International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 6th Feb 2020. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-09&issue=ISSUE-02

Title: HONEY ENCRYPTION WITH QUANTUM KEY DISTRIBUTION

Volume 09, Issue 01, Pages: 23-29.

Paper Authors

SRILATHA KOMAKULA, V.SHOBHA RANI

Chaitanya Deemed to be University

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

HONEY ENCRYPTION WITH QUANTUM KEY DISTRIBUTION

***SRILATHA KOMAKULA[1], **V.SHOBHA RANI[2]**

* Assistant Professor, Dept. Of Computer Science, Chaitanya Deemed to be University,

** Assistant Professor, Dept. Of Computer Science, Chaitanya Deemed to be University,
srilatha.kom@gmail.com shobhareddy19@gmail.com

ABSTRACT

We present a comprehensive survey of the Honey Encryption (HE) scheme. Honey Encryption is an encryption scheme that provides resilience against brute-force attack by serving up plausible-looking but fake plaintext for every invalid key used by an intruder to decrypt a message. Our goal is to furnish researchers with the framework of the scheme not just for implementation purpose but to identify the gaps in the scheme and answer the open questions that remain unanswered by the small set of research carried out since its inception. There are many hash cracking tools available which can easily crack these hashes when the passwords are weak. Weak passwords are not just the problem for hashing but also affect the security in Password-Based Encryption (PBE) scheme where the message is encrypted under a password. PBE is used to protect sensitive data and mostly used in Password Managers. Password Manager (PM) compiles small database of passwords and their associated accounts, and this database is encrypted with a user-selected master Password and is therefore vulnerable to brute force cracking of Master Password. In this review paper we have studied Honey Encryption (HE) which is a new encryption scheme that provides resilience against brute force attacks by ensuring that messages decrypted with invalid keys yield a valid-looking bogus message.

1. INTRODUCTION

The craft of deception is indispensable in the event of confronting an enemy. It enables an environment where an adversary is trapped into taking actions that consume/wastes his resources [1-3]. Employing deception and decoy techniques in network systems help to detect, trace, monitor and deter the activities of an adversary [4-6]. It is staged to make the adversary's life difficult where a false reality is projected as a reality to him.

Indeed, Sir Sun Tzu encapsulated the art of deception in a perspicuous sentence when he said, "*The art of war teaches us not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable*," [7].

We trace historical examples of the use of decoys to 1943 when the British found the

corpse of a homeless man and went through extraordinary length to fabricate his death and created a prior but fake existing personal life for him to deceive the Germans. His realistic but fake persona included him been a captain in the military, having a father whom he sends/receives letters from and a fiancee's letter and photo. Also, the British planted some fake papers on him indicating a false location for an Allied attack. Of course, the Germans found the dead man's body and the letters on him. They read the letters and believed everything on it based on the '*supposed*' evidence found on him. Subsequently, they diverted their attention and military warfare to some other region. Unknown to them, they were conned and the Allied troop landed. For a long time, the German military continued to think that there was a diversion even after the Allied troop landed. History has it that this British facade of the human decoy saved over forty (40) thousand Allied lives.

Honey encryption addresses the flaws of password-based encryption schemes and is currently employed in securing most password-based system in the form of honeywords. However, honey encryption has not been employed in most systems such as, its application for encrypting human written documents like e-mails, etc. The challenge is how do we create contextually and semantically correct decoy-message that can actually fool an attacker? Other problems like typo-safety have remained unaddressed. For instance, if a legitimate receiver mistakenly enters a wrong password. Given that decoy system can completely address brute force attack which standard encryption schemes are susceptible to, then there is need to foster research in this area. Moreover, the current advance made on quantum computers propels us to search for quantum-safe cryptosystems. Since all our encryption schemes are exclusively based on Mathematical problems which are established based on the difficulty of solving discrete logarithm and number factorization problem [18], this puts us at an 'encryptionless' state as soon as quantum computers solve all the underlying Mathematics used to secure our modern cryptosystems. The HE scheme is not devised under the computational difficulty of breaking them alone but the real trickery that cryptography was meant to be built upon [10-12]. Consequently, honey encryption if properly designed and implemented, will be a good supplementary encryption scheme for the quantum era. Therefore, the target of this paper is to capture and present a synopsis of the current state of Honey encryption scheme to identify the gaps in the scheme to enable its optimization for real-life applications.

2. RELATED WORK

Most of the systems with encryption use password-based encryption (PBE). These systems are susceptible to brute force guessing attacks. Honey encryption [5] aims to address this vulnerability by not allowing attackers to gain much information from password guessing. For each possible key, the system outputs a valid-looking decrypted message. So it is hard to tell which one is the correct password. This way honey encryption can protect sensitive data in

many applications. Honey encryption deceives attackers that the incorrectly guessed key is valid. Many luring technologies that also use the term honey have been proposed in the last 20 years. Honey tokens [6] are decoys distributed over a system. If any decoy is used, this means that a compromise is taking place. For example, honey words are passwords that are rarely used by normal users. Once a login attempt using a honeyword occurs, the system rises an alarm. Honeypots [3], Honeynet [7], and Honey farm [8] are luring systems that present many vulnerabilities. They are likely to become the targets for attackers. The objective of setting such systems is to study attackers' motivations, tools, and techniques. Honey encryption is also related to Format-Preserving Encryption (FPE) and Format-Transforming Encryption (FTE). In FPE, the plaintext message space is the same as the ciphertext message space. In FTE, the ciphertext message space is different from the message space. Honey encryption maps a plaintext message to a seed range in the seed space. Since the message space and the seed space are different, the ciphertext message space is different from the message space. While Vinayak and Nahala apply the honey encryption concept to MANETs to prevent ad hoc networks from the brute-force attack, Tyagi et al. [5] adopt the honey encryption technique to protect credit card numbers and a simplified version of text messaging. Most of these data in are from uniformly distributed message spaces. However, genomic data usually has highly nonuniform probability distributions. The GenoGuard mechanism [1] incorporates the honey encryption concept to provide information-theoretic confidentiality guarantees for encrypted genomic data. In [1, 5], a fixed distribution-transforming encoder (DTE) is utilised for encryption and decryption, so it is only suitable for binary bit streams or integer sequences, not for images and videos. Yoon et al. propose a visual honey encryption concept which employs an adaptive DTE so that the proposed concept can be applied to more complex domains including images and videos.

Jaeger et al. provide a systematic study of honey encryption in the low entropy key settings. They rule out the ability to strengthen the honey encryption notions to allow known-message attacks when attackers can exhaust the key space. Our paper focuses on applying the honey encryption technique to three new applications including citizens' identification numbers, mobile phone numbers, and debit card passwords. This data is vital private information that can cause serious damage to person's finance and/or reputation if stolen. Although honey encryption has been applied to a number of applications, due to the variety of message formats and probability features, the message space design needs to vary for new types of applications. The applications discussed in our paper are carefully selected, because later we will show that the protection honey encryption provides varieties for different applications: stronger for debit card passwords, weaker for mobile phone numbers. In our comprehensive design and implementation of the honey

encryption mechanisms for three different applications we cover small/large message spaces, uniformly/nonuniformly distributed message probabilities, and symmetric/asymmetric encryption mechanisms. As far as we know, our paper is the first one to study the performance of honey encryption. We discover the performance problem for large message spaces and present a performance optimisation for small message spaces. We also show that the capability of honey encryption to address the brute-force vulnerability could be lost if the message space has not been well designed and that this design needs to vary for different types of applications.

3. HONEY ENCRYPTION CONCEPT

Honey encryption protects a set of messages that have some common features (e.g., credit card numbers are such messages). A message set is called a message space. Before encrypting a message, we should determine the possible message space. All messages in the space must be sorted in some order.Then the probability of each message (PDF) that occurs in the space and the cumulative probability (CDF) of each message are needed. A seed space should be available for the distribution-transforming encoder (DTE) tomap each message to a seed range in the seed space (n-bit binary string space).The DTE determines the seed range for each message according to the PDF and CDF of the message and makes sure that the PDF of the message is equal to the ratio of the corresponding seed range to the seed space. The n-bit seed space must be big enough so that each message can be mapped to at least one seed. A message can be mapped to multiple seeds and the seed is randomly selected.

Let us consider using honey encryption to encrypt the coffee types, as shown in Figure 1. The coffee message space Mconsists of Cappuccino, Espresso, Latte, andMocha.These fourmessages are sorted alphabetically. Let us assumethat 4/8

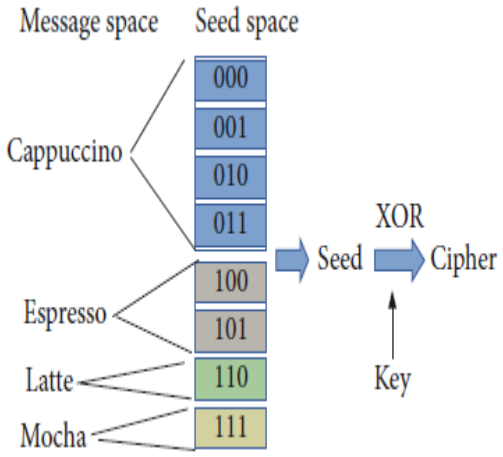

Figure 1: Honey encryption example.

people in Sydney like Cappuccino, 2/8 like Espresso, 1/8 like Latte andMocha. The seed space is a 3-digit space. According to those probability statistics, we map these four messages to four ranges in the seed space. When encrypting Cappuccino, theDTE randomly selects a seed fromthe corresponding seed range. The seed is XORed with the key and the ciphertext is generated.

For decryption, the ciphertext is XORed with the key to obtain the seed. Then the DTE inversely maps the seed to the original plaintext message. In the encryption process, a message could have multiple mapping

choices and the mapping is directional and random. However, since we sort plaintext messages in the message space and determine the seed range by the PDF and CDF of each message, it can be guaranteed that the seed ranges are arranged in the same order and the cumulative probability of the seed range in the seed space is equal to the cumulative probability of the message in the message space. Therefore, we establish an inverse table that consists of mappings of the cumulative probability to the plaintext message. Finding the seed, we can determine the seed range. Finding the seed range, we can determine the cumulative probability shared by the seed range and the corresponding plaintext message. Then by looking up the cumulative probability in the inverse table, we can find the original plaintext message and the ciphertext is decrypted.

4. DESIGN AND IMPLEMENTATION

We can design DTE as a common module that implements the encryption and decryption algorithms. For encryption, the DTE module takes in some parameters from the message space including the PDF and CDF probabilities of each message. Therefore, we abstract some interfaces for DTE to use when designing the message space. For decryption, the main task for DTE is to search the inverse table and find the correct plaintext message. Therefore, the message space implementation should provide interfaces for probabilities and the inverse table.

4.1. Message Space APIs. DTE maps the plaintext message

to a seed in a seed range. The starting point of the seed range is determined by the CDF of the message, while the end point of the seed range is determined by the PDF of the message. Therefore, we define an interface for the message space containing functions including the cumulative probability(mesg) function and the probability(mesg) function. These two functions accept a plaintext message as the parameter and output the CDF and PDF, respectively.

In decryption, DTE finds the plaintext message from the inverse table by looking up the cumulative probability of the seed. The inverse table is stored in a file. We define another function as get inverse table file name() in themessage space interface.The function returns the filename of the inverse table for DTE to look up and decrypt the ciphertext.

If the inverse table is not large, we can store the content in the memory when the system initiates.Then during decryption, the binary search method can be utilised to save time. However, if the inverse table size exceeds the available system memory, DTE needs to read the inverse table file line by line and find the plaintext by linear search.

4.2. DTE Implementation.

DTE maps the plaintext message into a seed range, randomly selects a seed from the range, and XORs the seed with the key to output the ciphertext.The beginning of the seed range is determined by the CDF and the end of the seed range is determined by the PDF. The seed is randomly selected from the range.

When decrypting a ciphertext, the ciphertext is XORed with the key to obtain the seed. Then DTE determines the location of seed in the seed range. The location is corresponding to a probability value which lies between the CDF of the message and the CDF of the next message in the message space. Every line in the inverse table contains a cumulative probability and its corresponding plaintext message. All lines are sorted by the cumulative probability. By searching the inverse table, the DTE can find the plaintext message given the cumulative probability determined by the seed.

CONCLUSION

In this work, we initiated the study of security notions for honey encryption schemes stronger than the previously proposed goal of resistance to message recovery attacks. We, first, proved that message recovery is always possible with a known-message attack. Formally proving this folklore result was more nuanced than expected. We then defined semantic security and non-malleability for honey encryption schemes with respect to targeted message distributions, and we showed that the simple constructions of encode-then-encrypt and encode-then-encipher achieve targeted distribution semantic security and targeted distribution non-malleability, respectively.

REFERENCES

[1] Omolara AE, Jantan A, Abiodun OI, Poston HE. A Novel Approach for the Adaptation of Honey Encryption to Support Natural Language Message. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* 2018 (Vol. 1).

[2] Disso JP, Jones K, Bailey S. A plausible solution to SCADA security honeypot systems. In *Broadband and Wireless Computing, Communication and Applications* (BWCCA), 2013 Eighth International Conference on 2013 Oct 28 (pp. 443-448). IEEE.

[3] Bringer ML, Chelmecki CA, Fujinoki H. A survey: Recent advances and future trends in honeypot research. *International Journal of Computer Network and Information Security*. 2012 Sep 1;4(10):63.

[4] Dhanji PK, Singh SK. Assault Discovery and Localizing Adversary in Remote Networks. *Indonesian Journal of Electrical Engineering and Computer Science*. 2018 Jan 1;9(1): 81-84

[5] Zhang D. Inconsistency: the good, the bad, and the ugly. In *Information Reuse & Integration*, 2009. IRI'09. IEEE International Conference on 2009 Aug 10 (pp. 182-187). IEEE.

[6] Pawlick J, Colbert E, Zhu Q. A Game-Theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy. *arXiv* preprint arXiv:1712.05441. 2017 Dec 14.

AUTHOR 1:

NAME: SRILATHA KOMAKULA
QUALIFICATIONS: MCA.,M.Tech.
DESIGNATION: Assistant Professor

DEPARTMENT: Computer Science
Chaitanya colleges(Autonomous)

AUTHOR 2:

NAME: V.SHOBHA RANI
QUALIFICATIONS: M.Sc(CS).,M.Tech.
DESIGNATION: Assistant Professor
DEPARTMENT: Computer Science
Chaitanya colleges(Autonomous)