**Mathematics and Computer Science**
2017; 2(3): 27-30
http://www.sciencepublishinggroup.com/j/mcs
doi: 10.11648/j.mcs.20170203.11

# Symmetric Volatile Shared Key Encryption: A Two-way Communication Shared Key Encryption

**Audai Al-Fandi, Yazeed Al-Howifer**

Department of Computer Science and Information System, Alma'Arefa Colleges of Science and Technology, Riyadh, Saudi Arabia

**Email address:**
121120527@student.mcst.edu.sa (A. Al-Fandi), 121120529@student.mcst.edu.sa (Y. Al-Howifer)

**To cite this article:**
Audai Al-Fandi, Yazeed Al-Howifer. Symmetric Volatile Shared Key Encryption: A Two-way Communication Shared Key Encryption. *Mathematics and Computer Science*. Vol. 2, No. 3, 2017, pp. 27-30. doi: 10.11648/j.mcs.20170203.11

**Received**: May 11, 2017; **Accepted**: May 25, 2017; **Published**: July 7, 2017

**Abstract:** When it comes to the subject of encryption of communication streams, two key exchange methods are widely used asymmetric and symmetric key algorithms. However, neither of these methods is perfect. This paper will an algorithm that changes keys periodically without exchanging them between parties of the communication, the paper will discuss how this algorithm will solve some of the drawbacks of the widely used asymmetric and symmetric key algorithms.

**Keywords:** Encryption, Asymmetric, Symmetric, Volatile Keys, Communication Algorithms

## 1. Introduction

One of the main security layers in every organization is communication security. And to ensure the security of communication, all necessary sorts of communication must be encrypted. This paper will discuss one of the most secure types of encryption, the asymmetric key Algorithm [1].

In 1970, British cryptographer James H. Ellis introduced the concept of "non-secret encryption" but could not implement the concept. Three years later, a colleague of Ellis, Clifford Cocks implemented the concept into what is known now as RSA encryption algorithm (which is made up of the initials of the names of the scientist who created it; Ron Rivest, Adi Shamir, and Leonard Adleman) [2]. However, this discovery was classified until 1997. The first time asymmetric key cryptosystem has been introduced to the public was in 1976 by Whitfield Diffie and Martin Hellman [3].

Since the introduction of asymmetric key algorithms to public, a variety of encryption algorithm was introduced like digital signature [4], key agreement [5] and other techniques that have been developed in the field of public key cryptography.

## 2. Problem Identification

Figure 1 demonstrates how Symmetric key works; sender and receiver agree on using a certain key that both parts know, the key will be used to both encrypt and decrypt the key.

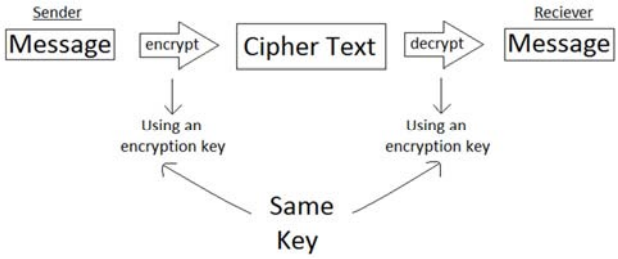


***Figure 1.*** *How Symmetric Key works.*

This approach is safe if the connection is secure and more efficient since the symmetric key is faster to decrypt and encrypt messages.

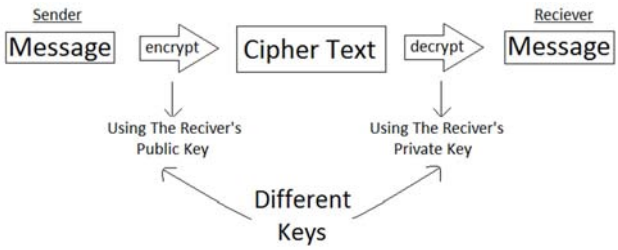


***Figure 2.*** *How Asymmetric Key works.*

Figure 2 shows how Asymmetric key algorithm work; unlike symmetric key algorithm, the sender and the receiver uses different keys to encrypt and decrypt the messages.

All parties in the Asymmetric key algorithm based have their own public key and private key, the sender uses the receiver's public key to encrypt the message, then the receiver uses his own private to decrypt the message.

Although this approach is more secure than symmetric key even in unsecured connections, it is costlier and slower.

The organization of this paper will be as follows; Asymmetric key problem identification will be covered in section 2, then the related work will be in section 3. After that the proposed solution in section 4. Finally, the conclusion and the final remarks will be covered in section 5.

Despite how secure public key algorithms are, it's not a perfect method of encryption and it has some drawbacks. Due to the length of the keys in this method, it is computationally costly compared to other encryption methods.

The public key due to its public nature is vulnerable to brute-force attacks. And since incoming data are encrypted with the same public key, a third party can impersonate the original sender and pass along false data to the key owner [6].

# 3. Related Works

In [7] P. Rasmi and V. Paul used a Hybrid cryptosystem that used asymmetric and symmetric key algorithm, the proposed solution was based on a Circle Symmetric key algorithm.

In [8], a patent by S. M. Matayas used a Hybrid cryptosystem that used both asymmetric and symmetric key algorithm based on control vectors. Which allows the cryptographic hardware to verify that the control vectors allow the use of the key.

A patent in [9] proposed a solution that uses symmetric key algorithms to encrypt and decrypt files however it derives the symmetric key from the private key of an asymmetric key algorithm.

In [10], the authors discuss a new asymmetric key algorithm that they claim is faster to process and harder against brute force attacks, A parallel-key cryptographic algorithm.

In [11], a hybrid asymmetric key algorithm based on two different implementations of RSA algorithm, RSA small-e, and efficient-RSA. Which according to the authors faster than the original RSA algorithm by 50% and more secure for cloud computing.

In [12], the author claimed a patent that uses a split private key algorithm, which creates keys that are made of portions of different keys.

In [13], the author used adaptive genetic algorithms to generate public and private key algorithm, which makes it hard to find the private key from the public key.

In [14], the author introduces an asymmetric key algorithm that generates private and public keys based on shared knowledge (domain parameters).

# 4. Proposed Solution

In this paper, we are proposing a solution to the previously mentioned problems that may arise during the communication process. The proposed solution will use a hybrid encryption method. In that, it'll use both the symmetric and asymmetric key encryption to communicate.

Since we have concluded that the asymmetric key encryption is safer, however, more computationally expensive and symmetric key encryption is more vulnerable but faster to encrypt and decrypt. We can use the best out of each of them in that we use asymmetric key encryption to exchange the symmetric key and then use the symmetric key encryption exchange encrypted messages.

However, using a symmetric key encryption is not completely safe. And the reason is because if hackers used a brute force attack to find the encryption key, they'll be able to decrypt all the remaining messages and/or even impersonate one of the two senders to pass messages to the other receiver.

In this paper, we will discuss a way to implement a dynamic shared key, the way to make the key dynamic and keep the encryption key identical in both parties, we need to use an equation to apply to the keys on both computers at the same time. An example scenario will be demonstrated below in Figure 3.

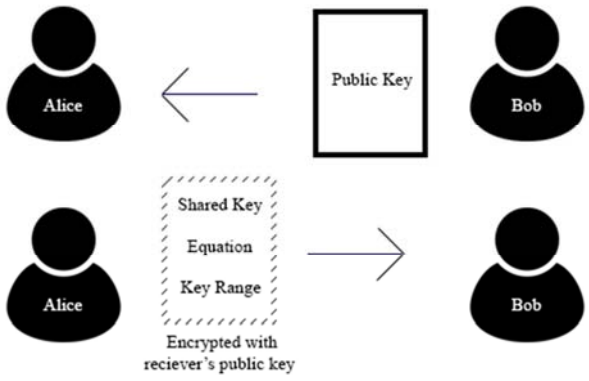


***Figure 3.*** *The Initiation of the Hybrid encryption.*

The Initiation of the connection is done through an asymmetric key encryption. In the scenario above Bob shares his public key with Alice to receive encrypted initiation information which includes the following:

(1) A symmetric key that will be used to encrypt messages for the rest of the session.
(2) An equation that will be used to alter the symmetric key.
(3) The range that symmetric key will be limited to.
(4) Optionally, the period of generating the symmetric key can be sent to be used to sync the changing the shared key.

Once the information has been exchanging the users can now communicate through an encrypted channel using the

symmetric key.

What this method differs from other encryption communication method is that the symmetric key will be changed every an agreed upon amount of time.

The example below will demonstrate an example of how the shared will changed with time using an arbitrary equation.

***Table 1.*** *A demonstration of how the shared key is changing.*

| Time of the session | Key Range | Equation | Shared Key |
|---|---|---|---|
| 00:00:00 | 51128 - 97311 | SK*S | 84297 |
| 00:07:00 | 51128 - 97311 | SK*S | 78799 |
| 00:14:00 | 51128 - 97311 | SK*S | 80626 |

In the table above the interval of changing the shared key is 7 minutes, the equation is SK*S, Where SK is the Shared Key and S is the elapsed seconds from the beginning of the session, and the range is from 51128 to 97311. All of these parameters of the communication were agreed upon during the initiation of the connection.

The way to calculate the new shared key within the range is by using the modulus equation to keep the result within the defined range as follows:

$$New\ SK = \big((SK \times S)\%(\Delta Range)\big) + Range\ Start \quad (1)$$

The advantage of this method is that it well work with a blind trust that both sides of the connection have the same shared key even though the shared key was exchanged only once. So even if a hacker succeeds in reverse engineering the shared key, the shared key will be changed by the end of the interval.

An outsider cannot predict the next shared key without having the rest of the parameters and knowing the exact duration of the session.

The implementation of this algorithm can differ based on what setting the programmer prefer. However, a general algorithm and flow chart is explained briefly in the following subsections.

### *4.1. Algorithm*

Algorithm 1: For generating a new shared key every I second

Dynamic Key Generation Algorithm
Input:
ST: Current Session Time in seconds,
SKc: Current Shared Key,
Rs: Range Start,
Re: Range End,
I: Interval
Output:
SKn: new Shared Key
if (ST % I) = 0 then //If the interval has elapsed
r = Re – Rs
SKn = ((SKc * ST) % r) + Rs
return SKn //Return the newly generated key
else
return SKc //Return the current key
end if

### *4.2. Flow Chart*

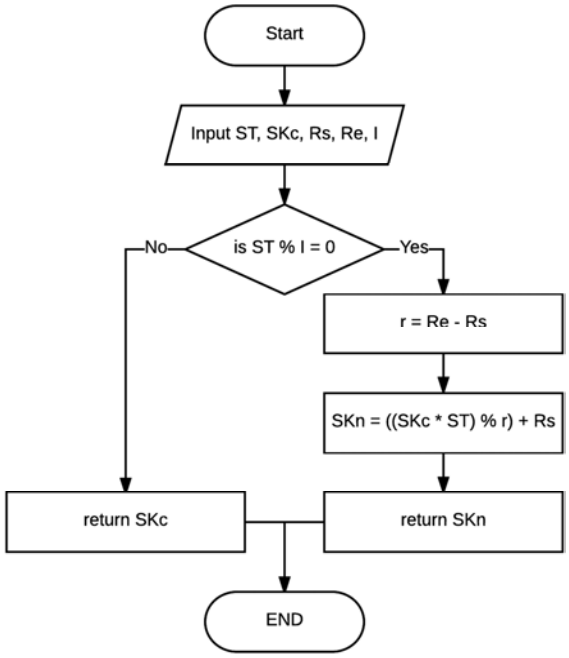


***Figure 4.*** *Dynamic key generation flow chart.*

## 5. Conclusion

The Proposed solution is demonstrating a concept of a blind trust between all sided of the connection whether they are two or more. The parameter of the connection initiation determines the level of security which can be easy or hard to predict based on the strictness of the parameters and the complication of the equation. The more complicated the equation is the more resource and processing expensive it will be.

The main parameter that will make it hard to predict the shared key is a dynamic agreed upon information that will be unknown and/or unavailable to an attacker. In this paper, the time of the session was chosen to be that parameter as it will be unknown to the attacker.

## References

[1] M. E. Whitman and H. J. Mattord, Principles of information security. Cengage Learning, 2011.

[2] S. Singh, "The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography."

[3] S. Garg and M. K. Rana, "A Review on RSA Encryption Algorithm."

[4] G. Oparnica, "Digital evidence and digital forensic education," Digital Evidence & Elec. Signature L. Rev., vol. 13, p. 143, 2016.

[5] H. Hou, "An authenticated certificateless key agreement protocol without bilinear pairing," in Computer Science and Service System (CSSS), 2011 International Conference on, 2011, pp. 2986-2989: IEEE.

[6] M. Blumenthal, "Encryption: Strengths and Weaknesses of Public-key Cryptography," CSRS 2007, p. 1, 2007.

[7] P. Rasmi and V. Paul, "A Hybrid Crypto System based on a new CircleSymmetric key Algorithm and RSA with CRT Asymmetric key Algorithm for E-commerce Applications," in International Conference on VLSI, Communication & Instrumentation. Kerala, 2011, vol. 9, pp. 14-18.

[8] S. M. Matyas et al., "Hybrid public key algorithm/data encryption algorithm key distribution method based on control vectors," ed: Google Patents, 1992.

[9] D. B. Cross, J. Gu, J. D. Benaloh, T. C. Jones, P. J. Leach, and G. D. Pittaway, "Deriving a symmetric key from an asymmetric key for file encryption or decryption," ed: Google Patents, 2007.

[10] T. Teerakanok and S. Kamolphiwong, "Accelerating asymmetric-key cryptography using Parallel-key Cryptographic Algorithm (PCA)," in Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on, 2009, vol. 2, pp. 812-815: IEEE.

[11] F. F. Moghaddam, M. T. Alrashdan, and O. Karimi, "A hybrid encryption algorithm based on RSA small-e and efficient-RSA for cloud computing environments," Journal of Advances in Computer Networks, vol. 1, no. 3, pp. 238-241, 2013.

[12] R. Ganesan, "Yaksha, an improved system and method for securing communications using split private key asymmetric cryptography," ed: Google Patents, 1996.

[13] P. G. Naik and G. R. Naik, "Asymmetric Key Encryption using Genetic Algorithm," International Journal of Latest Trends in Engineering and Technology (IJLTET), vol. 3, no. 3, pp. 118-128, 2014.

[14] C. Beeson, "Asymmetric key cryptosystem based on shared knowledge," ed: Google Patents, 2006.