Enhanced Honey Encryption Algorithm for Increasing Message Space against Brute Force Attack

Khin Su Myat Moe
Department of Computer Engineering and Information Technology
Yangon Technological University
Yangon, Myanmar
sumyat.3040@gmail.com

Thanda Win
Department of Computer Engineering and Information Technology
Yangon Technological University
Yangon, Myanmar
thanda80@gmail.com

Abstract –**In the era of digitization, data security is a vital role in message transmission and all systems that deal with users require stronger encryption techniques that against brute force attack. Honey encryption (HE) algorithm is a user data protection algorithm that can deceive the attackers from unauthorized access to user, database and websites. The main part of conventional HE is distribution transforming encoder (DTE). However, the current DTE process using cumulative distribution function (CDF) has the weakness in message space limitation because CDF cannot solve the probability theory in more than four messages. So, we propose a new method in DTE process using discrete distribution function in order to solve message space limitation problem. In our proposed honeywords generation method, the current weakness of existing honeywords generation method such as storage overhead problem can be solved. In this paper, we also describe the case studies calculation of DTE in order to prove that new DTE process has no message space limitation and mathematical model using discrete distribution function for DTE process facilitates the distribution probability theory.**

Keywords – data security, honey encryption, distribution transforming encoder, message space limitation

I. Introduction

Nowadays, many application systems widely uses password based encryption (PBE) for securing data communication because user's passwords are easily remember for performing encryption process. But, many users who apply PBE techniques usually use the weak or repeatedly passwords. Therefore PBE has weakness in brute force attacks and password guessing attacks [1].

A user data protection algorithm called honey encryption (HE) can generate valid looking plaintext if the attacker tries to decrypt the plaintext with wrong key or honeywords and it can strongly deceive the unauthorized users. HE process generates honey message if the attackers try to decrypt with any of a number of number of incorrect passwords. Otherwise, HE process produces correct ciphertext. HE turns every wrong password guess made by a hacker into a confusing dead-end [2].

The HE scheme is designed with a cryptographic primitive called the Distribution Transforming Encoder (DTE). The DTE is a set of encoding and decoding process, where encode takes a space of plaintext messages M as an input and returns a value in the seed space S of n-bit strings as output. Decoding process converts value in the seed space S of n-bit strings into plaintext. The DTE process takes the probability distribution theory for assigning the corresponding ratio of messages [3]. The example of existing honey encryption process is shown in Fig. 1.

In the honey encryption process, the message space M consists of the messages such as "chocolate, mint, strawberry and vanilla". This message space M is mapped into the seed space S_m using DTE process. The DTE process uses cumulative distribution function for mapping the message space M into seed space S_m. The resulting S_m are XOR with the secrete key to produce ciphertext. The DTE process using cumulative distribution function cannot send more than four messages because the cumulative distribution function cannot satisfy the probability theory in more than four messages [1]. To overcome this message space limitation weakness, we use discrete distribution function in DTE process.

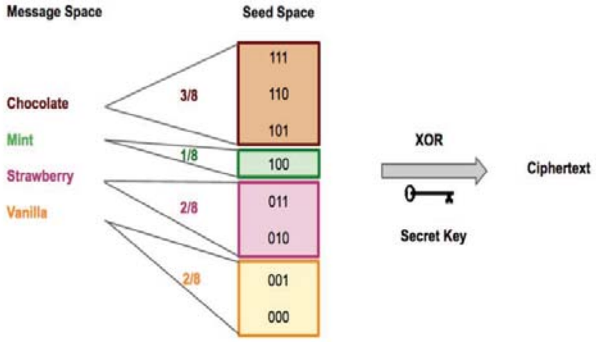

Fig. 1 Honey Encryption Process [1]

This paper is organized with the five sections. We discuss literature review about the honeywords generation algorithm and honey encryption process in section II and section III presents background theory of honeywords generation method and honey encryption algorithm using discrete distribution

978-1-5386-3555-1/18/$31.00 ©2018 IEEE

function. In section IV, we focus on the overview system flow of the proposed method and then finally we conclude the paper.

II. Literature Review

A. Jules and R. L. Rivest discussed the honeywords generation method that can produce at least 20 honeywords. The producing honeywords and user's real password are stored in the password file. If the adversary gets the password file in the database, he can't classify which password is real password [4]. However, this method can cause the storage overhead problem and typo safety problem because it produces many honeywords to store with the real password.

The new encryption method called honey encryption (HE) algorithm was developed by A. Jules and T. Ristenpart for preventing from brute force attack. HE is similar to the password based encryption algorithm (PBE) because it uses the user's passwords for encryption the private messages. So that HE can protect strongly the brute force attack. The main part of HE is distribution transforming encoder (DTE) and HE can apply in various different application such as credit card number, genomic data and etc [5][6]. Although HE protects the brute force attack, it has limitation in DTE process for placing the messages into the seed space.

A new encoding and decoding method called distribution transforming encoder (DTE) in honey encryption (HE) algorithm was proposed by Vinayak P.P, Nahala M.A described. The purpose of HE is to deceive attackers which use brute force attack. The attacker tries to get decrypted data by using brute force attack. That technique can give bogus messages to attackers if the attacker get user's password. The main innovation of HE is DTE [7]. The DTE in that system has limitation for distributing plaintext message into the seed space.

Therefore, we propose a system to overcome message limitation. This honeywords generation system can reduce the drawbacks of the existing honeywords generation algorithms such as reducing storage cost. Moreover, it can easily overcome the typo safety problem. For securing password files, we use new hashing and salting algorithm. The hashing and salting time is faster than the existing hashing and salting algorithm. Finally our system can overcome message space limitation using DTE compared with the existing honey encryption algorithm.

III. Background Theory

The honey encryption process includes honeywords generation process for key or passwords distribution and distribution transforming encoding (DTE) process for message distribution. In this section, we briefly introduce honeywords, honeywords generation algorithm and DTE process using discrete distribution function.

A. *Honeywords Generation Algorithm*

In order to prevent from unauthorized access, honeywords also called decoy password are stored with the user's real password in the database. Honeyword is a bogus that is created for deceiving the attackers if they get the password file in the database. The attackers can't know which password is real password by adding the honeywords to the password file. If an adversary attempts to login with a bogus password, the honey checker sends an alarm message to the system administrator for entering of honeywords. In this process, the combination storage of bogus passwords and real passwords are called sweetwords and the real user's password is called sugarword. Although the honeywords generation method can make the confusion to attackers, the database stores many password and it causes the storage overhead problem [8][9].

Therefore, in order to reduce the drawback of the existing algorithm, we propose a new algorithm called improved honeywords generation algorithm that can be solved storage overhead problem. In our improved honeywords generation algorithm, the system stores the sweetwords with two tables and it assigns the indexes of this sweetwords. For reducing the storage cost, we only save the index of honeywords that the other user's real password instead of generating honeywords and stored them in the password file. In this case, the indexes are randomly stored into the database. The user's real passwords are converted into the hash codes and store with the index of real password in one table. The second table are stored the username and indexes of honeywords.

B. *Distribution Transforming Encoder (DTE)*

The core creation of HE is DTE and it can perform the encoding and decoding process. The DTE encode takes as input message space that contain all plaintext message m€ M to the seed space S_m using cumulative distribution function in existing honey encryption algorithm [2]. Our proposed system uses discrete distribution function in mapping the seed space S_m for improving HE algorithm. Decoding process is slightly harder than encoding process. In the decoding process, DTE decodes the value of seed space s€ S and outputs a message m€ M. The encoding process by using discrete distribution function is shown in the following steps.

DTE Calculation Using Discrete Distribution function

Number of message= 5
For Distribution transforming encoder,
Seed space $S_m = 2^{n-1} = 2^{5-1} = 2^4 = 16$
{0000,0001,0010,0011,0100,0101,0110,0111,
1000,1001,1010,1011,1100,1101,1110,1111}

By using Discrete Distribution Function,
$m_1 = 1/16$
$m_2 = 4/16$
$m_3 = 6/16$
$m_4 = 4/16$
$m_5 = 1/16$

From the calculation, we can achieve the better result in HE algorithm if we use the proposed DTE process using discrete distribution function.

IV. System Flow of Proposed Method

Our proposed system mainly discusses two processes such as message distribution using DTE and key or passwords distribution. Fig.2 shows the example design of our proposed system.

In the example design process, the message space M consists of five messages (m) such as "Hi, Good evening, Morning, Good night, Hello". This message space is mapped to the seed space S_m using DTE process. The DTE process uses discrete distribution function for mapping the message space to the seed space S_m overcoming the message space limitation. In the key or password distribution module, key or passwords are mapped randomly into seed space S_k. The resulting seed space S_k are made XOR operation with seed space of message S_m to produce the ciphertext.

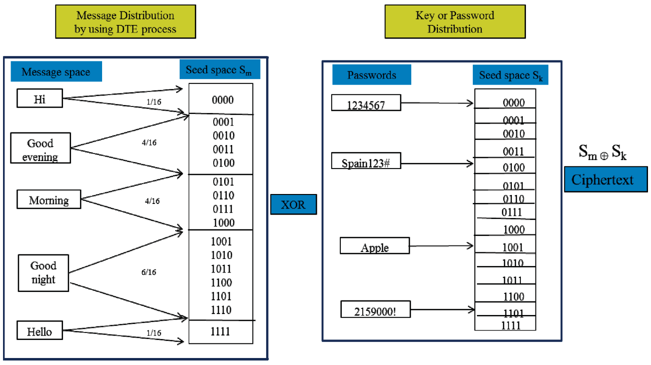

Fig. 2 Proposed System Design

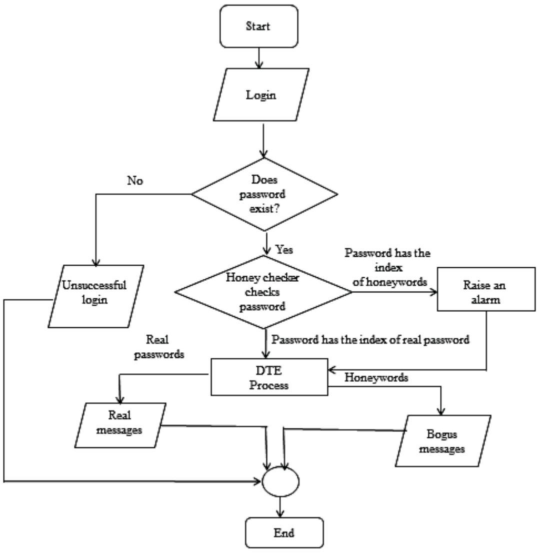

Fig. 3 Flowchart of the proposed system

Fig.3 describes the flowchart of our proposed system. This flowchart shows step by step procedures of our proposed system. In this process, the server checks whether the username or password exits in the database if the one of the users perform login process. If the user tries to enter the system, he can get three conditions. Firstly, he can get "unsuccessful login message" from the system if he isn't the member of this system. Secondly, if his password exists in the database, he performs the DTE process. In this condition, the server sends his password to honeychecker to know his password is honeyword or real password. He can get correct plaintext if his password is real password. Finally, if his password is real password, the honeychecker allows him to perform DTE process and he can get real messages.

A. Case Study for Proposed System

Consider the simple example of encoding the messages in Fig. 2. In this example, the message space M includes five messages such as "Hi, Good evening, Morning, Good night and Hello". According to the number of messages, probabilities are assigned to each message. By calculating, the seed space range of message M is 2^{5-1} $=2^4$ =16 and seed space is 4-bit strings. Depending on the results of these probabilities, we map each message to a seed space. In this process, we order randomly the messages. So, the messages would get the different range of seed space. The example of secrete keys or passwords are K = {1234567, Spain123#, Apple, 2159000!}.

Using discrete distribution function, probability of the messages is {1/16, 4/16, 6/16, 4/16, 1/16}. So, the seed space S_m of the message "Hi" is {0000} and "Good evening is {0001, 0010, 0011, 0100}. And then, seed space S_m of "Morning" is {0101, 0110, 0111, 1000} and "Good night" is {1001, 1010, 1011, 1100, 1101, 1110}. Finally, the seed space S_m of "Hello" is {1111}. According key distribution process, the seed space S_k of "1234567" is {0000} and "Spain123#" is {0100}. The next password "Apple" is {1001} and finally password "2159000!" is {1101}.

In this section, we describe three modules: encryption module, decryption module and attacker module. In this section, one people send the message to another people using HE algorithm. At this time, the third person or attacker tries to get messages between two people.

In the encrypting process, Suzy wants to send encrypting message "Hi" to Daniel under her secrete key as "2159000!". So, we compute seed space S_m by using DTE process and then the result S_m of "Hi" is (0000).Similarly, according to the key or password distribution, seed space S_k of password "2159000!" is (1101). And then, Suzy sends the resulting ciphertext (1101) by XORing the seed space S_m and S_k to Daniel. Daniel receives the ciphertext (1101) and he tries to get plaintext message by using decryption process.

The decryption process is harder than encryption process. In this process, Daniel tries to decrypt ciphertext C by using password "2159000!"of Suzy. Firstly, Daniel computes seed space of key distribution S_k and this S_k (1101) is XORed with the ciphertext (1101) to get seed space of message distribution S_m (0000). And then, he decodes S_m by using DTE process to get original plaintext message "Hi".

In the attacking process, attacker Suli wants to get Suzy message. So she attacks password file by using brute force attack. Suli gets password file but she doesn't classify which is Suzy password. So, she randomly chooses password "1234567" and tries to decrypt Suzy message. Suli computes seed space of "1234567" to get key distribution S_k (0000). And then she makes XOR process of ciphertext C and seed space S_k. The resulting seed space of message distribution S_m is (1101). And then, she decodes S_m by using DTE process to gets Suzy message. Finally, Suli gets "Good night" after decoding process of seed space of message distribution. In this process, she doesn't know which message is true message or decoy message. This encryption algorithm can make confusion for attackers and it can strongly protect brute force attack.

B. *Mathematical Model*

The core process of HE is DTE that assigns the plaintext message space M to the seed space S_m using discrete distribution function in our proposed system. The discrete distribution function is a branch of probability function and it needs to satisfy the properties of probability function. Firstly, the values y of probability function in the sample space Ω must lie between 0 and 1. The next theory is that the total values of f(y) by adding all values y must have 1. The following two properties are described as follows [10].

i. $f(y) \varepsilon [0,1]$ for all $y \varepsilon \Omega$

ii. $\sum f(y) = 1$

- *DTE using Discrete Distribution Function*

We denote S be seed space or sample space and the value of seed space is [0,1]. According to the discrete distribution function, we calculate to get one for the total value of probability P(y) over all values of message y in seed space S. Let n be total number of messages, y be number of '0' observed, M be the message space that contains number of messages and m be message. The following examples prove that DTE using discrete distribution function satisfies the properties of discrete probability theory. In this example we use number of five messages M=5. Firstly, we calculate seed space S.

S={0000,0001,0010,0011, 0100, 0101,0110, 0111,1000,1001, 1010,1011,1100,1101,1110,1111}

M	m1	m2	m3	m4	m5	Total
P(y)	1/16	4/16	6/16	4/16	1/16	1

- *DTE using Cumultative Distribution Fuction*

According to the cumultative distribution function, we should get one for the total result of probability P(y) over all values message y in seed space S to satisfy the properties of probability theory [2]. The following example describes that DTE using cumultative distribution function cannot satisfy the probability theory because the sum of probability P(y) on all the values of message in seed space S doesn't get one. In this example we use five messages M=5 and calculate the probability P(y).

M	m1	m2	m3	m4	m5	Total
P(y)	1/16	2/16	2/16	3/16	2/16	10/16

V. Conclusion

Honey encryption process and honeywords generation algorithm are strong encryption techniques for attackers to prevent the various attacks especially brute force attack. Our proposed honeywords generation algorithm can mostly reduce storage cost and typing mistake of users during entering of password. Besides, we propose a new way of DTE process by using discrete distribution function to overcome the vulnerability of existing DTE process for increasing message space in honey encryption algorithm. And then, we describe the detail case study analysis for new DTE process and mathematical model in order to prove that our DTE process can solve the message space limitation problem. Finally, our new DTE process and honeywords generation algorithm can strongly against the brute force attack by producing the meaningful fake messages.

References

[1] H. Choi, H. Nam and J.Hur. "Password typos resilience in honey encryption," *In Proc. of IEEE 2017 ICOIN*, 2017, pp-594-598.

[2] N. Tyagi, J. Wang, K. Wen and D.Zuo. "Honey encryption application," *In Proc. Computer and network Security*, Springer, 2015.

[3] A.E. Omolara, A. Jantan, O.I.Abiodun and H.E.Poston. "A novel approach for the adaptation of honey encryption to support natural language message," *In Proc. of the International MultiConference of Engineers and Computer Scientists IMECS 2018*, 2018.

[4] A. Juels, R.L. Rivest. "Honeywords: making password cracking detecatable," *in Proc.of the 2013 ACM SIGSAC Conference on Computer and Communciations Security*, 2013,pp. 145-160.

[5] A. Juels and T. Ristenpart. "Honey encryption: security beyond the bruteforce bound," *In Proc. Advances in Cryptology–EUROCRYPT 2014*, Springer, 2014, pp 293–310.

[6] Z. Huang, E. Ayday, J. Fellay, J.-P. Hubaux, and A. Juels. "Genoguard: protecting genomic data against brute-force attacks," *In 2015 IEEE Symposium on Security and Privacy (SP)* , 2015, pp 447–462.

[7] Vinayak P.P and Nahala M.A. "Avoiding brute force attack in MANET using honey encryption," *International Journal of Science and Research (IJSR),* vol-4, pp 83 85, 2013.

[8] N. Chakraborty and S. Mondal, "A new optimized honeyword generation approach for enhancing security and usability", arXiv preprint arXiv: 1509.06094v1, 2015.

[9] R.S. More and S.S. Konda. "Resilient security against hackers using enchanced encryption techniques: blowfish and honey encryption" *International Journal on Recent and Innovation Trends in Computing and Communication*, vol- 4 Issue: 6, June 2016.

[10] B.Addanki. "What is the difference between a probability density function and cumulative distribution function, 2015. [Online] Available: http:// www.quora.com