Encrypted Negative Password Using RSA Algorithm

Salwa P.B[1], Nice Mathew[2]

[1]M.Tech student Computer Science and Engineering,
[2]Asst.Professor, Department of Computer Science and Engineering, Indira Gandhi Institute of Engineering and Technology, Nellikuzhi, Kerala, India

---***---

Abstract - *Secure password storage is a in systems major fact based on password authentication, which has been widely used in authentication technique. Proposing a password authentication framework that is designed for secure password storage and it can be easily integrated into existing authentication systems. First, the received plain password from a client side is hashed using a cryptographic hash function (e.g., SHA-256).Then, hashed password is converted into a negative password. Finally, the received negative password is encrypted into an Encrypted Negative Password (abbreviated as ENP) using a symmetric-key algorithm (e.g., AES).Using multi-iteration encryption could be employed to further improve security. Both the cryptographic hash function and symmetric encryption make it difficult to crack passwords from ENPs. Finally, the received encrypted negative password is again encrypted using the RSA algorithm to improve the security of the password.*

***Key Words*: Encrypted Negative Password, Symmetric key algorithm, Hashed password**

1. INTRODUCTION

By the large development of the Internet, a huge number of online services have emerged, which password authentication is the most widely used authentication technique, for it is available at a low cost. Password security always attracts great interest from academia and industry. Because of careless behavior of the users password has been cracked, hence password authentication technique has been increasing. For instance, many of the users select weak passwords so that it can be reuse same passwords in different systems. Because they set their password according to their familiar vocabulary. It is very difficult to obtain passwords from high security systems. On the other side stealing authentication data tables (containing usernames and passwords) in high security systems is difficult.

1.1 OBJECTIVE

The aim of the paper is to enhance password security. When carrying an online guessing attack, there is a limit to the number of login attempts. However, passwords can be leaked from weak systems. Some old systems are more vulnerable due to their lack of maintenance. The passwords are often reused, adversaries may log into high security systems through cracked passwords from low security systems. There are lots of corresponding ENPs for a given plain password, which makes attacks (e.g., lookup table attack and rainbow table attack infeasible. The complexity analyses of algorithm and comparisons show that the ENP could resist lookup table attack and provide stronger password protection under dictionary attack. It is mentioning that the ENP does not introduce extra elements (e.g. salt). Most importantly, the ENP is the first password protection scheme that combines the cryptographic hash function, the negative password and the symmetric-key algorithm without the need of any for additional information except the plain password. The key certificates has been used to authenticate the user's key pair. Finally, the received encrypted negative password is again encrypted using the RSA algorithm to improve the security of the password.

1.2 SCOPE

By securing the password the online sites can provide security and protected from the cracking password. Passwords in the authentication data table presented in the form of hashed passwords. Processor resources and storage resources are becoming more and more abundant, so that the hashed passwords cannot resist pre computation attacks, such as rainbow table attack and lookup table attack. Moreover, they download and use attack tools without the need of any professional security knowledge. Some powerful attack tools, such as hashcat, Rainbow Crack and John the Ripper, provide functions, such as multiple hash algorithms, multiple attack models, multiple operating systems, and multiple platforms, which grand higher demand for secure password storage. In these situations, attacks are usually carried such as adversaries pre compute a lookup table, where the keys are the hash values of elements in a password list which contains frequent used passwords, and the records displayed are corresponding plain passwords in the password list. From the low security system generate a authentication data table .Finally, they search for the plain passwords in the lookup table with corresponding matching hashed passwords in the authentication data table and the keys in the lookup table. Then, by log into higher security systems through cracked usernames and passwords, they could steal more sensitive information of users. One of the main advantage that above lookup table attack is that the corresponding hashed password is determined for a given plain password. So that the lookup table could be quickly constructed, and the size of the lookup table could be large, which result in high success rate of cracking hashed passwords.

2. RELATED WORK

Some of the password protection schemes are hashed password, salted password and key stretching.

1) Hashed Password: The simple way to store passwords is to directly store plain passwords. However, this scheme presents a problem that once adversaries has been obtain the authentication data table, all passwords are immediately compromised. To store safely hash passwords using a cryptographic hash function, because it is infeasible to recover plain passwords from hashed passwords. The cryptographic hash function maps the data of arbitrary size to a fixed-size sequence of bits. In the authentication system that using the hashed password scheme, only hashed passwords are stored. However, hashed passwords doesn't resist lookup table attack. Rainbow table attack is more practical for its space-time tradeoff. Processor resources and storage resources are becoming richer, so the precomputed tables used in the above two attacks become large, so that adversaries obtain a higher success rate of cracking hashed passwords.

Table -1: Attending pattern examples

PATTERN EXAMPLE	PASSWORD EXAMPLE
Appending[0-9]	**Password1,princess1,angel1**
Appending 123	**Abc123,love123,red123**
Appending 1234	Abcd1234,abc1234,love1234
Appending DOT	Password,iloveyou,followingyou
Appending !	Iloveyou!, password!, rockypu!
Appending 101	Love101,zoey101,),sweet101

2) Salted Password: Salted password scheme used to resist precomputation attack. In this scheme, the plain password and a random data (called salt) is hashed through a cryptographic hash function. The salt is usually generated random, which shows that the hash values of the same plain passwords are almost different. The greater the size of the salt is, the password security is higher. However, under dictionary attack, salted passwords are still weak. By comparing with salted password, the ENP proposed guarantees the diversity of passwords without the need for extra elements (e.g., salt).

3) Key Stretching:. Key stretching were introduced to resist the dictionary attack. So that it can change weak password into enhanced password .It could increase the time cost required to every password attempt, so that the power of defending against dictionary attack is increased. In the ENP proposed, like key stretching, multi-iteration encryption is used to further improve password security under dictionary attack. Compared with key stretching, the ENP does not introduce extra elements (e.g., salt).

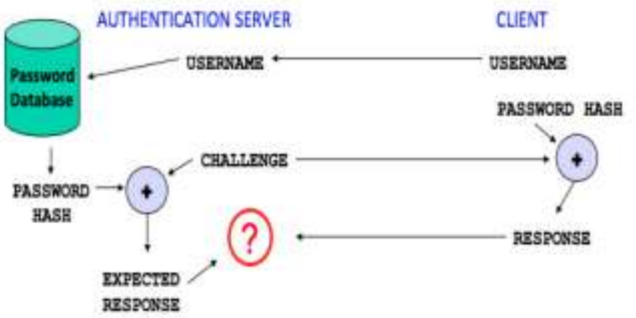

Fig -1: Password hashing

2.1 Negative Database

The compression of the complement of a positive database (denoted as DB) is stored in the NDB. Every entry in NDB contains three symbols: '0', '1', and '*'. The symbol '0' matches the bit 0, and the symbol '1' match the bit 1; the symbol '*' matches either the bit 0 or 1. Entry in an NDB consists of two kinds of positions: specified positions and unspecified positions. Positions where the symbols are '0' or '1' called specified positions, while positions where the symbols '*' called unspecified positions. Then, both '0' and '1' are specified symbols, and the '*' is the unspecified symbol. A sequence of bits is covered by an entry in an NDB; that is to say, the bits of the sequence are matches the symbols of the entry at the specified positions. If a sequence of bits is covered by one entry in an NDB, says that the sequence is covered by the NDB. If an NDB covers every entry in the (U-DB), says that the NDB is complete; otherwise, it is incomplete. The NDB converted from a DB with one entry is called the single NDB; otherwise, it is called the multiple NDB.

There are two types of NDB generation algorithms, one for the single NDBs and one for multiple NDBs. In the first type, Clause distribution control algorithm, 1-hidden algorithm, 2-hidden algorithm, q-hidden algorithm, hybrid algorithm, p-hidden algorithm, and K-hidden algorithm has been proposed successively. In the second type, the prefix algorithm, Randomize NDB (abbreviated as RNDB), multiple-solution algorithm proposed successively. These algorithms could also be used to generate single NDBs.

3. PROPOSED SYSTEM

The proposed framework two consist of the registration phase and authentication phase. The main framework to protect passwords in the authentication data table, the system designer should first select a cryptographic hash function and a symmetric-key algorithm, where the condition satisfied is that the size of the hash value of selected cryptographic hash function is equal to key size of the selected symmetric-key algorithm. In addition, cryptographic hash functions and symmetric-key algorithms could also be used in the ENP, which adequately indicates the flexibility of our framework.

3.1 Registration Phase

The registration phase consist of six steps.

(1) On the client section, a user enters his/her username and password. Then, the entered username and plain password are transmitted to the server through a secure channel;

(2) If the received username exists in the authentication data table, “The username already exists!” is displayed, which means that the server has rejected the registration, and the registration phase is terminated; otherwise, go to Step (3);

(3) Then the received password is hashed using the selected cryptographic hash function.

(4) The hashed password is then converted into a negative password using an NDB generation.

(5) The negative password is then encrypted to an ENP using the selected symmetric-key algorithm, where the key will be the hash value of the plain password. The additional option, multi-iteration encryption could be used for enhancing passwords.

6) The username and the resulting ENP get stored in the authentication data table and “Registration success” is displayed, means that the server has been accepted the registration request.

3.2 Authentication Phase

The authentication phase consist of five steps.

(1) On the client section, a user enters his/her username and password. Then, the username and plain password get transmitted to the server through a secure channel.

(2) If the received username does not exist in the authentication data table, then “Incorrect username or password!” is displayed, which means that the server get rejected the authentication request, and the authentication phase is terminated; otherwise, go to Step (3);

(3) Search for the authentication data table for the ENP corresponding to the received username;

(4) The ENP is then decrypted (one or more times according to the encryption setting in the registration phase) using the selected symmetric-key algorithm, where that the selected key is the hash value of the plain password; so, the negative password is obtained;

(5) If the hash value that has been received password is not the solution of the negative password, then “Incorrect username or password!” is displayed, which means that the server rejected the authentication request, then the authentication phase is terminated; otherwise, “Authentication success” is returned, that means that the server has accepted the authentication request.

ENPs could be obtained by the following steps. The received plain password (i.e., a sequence of characters) from a client is then hashed using a cryptographic hash function. Next, that received hashed password is converted into a negative password using an NDB generation algorithm, the negative password is then encrypted using a symmetric-key algorithm. So, the ENP is obtained. The solution of the negative password will be the hash value of the received plain password. In this processing, each component (i.e., the cryptographic hash function, the symmetric-key algorithm, and the NDB generation algorithm) is indispensable. The cryptographic hash function will converts plain passwords to hashed passwords; the fixed length property of that received hashed passwords offers convenience for the other subsequent encryption, since the length requirement for the secret key in the symmetric key algorithm; and other properties (such as avalanche effect and collision resistance) are also important factors of employing the cryptographic hash function. The reason behind this is the conversion from a hashed password to a negative password is not irreversible; therefore, if no encryption, when an adversary obtains a negative password, then immediately obtains the corresponding hashed password, which makes the strength of the ENP equivalent to that of the hashed password.

4. CONCLUSION

This paper proposed a password protection scheme called ENP, and presenting the password authentication framework based on the ENP. In this framework, the entries on the authentication data table are ENPs. In the end, analyzed and compared the attack complexity of hashed password, salted password, key stretching and the ENP. The results will show that the ENP could resist lookup table attack and provide stronger password protection under dictionary attack. It is clear that the ENP does not need extra elements (e.g., salt) while resisting lookup table attack. In the future, other NDB generation algorithms will be studied and introduced to the ENP to further improving password security. Furthermore, other techniques, also can be used such as multi factor authentication and challenge–response authentication, will be introduced into our password authentication framework. For securing the encrypted negative password is then encrypted using the RSA algorithm.

REFERENCES

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, “Passwords and the evolution of imperfect authentication,” Communications of the ACM, vol. 58, no. 7, pp. 78–87, Jun. 2015.

[2] M. A. S. Gokhale and V. S. Waghmare, “The shoulder surfing resistant graphical password authentication technique,” Procedia Computer Science, vol. 79, pp. 490–498, 2016.

[3] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic passwordmodels,” in Proceedings of 2014 IEEE Symposium on Security and Privacy, May 2014, pp. 689–704.

[4] A. Adams and M. A. Sasse, “Users are not the enemy,” Communications of the ACM, vol. 42, no. 12, pp. 40–46, Dec. 1999.

[5] E. H. Spafford, “Opus: Preventing weak password choices,” Computers & Security, vol. 11, no. 3, pp. 273–278, 1992.

[6] Y. Li, H. Wang, and K. Sun, “Personal information in passwords and its security implications,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2320–2333, Oct. 2017.

[7] D. Florencio and C. Herley, “A large-scale study of web password habits,” in Proceedings of the 16th International Conference on World Wide Web. ACM, 2007, pp. 657–666.

[8] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, “Designing password policies for strength and usability,” ACM Transactions on Information and System Security, vol. 18, no. 4, pp. 13:1–13:34, May 2016.

[9] D. Wang, D. He, H. Cheng, and P. Wang, “fuzzyPSM: A new password strength meter using fuzzy probabilistic context-free grammars,” in Proceedings of 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Jun. 2016, pp. 595–606.

[10] H. M. Sun, Y. H. Chen, and Y. H. Lin, “oPass: A user authentication protocol resistant to password stealing and password reuse attacks,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 651–663, Apr. 2012.

[11] M. Zviran and W. J. Haga, “Password security: An empirical study,” Journal of Management Information Systems, vol. 15, no. 4, pp. 161– 185, 1999.

[12] P. Andriotis, T. Tryfonas, and G. Oikonomou, “Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method,” in Proceedings of Human Aspects of Information Security, Privacy, and Trust. Springer International Publishing, 2014, pp. 115– 126.

[13] D. P. Jablon, “Strong password-only authenticated key exchange,” SIGCOMM Computer Communication Review, vol. 26, no. 5, pp. 5–26, Oct. 1996.

[14] J. Jose, T. T. Tomy, V. Karunakaran, A. K. V, A. Varkey, and N. C. A., “Securing passwords from dictionary attack with character-tree,” in Proceedings of 2016 International Conference on Wireless Communications, Signal Processing and Networking, Mar. 2016, pp. 2301–2307.

[15] A. Arora, A. Nandkumar, and R. Telang, “Does information security attack frequency increase with vulnerability disclosure? an empirical analysis,” Information Systems Frontiers, vol. 8, no. 5, pp. 350–362, Dec. 2006.