International Journal of Scientific Research and Engineering Development-– Volume3 Issue 1, Jan-Feb 2020

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Password Authentication Framework Based on Encrypted Negative Password

Jitty Merin Mathew[1], Bibin Varghese[2], Smita C Thomas[3]

[1](P G Scholar, Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta)

[2] (Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta)

[3](Research Scholar, Computer Science and Engineering, Vels University,Chennai)

----------**********----------

Abstract:

The negative database (NDB) is a new technique for privacy preserving and information hiding. It hides information by storing the complementary set instead of the original data. In order to protect the hidden information, NDBs should be hard-to-reverse. This paper propose a K hidden algorithm for generating hard-to-reverse NDBs (called K hidden-NDBs)

Keywords — **Authentication, lexicon attack, lookup table attack, negative database, secure password storage.**

----------**********----------

I. INTRODUCTION

The rule quality analyses and comparisons show that the ENP may resist operation table attack and supply stronger positive identification protection underneath lexicon attack.ENP doesn't introduce additional parts (e.g., salt); besides this, the ENP could still resist precomputation attacks. 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 for additional information except the plain password. The existing system actually uses the simplest mechanism of all the other techniques. The plain positive identification is simply encrypted and keep within the info..This mechanism is highly insecure and you can also find that it is easy to attack and get the password. The other main mechanism which is used till date is the hashing mechanism where in the plain password is hashed using hashing algorithms such as the Secure Hash Algorithm or the Message Digest Algorithm. Comparing to the previous mechanism it provides more security and also it doesn't provide the actual password but the hashed value of the password. But the plain password can be from the hashed value from the rainbow table attack and lookup table attack. Thus to reduce the vulnerability and risk, Encrypted Negative Password System is used. Owing to the event of the net, a massive range of on-line services have emerged, in which password authentication is the most widely used authentication technique, for it is available at a low cost and easy to deploy.

II. LITERATURE REVIEW

Theory on passwords has lagged behind practice, where large providers use back-end smarts to survive with imperfect technology. Passwords will continue as a useful signal for the foreseeable future, where the goal is not impregnable security but reducing harm at acceptable cost. Passwords offer plenty of examples of divergence between theory and practice; estimates of strength, models of user behaviour and password-composition policies that work well in theory generally remain unsupported by evidence of reduced harm in practice and have in some cases been directly

 ©IJSRED: All Rights are Reserved

contradicted by empirical observation. Yet large Web services appear to cope with insecure passwords, largely because shortcomings can be covered up with technological smarts in the back end. This is a crucial, if unheralded, evolution, driven largely by industry, which is experienced in data-driven engineering [2].

Nowadays computer as well as information security is the most significant challenge.Authorized users should access the system or information. Authorization can't occur without authentication. For this authentication various techniques are available. Among them the most popular and easy is the password technique. Password ensures that computer or information can be accessed by those who have been granted right to view or access them. Traditional password technique is a textual password which is also called alphanumeric password. But these textual passwords are easy to crack through various types of attack. So to overcome these vulnerabilities, a graphical password technique is introduced. As name suggests in this technique images (pictures) are used as a password instead of text. Also psychological study says that human can easily remember images than text. So according to this fact, graphical passwords are easy to remember and difficult to guess. But because of graphic nature, nearly all the graphical password techniques are vulnerable to shoulder surfing attack. It can be useful for smart held devices like smart phones, PDA, iPod, iPhone etc. Today, authentication is achieved through the use of password technique [3].

Proposed approach make three contributions in this paper. The first is to introduce probability-threshold graphs for evaluating password datasets. The second is to introduce knowledge and techniques from the rich literature of statistical language modelling into password modelling. It also identifies new issues (such as normalization) that arise from modelling passwords, and a broad design space for password models, including both whole-string models and template-based models. Third, there have conducted a systematic study of many password models, and obtained a number of findings. In particular, show that the PCFGW model, which has been assumed to be the state of the art and has been widely used in password research, underperforms whole-string Markov models in the experiments [4].

III. EXISTING SYSTEM

In existing system there is no easy-to reverse complete single NDB generation algorithms (i.e., the prefix algorithm with permutation, and the variant of the prefix algorithm, to generate negative passwords. Typical Password Protection Schemes: 1) Hashed Password: The simplest scheme to store passwords is to directly store plain passwords. However, this scheme presents a problem that once adversaries obtain the authentication data table, all passwords are immediately compromised. To safely store passwords, a common scheme is to hash passwords using a cryptographic hash function, because it is infeasible to directly recover plain passwords from hashed passwords. The cryptographic hash function quickly maps data of arbitrary size to a fixed-size sequence of bits. In the authentication system using the hashed password scheme, only hashed passwords are stored. However, hashed passwords cannot resist lookup table attack. Furthermore, rainbow table attack is more practical for its space-time tradeoffs. Processor resources and storage resources are becoming richer, which makes the precomputed tables used in the above two attacks sufficiently large, so that adversaries could obtain a higher success rate of cracking hashed passwords.

2) Salted Password: To resist precomputation attacks, the most common scheme is salted password. In this scheme, the concatenation of a plain password and a random data (called salt) is hashed through a cryptographic hash function. The salt is usually generated at random, which

 ©IJSRED: All Rights are Reserved

ensures that the hash values of the same plain passwords are almost always different. The greater the size of the salt is, the higher the password security is. However, under dictionary attack, salted passwords are still weak. Note that compared with salted password, the ENP proposed in this paper guarantees the diversity of passwords without the need for extra elements (e.g., salt).

3) Key Stretching: To resist dictionary attack, key stretching, which converts weak passwords to enhanced passwords, was proposed. Key stretching 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 in this paper, like key stretching, multi-iteration encryption is used to further improve password security under dictionary attack, and compared with key stretching, the ENP does not introduce extra elements (e.g., salt). ENP can resist pre computation attack.

4) Negative Database: In the NDB, the compression of the complement of a positive database (denoted as DB) is stored. As described, U = n denotes the universal set of n-bit sequences; x ∈ U denotes an n-bit sequence; DB denotes a positive database that contains m entries; then NDB stores the compression. Every entry in an NDB contains three symbols: '0', '1', and '*'. The symbol '0' only match the bit 0, and the symbol '1' only match the bit 1; The symbol '*' can match either the bit 0 or 1. Every entry in an NDB consists of two kinds of positions: specified positions and unspecified positions. Positions where the symbols are '0' or '1' are called specified positions, while positions where the symbols are '*' are called unspecified positions. Accordingly, both '0' and '1' are specified symbols, and the '*' is the unspecified symbol. A sequence of bits is covered by one entry in an NDB; that is to say, the bits of the sequence are matched by the symbols of the entry at the specified positions. If a sequence of bits is covered by one entry in an NDB, we say that the sequence is covered by the NDB. If an NDB covers every entry in the (U-DB), we say that the NDB is complete; otherwise, it is incomplete. The NDB converted from a DB with only one entry is called the single NDB; otherwise, it is called the multiple NDB.

IV.PROPOSED SYSTEM

This paper proposes a K-hidden algorithm to generate hard-to-reverse K-NDBs (called K-hidden-NDBs for simplicity). The K-hidden algorithm is more fine-grained than the q-hidden algorithm because K−1 parameters are used to control the distributions of different types of entries in NDBs. Although some NDB generation algorithms have been proposed, most of them cannot control or adjust the distributions of different types of entries in NDBs arbitrarily according to variable security and utility requirements in realworld applications. The p-hidden algorithm can control the distributions in a 3-NDB arbitrarily, but its maximal security strength is limited when the string length is fixed. Typically, we define the degree of freedom (DoF) of an algorithm as its independent parameters number for controlling the distributions of different types of entries in NDBs. The DoF of an NDB generation algorithm is related to the ability of generating NDBs with different levels of security and utility.

It consist of the following phases.

A. Registration Phase

The registration phase is divided into six steps.

(1) On the client side, a user enters his/her username and password. Then, the 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 returned, which means that the server has rejected the registration request, and the registration phase is terminated; otherwise, go to Step (3).

 ©IJSRED: All Rights are Reserved

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

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

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

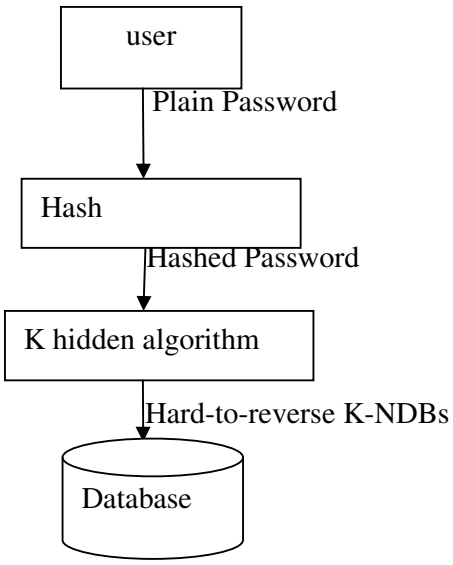

Fig.1 The data flow diagram of the generation procedure of the hard to reverse NDB

(6) The username and the resulting ENP are stored in the authentication data table and "Registration success" is returned, which means that the server has accepted the registration request.

B. Authentication Phase

The authentication phase is divided into five steps.

(1) On the client side, a user enters his/her username and password. Then, the username and plain password are 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 returned, which means that the server has rejected the authentication request, and the authentication phase is terminated; otherwise, go to Step (3).

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

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

(5) If the hash value of the received password is not the solution of the negative password (verified by Algorithm 1 or Algorithm 2), then "Incorrect username or password!" is returned, which means that the server has rejected the authentication request, and the authentication phase is terminated; otherwise, "Authentication success" is returned, which means that the server has accepted the authentication request.

The proposed work describes as follows:
The user given plain password is given to a hash function to get hashed password. Then perform K hidden algorithm to get a hard to reverse K- NDBs. It is then stored to a database.

V. CONCLUSION

This paper proposes the K-hidden algorithm to generate hard-to-reverse K-NDBs (called K-hidden-NDBs for simplicity). The K-hidden algorithm is more fine-grained than the q-hidden algorithm because K−1 parameters are used to control the distributions of different types of entries in NDBs.

REFERENCES

[1] Wenjian Luo, Yamin Hu, Hao Jiang, and Junteng Wang, "Authentication by Encrypted Negative Password", *IEEE Transactions on Information Forensics and Security,* vol - 14, no.1, pp: 114 - 128, Jan 2019.

 ©IJSRED: All Rights are Reserved

[2] 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.

[3] 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.

[4] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models", *in Proceedings of 2014 IEEE Symposium on Security and Privacy*, pp: 689 - 704. May 2014.

[5] 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.

[6] 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.
.

 ©IJSRED: All Rights are Reserved