International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Review on Honey Encryption Technique

Nahri Syeda Noorunnisa[1], Dr. Khan Rahat Afreen[2]

[1] Department of Computer Science and Engineering, Marthawada Shikshan Prasark Mandal‟s Deogiri Institute of Engineering & Management Studies, Aurangabad Maharashtra state, India 2015-16

[2] Associate Professor, Department of Computer Science and Engineering, Marthawada Shikshan Prasark Mandal‟s Deogiri Institute of Engineering & Management Studies, Aurangabad Maharashtra state, India 2015-16

Abstract: ***Users mostly select common passwords which are easy to remember and easy to guess. Passwords are often protected in the database in the form of cryptographic hash function. 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.***

Keywords: Password-based Encryption, Password Managers, Brute force attack, Honey Encryption.

1. Introduction

Most of the users select the passwords that are very simple and therefore easy to remember. Thus the problem is that, if it is easy to remember it is also easy to be predicted by the attacker. Most users would like to pick one password and use it for all their accounts, never change it and write it down for future references.

About 32 million clear-text passwords were exposed when an attacker was able to break into the database of RockYou.com which provides services and applications for social networking sites, through SQL vulnerability. [1] This breach provided a proof that consumers repeatedly use easy to guess login credentials. After analyzing the data it was found that the top ten common password that were used are 123456, 12345, 123456789, password, iloveyou, princess, rockyou, 12345678, abc123. [1] The trivial nature of the top ten exposed passwords is bad enough, but more worse is that nearly 50 percent of passwords which were exposed from RockYou breach used name, slang words or dictionary words. If these login names and passwords become easy to guess then it becomes more likely that the attackers or hackers will be able to break into accounts using various attacking techniques such as Brute force, Dictionary attacks and readily available password cracking tools.

The common selection of passwords by the users which are easy to remember is the main reason behind the development of Honey Encryption (HE). The term „honey‟ in computer security, is commonly used to denote a false resource which is designed to deceive or lure an attacker. For example, Honeypots are the servers which are designed to attract the attackers for observation and study. Honey Encryption technique creates a cipher text that when decrypted with an incorrect key or passwords yields a valid looking bogus message and so the attackers can‟t tell when the decryption has been successful.

2. Literature Survey

2.1 The problem of weak passwords:

The compromise of database of passwords of RockYou, [1] Yahoo, [2] Adobe, [3] LinkedIn [4] revealed that simply storing the passwords in plaintext in vulnerable to attacks. For securing the passwords, the passwords are now stored in database in the form of cryptographic hash function rather than being stored in the form of plaintext. Hashing is done by using a cryptographic hash function which is irreversible.

2.2 Password Hashing

Hash algorithms are one-way cryptographic functions which turn any amount of data into a fixed length „fingerprint‟ that cannot be reversed back to the original data.[5] Hash algorithms also have a property that if the input changes by even a small amount, the resulting hash changes completely from the original one.

There are many techniques to crack the plain hashes and obtain the original passwords. And therefore simply hashing the passwords does not meet the need for good security. Following are some of the common attacks which are used to crack plain text password hashes:

Dictionary and Brute force attacks: These are two most common ways of sussing passwords. One of the simplest way to crack the hash is to guess the password, then hash each guess followed by checking that guess‟s hash equals the hash that is being cracked; if the hashes are equal, the guess is password.

Lookup tables: An extremely effective method for cracking many hashes of same type very quickly is using lookup tables where the basic idea is to pre-compute the hashes of password in a password dictionary and store them and their corresponding passwords in the lookup table data structure.

Rainbow Tables: Rainbow table implements the time-memory trade-off technique and it is quick and effective way of doing cryptanalysis. [6] They are similar to look up tables with the exception that they compromise the hash cracking speed in order to make the lookup tables smaller. Because they are smaller, the solutions to more password hashes can be stored in the same amount of memory space, making rainbow tables more effective.

2.3 Adding Salt to Hashes

The method of storing a simple hash of a password does not ensures that the passwords are stored securely. Two of strengths of hashes are also their largest potential weakness: they are very small to store and quick to generate. The solution to this is to use the method of „salting", which means hashing more than just the user"s passwords. In the process of salting, the hashes are randomized by appending or pre-pending a random string called a „salt", to the password before hashing. As a result of this, the same password hash gets transformed into a completely different string every time.

2.4 Hash Cracking Tools

Now-a-days many hash cracking tools are easily available. These tools enable the attackers to easily crack the hashes when the underlying passwords are weak. These tools exploit the knowledge of how the users typically compose their passwords. Hash cracking tools are the major reason behind the disclosure of hashed passwords- for example, the 2014 breach of Yahoo. [7]

Some of the hashes cracking tools are:

- John The Ripper: One of the world"s best passwords cracking tool is John the Ripper which is free and Open Source software. It is strictly command line and for Linux Operating System. [8]
- Ophcrack: One of the free Rainbow Table based password cracking tool for Windows is Ophcrack. It can also be used on Linux and Mac systems. [8]
- Brutus: Brutus is an online password cracker and is considered by many as the fastest online password cracker. [8]
- RainbowCrack: RainbowCrack software uses rainbow table to crack hashes. It uses the process of large scale time-memory trade-off for effective and fast password cracking. It is available for Linux and Windows Operating System. [9]

3. Password Based Encryption

Vulnerable passwords are not just the problem for hashing but they also impact user"s ability to encrypt sensitive data using Password-based Encryption (PBE) scheme. PBE carries the same vulnerability to guessing attacks as Hashing. The technique of PBE consists of an encryption function *enc()* and a corresponding decryption function *dec()*. A message M is encrypted under a password P as,

$$\text{Ciphertext } C = enc\ P(M)$$

The message can be decrypted as,

$$M = dec\ P(C)$$

Given a decryption attempt using an incorrect password $P' \neq P$, *dec P(C)* outputs an error message, which makes it clear that we have entered an incorrect password. [10]

Thus with the traditional approach, explicitly giving an error message notifies that the password entered is incorrect. Authenticated encryption also means that adversaries making password-guess attempts against a PBE ciphertext knows when they have decrypted successfully.

Although the passwords are considered secure, but if enough computations are done then the passwords are vulnerable to brute-force attacks. The decryption of a ciphertext through brute force guessing of passwords can be confirmed with a valid-looking message output, but more importantly, an invalid-looking output as confirmation of an unsuccessful attempt.

4. Password Managers

A Password Manager (PM) helps a user in managing their passwords and associated accounts in a secure manner. Password manager"s stores encrypted passwords and it requires the user to create a master password. A master password is a user selected strong password which is used to encrypt the password database and later grants the user access to the entire password database. The primary function of the password managers is to store and remember all the user passwords and its associated accounts so that the user will not have to remember. [11] It stores the user passwords and also the user"s personal information in an encrypted file which will help in protecting the confidential data of user from the attackers.

The encrypted file can be accessed only through the use of user selected master password, which means that user will have to remember only a single master password, and all the remaining passwords and the other forms of data in the encrypted file will be remembered by the password manager. Password Based Encryption scheme is used to protect sensitive information and is notably used in Password Managers. Many users store and protect their passwords in password managers, such as DashLane, LastPass, or Apple"s iCloud keychain. Password Managers provides a database of passwords and their associated accounts and this database is encrypted with a user-selected master password. If such a database is breached, then brute-force cracking of master password will yield all the user passwords.

Although a password manager greatly reduces the burden of the user in remembering the password, but it also introduces a point of failure in this. If an attacker obtains a encrypted vault of passwords it can mount offline brute force attacks and attacker achieves success, then this will lead to the compromise of all the passwords stored in the vault. [12]

5. Honey Encryption

Honey Encryption technique was developed by Ari Juels, former chief scientist of the RSA, and Thomas Ristenpart from the University of Wisconsin. [10] Honey Encryption is best-suited in the situations where the encrypted data is obtained from the passwords. If an attacker tries to carry out brute force attack then using the Honey Encryption security tool makes it complicated for an attacker to know if he has correctly guessed a password or encryption key. If Honey Encryption has been used, then however, the wrong guesses of an attacker generates similar results that appear to be real. Because each incorrect guess generates a plausible looking result, thus the attacker is misguided by honey Encryption. For an example, if an attacker tries to get a credit card number by making 1000 attempts, then for all the 1000 attempts he will be getting 1000 fake credit card numbers. Each decryption is going to look as plausible as other. The attacker has no way to distinguish a priori which is correct.

5.1 Distribution Transforming Encoders

The DTE (Distributed Transforming Encoding) is the main idea behind pure honey encryption technique. Honey encryption manages the space of plaintext via DTE. Let the probability distribution over the message space be p over the message L.

The distribution transforming encodes the message L as a K bit seed $S \in \{0, 1\}K$ and decodes the message by inverse DTE method, *decode (S)* $=L$. DTE is a good model of the message distribution. The internal structure of the HE includes DTE encryption and DTE decryption .The two algorithms describes the net functioning of the Honey Encryption.

Honey Encryption Algorithm:
$H \leftarrow \text{Enc}(X, L)$
$S \leftarrow \$$ *encode (L)*
$R \leftarrow \$ \{0, 1\} n$
$S' \leftarrow H(R, X)$
$C \leftarrow S' \oplus S$

Honey Decryption Algorithm:
$H \leftarrow \text{Dec}(X, (R, C))$
$S' \leftarrow H(R, X)$
$S \leftarrow C \oplus S'$
$L \leftarrow$ *decode(S)*
Return L

H is a cryptographic hash function, X is a key, L is a message, S is a seed, R is a random string, C is a cipher text and $\leftarrow\$$ indicates that Honey Encryption algorithm may use some number of uniform random bits. When the Honey Encryption is applied to the plaintext message L, it first encodes the message L to S and then encrypts S by a key X using suitable symmetric encryption algorithm .The above algorithms describes these steps clearly, high message recovery security is provided by Honey encryption. This functioning can be described by an example encrypting soft drink flavours. This example includes three flavours such as Apple, Mango and Orange. These encrypted items will have a two bit string such as {00, 01, 10, 11} etc.

Honey encryption can be described by the following example. Let's assume Bob want to encrypt his favourite soft drink flavour L= Mango that is to be send to Alice under a secret key =0000 that is shared with Alice. Bob constructs a flavour soft drink DTE that maps the message L into the space of 2-bit strings {00, 01, 10, 11}. The working is like via DTE the encoded Apple will have the value 00 and encoded Orange will have the value 10 or 11 which is randomly chosen .The message encoded by Bob that is Mango is having the value 01. Bob selects a random string R and computes $S'=H(R, X)$ and assume that $S'=(R, 0000) =11$ and then the Bob computes $C=11 \oplus 01=10$ and it is forwarded to Alice. [13]

Alice decrypts C by the key that has been shared by the Bob that is key X=0000. So $S'=H(R, 0000) =11$, and $S=C \oplus S'=10 \oplus 11=01$ and the encode (01) = Mango and the message is successfully recovered by the Alice. Suppose an attacker Eve tries to decrypt it. He doesn't know the key that is used so he assumes key such that of 1432, $H=(R, 1432) =$ 00 and then $S''=C \oplus S'=10$, and by decoding it he will get decode (10) = orange. Thus the attacker is fooled by this new type of encryption.

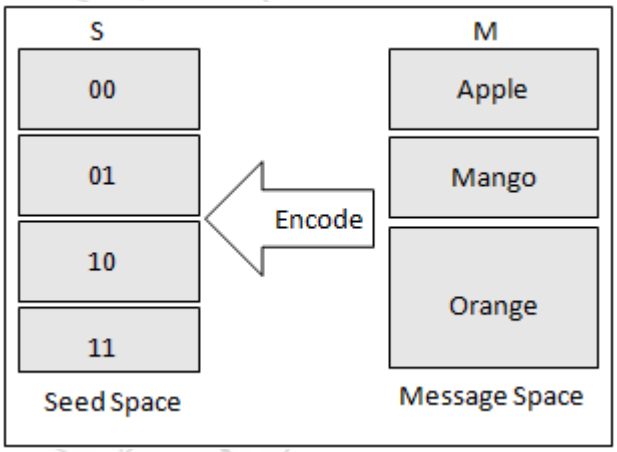

Figure 1: DTE Mapping

The message "apple" (with $p_m = 1/4$) maps to 00, "mango" (with $p_m = 1/4$) maps to 01, and "orange" (with $p_m = 1/2$) maps to {10, 11}; p_m is a probability distribution over the message space.

References

[1] RockYou hack exposes names, passwords of 32M accounts | Computerworld. [Online]. Available: http://www.computerworld.com/article/2522045/security0/rockyou-hack-exposes-names--passwords-of-30m-accounts.html. [Accessed: Nov 5 2015]
[2] Yahoo Hacked, 45,000 passwords posted online – CNN.com. [Online]. Available: http://edition.cnn.com/2012/07/12/tech/web/yahoo-users-hacked/ [Accessed: Nov 5 2015].
[3] Number of Adobe Accounts Hacked Now up to 150M, Check Yours. [Online]. Available: http://petapixel.com/2013/11/07/number-adobe-accounts-hacked-now-150m-check/ [Accessed: Nov 5 2015].

[4] More than 6 million LinkedIn passwords stolen. [Online]. Available: http://money.cnn.com/2012/06/06/technology/linkedin-password-hack/ [Accessed: Nov 5 2015].
[5] Pritesh N. Patel, Jigisha K. Patel and Paresh V. Virparia, "A Cryptography Application using Salt Hash Technique", Volume 2, Issue 6, June 2013
[6] Philippe Oechslin, Laboratoire de Securit´e et de Cryptographie (LASEC) Ecole Polytechnique F´ed´erale de Lausanne Facult´e I&C, 1015 Lausanne, Switzerland, "Making a Faster Cryptanalytic Time-Memory Trade-Off ". D. Boneh (Ed.): CRYPTO 2003, LNCS 2729, pp. 617–630, 2003.
[7] Yahoo Hacked and How to Protect Your Passwords. [Online]. Available: http://www.forbes.com/sites/jameslyne/2014/01/31/yahoo-hacked-and-how-to-protect-your-passwords/ [Accessed: Nov 10 2015]
[8] Hack Like a Pro: How to Crack Passwords, Part 1 (Principles & Technologies). [Online]. Available: http://null-byte.wonderhowto.com/ how-to/hack-like-pro-crack-passwords-part-1-principles-technologies-0156136/ [Accessed: Nov 10 2015].
[9] RainbowCrack- Crack Hashes with Rainbow Tables. [Online]. Available: http://project-rainbowcrack.com/ [Accessed: Nov 10 2015].
[10] Ari Juels, Thomas Ristenpart, "Honey Encryption-Encryption beyond Brute Force Barrier" IEEE Security and Privacy July/August 2014.
[11] Ambarish Karole, Nitesh Saxena, and Nicolas Christin, "A Comparative Usability Evaluation of Traditional Password Managers"
[12] Rahul Chatterjee, Joseph Bonneauy, Ari Juels, Thomas Ristenpart, "Cracking-Resistant Password Vaults using Natural Language Encoders" .
[13] Vinayak P P, Nahala M A, "Avoiding Brute Force attack in MANET using Honey Encryption", IJSR Volume 4 Issue 3, March 2015.