Secure Data Encryption Through a Combination of AES, RSA and HMAC

Eman Salim Ibrahim Harba
Computer Unit and Internet
College of Arts, University of Baghdad
Baghdad, Iraq
emanharba_121212@coart.uobaghdad.edu.iq

***Abstract*—Secure file transfer based upon well-designed file encryption and authorization systems expend considerable effort to protect passwords and other credentials from being stolen. Transferring and storing passwords in plaintext form leaves them at risk of exposure to attackers, eavesdroppers and spyware. In order to avoid such exposure, powerful encryption/authentication systems use various mechanisms to minimize the possibility that unencrypted credentials will be exposed, as well as be sure that any authentication data that does get transmitted and stored will be of minimal use to an attacker. In this paper we proposed a method to protect data transferring by three hybrid encryption techniques: symmetric AES algorithm used to encrypt files, asymmetric RSA used to encrypt AES password and HMAC to encrypt symmetric password and/or data to ensure a secure transmitting between server-client or client-client from verifying in-between client and server and make it hard to attack by common attacked methods.**

Keywords: Cryptography; Data Encryption; AES; RSA; HMAC

I. INTRODUCTION

Key based encryption can be categorized into two types: symmetric key and asymmetric key [1]. The algorithms of symmetric-key, known as one-key, single-key, and private key encryption are a type of cryptography that uses a public and private algorithm to implement encryption/decryption. Commonly used symmetric algorithms include Blowfish, DES, TEA, TDES, IDEA, CAST5, AES (Rijndael), RC6, Two Fish, MARS and Serpent. The asymmetric key algorithms, also referred to as public key encryption, use two keys, one called the public key and on called the private key. Common asymmetric algorithms include: RSA, PGP, SSH and SSL [2]. In addition to the previous two types of encryption, there is a third type called hash which focuses on securing authentication. However traditional algorithms of this kind (such as SHA and MD5) have gradually become obsolete. In spite of their improvements, once an attacker is able to get a hash and successfully provide it to the authentication server, he can assume whatever security identity is associated with this hash [3]. To protect stored hashes, various tool have been proposed typically based on using LSA (Local Security Authority). Several works focus on enhancing hashing security, mainly employing Message Authentication Code commonly known as MAC. Typically, MAC is used between two sides that share a secret key in order to verify transferred data. This method is known as HMAC (Hash-based Message Authentication Code) [4].

HMAC provides the client and server with a private and public key. The public key is known, while the private key is known only in the specific server and client. The client generates a specific hash or HMAC for each request via combing the hashing and request data together with a private key and then transmitting it as portion of a request to server. When the server receives this request it will regenerate a unique HMAC. The client authorization is based on the comparison of the two HMACs. This technique is usually known as secret handshake [3]. The reason that makes HMAC safer compared to MAC is that the message and the key are hashed in separate steps. The sender appends an authentication tag to the data calculated as a function of shared key and the data. At the receiver side, the receiver recomputes and compares the authorization tag on the message received [4]. An effective wireless application protocol called WTLS has been presented in [5]. In [5], HMAC has been used in WTLS to support the unique demands for verification with security of high-level strength. The presented design is based upon used hash function type SHA-1. The results of the implementation of both the SHA-1 and HMAC are compared with other relevant studies, and it is shown that the proposed system performs better. In [2], authors provided an extensive comparison analysis of several existing symmetric cryptographic algorithms based upon their limitation, scalability, reliability, security, flexibility and their architecture. Through that analysis it was noticed that AES algorithm perform the best in terms of encryption performance, flexibility, security and memory usage. Although the other algorithms were also competent, most of them had a tradeoff between memory usage and the time required for encryption.

In [6], authors implemented image and text encryption along with decryption. The text encryption uses 128-bit key size and also plaintext. Every word or space is changed into an 8-bit sequence. Thus, maximum overall 16 positions are identified by this code. AES encryption algorithm in CFB mode is used for image encryption. The PKCS5Padding method is used. Results verify the superior speed of AES. In [1], authors presented a comparison study of encryption standards AES, DES and RSA considering different parameters

including memory usages and computation time. A cryptographic tool has been used for testing. Depending on the text files used and the experimental result it is shown that AES and DES algorithm uses minimum encryption time compare to RSA. In addition, DES has the lowest memory usage, and the time of encryption between AES and DES algorithm has a very minor difference. When data size increases then the asymmetric cryptographic algorithm performs slower compare to the symmetric algorithm. In [7], authors presented a new approach to saving files in the cloud. They used RSA and AES algorithms for protecting data and connection based upon various keys in encryption and decryption. They also used an SHA1 algorithm to protect the hash table of data. Their model handles the total cloud system security to protect data. AES is used for signing in and RSA to encrypt data on storage. The SHA1 function is used to hash the key on the system. A key managing center is used as a third party to send out keys in all stages.

In the present paper, three methods are combined: a symmetric-key algorithm type (AES) has been used to encrypt data, standard type asymmetric cryptography (RSA) has been used to encrypt AES key and hash type (HMAC) has been used in between the two sides (that share a secret key) in order to authenticate transferred information.

II. THEORETICAL BACKGROUND

A symmetric key algorithm is an encryption technique where both receiver and sender share the same key. DES and AES are the general types of Symmetric-key algorithms. Ciphers are performed as either stream or block ciphers. A block cipher considers inputs in blocks of plaintext rather than individual characters [8]. The main advantages of symmetric cryptography are strength (due to its speed it can encrypt vast amount of data) and availability whereas its main weaknesses are key implementation and management and limited distribution security [9]. Other block ciphers have been released and several have already been broken (e.g. FEAL [10]). The DES algorithm is a block cipher with a a 56-bit key and 64-bit block size. It involves 16-rounds of permutation and substitution. In every round, key and data bits are XORed, permutated, shifted, and sent through a group of lookup tables. Decryption is basically the same process, performed backwards [6]. It is considered vulnerable to attacks employing linear cryptanalysis, differential cryptanalysis and brute force. The AES algorithm is a 'symmetric block cipher' that can be decrypted using the original encryption key. AES uses 10, 12 or 14 rounds that involve mixing, substitution permutation, and key adding [11]. The size of the key may be 128 bits, 192 bits or 256 bits. It main weaknesses are cipher limitations, requiring additional code and cycles to be implemented on a smart card and the use of numerous tables and large code in terms of software. It is considered vulnerable to the XLS attack.

An asymmetric cryptography algorithm uses two keys which are often called public and private. The public one is utilized for encryption function and the private one is utilized for decryption. Both of these keys are mathematically related. The public key may be revealed without compromising system security [8]. Its main advantages are its strength and that it offers efficient scalability and that it allows the distribution of the public key. Its main weakness is that it is computationally intensive and slow (thousand times or more slower than the symmetric).

An RSA algorithm is a generally used public key algorithm. It is the most commonly used asymmetric cryptography algorithm. It can be used for the encryption of small blocks of data or in key exchange digital signatures. RSA uses a variable key size and a variable encryption block size. The key pair come from a highly large number, which is the product of 2 prime numbers whose selection is based on special rules. RSA is commonly used for developing secure communication channels and for digital signatures [6]. Its main disadvantages are the large number of bits required and the fact that if one knows enough of the parameters used, he can calculate the others. RSA is used in security protocols like: SSH-terminal connection security, email security (PGP), data security IPSEC/IKE-IP, service security (SILC-conferencing) and web security - transport data security (TLS/SSL). It is considered vulnerable to lattice based attacks such as attacks on small secret keys, attacks on stereotyped messages, factoring with a hint and factorization attacks.

Hash cryptography is a technique that does not use a key. Instead, a fixed-length hash value is computed based on the plaintext which makes it impossible to be recovered for either length of the plaintext or the contents. Hash algorithms are generally employed to produce a digital fingerprint of a file's contents. It is generally used by several OS to encrypt passwords [3]. SHA includes a variety of algorithms such as SHA-1 (SHA160), SHA256, SHA385 and SHA512, in which the number is the hash length in bits. SHA is used widely in several algorithms of public-key cryptographic, such as in Digital Signature Algorithm (DSA) [12]. The HMAC algorithm is a method to modify hash functions to give cryptographic security. Basically, a unique part of data is hashed together with a key in many steps so that increased effort will be required to reverse the function. The advantages of HMAC is that data can be authenticated through a certain key among the inputs to the function. In normal hash function, the result is always the same for a specific input, nevertheless because a HMAC uses the key as an additional input the output depends upon both the data and the key [4]. An HMAC can be used to check if some a part of the data has been modified or not. Whenever the parties share a key, they compute the HMAC of the data using this key each before transmitting and after reception. If HMAC of received message meets the expected value, then the key (or data) won't be modified and could be considered authentic and if HMAC is different from the expected value, then the key (or data) seems to have changed. The attacker can't easily customize the data and recalculate the HMAC even when simple hash functions are used. If the key is properly chosen, HMAC can be a rather powerful method [13].

III. SYSTEM MODEL

The proposed model depends on three hybrid functions that are combined with each other to produce a strong system

protection. The AES function is used to encrypt the data symmetric AES algorithm to encrypt files, asymmetric RSA to encrypt AES password and HMAC to encrypt the authentication between server-client or client-client [14]. Figure 1 illustrates the system.

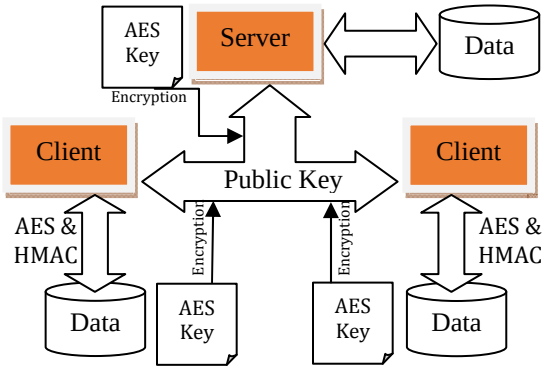

Fig. 1. Block diagram of the proposed system.

As shown, the data runs under three levels of security using the three algorithms in combination:

- Receiver uses the RSA algorithm to generate the key pairs (public and private keys), and send the public key value to the sender.
- Sender uses the AES algorithm to encrypt data using cipher keys of 256 bits length.
- Asymmetric-key algorithm (RSA algorithm) is used to encrypt the symmetric key with receiver's public key.
- A HMAC SHA256 is used to calculat4 the MAC that includes a hash function combination with a secret cryptographic key.
- The ciphertext produced by encrypting the plaintext then appends MAC from the encrypted plaintext.
- Receiver uses the stored key (private key) to decrypt the AES key, and after that decrypt data using the AES key.

AES is usually high speed and requires low RAM, so it is very useful for encrypting data, but since it's the same key for both encryption and decryption, there is a big problem of key transport from the encryption side (sender) to the decryption side (receiver). RSA is used to protect the encryption key. Instead of using an expensive RSA provider we generated a stand-alone RSA algorithm that generates two key (private and public). Private key stays on the receiver side and the public key is sent to the sender side. The sender will use this key to encrypt data. This will be useful against passive attackers but not against active attacks, so MAC is used to protect encrypted data. Supposing the MAC shared secret hasn't been compromised, we need to be capable to deduce if a given ciphertext is definitely authentic or has been forged. So, the flexibility of the cipher scheme ensures the elimination (through the MAC code) of any invalid cipher text. MAC doesn't give any information on the plaintext due to the fact that the output of the cipher appears randomly. Basically, it does not carry any structure from the plaintext to the MAC code.

The basic steps of the algorithm are as follows:

1. insert data for encryption.
2. generate RSA key pair
3. insert encryption key
4. use random salt to block pre-generated weak password attacks. The salt bit size is 64. At first salt1 is created then derived and used for crypto key in AES and also slat2 is created then derived and used for authentication key for HMAC.
5. apply AES with a 128 block bit size and 256 key bit size.
6. AES encryption and HMAC authentication of data

IV. Results

The sender and receiver operation is simulated by first using the RSA algorithm to generate the two keys (public and private) and save each one as two files. Figure 2 shows the values of the AES key, the RSA public key and the HMAC key. The encrypted key results are shown in Figure 3. The encryption program used (Figure 4) uses AES & HMAC algorithms to encrypt data. On the other hand, the symmetric key itself is encrypted through a key-encryption program by uploading the public key to it. The size and encryption time request is tested by using a text file or by typed text. The size in bytes is firstly calculated and then the results after applying AES algorithm with the defined symmetric key and then by applying HMAC algorithm. Figure 4 shows the text data before encryption and the result after applying AES and HMAC algorithm. As shown, size of the text has increased from 144 bits to 784 and the encryption time is under 1 second.

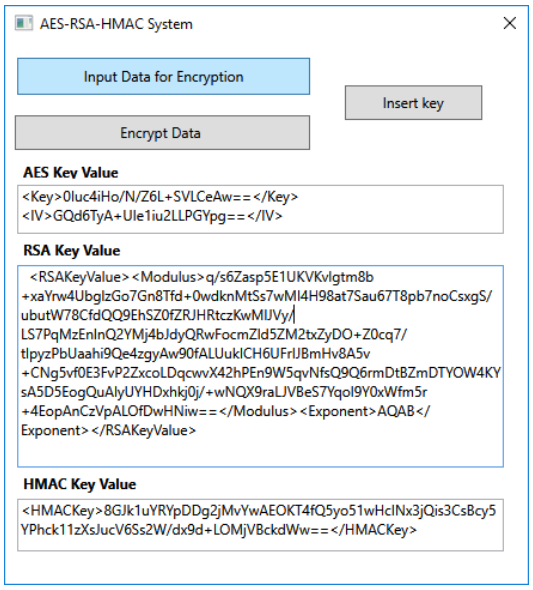

Fig. 2. Keys values of the three algorithms.

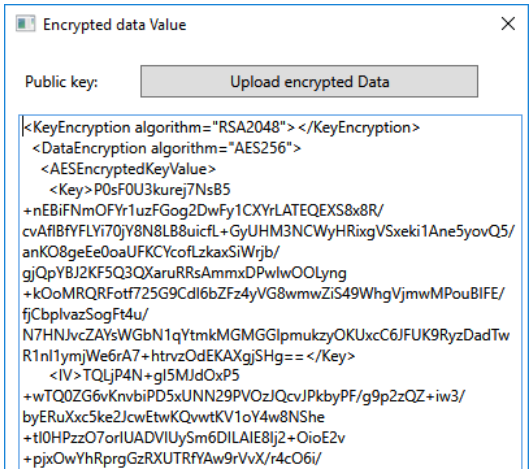

Fig. 3. Encrypted data value and uploading the public key.

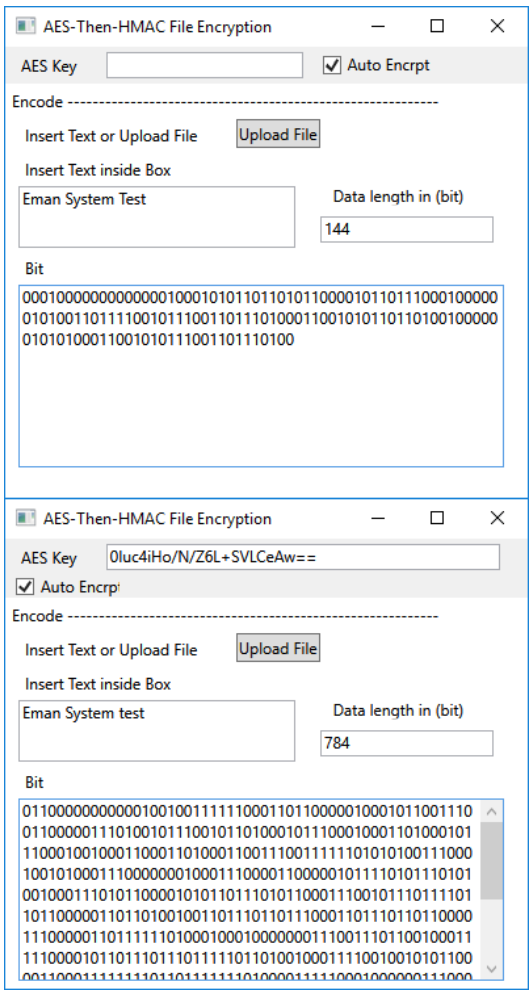

Fig. 4. Text data size. Up: before AES-HMAC encryption and down: after AES-HMAC encryption.

In the second test, a large text file (136800 byte) was used Figure 5shows the text data before encryption and after applying AES and HMAC algorithm. From Figures 4 and 5, it can be seen that there is a change in size due to adding encryption which is 256bit AES and 256bit HMAC, but the change in size is still very small and near the AES encryption size. The time requests are also near the AES encryption of the same condition.

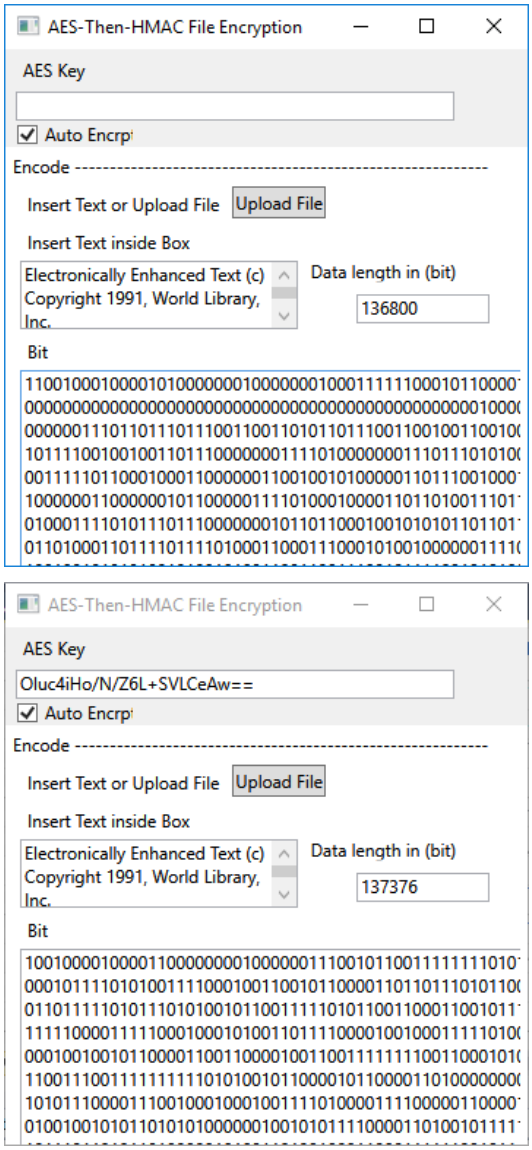

Fig. 5. Text data size. Up: before AES-HMAC encryption with and down: after encryption with AES-HMAC.\

Fig. 6. Conclusion

In this work, it three types of encryption have been combined in order to exploit the advantages of each one to build a high security system. AES is used to encrypt sent data, exploiting its high encryption speed and its low RAM requirements. RSA is used to protect the encryption key from getting stolen by generating two keys (a private and a public one). MAC is used to protect the encrypted key or the data. The key and encrypted data are sent to the receiver and get decrypted by using the private. The overall encryption run is

simple and fast with low computational requirements and provides high system security.

REFERENCES

[1] P. Prajapati, N. Patel, R. Macwan, N. Kachhiya, P. Shah, "Comparative Analysis of DES, AES, RSA Encryption Algorithms", International Journal of Engineering and Management Research, Vol. 4, No. 1, pp. 292-294, 2014

[2] M. Ebrahim, S. Khan, U. Bin Khali, "Symmetric Algorithm Survey: A Comparative Analysis", International Journal of Computer Applications, Vol. 61, No. 20, pp. 12-19, 2013

[3] G. C. Kessler, "An Overview of Cryptography" in the Handbook on Local Area Networks, Auerbach, 1998

[4] P. Jungles, M. Simos, B. Godard, J. Bialek, M. Bucher, C. Waits, W. Peteroy, T. Garnier "Defending Against Pass-the-Hash Attacks, Mitigating Pass-the-Hash and Other Credential Theft", The Microsoft Security Intelligence Report (SIR), Microsoft Corporation, 2014.

[5] R. Berry, K. Berry, A. Kumar, "Review on Network Security and Cryptography", International Journal of Innovative Research in Technology, Vol. 3, No. 7, pp. 44-53, 2016

[6] G. Selimis, N. Sklavos, O. Koufopavlou, "VLSI Implementation of The Keyed-Hash Message Authentication Code for The Wireless Application Protocol" 10th IEEE International Conference on Electronics, Circuits and Systems, United Arab Emirates, December 14-17, 2003

[7] M. S. Abutaha, A. A. Amro, "Using AES, RSA, SHA1 for Securing Cloud", International Conference on Communication, Internet and Information Technology, Madrid, Spain, 2014

[8] V. Agrawal, S. Agrawal, R. Deshmukh, "Analysis and Review of Encryption and Decryption for Secure Communication", International Journal of Scientific Engineering and Research, Vol. 2, No. 2, Art. No. J2013115, 2014.

[9] N. Settia, "Cryptanalysis of Modern Cryptographic Algorithms", International Journal of Computer Science and Technology, Vol. 1, No. 2, pp. 166-169, 2010.

[10] A. J. Menezes, P. C. van Oorschot , S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 2001

[11] D. P. Joseph, M. Krishna, K. Arun, "Cognitive Analytics and Comparison of Symmetric and Asymmetric Cryptography Algorithms" International Journal of Research Studies in Computer Science and Engineering, Vol. 2, No. 3, pp. 63-68, 2015

[12] Y. S. Solanki, "Performance Based Design and Implementation of a SHA-1 Hash Module on FPGA", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, No. 12, pp. 391-393, 2012

[13] C. Knopf, Cryptographic Hash Functions, Thesis, Leibniz Universität, Hannover Institut für Theoretische Informatik, 2007

[14] NIST, Secure Hash Standard, Federal Information, Processing Standards Publication 180-2, 2002