>
<td>×</td>
</tr>
<tr>
<td>Optimal feasibility for bulky data sets (Audio, Video)</td>
<td>×</td>
</tr>
<tr>
<td>Non-repudiation and Fake Modifications</td>
<td>✓</td>
</tr>
<tr>
<td>Origin authenticity of sender and receiver</td>
<td>✓</td>
</tr>
<tr>
<td>Applicability of Forgery Attack</td>
<td>×</td>
</tr>
<tr>
<td>Applicability Password (session key) Guessing Attack</td>
<td>×</td>
</tr>
<tr>
<td>Customer Satisfaction</td>
<td>×</td>
</tr>
<tr>
<td>Access of Third party over Keys Digest (Hash values)</td>
<td></td>
</tr>
<tr>
<td>Memory Requirement</td>
<td>High</td>
</tr>
<tr>
<td>Electric Power (energy) consumption</td>
<td>High</td>
</tr>
</tbody>
</table>
Figure 1: Generic Hybrid Encryption System (GHES) Framework

Figure 2: Encryption Time of Different Cryptosystems
Figure 3: Key Generation, Signature Generation and verification Time

<table>
<thead>
<tr>
<th>Time in ms</th>
<th>Signature Generation</th>
<th>Signature Verification</th>
<th>Key Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RSA</td>
<td>ECC</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>13</td>
<td>170</td>
</tr>
<tr>
<td>157</td>
<td>235</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Decryption Time of Different Cryptosystems