Protected Identifications in Converted arrangement by using different Algorithm Technique.

[1] Pooja Chharge [2,] Sayali Ghumatkar [3]Diksha Khatal [4] Priyanka Rajguru [4] Prof . P.P Gawali

Abstract:

Protected password storage is a very important aspect in systems based on password authentication. There are some protection flaws in authentication method. Due to this password can be still easily cracked. Therefore, we are going to build up a system that will provide a way of storing password by using cryptographic functions. System will adopt a framework to avert password in data table. In this paper, we recommend a password authentication framework that is designed for protected password storage and could be easily integrated into existing authentication systems. The system consists of two phases: the registration phase and authentication phase. In registration phase, user enters username and password. The received password will be transformed to hash value by using Elliptic curve cryptography (ECC) algorithm. This hash value will be then transformed into negative password using negative password generation algorithm. The negative password will be then converted into Encrypted Negative Password by using selected symmetric-key algorithm i.e. Attribute Based Encryption (ABE). While encryption key will be hash value of simple password. This ENP will be then hold to authentication data table. Most prominently, the ENP is the first password security scheme that combines the cryptographic hash function, the negative password and the symmetric-key algorithm, without the necessity for extra information except the simple password.

IndexTerms **- Component,formatting,style,styling,insert.**

I. INTRODUCTION:

These days computer as well as information protection is the most important challenge. Allowed users should access the system or information. Permission can't occur without authentication. For this authentication various techniques are existing. Among them the most admired and simple is the password method. Password ensures that computer or information can be accessed by those who have been arranged right to vision or access them. The progress of the Internet, a vast number of online services have emerged, in which password authentication is the most widely used authentication method, for it is accessible at a low cost and easy to deploy. Hence, password security always attracts large interest from business. even with great study achievements on password security, passwords are still cracked since users' not careful behaviors. For example, many users often select pathetic passwords they tend to use again same passwords in different systems they usually set their passwords using familiar words for its expediency to keep in mind. In addition, system problems may cause password compromises. It is very tricky to obtain passwords from high security systems. After obtaining authentication data tables from weak systems, adversaries can carry out offline attacks. Passwords in the authentication data table are usually in the form of hashed passwords. However, because processor resources and storage resources are becoming more and more abundant, hashed passwords cannot resist pre computation attacks, such as rainbow table attack and lookup table attack. In this paper, a password security scheme called Encrypted Negative Password (ENP) is proposed, which is based on the Negative Database (NDB) cryptographic hash function and symmetric encryption, and a password authentication framework based on the ENP is presented. The NDB is a new

security technique that is inspired by biological resistant systems and has a wide range of applications. Symmetric encryption is usually deemed unsuitable for password protection. Because the secret key is usually common by all encrypted passwords and stored together with the authentication data table, once the authentication data table is stolen, the shared Key may be stolen at the similar time. Thus, these passwords are right away compromised. However, in the ENP, the Secret key is the hash value of the password of each user, so it is almost dissimilar and does not need to be specially generated and stored. as a result, the ENP enables symmetric encryption to be used for password protection. As an implementation of key stretching multi-iteration symmetric encryption is introduced to further improve the strength of ENPs. Compared with the salted password scheme and key Stretching, the ENP guarantees the variety of passwords by itself without introducing additional elements.

To summarize, the major contributions of this paper are as follows:

(1) We recommend a password security scheme called ENP, and we suggest two implementations of the ENP: ENPI and ENPII, including their generation algorithms and verification algorithms. Furthermore, a password authentication framework based on the ENP is presented.

(2) We examine and measure up to the attack difficulty of hashed password, salted password, key stretching and the ENP. The results show that the ENP could resist lookup table attack without the need for extra elements and provide stronger password security under vocabulary attack.

Keywords
Authentication, vocabulary attack, lookup table attack, negative database, protected password storage.

Related Work:

A. Typical Password Security Schemes

1) Hashed Password:

The simplest scheme to hold passwords is to directly store simple passwords. However, this scheme presents a problem that once adversaries obtain the authentication data table, all passwords are right away compromised. To securely store passwords, a common scheme is to hash passwords using a cryptographic hash function because it is infeasible to straight recover simple passwords from hashed passwords. The cryptographic hash function rapidly maps data of random size to a fixed-size series of bits. In the authentication system using the hashed password scheme, only hashed passwords are stored. However, hashed passwords cannot oppose lookup table attack. Furthermore, rainbow table attack is more practical for its space-time exchange.

Processor resources and storage resources are becoming richer, which makes the pre computed tables used in the above two attacks adequately huge, so that adversaries could obtain a higher success rate of cracking hashed passwords.

2) Salted Password:

To oppose pre computation attacks, the most common scheme is salted password. In this scheme, the concatenation of a simple password and a random data (called salt) is hashed through a cryptographic hash function. The salt is usually generated at random, which ensures that the hash values of the same simple passwords are almost always different. The greater the size of the salt is, the higher the password security is. However, under vocabulary 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 improved passwords, was proposed. Key stretching could increase the time cost necessary to every password attempt, so that the control of defending against vocabulary 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).

B. Negative Database:

Some concepts of NDB are given below. Every entrance in an '*'. 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 unnamed 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 sheltered 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 entrance is called the solo NDB; otherwise, it is called the several NDB.

Motivation:

We are developing this system for providing more safety to responsive data. We investigate and evaluate the attack difficulty of our scheme with that of typical password storage schemes (i.e., Hashed password, Salted password) below lookup table attack and vocabulary attack. To offer a technique for avoidance of data tables to keep user's responsive information safe from various attacks. We are going to use grouping of various techniques to convert plain text information into encrypted information.
We are developing this system for given that more security to sensitive data. We examine and compare the attack complexity of our scheme with that of typical password storage schemes (i.e., Hashed password, Salted password) under lookup table attack and vocabulary attack. To give a method for prevention of data tables to keep user's responsive information secure from various attacks. We are going to use grouping of various techniques to convert plain text information into encrypted information.

ECC Algorithm Flow :-

Input : Plain Password.

Output : True or False

Step 1 : I/p as plain password.

Step 2 : Convert input plain password into Byte format

```
byte[] I/p1 = I/p.getBytes();
```

Step 3 : Convert input I/p1 into Hex to String

```
pass2=Hex.toHexString(msgHash);
```

Step 4 : Convert input pass2 into 16 byte long

Pseudo Algorithm Flow :-

Input : a hashed password hashP;
a negative password np
Output : true or false
1: m LENGTH(hashP)
2: for i 1 to m with stepsize of 1 do
3: if NUMBEROFSP(npi) 6= i then
4: return false
5: end if
6: end for
7: for i 1 to m with stepsize of 1 do
8: if NUMBEROFSP(npi) 6= 1 then
9: return false
10: end if
11: k INDEXOFSP(npi)
12: x[k] :TOBIT(npi[k])
13: for j i + 1 to m with stepsize of 1 do
14: if npj [k] 6= TOSYMBOL(x[k]) then
15: return false
16: end if
17: npj [k] '*'
18: end for
19: end for
20: if x = hashP then
21: return true
22: else
23: return false
24: end if

ABE Algorithm Flow :

Input : Negative Password.

Output : True or False

Step 1 : I/p of Pseudo password.

Step 2 : Convert input of NP password into cipher

```
keyCph = Bswabe.enc(I/p, policy);
```

Step 3 : Convert input I/p1 into Hex to String

```
cphBuf = SerializeUtils.bswabeCphSerialize(keyCph);
```

Step 3 : Passed cphBuf to AES Coder

aesBuf = AESCoder.*encrypt*(m.toBytes(), cphBuf);

Step 4 : Apply condition to encryption of password then encrypt password.

Comparison:

In existing system first, the received plain password from a client is hashed through a cryptographic hash function (e.g., SHA-256). Then, the hashed password is converted into a negative password. Finally, the negative password is encrypted into an Encrypted Negative Password (abbreviated as ENP) using a symmetric-key algorithm (e.g., AES). In out project instead of SHA-256 Algorithm we used ECC algorithm and instead of AES we used ABE Algorithm for encryption.

Advantage:

- The advantage of this system is to make more secure password by using multilayer encryption.

Literature Survey

Paper Title	Year	Objective	Limitation
Intelligent Phishing Website Detection using Random Forest Classifier	2017	In this paper, an intelligent system to detect phishing attacks is presented. We used different data mining techniques to decide categories of websites: legitimate or phishing. Different classifiers were used in order to construct accurate intelligent system for phishing website detection.	The limitation of this approach is that the extracted features are only for detection of phishing WebPages. They achieved 97.5% classification accuracy
Visual Similarity-based Phishing Detection Scheme using Image and CSS with Target Website Finder	2017	The detection of phishing websites and identifying their target are imperative. In this paper, we propose visual similarity-based phishing detection scheme using image and CSS with target website finder. To remedy first shortcoming, we focus on the fact that legitimate websites are often linked by other websites and regard such website as legitimate and store the screenshot and CSS in the database.	As future works, we will blush up the proposed scheme. In the current state, although the websites linked by at least one website are registered in white list, we consider it is necessary to set optimal threshold or some conditions. And then, we will try reduce False Positive by selecting proper CSS.
The Shoulder Surfing Resistant Graphical Password Authentication Technique	2016	A proposed system provides a strong security against brute force and guessing attacks as it has a good combination of two types of graphical passwords. It is difficult to guess the password system by a person or by a computer by trying millions of possibilities.	Some limitations of graphical password techniques and the major limitation observed is that, it is vulnerable to shoulder surfing attack as images are used as a password. Shoulder surfing means watching over the person's shoulder to get the password.
Detection of Phishing Websites Using C4.5 Data Mining Algorithm	2017	The main aim of phishing is that to steal sensitive information of user such as password, username, pin number, etc. Victims of	One of the data mining algorithm C4.5 has been analyzed and the accuracy obtained for the classifier generated by this algorithm is

		phishing attacks may uncover their money related delicate data to the attackers who may utilize this data for budgetary and criminal exercises.	82.6%. As data mining algorithm can determine phishing websites in real time so this accuracy rate of C4.5 is quite significant.

Mathematical Model

Mathematical model set theory S= {s, e, X, Y, Φ}
s= Start of the program
1. Register/Login into the system
2. Provide Plain Password.
e= End of the program
Identify the Encrypted Password
X= input of the program= {P, R, Q}
P = Plain Password
R= Hash Value
Q = Using ABE Encrypt the hash value
Y = Output of program = Fully Encrypted Password.
First, users provide the specific Plain Password.
Let A be the set of categories
Overall the plain password is converted into hash value then encrypted it into By ABE
Y= E1+E2+... +Em / m
Where M is number of overall Encrypted Password.

System Architecture:

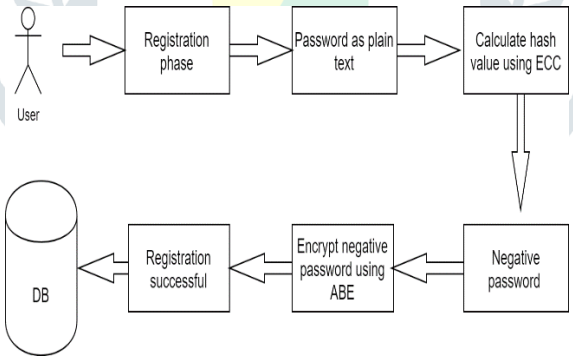
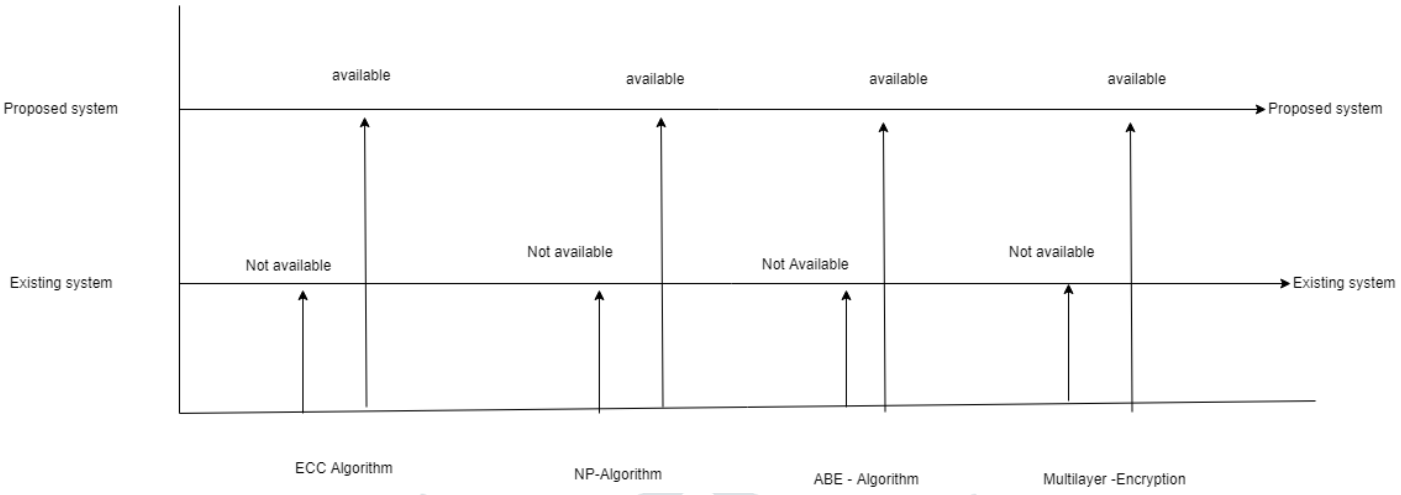


Figure shows the flow of our System. The system consists of two phases: The registration phase and authentication phase. In registration phase, user enters username and password in plain text. The received password will be converted to hash value by using ECC algorithm. This hash value will be then converted into negative password using negative password generation algorithm. The negative password will be then converted into Encrypted Negative Password (ENP) by using selected symmetric-key algorithm i.e. Attribute Based Encryption (ABE). While encryption key will be hash value of plain password. This ENP will be stored to authentication data table. If the user password is match with Authentication System then Registration will be Successful.

Existing – Proposed System Graph :



Conclusion:

In this paper, we propose a password authentication framework that is planned for protected password storage and could easily integrated into presented authentication systems. We are going to propose password protection scheme called Encrypt Negative Password (i.e., ENP) and records in authentication data table are Encrypted Negative Passwords. In the end, we analyzed and compared the attack complexity of hashed password, salted password, key stretching and the ENP. The results illustrate that the ENP could resist lookup table attack and provide stronger password security under vocabulary attack. It is worth mentioning 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 recover password security.

Future Work :

- In future you can add modified updated different encryption algorithm for more secure field of password.

Result :

- More secure encrypted password .
- Detecting phishing site bye NB algorithm.

ACKNOWLEDGMENT:

It gives us great pleasure in presenting the preliminary project report on '**Protected Identifications in Converted arrangement by using different Algorithm Technique.**

'.

I would like to take this opportunity to thank my internal guide for giving me all the help and guidance I needed I am really grateful to them for their kind support. Their valuable suggestions were very helpful. I am also grateful to HOD, for his indispensable support and suggestions.

Name of Students

[1] Sayali Ghunatkar [2,] Diksha Khatal [3]Priyanka Rajguru, [4] Pooja Chharge

REFERENCES:

[1] R. Liu, W. Luo, and L. Yue, "The p-hidden algorithm: hiding single databases more deeply," Immune Computation, vol. 2, no. 1, pp. 43–55, Mar. 2014.

[2] D. Zhao, W. Luo, R. Liu, and L. Yue, "A fine-grained algorithm for generating hard-to reverse negative databases," in Proceedings of 2015 International Workshop on Artificial Immune Systems, Jul. 2015, pp. 1–8.

[3] D. Zhao and W. Luo, "One-time password authentication scheme based on the negative database," Engineering Applications of Artificial Intelligence, vol. 62, pp. 396–404, 2017

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

[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] 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.

[7] 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.

[8] 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.

[9] D. Wang, D. He, H. Cheng, and P. Wang, "fuzzy PSM: 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] S. Boonkrong and C. Somboonpattanakit, "Dynamic salt generation and placement for secure password storing," IAENG International Journal of Computer Science, vol. 43, no. 1, pp. 27–36, 2016.