HONEY CHATTING: A NOVEL INSTANT MESSAGING SYSTEM ROBUST TO EAVESDROPPING OVER COMMUNICATION

Joo-Im Kim and Ji Won Yoon

Center for Information Security Technologies (CIST)
Korea University, Seoul, Republic of Korea
{jooimkim, jiwon_yoon}@korea.ac.kr

ABSTRACT

There have been many efforts to strengthen security of *Instant Messaging* (IM) system. One of the typical technologies is the conventional message encryption using a secret or private key. However, the key is fundamentally vulnerable to a *brute-force attack*, causing to acquire the original message. In this respect, a countermeasure was suggested as the way to generating plausible-looking but fake plaintexts, which is called *Honey Encryption* (HE). In this paper, we present a HE-based statistical scheme and design a *Honey Chatting* application, which is robust to *eavesdropping*. Besides, we verify the effectiveness of the *Honey Chatting* by comparing the entropy of decrypted messages through experiments.

Index Terms— Instant Messaging, Honey Encryption, Brute-force attack, Eavesdropping

1. INTRODUCTION

Nowadays we frequently use *Instant Messaging* (IM) system for communication. While it provides us convenience to interact with others, there exist some side effects like an invasion of privacy through eavesdropping. Thus, popular IM systems such as Telegram and Threema have strengthened their security by message encryption. It makes the message unreadable to anybody except the sender and the receiver. However, it is still weak in a *brute-force attack* because the security of cryptosystem depends on the key size which has potential vulnerability to be cracked. There are 2^n possible keys with a n-bit key, so the attacker needs to try 2^{128} operations for cracking 128-bit key. In the case of *Password based encryption* (PBE), the necessary number of operations for cracking the key are much less by the fact that user-chosen passwords are tend to be weak, which results in the small key space.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2013R1A1A1012797).

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the "The Types of employment contract to support master's degree in Information Security" supervised by the KISA (Korea Internet Security Agency).

From this point of view, one of the countermeasures to the *brute-force attack* is *Honey Encryption* (HE). It is encoding and decoding scheme which can be used together with encryption/decryption scheme. Therefore, it seems as if HE is much closer to coding algorithms of signal processing rather than encryption algorithm of cryptography since HE focuses on encoding and decoding scheme although it is initially introduced in the cryptography conference. HE is used with the conventional encryption technology, and the main purpose is to make it difficult to distinguish a true output message from other fake output messages [1]. The contrast between the output messages provides a clue as the validity of the key to the attacker, leading to success of the *brute-force attack*. For instance, through decryption with key, the attacker would obtain a desired plaintext if the key is correct, while the attacker gains the other false results if the key is wrong. Threrfore, HE plays important roles in confusing the attacker by generating plausible-looking results.

There are various concepts relevant to *Honey Encryption*. A deniable encoding based on the stochastic language model was in [2]. In addition, an article introduced a structural coding scheme [3]. For a practical goal, Juels and Ristenpart presented a method for generating bit strings such as credit card number and RSA secret key [1]. Also, there was Visual Honey Encryption scheme for multidimensional data [4] and cracking-resistant password vaults which store user's password [5]. However, those methods are hard to be applied to the text message of *Instant Messaging* system due to the different output data form. We accordingly use processing technology, *statistical coding scheme*. In this paper, we propose a type of chatting system applying the HE scheme focusing on the message exchange. Our key contributions are as follows:

- We introduce a new secure chatting application robust to *eavesdropping* by applying the HE scheme, and we call it *Honey Chatting*.
- We compare the *entropy* between decrypted messages generated by the HE and conventional ASCII scheme.
- We explain the valid range of *message length* when applying the HE scheme.

The rest of this paper is organized as follows: background knowledge about instant message, honey encryption, and entropy are in section 2. We propose a threat model and the concept of our approach in section 3 and 4, and we introduce the honey chatting application in section 5. In section 6, we show results of experiment related to entropy. Finally, in section 7, we conclude with overall summary of our approach.

2. BACKGROUND

2.1. Instant Messaging Security

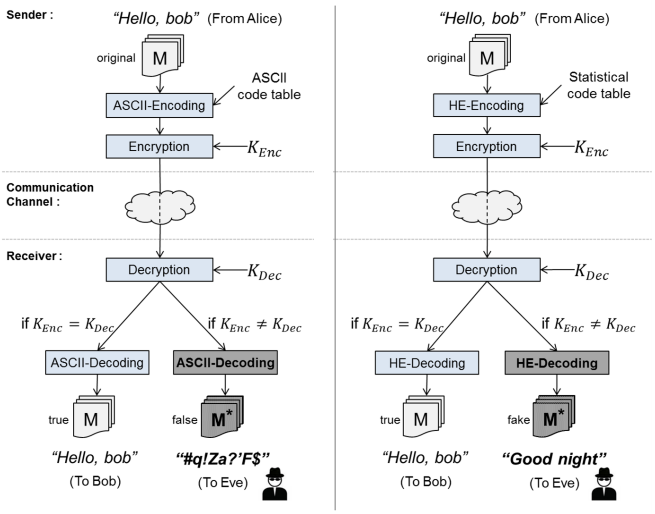
Instant messaging (IM) is a private network communication which transmits real-time texts between two or more users. There are considerations about IM security such as data transfer, user authentication, etc [6]. When it comes to data transfer, an original message is encrypted with cryptographic algorithm like AES before sending, and a user having a fair key can decrypt it. In client-server architecture of many IM systems, chat messages pass through central servers. Thus unencrypted messages including private data might be easily exposed to providers. That's why the message encryption technology like End-to-End Encryption comes to the fore as IM security technology. But there still exist threats by *brute-force attack* that exploits the essential vulnerability of key.

2.2. Honey Encryption

In order to improve the drawback of the conventional password based encryption (PBE) with low-entropy passwords, Juels and Ristenpart introduced *Honey Encryption* (HE) [1]. The main idea is that encryption of plaintext *M* is randomized with a password *k*, and decryption of ciphertext results in plausible-looking plaintext *M´* with wrong password *k'*. They construct a *distribution-transforming encoder* (DTE) for encoding and decoding of message as bit string, denoted DTE = (encode, decode). In brief, overall process is HE[DTE,SE]=(HEnc,HDec) where SE means conventional symmetric encryption. The ciphertext is C = HEnc(k, M) and decryption works M = HDec(k, C) or M' = HDec(k', C).

2.3. Entropy

Entropy is a measure of the uncertainty of a dataset in information theory. If the dataset is composed randomly, its entropy would be high. The entropy $H(X)$ of a discrete random variable X is $H_b(X) = -\sum_{x \in \mathcal{X}} p(x) \log_b p(x)$. Here, $p(x)$ is the probability mass function which denotes $\Pr\{X = x\}, x \in \mathcal{X}$ [7]. Note that b is the base of the logarithm, so b=2 in bits and b=26 in lower-case letters. In addition, entropy has been used in several ways to identify encrypted packet or detect the anomaly and worm [8, 9]. Since the purpose of cryptographic algorithm is to protect the original data from prediction, the encrypted bit stream would have high entropy which indicates uniformly distributed random variables.



(a) ASCII scheme (b) HE scheme

Fig. 1. Eve obtains *random* ASCII characters in (a), which means the key is wrong. On the other hand, in our proposed system (b), Eve gains a *plausible-looking* fake message.

3. THREAT MODEL

There might be a threat that a *brute-force attacker* obtains an encrypted message on the communication channel of IM system and tries to decrypt it by using every possible key. The attacker could notice the difference of entropy between decrypted messages with wrong keys and with a real key, and then find the original message. It's because the correctly decrypted message is mostly composed of chat texts like alphabets, digits, or some special characters while the messages decrypted with wrong keys have irregular random characters. Therefore, our goal is to make chat message *indistinguishable* when the attacker decrypt ciphertext and check if it is valid.

4. PROPOSED APPROACH

4.1. Concepts

Fig.1 shows the overall process of message transmission. (a) is the conventional encryption and decryption process using ASCII code, while (b) uses HE scheme as encoding and decoding method. In (b), a code table is made from the *statistical coding scheme* using *n-gram language model* of text corpus, and the sender and the receiver share it. The sender's message M is encoded using the code table and encrypted with K_{Enc}. It passes through the communication channel such as Internet. The receiver decrypt it with K_{Enc} and decode it using the same code table. If $K_{Enc} = K_{Dec}$, the receiver can obtain a *true* message in both cases. If $K_{Enc} \neq K_{Dec}$, however, an *random* message is appeared in (a) while a *plausible-looking* fake message is generated in (b).

4.2. N-gram Language Model

Chat messages can be represented by the *n-gram language model*. It is a probabilistic language model widely used to predict the next character in a sequence [10]. For example, the 5-gram model of the sentence "you are beautiful" has the probability of five subsequent characters, "you a", "ou ar", "u are", and so on. If we have a sequence "you ", then the following characters are likely to be "are" more than "is".

In n-*gram model*, the probability $P(x_1, x_2, \cdots, x_m)$ of discrete stochastic process $x_1 x_2, \cdots, x_m$ whose length is m can be expressed in the form of (n-1)th-order *Markov model*, so that $P(x_1, \cdots, x_m) \approx \prod_{i=1}^{m} P(x_i | x_{i-(n-1)}, \cdots, x_{i-1})$. By using this, we can obtain the probability of consecutive characters, and can build a code table using the *statistical coding scheme* introduced in the next section.

4.3. Statistical Coding Scheme

In the HE scheme proposed [1], messages are encoded using *distribution-transforming encoders* (DTE) limited in the bit stream and integers. Meanwhile, there is an approach to build the *cumulative massive function* (CMF) for image data [4]. The method to construct CMF as the code table can also be applied to the set of the message's characters. Therefore, we build the CMF of *i*-th character of input message as follows:

$$p_{\mathrm{cmf}}^{(\mathrm{i})}(c_k) = \sum_{k=0}^{S} \frac{p(x_i = c_k | \mathbf{x}_{i-1:i-n})}{\sum_{j=0}^{S} p(x_i = c_j | \mathbf{x}_{i-1:i-n})} \qquad (1)$$

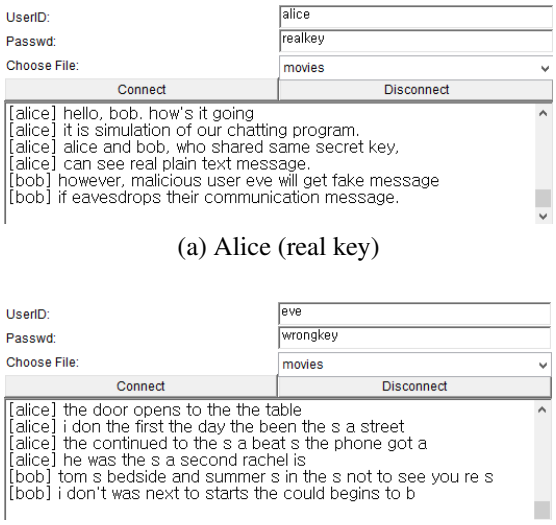
where S is the number of possible character set in the code table and n is the order of *Markov process*. The conditional posterior of *i*-th character $p(x_i | \mathbf{x}_{i-1:i-n})$ indicates that *i*-th character of message is influenced by near $n-1$ characters. After calculating the probability of each character, we need to adjust the probability of them because they have different weight relative to the frequency of appearance.

This CMF is served as a code table and shared between the sender and the receiver, converting hexadecimal numbers to characters and vice versa. When the sender transmits a message, it is encrypted with a key after encoding by referring to the code table constructed by using a *statistical coding scheme*. In addition, in a receiver's side, the encrypted message is decoded with the same code table which the sender has. Although we use AES in our *Honey Chatting* described later, it is possible to use any encryption method (e.g., RSA and DES) with the *statistical coding scheme*.

5. HONEY CHATTING APPLICATION

5.1. Structure and Simulation

We build a simple chatting system between users similar to an actual *Instant Messaging* system. It uses socket programing based on Java language and has a central server delivering message from the sender to the receiver. At the client-side, both users should enter their secret password before starting communication. Their messages are processed with *statistical coding scheme* and *password-based encryption* (PBE). In real situation, the PBE can be replaced with other methods.



(a) Alice (real key)

(b) Eve (wrong key)

Fig. 2. This is simulation of *Honey chatting* program. (a) is chat messages between Alice and Bob with real shared key, and (b) is eavesdropped chat messages of Eve with wrong key.

Fig.2 shows the simulation of our application. It is a chat room which two fair users, Alice and Bob, are participating in. Suppose that each user shares the same password as a secret key, and a malicious attacker Eve is trying to eavesdrop their chat. No matter what the wrong password is entered, she could see *plausible-looking* plain texts which are not real. Thus, she cannot sure whether the conversation between Alice and Bob is true or not. Consequently, Eve needs to do additional work to acquire their real message among fake messages, such as considering their way of talking, contents, etc.

5.2. Text Corpus and Generating Messages

It is important to choose text corpus as the basis of the coding scheme for training data, since the output fake messages of HE with wrong key is influenced by literary style of chosen text corpus. The reason is that we build a code table based on *Markov process*, the probability of consecutive characters appearing. Accordingly, both sender and receiver have to share the same code table for exact conversion or their messages. In *Honey Chatting* application, we select text database such as movie subtitles or fictions including much dialogue rather than description in order to make fake messages of HE to look more like chat messages. For the practical use in real world, we should consider context and grammar of the sentences to make messages be natural, and the available charac-

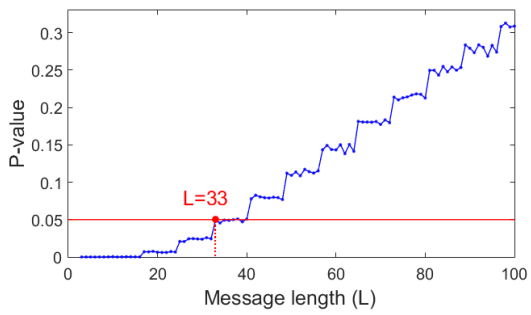


Fig. 3. The valid message length in Honey Chatting. $H(M)$, the entropy of message, is included in the 95% confidence range of $H(M^*)$ when message length is $L \geq 33$.

ter set should be increased which is now 30 characters: letters (a-z), space, period, and comma.

6. EXPERIMENT

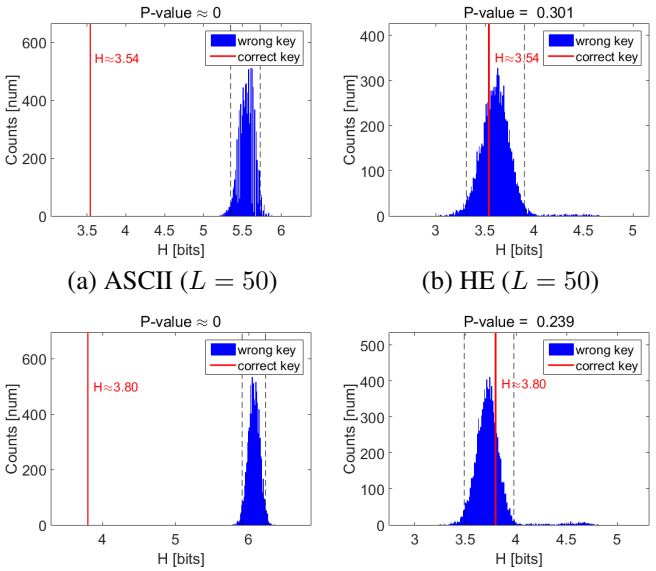
6.1. Experiment Procedures

We conduct a *significance test*, also called *hypothesis test*, to show difference between decrypted text with wrong and real key applying HE scheme. As a preparation, we make a plaintext M and ciphertext $C = Enc_k(M)$ with a correct key k. And then we repeatedly decrypt C with a wrong key k^* and obtain a wrong message $M^* = Dec_{k^*}(C)$ for 10,000 times. Here, a wrong key k^* is randomly generated each time.

Now we proceed with *significance test* as follows. The null hypothesis H_0 is *"There is no difference of entropy between M and M^*s"*, which can be interpreted into the meaning that the entropy of M is included in the scope of M^*s entropy. Thus, the alternative hypothesis H_a is *"There is difference of entropy between M and M^*s"*, which indicates they can be *distinguished*. We consider the entropy distribution of wrong messages M^*s as *test statistic*, while the entropy of real message M as *observed value*. The small *P-value* represents that the observed data M could not be included in the range of M^*s, so we reject H_0 and accept H_a[11].

6.2. Experiment Analysis

Now, the next question is what additional factors may influence to the effectiveness of the Honey chatting application. We found that one of the most serious factors is the chosen length of the message. Fig.3 represents the minimum length of message whose *P-value* is above 0.05. The entropy of message $H(M)$ is included in the 95% confidence range of $H(M^*)$ when message length is $L \geq 33$. Applying the HE scheme would be less effective if the length is smaller than this, because it is easy to distinguish M from M^*s. Here, the threshold of message length depends on the text corpus, movie subtitles and fictions in this experiment.



(a) ASCII ($L = 50$) (b) HE ($L = 50$)

(c) ASCII ($L = 100$) (d) HE ($L = 100$)

Fig. 4. The entropy difference between messages of length L. (a), (c) use conventional ASCII coding, while (b), (d) is using our proposed HE scheme.

Fig.4 demonstrates the difference of entropy between ASCII and HE decoding scheme about messages M and M^*s of length L . Each figure indicates the frequency of entropy $H(message)$. We calculate and compare entropy of M and 10,000 M^*s. The bell shaped blue bar is $H(M^*)$ with the wrong key, and the red solid line is $H(M)$ with the real key. The dotted line is the 95% confidence intervals. It is obvious that $H(M)$ is out of range of $H(M^*)$ in (a) and (c), whereas $H(M)$ is included in $H(M^*)$ in (b) and (d). Going back to the hypothesis above, we reject H_0 and accept H_a because the *P-value* in (a) and (c) is significantly small. In this case, there are *clear distinction* between M and M^*s. However, moderately large *P-value* in (b) and (d) shows that observed data M is agreed with H_0. It means that M is *similar* with M^*s, so the brute-force attacker could not notice success.

7. CONCLUSION

There are many *chatting systems*, which enhance security with technology such as message encryption. But it has the fundamental vulnerability related to the key. In other words, it might be possible to crack the key and snoop the content using a *brute-force attack* by a computationally-unbounded attacker. In this paper, we proposed a new type of secure chatting system, *Honey Chatting*. By generating *plausible-looking* messages when trying to attack encrypted messages, we can confuse the attacker and prevent him/her from achieving the actual content. Through this approach, we could build a messaging system which is robust to *eavesdropping*.

8. REFERENCES

[1] A. Juels and T. Ristenpart, "Honey encryption: Security beyond the brute-force bound.," *IACR Cryptology ePrint Archive*, vol. 2014, pp. 155, 2014.

[2] J. W. Yoon and H. K. Noh, "Deniable encryption system and method," Jan. 22 2015, WO Patent App. PCT/KR2014/006,579.

[3] H. J. Jo and J. W. Yoon, "A new countermeasure against brute-force attacks that use high performance computers for big data analysis," *International Journal of Distributed Sensor Networks*, 2015.

[4] J. W. Yoon, H. S. Kim, H. J. Jo, H. L. Lee, and K. S. Lee, "Visual honey encryption: Application to steganography," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, New York, NY, USA, 2015, IH&MMSec '15, pp. 65–74, ACM.

[5] R. Chatterjee, J. Bonneau, A. Juels, and T. Ristenpart, "Cracking-resistant password vaults using natural language encoders.," in *IEEE Symposium on Security and Privacy*. 2015, pp. 481–498, IEEE Computer Society.

[6] R. B. Jennings III, E. M. Nahum, D. P. Olshefski, D. Saha, Z. Y. Shae, and C. Waters, "A study of internet instant messaging and chat protocols," *IEEE Network*, vol. 20, no. 4, pp. 17, 2006.

[7] T. M. Cover and J. A. Thomas, *Elements of information theory*, Wiley-Interscience, 2006.

[8] P. Dorfinger, G. Panholzer, and W. John, "Entropy estimation for real-time encrypted traffic identification.," in *TMA*, Jordi Domingo-Pascual, Yuval Shavitt, and Steve Uhlig, Eds. 2011, vol. 6613 of *Lecture Notes in Computer Science*, pp. 164–171, Springer.

[9] A. Wagner and B. Plattner, "Entropy based worm and anomaly detection in fast ip networks.," in *WETICE*. 2005, pp. 172–177, IEEE Computer Society.

[10] W. B. Cavnar and J. M. Trenkle, "N-gram statistics for natural language understanding and text processing," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1979, vol. 2, pp. 164–172.

[11] A. Agresti and B. Finlay, *Statistical Methods for the Social Sciences*, Pearson Prentice Hall, 2009.