Multi-Level Encryption Framework

Ahmad Habboush
Faculty of Computer and Information Technology
Jerash University
Jerash, Jordan

***Abstract*—Multi-level encryption approaches are becoming more popular as they combine the strength of multiple basic/traditional approaches into a complex one. Many multi-level encryption approaches have been introduced for different systems, like Internet of Things, sensor networks, big data, and the web. The main obstacles in building such approaches are to have a secure as well as a computationally efficient multi-level encryption approach. In this paper, we propose a computationally efficient multi-level encryption framework that combines the strength of symmetric, the encryption algorithm AES (Advance Encryption Standard), Feistel network, Genetic Algorithm's Crossover and Mutation techniques, and HMAC. The framework was evaluated and compared to a set of benchmark symmetric encryption algorithms, such as RC5, DES, and 3-DES. The evaluation was carried out on an identical platform and the algorithms were compared using the throughput and running time performance metrics and Avalanche effect security metric. The results show that the proposed framework can achieve the highest throughput and the lowest running time compared to the considered benchmarked symmetric encryption algorithms and passes the avalanche effect criterion.**

Keywords*—Multi-level encryption; Advance Encryption Standard (AES); Feistel encryption; symmetric encryption algorithm

I. Introduction

Nowadays, different types of communication are available at the same time such as people to people, people to objects, and objects to objects. Communication is used worldwide and it is spread in almost all areas; such as industry, academia, health…etc. One of the main challenges in communication is security.

As the Internet or any network channels are considered to be unsafe, the aim of having security measurements is to ensure the data are transmitted and received without being stolen, manipulated or deleted. The most commonly used security measurement for network communication is encryption [1].

Encryption is the process of manipulating a plain text message into a ciphered one, usually this process is done with the use of a key. Based on the type of the key used in the encryption/decryption process, the encryption algorithms can be categorized in to symmetric key and asymmetric key. In symmetric key algorithms, the same key can be used to encrypt/decrypt a plain text. Consequently, these algorithms are secure and computationally efficient. Examples of such algorithms: AES, RC6, MARS, Bluefish, DES, and 3-DES [2]-[4]. On the other hand, asymmetric encryption algorithms require two keys for encryption and decryption, thus, making the generation of these keys computationally high and not efficient for encrypting large data. Example of asymmetric encryption algorithm is RSA [5].

With the advancement of technology and computation power, hackers have developed different models to attack such basic encryption algorithms. One of the solutions to deal with this problem is to build a complex model that consists of multi-level encryption algorithms [6]-[9]. The idea is to combine the strength of multiple basic encryption algorithms together to build a complex, sophisticated encryption approach. Two challenges are facing such approaches; computation efficiency and security. The earlier is needed when sending large data and the later ensures that the combination between these levels (algorithms) will lead into a complete system that is secure. Many multi-level encryption approaches were proposed in literature, but they were suffering from computational or security problems or being specified for a certain type of networks.

In this paper, we propose a multi-level encryption framework that is secure and computationally efficient. The framework consists of using a symmetric encryption algorithm AES, Feistel network, Crossover and Mutation techniques from genetic algorithm, and HMAC for encrypting and decrypting data.

The proposed framework was evaluated against benchmarked symmetric encryption algorithms like: RC6, DES, and 3-Des for performance and security metrics. In terms of performance, the proposed framework is compared against them using throughput and running time metrics. The results show that the proposed framework has the highest throughput and the lowest running time. For security purposes, the Avalanche effect is used. The proposed framework has the highest avalanche effect and passes that criterion.

The rest of the paper is as follows: Section 2 review some related work. Section 3 introduces the system framework. Section 4 presents the experimental evaluation. Finally, Section 5 concludes the paper.

II. Related Work

A. Selecting a Template

Many algorithms and approaches have been proposed to deal with the encryption/decryption problem. Besides the classical symmetric and asymmetric encryption algorithms, new encryption algorithms categories are being introduced in recent years. Some of these categories include: Homomorphic encryption [10]-[14], attributes-based encryption [15]-[20],

and multi-level encryption algorithms. In this section, some recent approaches that uses multi-level encryption are introduced.

S. Aljawarneh, et al. [6] proposed a resource efficient encryption algorithm for multimedia big data. The algorithm is composed of a framework with multi-level encryption that includes: Fiestel encryption scheme, AES with Sbox, and genetic algorithm. There is no key generation since the key is generated from the plain text. The authors propose a multi-threaded version of the scheme to increase performance [7]. The main drawback of this scheme is that the scheme does not preserve confidentiality. If two senders are using this scheme, they can encrypt/decrypt the messages from the other sender.

S. Masadeh, et al. [8] introduced a multi-level security approach that consists of PGP, WIFI, and HMAC systems for authentication and communication encryption. The approach uses the PGP to generate messages and the message is encrypted using the public key of the receiver.

S. Aljawarneh, et al. [9] proposes a multi-level encryption system for WiFi called secure WiFi. The system is composed of Feistel encryption and HMAC. Their proposed system uses 64bit blocks to be encrypted which are not suitable for encrypting large data.

III. System Model

The system model is similar to the one proposed by [6] with an additional security level to solve their confidentiality problem. Fig. 1 shows our proposed system model. The original file to be encrypted is split into chunks, where each one has the size of 256bit. The 256bit plaintextchunksis divided into two equal 128bit blocks. The first block is encrypted using a 3-level non-key Feistel network. This encryption is composed of bit and bytes shifting and rotation processes. The second 128bit block is encrypted using Advance Encryption Standard (AES) where the key for this encryption is the output of the Feistel network encryption. The third step is to use the crossover and mutation techniques from genetic algorithm to scrapple the two blocks together. Lastly, we use HMAC hash function to encrypt the result using a hash key. The resulted encrypted file is sent over network channel to the receiver for encryption. The decryption process is opposite of the encryption process. Next, we explain these levels in more details.

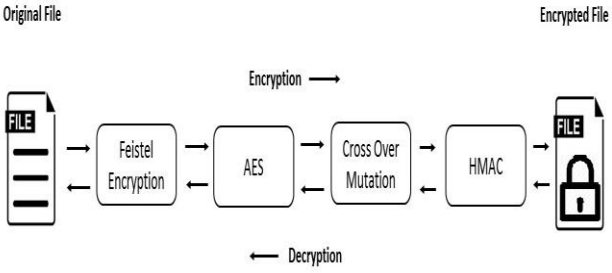

Fig. 1. Proposed framework.

A. Feistel Encryption

As a start, the file is divided into 256bit chunks and each chunk is encrypted using the proposed framework. For a chunk, the first step is to split it into two 128bit blocks (right and left). The left block is encrypted using 3-level Feistel network as shown in Fig. 2. The first level splits the block into individual 16 bytes and defines the highest and lowest bits within a byte. The second level shifts right the highest bits within each byte, so they become the lowest bits. At the same time, the lowest bits of each byte are rotated to become the highest bits of another byte as shown in the figure. Lastly, the bytes are shifted and rotated and combined to form a ciphered key that will be used in the AES encryption.

B. Advance Encryption Standard (AES)

The Advance Encryption Standard (AES) is a symmetric encryption algorithm that needs only one key for encryption and decryption. AES is considered to be fast and secure encryption algorithm. It uses substitution and permutation network where it iterates the encryption process for 10 rounds to generate the cipher text. In AES, the size of the key is very crucial since it will determine the number of encryption rounds. In general, AES uses 10 rounds for 128bit key and 12 rounds for 192bit key.

In our proposed framework, AES will use the ciphered key generated from the Feistel encryption as a key to encrypt the right 128bit block. The result will be an encrypted version of the right block.

AES consists of three main stages: Add Round Key, Rounds, and Final Round. The add key round or the initial round uses a bitwise XOR to combine the block of the round key with each byte of the state.

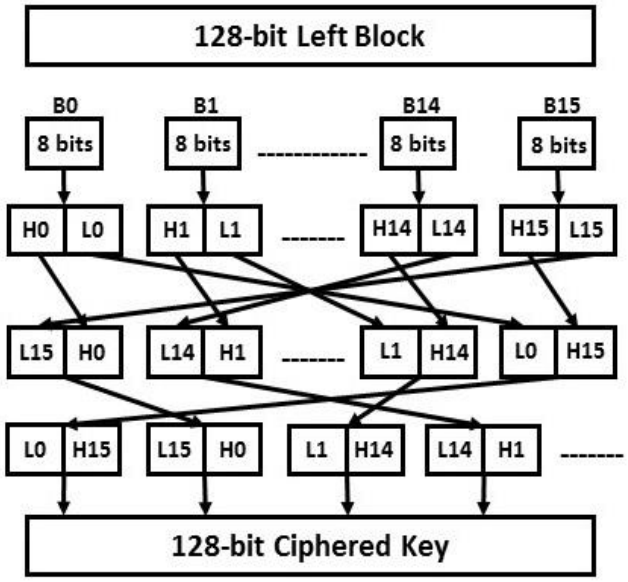

Fig. 2. Feistel encryption.

The rounds stage is mainly consist of four main substages: the sub-byte process, shift rows process, mix columns process, and the add round key process. In our framework, these substages are repeated for 8 times. Lastly, the final round

which is similar to one round of the rounds stage with a small difference that the mix columns process substagewill not be performed. By the end of the AES encryption level, both left and right blocks are encrypted.

C. Cross Over and Mutation

The aim of this stage is to scatter the encrypted block generated from the Feistel encryption and the one generated by the AES. The cross over and mutation techniques are parts of genetic algorithm. The cross over is an operation where two blocks are changed together in a random way. Fortwo blocks, cross over starts by randomly choosing the cross over points. These points are recorded and stored for the decryption process. Next, the blocks from the cross over points to the least significant bit are interchanged. For example: for two blocks 010100101011 and 100101010101. If the cross over point is 8, the resulting blocks are: 010100100101 and 100101011011.

The mutation operation is used in genetic algorithm to maintain the generic diversity of results from one generation to another in a random way. For two blocks, the mutation operation starts by selecting one mutation point in a random way. At each block, the bit corresponding to the mutation point is flipped and the mutation point is recorded for the decryption process. An example of the mutation operation: for two blocks 010100101011 and 100101010101. If the mutation point is 8 then the resulting two blocks are 010100100011 and 100101011101.

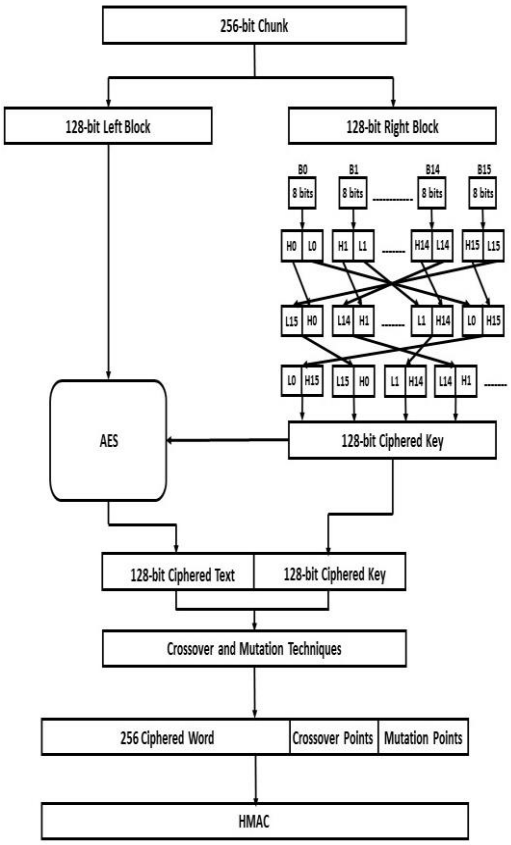

Fig. 3. Architecture of the proposed encryption framework.

The crossover and mutation operations are repeated for many times. In our experiments we repeat these operations for 5 times. By the end of this stage, the encrypted data that need to be sent to the receiver includes: the encrypted 256bit chunk, the cross over points, and the mutation points.

D. HMAC

HMAC is message authentication code that involves the use hash function and key. The strength of HMAC is based on the strength of the hash function used. In our proposed framework we use SHA-1 hash function for authenticate the encrypted message. The aim of this stage is to maintain confidentiality of the encrypted messages. By the end of this stage, the encrypted message is ready to be sent to the receiver. Fig. 3 shows the architecture of the encryption part of our proposed framework.

Note that the decryption process is the opposite of the encryption process. The receiver just need to have the key of for HMAC and perform the decryption in a reverse order of the encryption process. Fig. 4 presents the architecture of the decryption process of our proposed framework.

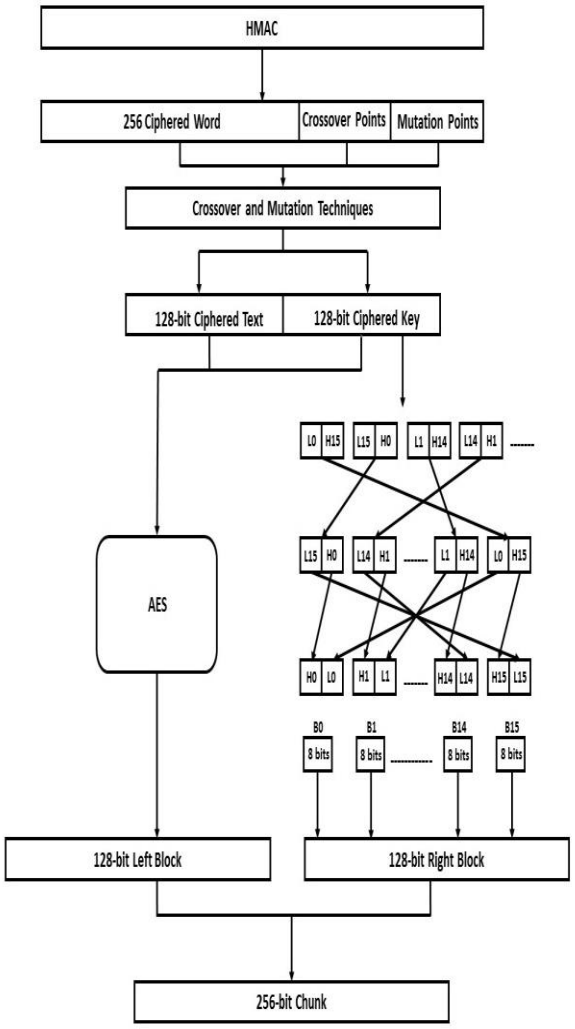

Fig. 4. Architecture of the proposed decryption framework.

IV. Experimental Evaluation

The evaluation is conducted to show the efficiency and performance of the proposed framework. The proposed framework was implemented using JAVA. All experiments were conducted on an identical platform; a Windows based machine that is equipped with 32 Gb of memory and an Intel i7 4770 3.4 GHz CPU.

The proposed framework results were compared to a number of symmetric encryption algorithms such as CR6, DES, 3-DES using file sizes ranges from 1MB to 1GB. The encryption running time and throughput performance metrices are used in this evaluation. For security evaluation, we compare the Avalanche effect security metric.

Fig. 5 (a)-(e) illustrates the encryption running time for our proposed framework compared to RC6, DES, and 3-DES encryption algorithm for files range from 1 MB to 1GB. The results clearly show that the proposed framework outperforms all other encryption algorithms for file sizes 1 MB to 1 Gb. AES is very fast and secure as it has very strong resistance against attacks. On the other hand, RC6 is less secure, needs more rounds, and it uses extra multiplication operation that increases the encryption running time. DES uses a 56bit key, has a S-Box structure where the encryption operation needs to use a lookup mechanism. This lookup mechanism will lead to a slow software implementation. The 3-DES is the slowest encryption algorithm, it uses 168bit key size and it runs DES three times. Fig. 6 shows the average encryption running time for all ranges.

The encryption process throughput for files range from 1MB to 1GB is shown in Fig. 7 (a)-(e). Similar to the encryption running time, our proposed framework has the highest throughput in comparison with the other encryption algorithms. Two main factors affect the throughput: the file size and running time. As our proposed framework has the lowest running time, it is expected that it will provide the highest throughput. It should be noted that the results only show the encryption process since the decryption process is just the reverse of the encryption process and it provides almost the same results. Fig. 8 shows the average encryption throughput for all ranges.

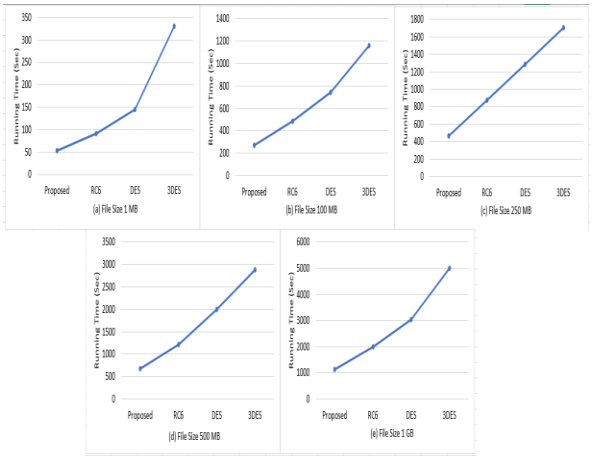

Fig. 5. Average Encryption part running time for file ranges: (a) 1MB (B) 100MB (c) 250MB (d) 500MB (e) 1GB.

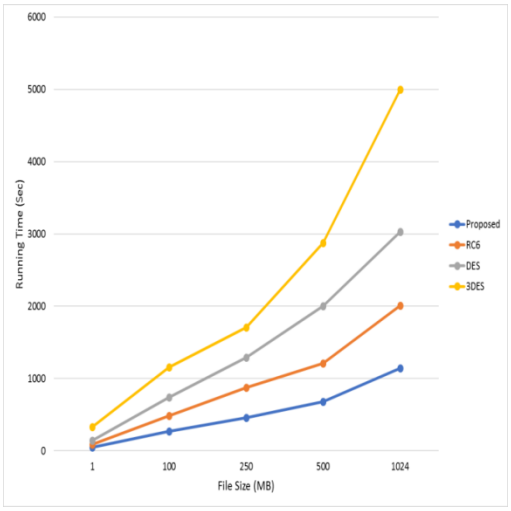

Fig. 6. Average encryption running time for all ranges.

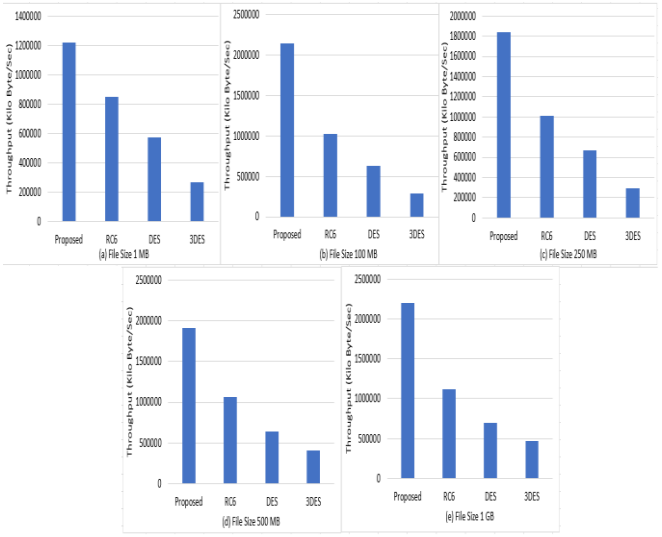

Fig. 7. Average throughput for file ranges: (a) 1MB (B) 100MB (c) 250MB (d) 500MB (e) 1GB.

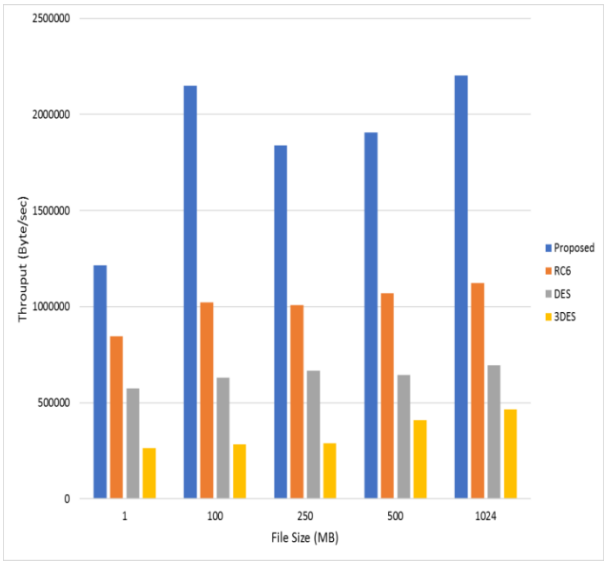

Fig. 8. Average encryption throughput for all ranges.

To evaluate security, the Avalanche effect is calculated for our proposed framework and other symmetric encryption algorithms. The idea of the Avalanche effect is to show the effect of changing one bit in the plaintext to the ciphertext. This will show how solid the encryption algorithm is against cracking and hacking threats and real time attacks. For an algorithm to pass the avalanche effect criteria, changing one bit in the plaintext is expected to affect half of the bits in the cipher text [21].

Fig. 9 shows the avalanche effect of the proposed framework compared to benchmarked algorithms; RC6, DES, 3-DES. The avalanche effect of our proposed framework was 58% compared to 44% for RC6, 27% for DES, and 35% for 3-DES. From these results, it can be seen that the proposed framework is the only encryption that satisfies the avalanche effect criterion.

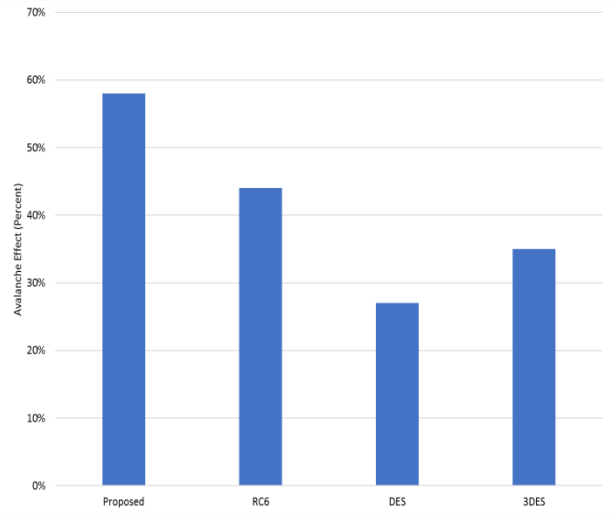

Fig. 9. Avalanche effect.

V. Conclusion

Multi-level encryption algorithms are popular since they combine the strength of many encryption techniques at the same time. In this paper, we proposed a multi-level encryption framework that combines the strength of Feistel encryption, AES, Crossover and mutation, and HMAC. The framework was evaluated against symmetric encryption algorithm CR6, DES, 3DES for performance and security metrics. The results show that the proposed framework has the lowest running time, highest throughput, and passes the Avalanche effect criterion.

References

[1] William Stallings "Network Security Essentials (Applications and Standards)", Pearson Education, 2004.

[2] Mohan, H.S. and Reddy, A.R., 2011. Performance analysis of AES and MARS encryption algorithms. IJCSI International Journal of Computer Science Issues, 8(4), pp.1694-0814.

[3] Ebrahim, M., Khan, S. and Khalid, U.B., 2014. Symmetric algorithm survey: a comparative analysis. International Journal of Computer Applications (0975 – 8887) Volume 61– No.20, January 2013.

[4] Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard." Dr. Dobb's Journal, March 2001.

[5] Vincent, P.D.R., 2016. RSA Encryption Algorithm-A survey on its various forms and its security level. *International Journal of Pharmacy and Technology*, *8*(2), pp.12230-12240.

[6] Aljawarneh, S. and Yassein, M.B., 2017. A resource-efficient encryption algorithm for multimedia big data. Multimedia Tools and Applications, 76(21), pp.22703-22724.

[7] Aljawarneh, S. and Yassein, M.B., 2017. A multithreaded programming approach for multimedia big data: encryption system. Multimedia Tools and Applications, pp.1-20.

[8] Masadeh SR, Azzazi A, Alqaralleh BA, Al Sbou AM. A novel Paradigm In Authentication System Using Swifi Encryption/ Decryption Approach. International Journal of Network Security & Its Applications. 2014 Jan 1;6(1):17.

[9] Aljawarneh S, Masadeh S, Alkhateeb F. A secure wifi system for wireless networks: an experimental evaluation. Network Security. 2010 Jun 30;2010(6):6-12.

[10] Brakerski, Z. and Vaikuntanathan, V., 2014. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, *43*(2), pp.831-871.

[11] Brakerski, Z., Gentry, C. and Vaikuntanathan, V., 2014. (Leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT), 6(3), p.13.

[12] Ducas, L. and Micciancio, D., 2015, April. FHEW: bootstrapping homomorphic encryption in less than a second. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 617-640). Springer, Berlin, Heidelberg.

[13] Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M. and Wernsing, J., 2017. Manual for using homomorphic encryption for bioinformatics. Proceedings of the IEEE, 105(3), pp.552-567.

[14] Lepoint, T. and Naehrig, M., 2014, May. A comparison of the homomorphic encryption schemes FV and YASHE. In International Conference on Cryptology in Africa (pp. 318-335). Springer, Cham.

[15] J. S. Su, D. Cao, X. F. Wang, Y. P. Su, and Q. L. Hu, "Attribute-based encryption schemes," Journal of Software, vol. 6, pp. 1299–1315, 2012.

[16] Q. Tang and D. Ji, "Verifiable attribute-based encryption," International Journal of Network Security,

vol. 10, no. 2, pp. 114–120, 2010.

[17] G. Wang, Q. Liu, and J.Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proceedings of the 17th ACM conference on Computer and communications security, pp. 735–737, 2010.

[18] G.Wang, Q. Liu, J.Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," Computer & Security, vol. 30, pp. 320–331, 2011.

[19] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," Public Key Cryptography V PKC, vol. 6571 of LNCS, pp. 53–70, 2011.

[20] Yao X, Chen Z, Tian Y. A lightweight attribute-based encryption scheme for the Internet of Things. Future Generation Computer Systems. 2015 Aug 1;49:104-12.

[21] Heys HM, Tavares SE. Avalanche characteristics of substitution-permutation encryption networks. IEEE Transactions on Computers. 1995 Sep;44(9):1131-9.