Mathematics and Computer Science
2017; 2(3): 31-34
http://www.sciencepublishinggroup.com/j/mcs
doi: 10.11648/j.mcs.20170203.12

Hybrid Symmetric Volatile Encryption Algorithm Using Array

Mostafa Al-Romi, Raid Al-Malki

Department of Computer Science and Information System, AlMa'arfa Colleges of Science and Technology, Riyadh, Kingdom Saudi Arabia

Email address:
131120277@student.mcst.edu.sa (M. Al-Romi), 141120449@student.mcst.edu.sa (R. Al-Malki)

To cite this article:
Mostafa Al-Romi, Raid Al-Malki. Hybrid Symmetric Volatile Encryption Algorithm Using Array. *Mathematics and Computer Science.* Vol. 2, No. 3, 2017, pp. 31-34. doi: 10.11648/j.mcs.20170203.12

Received: May 11, 2017; **Accepted**: May 24, 2017; **Published**: July 7, 2017

Abstract: This paper demonstrates the symmetric key algorithm and describes how this algorithm works. Also, discuss the most common challenges that facing this cryptosystem and how to solve these aspects, and discuss two major kinds of attackers, also discuss some prior works that related with Symmetric Key Algorithm, also present the proposed solution.

Keywords: Symmetric Key, A Public Key, Cryptosystem, Double Key, Asymmetric Key

1. Introduction

The great technological progress, the development of communication media, and various communication link the world with each other depended on sending various types of data through the networks [1]. All of this progress in the communication field, and exchange a huge of data, led to a threat to leak this data [2]. Also, this progress led to access the data by the attackers, and thus became the urgent need to preserve the security of this information [1, 2]. Also diversified this information on the degree of confidentiality from regular public data to government information and statistics, state budgets and very serious and confidential information [3].

2. Problem Identification

All kinds of information and data have been transmitted and saved often through computer networks in different types and locations. The progress of transmitting a huge of information lead to a significant threat to the safety and security of data transferring in case there is any threat in Symmetric Key Algorithm [4].

2.1. Attackers Type

Generally, most security methods work to avoid and detect some types of attackers, figure 1 shows main major types of these attackers.

2.1.1. Passive Attack

Is an attack that does not produce an unauthorized change in data, such as when the attacker reviewing or copying the data [5]. The most prominent example of the Passive attack is (Sniffing) Sniffing is Sometimes called "Listening", and it's happening in networks when the attacker catches the packets and analyzed it to find out information about the devices for each of the sender and receiver, and possibly the content of the packets [5].

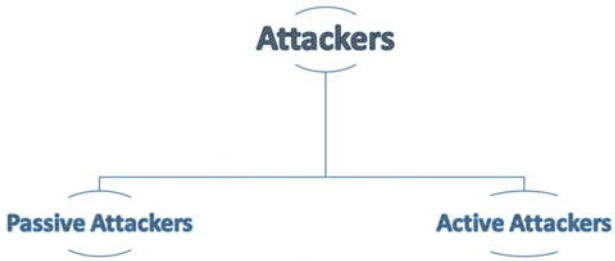

Figure 1. *Attackers Types.*

2.1.2. Active Attack

An attack which has produced an unauthorized change in data, such as manipulation, add or delete the files [6]. The most prominent example of active attack is (Spoofing) Spoofing is the attack which is based on falsification of identities to control the data [6].

2.2. Cryptography Methods

Cryptography is a method of encrypting and decrypting the

data. Cryptography system is a science to protect the information from the attacker by encryption the data [4]. Cryptography can make you transmit the data over untrusted network like the internet with some protection [4]. There are two main types of Cryptography methods:

2.2.1. Symmetric Key

Symmetric Key cryptosystem is an algorithm that using a single key as shown in figure 2 [3].

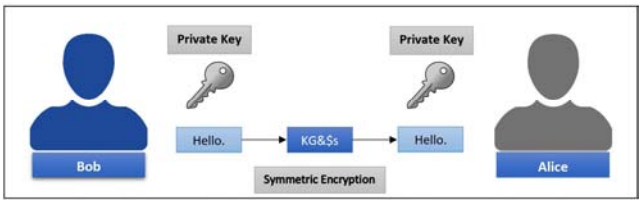

Figure 2. *Symmetric Key Cryptosystem.*

2.2.2. Asymmetric Key

Asymmetric key cryptography is also named as double key due to use two types of keys as shown is figure 3 [7].

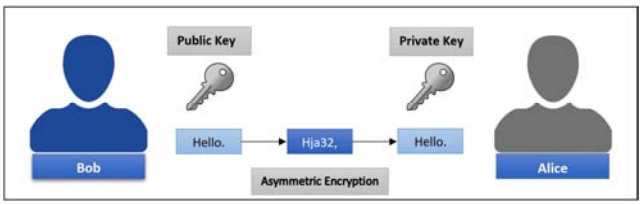

Figure 3. *Asymmetric Key cryptography.*

This paper will discuss Symmetric Key cryptosystem, and how it's work and some point that revolves around this topic.

Symmetric Key cryptosystem is an algorithm to encrypt the transmitted data by using the same key (Single key) for both encrypting and decrypting to establish a secure communication. Both sender and receiver must select a symmetric key Algorithm. This requires a secure key that sends from sender to receiver separately. that take variable length input string and converts it into a fixed-length binary sequence[1, 3]. Figure 4 describe transfer the data via different mediums by using Symmetric key cryptosystem way.

Figure 4. *Transfer The Data Using Symmetric Key Cryptosystem.*

This paper will discuss the trust and the keys distribution issues. The sender and receiver share a symmetric key there will be implied condition between each other it is (trust). So if the receiver lost the key to an attacker the sender may be not informed [2, 3]. On another hand, if any attacker listening to the connection the attacker may catch the key due to that the key sending separately from the message [2]. These challenges are facing a modern communication today. The people today needs to exchange information and communicate with each other with more trust way.

The organization of this paper will be as follows; Symmetric Key Cryptosystem aspects will be covered in section 2, then the related work will be in section 3. After that the proposal algorithm in section 4. Finally the conclusion in section 5.

3. Prior Works

Symmetric Key is Also known as Private Key Encryption, where the parties must agree on the same encryption key, which often sent to each other across the network.

In [8] Symmetric Key has two types of encryption algorithms, including stream ciphers and block ciphers, these types provide bit by bit and block encryption one by one.

“Aldar” discuss Hill Cipher and Iterated Hill Cipher (IHC) encryption schemes, which enable data to be processed directly in encrypted form. Moreover, these schemes have a good property, that encrypted data can have a number of different representations [9].

In [10] “Himani” and “Monisha” discuss various symmetric cryptosystems, including Blowfish cipher which is very fast cryptosystem, and one of the tremendous encryption algorithms, which encrypts data on 32-bit microprocessors at a rate of 18 clock cycles per byte, and has a key length that can vary from 32 bits to a maximum of 448, and can run in less than 5 KB of memory.

“Asoke”, “Saima” and “Meheboob” examine and use Random Key Encryption Algorithm, which is essentially streaming cipher method, and it may take a huge amount of time if the file size is large, also the encryption number is large especially that matrix may be generated in 256 Ways [11].

In [12] the authors investigate the block cipher implementation in Reshape hardware has been presented and a wide range of block ciphers was examined. This description led to a set of requirements used to develop COBRA, a Reshape architecture designed to achieve efficient block cipher implementations.

“Burke”, “McDonald” and “Austin” discusses DES, AES, and Blowfish: Symmetric Key Cryptography Algorithms Simulation-Based Performance Analysis. The authors focus on providing a performance analysis of symmetric key cryptography algorithms: Data Encryption Standard (DES), Advanced Encryption Standard (AES), and Blowfish [13].

“Thakur” and “Kumar” conclude that in present work they use the maximum encryption number = 64 and maximum randomization number = 128. The present work is an exchange method and can be used to replace a character by any of the 256 characters [14].

In [15] the authors performed a comparison test by implementing both symmetric-key and public-key primitives on the popular sensor motes. Experiment results indicate public key based protocol is more advantageous than the symmetric key in terms of the memory usage, message complexity, and security resilience.

4. Proposed Solution

This paper discusses some issues that impact negatively on sending and receiving the data by secure way using Symmetric Key Algorithm. In this section, there are two algorithms that will use to solve the problems that mentioned in problem definition section, first algorithm is Encrypt Key Using Array (EKUA) which it works to encrypt the key, and the second algorithm is Decrypt Key Using Array (DKUA) which it works to decrypt the key by receiver.

When the sender sends multiple messages with a single key using Symmetric Key Algorithm, all messages are at risk if the key is caught by the attacker, on the other hand, if the receiver lost the key to an attacker all messages will be at risk.

The proposed solution is a way to safely transfer the key through using the array and save only one character in each index. This process is intended to perform some operations that affect only the odd or even indexes based on the time of the session. For example: if the session time is (02:38:57) the operation will execute on the odd indexes due to the one's category in the seconds is (7).

After selecting the target indexes, now we will execute an algorithm to encrypt the key before initiating the connection. Figure 5 demonstrate the mathematical algorithm for encrypting the key.

$$EIV = IV_R \times (S \div I_N) \quad (1)$$

EIV = Encryption Index Value
IV_R = Index Real Value
S = second
I_N = Index number

This algorithm work on encrypting the indexes except index zero, where the IEV is the value of index after executing the algorithm, and IV_R is the American Standard Code for Information Interchange (ASCII) code of the index value, and S is the second number that given from transfer time in the packet, S will be changing to (10) if the value of it equal zero, and finally I_N is the number of selected index.

The decryption of the key in the destination side will use the same parameter but in the different formula to retrieve the original values of the key and make it valid to decrypt the packet but in the different formula. Figure 6 will show the algorithm that using to decrypt the encryption key.

$$IV_R = IEV \div (S \div I_N) \quad (2)$$

This algorithm will help to make symmetric key more secure by using the indexes of the array which it changing every time with different ways.

4.1. Algorithms

Figure 5 demonstrate the flow chart of (EKUA) algorithm to demonstrate how the algorithm 1 works.

Figure 6 show the flow chart of (DKUA) algorithm to demonstrate how the algorithm 2 works.

Start EKUA
Input: Real Array, Send Time (ST)
Output: Encrypted Array
Read: ST
    If (Second is even) then
        $i = 2$
    Else
        $i = 1$
    End if
For (is i less than Array length? step+2)
    $EIV = IV_R \times (S \div i)$ ..... (1)
    Next i
End for
End EKUA

Algorithm 1. *Encrypt Key Using Array (EKUA).*

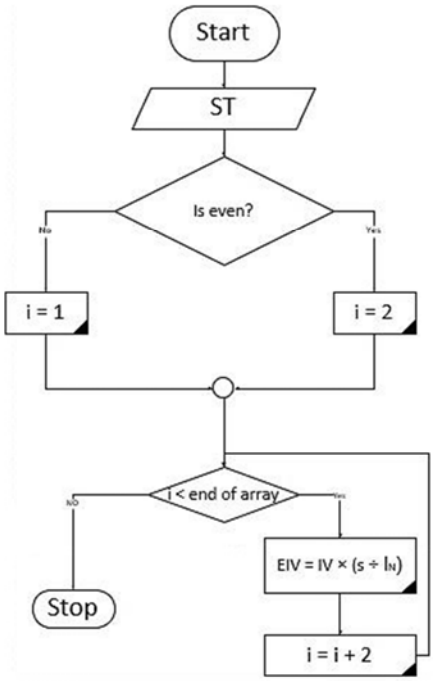

Figure 5. *Flow Chart of Encrypt Key Using Array (EKUA).*

Start DKUA
Input: EKUA, Send Time (ST)
Output: Decrypted Array
Read: ST
    If (Second is even) then
        $i = 2$
    Else
        $i = 1$
    End if
For(is i less or equal than Array length? step+2)
    $IV_R = IEV \div (s \div i)$ ..... (2)
    Next i
End for
End DKUA

Algorithm 2. *Decrypt Key Using Array (DKUA).*

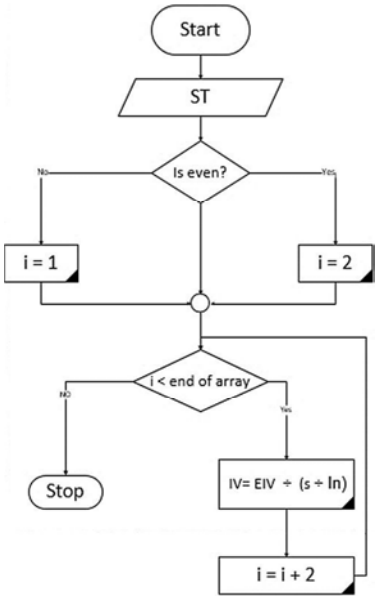

Figure 6. *Flow Chart of Decrypt Key Using Array (DKUA).*

4.2. Example

This example will explain how the EKUA and DKUA will works:

Table 1. *Real Array.*

1	2	3	4
A	B	C	D

'A' in ASCII code = 41, 'B' in ASCII code = 42, 'C' in ASCII code = 43, 'D' in ASCII code = 44.

ST = 20:44:22
The seconds is even!
The above array will be encrypted by using equation (1):

$$EIV = 42 \times (22 \div 2) = 462$$

$$EIV = 44 \times (22 \div 4) = 242$$

Then the algorithm will save these values in index one and three.

Table 2. *Encrypted Array.*

1	2	3	4
A	441	C	231

Then send it to the receiver after that receiver will decrypt the array by using equation (2):
$IV_R = 462 \div (22 \div 2) = 42$ this value = B in ASCII code.
$IV_R = 242 \div (22 \div 4) = 44$ this value = D in ASCII code.

5. Conclusion

The proposed solution discusses the symmetric key trust and key distribution aspects, the salvation depends on place the encryption key to an array with one character in each field, and chooses the target indexes based on the session time value, and use the proposed algorithm to encrypt the key. This will make the key transferring via a network by encrypted form. In the destination side, the decryption algorithm will change each value to the original value to make the key valid to decrypt the message.

References

[1] Saranya, K., R. Mohanapriya, and J. Udhayan, *A review on symmetric key encryption techniques in cryptography.* International Journal of Science, Engineering and Technology Research, 2014. 3 (3): p. 539-544.

[2] Rejani, R. and D. Krishnan, *Study of symmetric key cryptography algorithms.* International Journal of Computer Techniques, 2015. 2 (2): p. 45-50.

[3] Chandramathi, S., et al., *An overview of visual cryptography.* International Journal of Computational Intelligence Techniques, ISSN, 2010: p. 0976-0466.

[4] Al-Vahed, A. and H. Sahhavi, *An overview of modern cryptography.* World Applied Programming, 2011. 1 (1): p. 3-8.

[5] Serjantov, A. and P. Sewell. *Passive attack analysis for connection-based anonymity systems*. in *European Symposium on Research in Computer Security*. 2003. Springer.

[6] Gilbert, H., M. Robshaw, and H. Sibert, *Active attack against HB/sup+: a provably secure lightweight authentication protocol.* Electronics Letters, 2005. 41 (21): p. 1169-1170.

[7] Wu, Q., et al. *Asymmetric group key agreement*. in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 2009. Springer.

[8] Surya, E. and C. Diviya, *A Survey on Symmetric Key Encryption Algorithms.* International Journal of Computer Science & Communication Networks, 2012. 2 (4): p. 475-477.

[9] Chan, A.-F. *Symmetric-key homomorphic encryption for encrypted data processing*. in *Communications, 2009. ICC'09. IEEE International Conference on*. 2009. IEEE.

[10] Agrawal, H. and M. Sharma, *Implementation and analysis of various symmetric cryptosystems.* Indian Journal of Science and Technology, 2010. 3 (12): p. 1173-1176.

[11] Nath, A., S. Ghosh, and M.A. Mallick. *Symmetric Key Cryptography Using Random Key Generator*. in *Security and Management*. 2010.

[12] Elbirt, A. J. and C. Paar, *An instruction-level distributed processor for symmetric-key cryptography.* IEEE Transactions on Parallel and distributed Systems, 2005. 16 (5): p. 468-480.

[13] Burke, J., J. McDonald, and T. Austin, *Architectural support for fast symmetric-key cryptography.* ACM SIGARCH Computer Architecture News, 2000. 28 (5): p. 178-189.

[14] Thakur, J. and N. Kumar, *DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis.* International journal of emerging technology and advanced engineering, 2011. 1 (2): p. 6-12.

[15] Wang, H., et al. *Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control*. in *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*. 2008. IEEE.