Taiz University Research Journal, Volume 34 jan2023 ISSN: 2985-7848
Taiz University
2nd ICTSA -2022 Proceedings
17 -19 Dec, 2022, Taiz University, Taiz, Republic of Yemen.

Reducing The Risk Of Forgotten Long Secret Key

D. Saleh Noman Alassali[1] , D. Mohammed Mokred Nagi[2]

[1]Dept. Computer Science, Faculty of computer and Technology, Sheba Region University, Marib, Yemen
[2]Dept. Computer Information system, *Faculty of* AL-JAWF, Sheba Region University, Marib, Yemen

Email address:
mmnsyemen2@gmail.com (Saleh Noman Alassali)
mmnsyemen@gmail.com (Mohammed Mokred Nagi))
Corresponding author Mohammed Mokred

To cite this article:
Authors Name. Paper Title. *ICTSA – 2021 Proceedings*, 2021, pp. x-x.

Received: MM DD, 2020; **Accepted**: MM DD, 2020; **Published**: MM DD, 2021

Abstract:

Nowadays, most peoples use PCs, and save sensitive information in their PCs. Some of users use encryption/decryption techniques to hide his/her own sensitive information, and other not use cryptography at all. But a lot of the users use unsuitable secret keys. They use either short, or weak keys, because each of short or weak keys are easy to be remembered. Those users may don’t know the risk of using weak or short keys, and they may don’t know that the security produced from cryptography are directly proportional to the quality and the length of the secret key used along with the used algorithm. Using suitable secret key, may necessitate some users to save the secret key in a file or to write it in some place, which in turn weaken the associated security. Reducing the problem of keeping long key secretly without forgotten it and without saving it in a file is a valuable problem. This paper suggests a method to generate a suitable secret keys from ***passphrases*** *along with salt ’short secret key’ by using one way hash function. In the suggested method, the user can exploit some files saved in his/her PC as passphrases, and generates the actual secret key by selecting one or more of the passphrase files and hashing it/them along with the salt. The output of the hashing can be used as an actual user secret key, called message digest code. By this way, the user will remember only small secret key*.

Keywords: passphrases; encryption/decryption; hash function; Symmetric; Asymmetric; Message Digest Code; Confidentiality

1. INTRODUCTION

In computer environment, information can be easily saved and retrieved. Encryption/decryption techniques are effective tools to hide sensitive information. The security produced by any encryption/decryption technique, is directly proportional to the quality and the length of the used secret key. Nowadays, all good encryption/decryption algorithms required long secret keys. For example, if a user uses Advance Encryption Standard, AES algorithm of 256-bits, he has to use 256-bits key for encrypting/decrypting his/her sensitive data or information, that means the secret key consisted of 32 symbols. The user should remember 32 symbols, the user vulnerable to forget this secret key. [1],[2]

1.1. Problem statement

Indeed, one problem may arise of using long secret key. How does a user keeps suitable secret key for long time? The user may necessitate to write the secret key. Short or weak keys ease to be remembered, but long or strong keys dose not. Writing a secret key in any place, or saving it in PC, may be weaken the security produced by that key. [2]

1.2. Main objective

Generating a suitable key from passphrases along with small secret key called salt, using suitable one way hash function. This paper suggests a method to generate a suitable key from passphrases along with secret key called salt, using suitable one way hash function. Actually any user has several files saved in his/her PC. Tow suitable files can be used as passphrases inputs to one way hash function algorithm along with salt ’short secret key’. The output of the one way hash function algorithm called Message Digest Code, can be used as the actual user secret key. This paper is organized into four sections, Introduction is in Section I, related works is in Section II. The suggested method is in Section III, Implementation is in Section III.2, and Conclusion is in Section IV.

2. RELATED WORK

This section gives a brief description to the information confidentiality using cryptography and gives the main concept of one way hash functions.

2.1. Logical security

All information and systems inside computer are secured by logical security. Logical security consisted of several aspects, like encryption/decryption, privileges,. , and access control. Any computer system has access control system ACS. ACS operates through accounts. So the ACS requires opening accounts to every user, that is to make the user able to login the computer. Any account needs requirements, like user-name, password, suitable privileges, and so on. All account's requirements should be saved in the database manager system during the opening of the account. Every account should have suitable privileges. These privileges are granted by the ACS manger, according to the associated tasks/operations related to that account. Indeed, ACS is important part of the logical security. The ACS uses user-name and password to prevent unauthorized user from login either to the computer or to the unauthorized user account. [2],[3]

2.1.1. Source of the logical security

In general, most computer systems and information are secured by the password or secret key, and any type of identity authentications are depended on the passwords or secret keys used as user's identities. The security of the password or secret key is depended on symbols randomization and the symbols quantity. The security produced is directly proportional to the quantity of the symbols, and the accuracy of the randomization of that symbols' used as a password or a secret key' along with the associated algorithm. But these type of securities may are not sufficient specially for sensitive data/information. May users left their screens opened, for a moments, beside his/her partners, in their organization during he/she is doing other works, vulnerably sensitive data/information to risk. Also a lot of attacks and risks may penetrate sensitive data/information. So sensitive data/information should be encrypted before saving it in computer or in any one of computer devices.

2.1.2. Encryption/ decryption

Cryptography is the science of converting readable data/information to unreadable data/information and vice versa. Encryption is the process of transforming plaintext into ciphertext. Decryption is the process of transforming ciphertext into plaintext. There are two types of cryptosystem: a symmetric and an asymmetric cryptosystem. A symmetric cryptosystem is one that uses only one key for encryption and decryption. An asymmetric cryptosystem is one that uses one key called public key for encryption and other key called private key for decryption. Figure 1 shows the diagram of encryption/decryption.[2],[3]

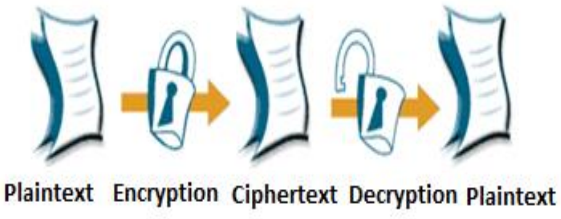

Figure 1: *the diagram of encryption/decryption using secret key.*

2.1.3. Data/information confidentiality

Data/information confidentiality refers to protection against unauthorized data/information access. Confidentiality is the protection of transmitted or stored data/information from passive attack. Confidentiality ensures that the programs, data/information saved in a computer system are reading only by authorized parties. So the cryptography can be used efficiently to fulfill data/information confidentiality.

The security or confidentiality produced by cryptography depend on the algorithm used and the secret key. In general the cryptography's algorithms are studied and analysed by several specialties of scientists and then approved. In the cryptography field, the algorithms are usually known, but the keys should be kept secretly. So the security gained from cryptography based on the randomization of the content of the secret key's construction, and the length of the secret keys. Figure 2 shows the diagram of encryption/decryption.

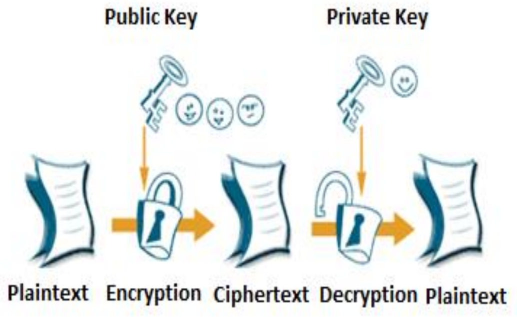

Figure 2: *the diagram of encryption/decryption using public key.*

2.2. One way hash function

A one way hash function algorithm, operates on an arbitrary length pre-image message. It returns a fixed-length hash value. The output of one way hash function algorithm is called Fingerprint or Message Digest code.

In computer field the Fingerprint and Message Digest can be used interchange. This paper will use Message Digest terminology.

2.2.1. Message Digest Code

Message Digest Code can be defined as a unique code corresponding to unique data. The Message Digest Code in its simple frame is a unique code generated by a person/machine. Message Digest Code is usually generated by any one way hash function algorithm.

The security produced by one way hash function algorithm depend on the nonlinear relationship between the inputs and the outputs of the hashing algorithm used. [3],[4]

2.2.1.1. Essential characteristics of Message Digest

A one way hash function, denoted by H(M), operates on an arbitrary length pre-image message, M. It returns a fixed-length hash value, h.

h= H(M), where h is of length m.

Many hash functions can take an arbitrary-length input and return an output of fixed length, but one-way hash functions have additional characteristics that make the one-way cod generation.

- Given message M, it is easy to compute h.
- Given h, it is hard to compute M such that H(M)=h.
- Given M, it is hard to find another, M^1 such that $H(M)=H(M^1)$. [3],[4]

So the Message Digest code should satisfy the following conditions:

$y = f(x)$ easy to compute. (1)

$x = f^{-1}(y)$ hard to compute. (2)

$y^{-} \neq f(x)$ and $y \neq f(x^{-})$ [3],[4] (3)

2.3. Passphrase

Passphrase is a collection of data with arbitrary-length. Passphrase is not necessary to be kept secretly, it can be saved in the user's PC. From the security view point, the passphrase's contents should have reliable meaning like computer programs, verses of poetries or wisdom text. Passphrases are used as the row material inputs to the one way hash function algorithm along with the salt. [5]

2.4. Weakness may be associated with secret keys

Cryptography always needs suitable algorithm and suitable secret key. Most algorithms used in cryptography were studied and analysed by may specialist, and the probability of the defect in the algorithm used is may be very small. But using short or weak secret key, vulnerable to brute force or guessing attacks. Suitable secret key means difficult to be guessed, and long enough, i.e. has large space-key which has immunity against brute force attack. [1],[2],[3]

2.4.1. Brute force attack

Brute force in its simple definition is a way in which the cryptanalyst tries to cover all the key-space to extract the plaintext from the cipher-text. This way suitable with short keys.

2.4.2. The security gained by the secret key length

The security gained by the length of the keys based on the key-spaces. The produced security is directly proportional to the key-spaces, if the key-spaces not sufficient, there is a weakness in security gained. But there is a limitation to the length of the secret key that the user can be able to memorize the selected key's symbols.

In many situations, someone can impersonate an account of another user, in the same organization and try to gain access right of the that user, by exploiting inadvertency of that user and stoles some files which contain sensitive information and exploiting available algorithms and the high computations powers to break the secret key. That is maybe done by covering all the probabilities of existing secret key symbols. Indeed, any secret key is consisted of set of symbols which exist in the ordinary keyboard, that around 100 symbols. [4],[5],[6]

Actually, most users use secret key around 15 symbols, because of long secret key cannot be remembered easily. Then the corresponding secret key space is around 100^{15}. Nowadays, the high computations powers can exhaust 100^{15} and cover all the probabilities of existing secret key symbols, especially in decryption. In decryption process, the row material of encrypted information will be available to the opponents, and the opponents may use the brought force methods to retrieve the encrypted information. Also the opponents can exploit the available algorithms found in the net and the high computations powers found in the net also, to cover all the probabilities of existing secret keys within short period. So short secret key is not sufficient, and not applicable to the recent algorithms like AES, or SHA-256 bits, that require 32 symbols= 256-bits.

2.4.3. Guessing attack

Weak-key vulnerable to guessing attack, weak-key, like 101010…… sequence of ones and zeros , or ababab.. sequence of a's and b's, or something like that, or any strings have meaning vulnerable to guessing attacks. For example this key "qwertyuiop[asdfghjkl" is very weak, it consisted of 20 characters, but this symbols corresponding to the 2nd and 3th rows of the keyboard, or telephone numbers. But in many cases, the cryptanalyst tries to guess the secret key using dictionary words or some other technics.

Indeed, cryptanalyst uses several types of attacks through which reduce the total cost of breaking the cipher-text. Cryptanalyst will try to use the behaviour conduct and the curriculum vitae of the user to guess the secret key and retrieve secret information. [2],[3],[4]

3. Description of the suggested method

This section gives a description to the suggested method which reduces the number of symbols used to construct suitable keys at the user. One way hash function is used to generate the required symmetric keys. The following subsections describe generating suitable key in the suggested method. One way hash function accepts tow inputs 1: an arbitrary length file text, and 2: a secret key. The output from the one way hash function is a fixed-length hash value, called Message Digest Code MDC, correspond to the inputs. If one character, or one bit changed in the inputs, the output MDC will be changed correspond to that inputs changes.[4]

3.1. Generating suitable secret key

To overcome the problem associated with using short or weak secret key, the suggested method is generating actual secret keys, based on suitable one way hash function algorithm, using suitable passphrases along with secret key called salt. In the suggested method, the user should select suitable passphrases files from his/her directory in his/her PC. The selected passphrases are used as a row material inputs to the one way hash function algorithm along with the salt. The output of the hashing 'MDC' used with other passphrase as input to the hash function algorithm again. The output MDC will be used as a user secret key. This way reduces the problem of keeping long key secretly without writing it. [6],[7],[8],[9]

The following steps describe the processes of generating suitable user secret key:

Step1. The user saves suitable passphrases files in his/her PC. Only two of passphrases files will be used as a row material for generating a suitable user secret key. The other passphrases files will be used only for camouflage.

Step2. The user select the first passphrase file as an input to the one way hash function algorithm, and enters his/her secret key. The hash function algorithm SHA-256 bits is suitable to be used.

Step3. The user select the second passphrase file along with the output of step2, and the result used as an input to the one way hash function algorithm again. The output of hashing used as a user secret key. [7],[8]

The block diagram of Figure 3. summarizes the suggested method to generate the user secret key.

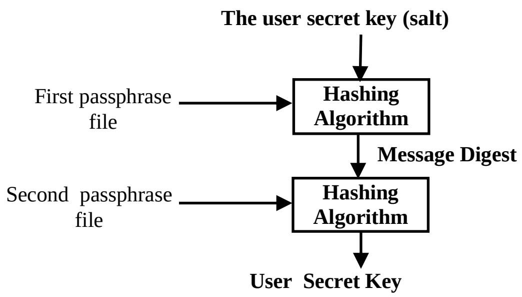

Figure 3: *Block Diagram of Secret Key Generation.*

3.2 Implementation

The suggested key generation model is implemented using C++ compiler under Windows environment. In this implementation, AES of 256-bit is used as a symmetric encryption algorithm, and SHA-256-bit is used as a hashing algorithm for generating the required keys.

The following are simples of the generated secret key with 256-bit.

```
1101011010110010101001101110111010010110
0110110010101010101101111011011001101010
0111001011101101001111011100110011100010
1000110111001100001010011011011010010101
1011110001100100101001001100111000101100
1111011011011101110101001100011110001001
1010111101011101
```

```
1010011011101001000111011001010010111011
1011110100101110010011011101100111101010
0101111010011100110111101110011100111010
1111010101010010100101011100101010001101
0110111101010100010011110111011011100111
1101110111011010100110110111011101110001
1000110100110001010010010101010001011010
0100100110101100
```

```
1011010001000111010111101011101101100111
1000100100110110001001010101001100111001
0100101010100100011001011010010010010110
0001011110010101001110111011111000010100
1110010001001001111011101011101110001001
1110111010010011010111101110011011110011
1101001011011001
```

Figure 4.: *user code*

3.3. Security gained and Verification

The generated keys have both required specifications, good qualities to some extent, and length, because it is too difficult to guess any key of them. The long secret key is verified using simulation program. Figure shows three different keys generated using same passphrase files with slight modification with every key's generation. Each generated key with length 256 bits. The first passphrase was the following:

" The user select the second passphrase file along with the output of step1, and the result used as an input to the one way hash function algorithm again. The output of hashing used as a user secret key. The block diagram of Figure.3 summarizes the suggested method to generate the user secret key. The garbage data used are " may be tow rew stop takensequencialy in the data of the initial to day i want many student all of you IN THE NAME of alla alrrahman arrheem I want you to write some messages. come to me please 10 becas of the. random numbers must be non repeated sequencialy in the data of the initial value !"

The second passphrase was the same first, only with slight modification to the word step1 in the second line to the word step2 ".

The third passphrase was the same first, only with slight modification to the word step2 in the second line to the word step3 ".

As mention above, only the modifications were on the word step1, step2 and step3 in the inputs. The outputs show big deference's associated with slight modifications in the used passphrases.

The suggested method has security gained as follow:

1. Security gained corresponding to the content of the first passphrase, because the first passphrase may contain any data, related to any directory.
2. Security gained corresponding to the content of second passphrase, because the second passphrase should contain different data, and it may be related to any directory.
3. Security gained corresponding to the salt, because the salt should be kept secretly.
4. Security gained corresponding to the length of the generated key, because generated key with length 256 bits, this length is sufficient.
5. Security gained corresponding to the goodness of the generated key, because generated key has immunity against guessing attack.

Disadvantages of the suggested method

The main disadvantages of the suggested method as follow:

1. Time required for searching about the used passphrases. Because of, from security view point, it is better to keep the used passphrases in different directory, and not be related to the same directory.
2. Extra execution time required to convert to the content of the passphrases, from text to binary system. Because of, from security view point, it is better to keep the used passphrases as a text.
3. Extra execution time required to the used one way hash function algorithm.

4. Conclusion

This paper focus on the problem of saving long secret key, and suggests reducing the number of symbols of secret key by using passphrase files stored at the user account along with secret key. The main advantage of the suggested method is that the user not necessary to save long secret key, but he has to save passphrase files at his/her account. Instead of that, it store only short key securely, and then generate long key on demand.

REFERENCES

[1] W. Stallings, "Cryptography and Network Security", principles and practices, 7th Edition, Pearson Prentice Hall, 2017.

[2] A.J. Menezes, P.C. Van Oorschot and S.A. Vanstone, "Handbook of Applied Cryptography", 5th Edition, CRC Press, Inc, United States, 20010.

[3] Douglasr. Stinson (2002) "Cryptography: Theory and Practice" Department of Combinatory and Optimization University of Waterloo, Waterloo, Ontario Canada. 2nd Edition, Chapman & Hall/CRC.

[4] A. Aasarmya and S. Agarwal, "Improving Security for Data Migration in Cloud Computing using Randomized Encryption Technique", International Journal of Computer Sciences and Engineering,Vol.7, Issue.8, pp.39-43, 2019.

[5] Y.A. and S.N. Alassali," An Improved Key Distribution Protocol Using Symmetric Key Cryptography", International Journal of Computer Sciences and Engineering (IJCSE),Vol.8, Issue.11, pp.21-26, 2020.

[6] Guang-Gong and Golomb-SW Commun. Sci. Inst., University of Southern California, Los Angeles, CA, USA IEEE-Transactions-on-Information-Theory. Vol.45, No..6; Sept.1999; p.2065-73 1999 Journal-article.

[7] Shalini and M. Kushwaha, "Mutual Authentication and Secure Key Distribution in Distributed Computing Environment", International Journal of Advanced Research in Engineering and Technology (IJARET), Vol.11, Issue.5, pp.378-390, 2020.

[8] K. Liu, J. Ye and Y. Wang, "The Security Analysis on Otway-Rees Protocol Based on BAN Logic", IEEE 4th International Conference on Computational and Information Sciences (ICCIS), Chongqing, China, pp.341-344, 2012.

[9] S. Verma, R. Choubey and R. Soni, "An Efficient Developed New Symmetric Key Cryptography Algorithm for Information Security", International Journal of Emerging Technology and Advanced Engineering, Vol.2, No.7, pp.18-21, 2012.