

Secured Password Storage and Authentication by Encrypted Negative Password

Dr. V. GOUTHAM¹ B. RUTHVIK REDDY² L. VAMSHIDHAR GOUD² N. SHIVIVISTA²

¹Professor, Department of CSE, ²Final year B.Tech. Students, Department of CSE,
Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India

Abstract

Secure password storage is a vital aspect in systems based on password authentication, which is still the most widely used authentication technique, despite its some security flaws. In this paper, we propose a password authentication framework that is designed for secure password storage and could be easily integrated into existing authentication systems. In our framework, 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), and multi-iteration encryption could be employed to further improve security. The cryptographic hash function and symmetric encryption make it difficult to crack passwords from ENPs. Moreover, there are lots of corresponding ENPs for a given plain password, which makes precomputation attacks (e.g., lookup table attack and rainbow table attack) infeasible. The algorithm complexity analyses and comparisons show that the ENP could resist lookup table attack and provide stronger password protection under dictionary attack. It is worth mentioning that the ENP does not introduce extra elements (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.

1.Introduction

1.1 Motivation:

Owing to the development of the Internet, a vast number of online 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. Hence, password security always attracts great interest from academia and industry. Despite great research achievements on password security, passwords are still cracked since user's careless behaviours. For instance, many users often select weak passwords they tend to reuse same passwords in different systems, they usually set their passwords using familiar vocabulary for its convenience to remember. In addition, system problems may cause password compromises. It is very difficult to obtain passwords from high

security systems. On the one hand, stealing authentication data tables (containing usernames and passwords) in high security systems is difficult. On the other hand, when carrying out an online guessing attack, there is usually a limit to the number of login attempts. However, passwords may be leaked from weak systems. Vulnerabilities are constantly being discovered, and not all systems could be timely patched to resist attacks, which gives adversaries an opportunity to illegally access weak systems. In fact, some old systems are more vulnerable due to their lack of maintenance. Finally, since passwords are often reused, adversaries may log into high security systems through cracked passwords from systems of low security. 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 precipitation attacks, such as rainbow table attack and lookup table attack. Note that there is a trend of generalization of adversaries, because anyone could obtain access to information on vulnerabilities from vulnerability databases, such as the Open Source Vulnerability Database (OSVDB), National Vulnerability Database (NVD), and the Common Vulnerabilities and Exposures (CVE), and then make use of these information to crack systems. Moreover, they could download and use attack tools without the need for very professional security knowledge. Some powerful attack tools, such as hashcat, Rainbow Crack and John theRipper, provide a variety of functions, such as multiple hash algorithms, multiple attack models, multiple operating systems, and multiple platforms, which raises a higher demand for secure password storage. Hence, there is a need for a password protection scheme which would resist to all these attacks.

1.2 Problem Definition

Despite of great achievements on password security, passwords are still cracked due to weak systems. In these situations, adversaries may gain opportunity to illegally access weak systems which usually carried out as follows. First, adversaries precompute a lookup table, where the keys are the hash values of elements in a password list containing frequently-used passwords, and the records are the corresponding plain passwords in the password list. Next, they obtain an authentication data table from low security systems. Then, they search for the plain passwords in the lookup table by matching hashed passwords in the authentication data table and the keys in

the lookup table. Finally, the adversaries log into higher security systems through cracked usernames and passwords, so that they could steal more sensitive information of users and obtain some other benefits. A considerable number of attacks are carried out in this way, so that adversaries could obtain passwords at a low cost, which is advantageous to their goals. So, password authentication framework is essential for secure password storage.

1.3 Objective

This study lays out creation of password protection framework that is designed for secure password storage and it should be easily integrated into existing authentication system. Here we make use of Cryptographic hash function and Symmetric encryption to improve security. This password authentication provides high security and it makes difficult for the adversaries to crack the passwords from ENP's.

- To implement a computationally light weight efficient password protection scheme called Encrypted Negative Password (abbreviated as ENP)
- To implement the solution in such a way that it should be easier to integrate this with existing systems
- To prove that the proposed scheme provides a strong security against various kinds of attacks.
- To provide an efficient user interface access to the clients to access the portal
- To deploy the project over the cloud so that it can be accessed from various geographical location from any device.

2. Existing System

The following are the typical Password Protection Schemes

- 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 tradeoff. 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.

- **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 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).
- **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).

2.1. Disadvantages Of Proposed System

- System is not secured due to lack of improved dynamic Key-Hashed Message Authentication Code function (abbreviated as d-HMAC).
- Password protection scheme called Encrypted Negative Password is absent.

3. Proposed System

In the proposed system, a password protection scheme called Encrypted Negative Password (abbreviated as ENP) is proposed, which is based on the Negative Database (abbreviated as 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 immune systems and has a wide range of applications. Symmetric encryption is usually deemed inappropriate for password protection. Because the secret key is usually shared

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 same time. Thus, these passwords are immediately compromised. However, in the ENP, the secret key is the hash value of the password of each user, so it is almost always different and does not need to be specially generated and stored. Consequently, the ENP enables symmetric encryption to be used for password protection. As an implementation of key stretching, multiiteration encryption is introduced to further improve the strength of ENPs. Compared with the salted password scheme and key stretching, the ENP guarantees the diversity of passwords by itself without introducing extra elements (e.g., salt). To summarize, the main contributions of this paper are as follows: The system also proposes a password protection scheme called ENP, and we propose 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. The system analyses and compares the attack complexity 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 .

3.1. Advantages of Proposed System

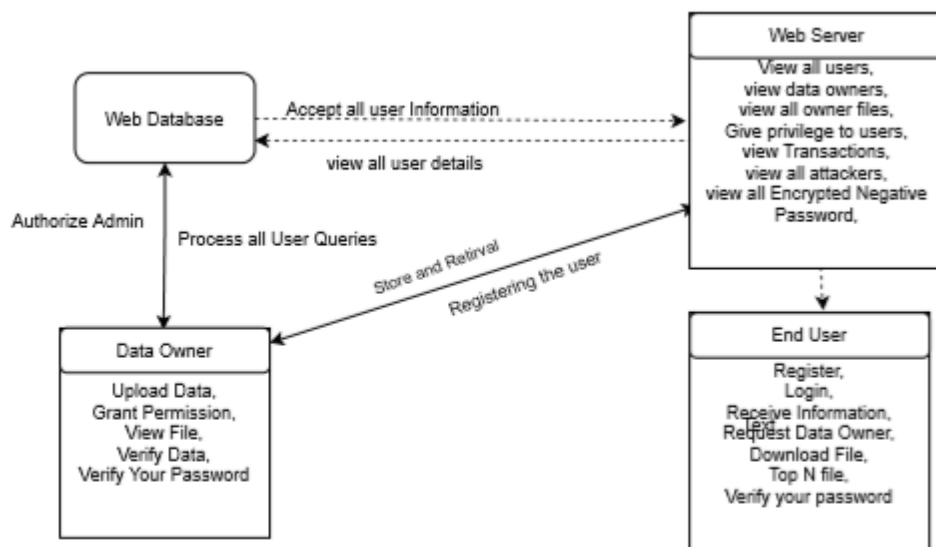
- The system is more effective due to improved dynamic Key-Hashed Message Authentication Code function (abbreviated as d-HMAC) was proposed for password storage.
- The system more powerful password scheme by dynamic salt generation and placement are used to improve password security.

It is the purpose of the new system to address all the problems plaguing the present system. The proposed system will also have some other features such as:

1. This password protection scheme will resist to various attacks.
2. The use of cryptographic hash function and symmetric key encryption makes difficult to crack passwords from ENP's.
3. Files can be downloaded securely.
4. Personal files of users are stored securely.

4. Architecture

This password authentication framework which is designed for secure password storage could be easily integrated into existing authentication system. The cryptographic hash function and symmetric-key algorithm make difficult to crack the passwords. Hence security is the most important future of the application.



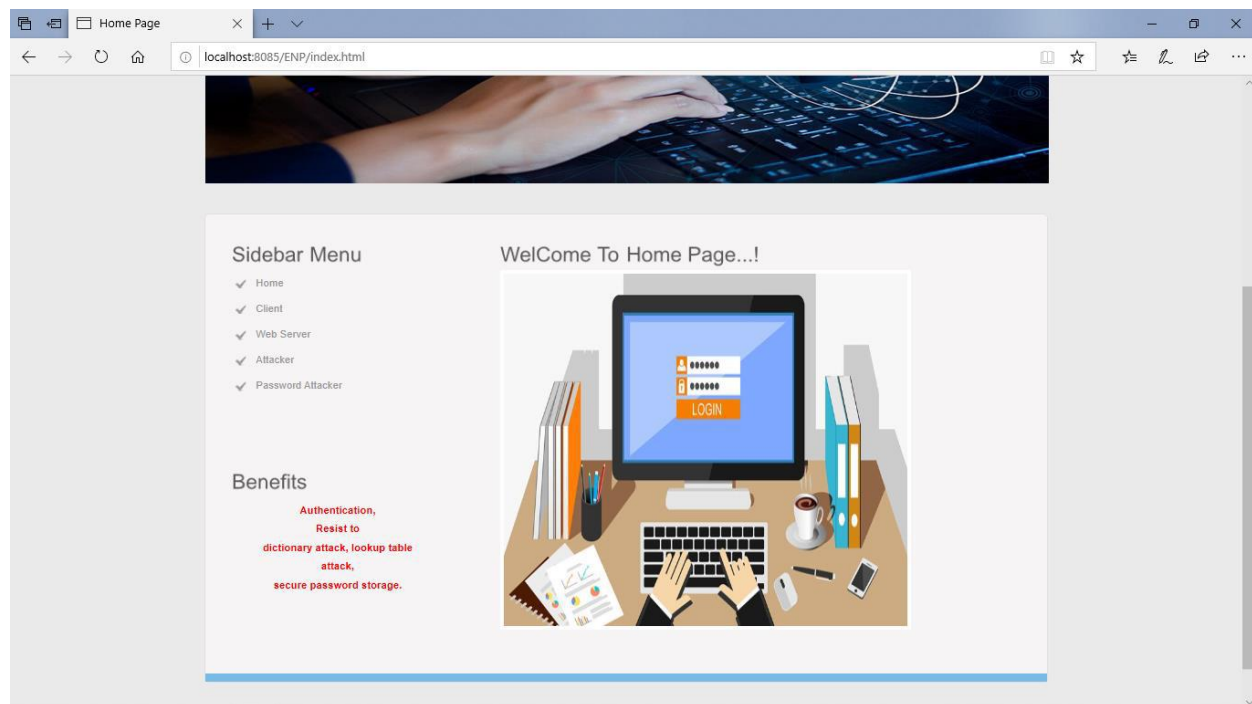
5. Implementation And Results

5.1 Introduction

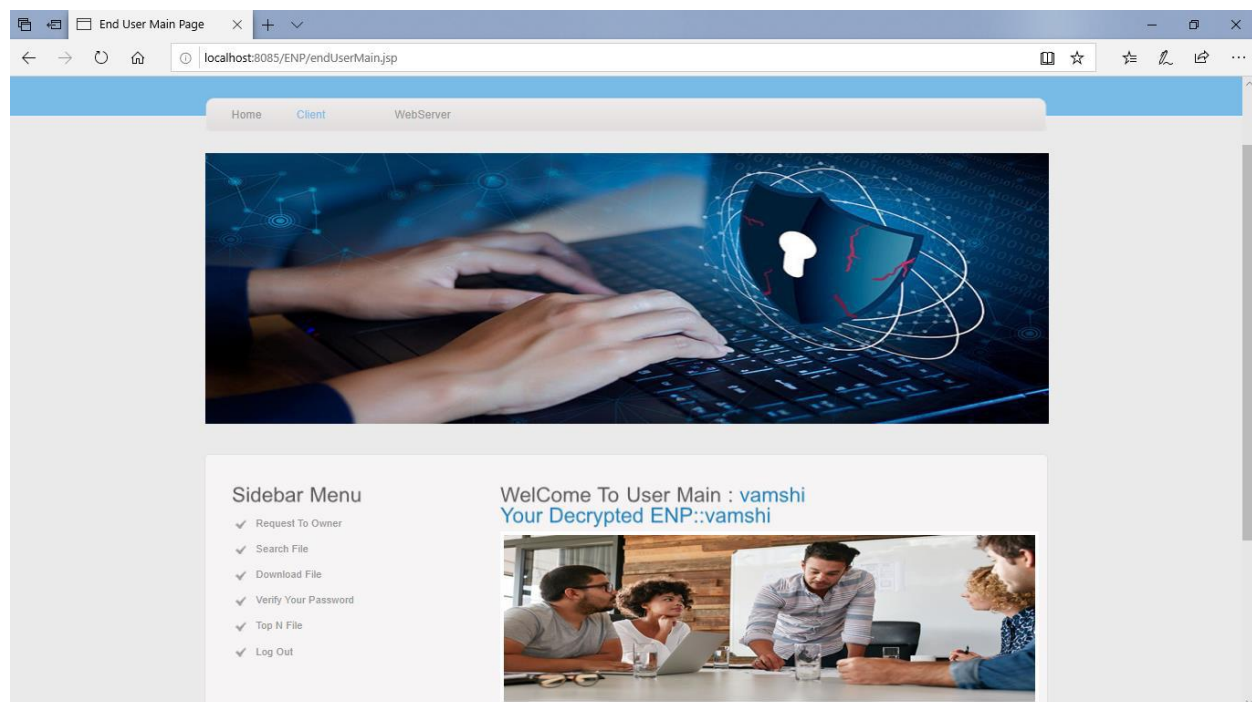
Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.2. Output Screens

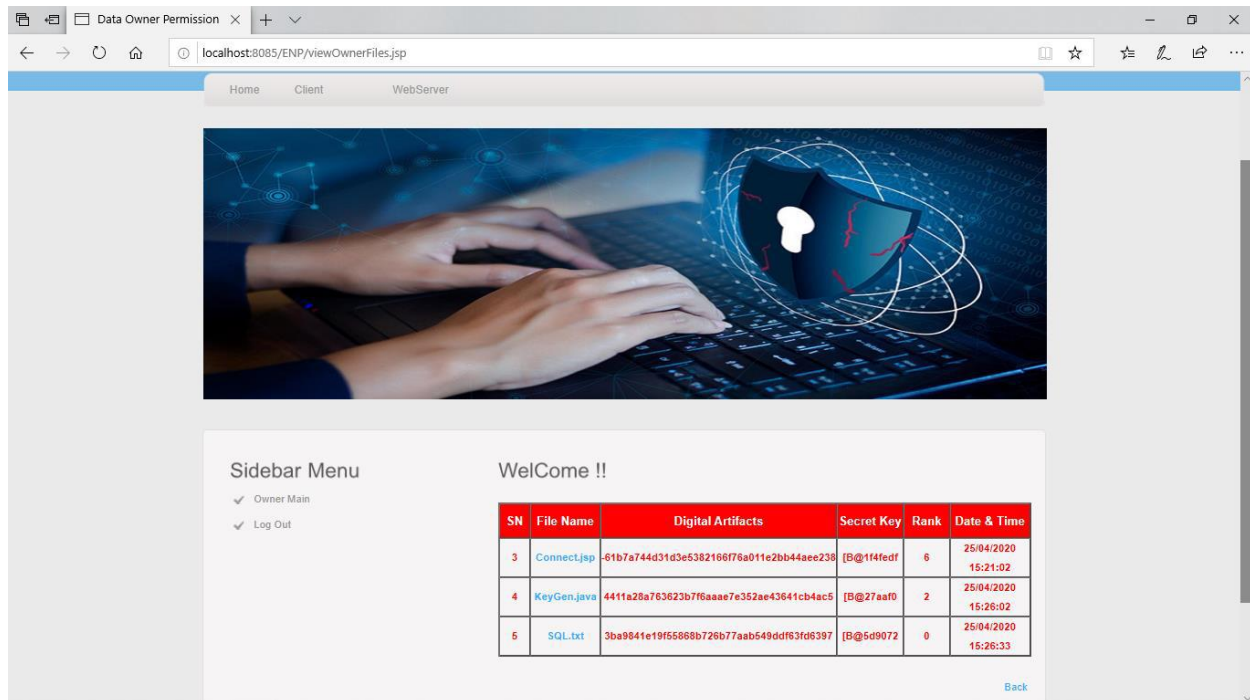
Homepage



Owner page

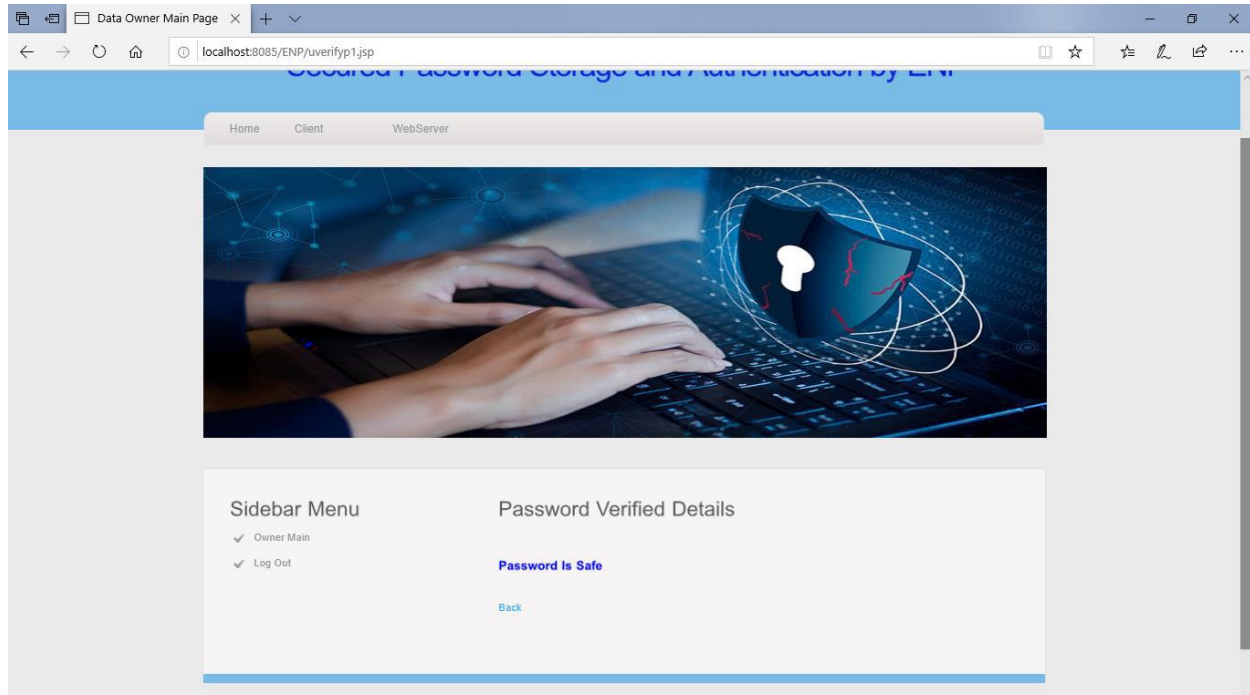


View files page



SN	File Name	Digital Artifacts	Secret Key	Rank	Date & Time
3	Connect.jsp	-61b7a744d31d3e5382166f76a91fe2bb44ae238	[B@1f4fedf	6	25/04/2020 15:21:02
4	KeyGen.java	4411a28a763623b7f6aaae7e352ae43641cb4ac5	[B@27aa10	2	25/04/2020 15:26:02
5	SQL.txt	3ba9841e19f55888b726b77aab549ddf63fd6397	[B@5d9072	0	25/04/2020 15:26:33

Verify password page



Password Verified Details

Password Is Safe

[Back](#)

5.3.Result Analysis

At the end of this project work, we were able to design and develop a password protection scheme. In this password protection scheme we have combined cryptographic hash function, negative password and symmetric key algorithm. This work also will serve as a stepping-stone for people who wish to research more on this topic.

5.4.Design of Test Cases and Scenarios:

USER LOGIN:

TEST CASE	INPUT	EXPECTED BEHAVIOUR	OBSERVED BEHAVIOUR	STATUS SUCCESS
1.	Enter the incorrect login id ,password	Displays message “ invalid user name and password”	Message “ invalid user name and password” is displayed.	Success
2.	Enter blank values	Displays an alert message “enter valid value for corresponding field”	An alert message “enter valid value for corresponding field” is displayed.	Success
3.	Enter the correct login id and password	Displays authorized action to user	User page is displayed.	Success
4.	Trigger back button	Page is redirected to the home page	Home page is displayed	Success

Table. Testcase for User Login

TEST CASE	INPUT	EXPECTED BEHAVIOUR	OBSERVED BEHAVIOUR	STATUS
1.	Enter the incorrect field values	Displays an alert message “ invalid user name and password”	An alert message “ invalid user name and password” is displayed	Success
2.	Enter blank values	Displays an alert message “enter valid value for corresponding field”	An alert message “enter valid value for corresponding field” is displayed	Success
3.	Trigger submit button	All values are stored in database and Page is redirected to the login page	Values stored successfully and redirected to login page.	Success

Table. Test case for registration

TEST CASE	INPUT	EXPECTED BEHAVIOUR	OBSERVED BEHAVIOUR	STATUS
1.	Click on logout.	Session has to be destroyed.	Session destroyed.	Success

Table. Testcase for public logout

TEST CASE	INPUT	EXPECTED BEHAVIOUR	OBSERVED BEHAVIOUR	STATUS
1.	Enter the incorrect login id ,password and select police station name	Displays message “invalid user name and password”	message “invalid user name and password is displayed	Success
2.	Enter blank values	Displays an alert message “enter valid value for corresponding field”	an alert message “enter valid value for corresponding field” is displayed	Success
3.	Enter the correct login id and password	Displays authorized action to Administrator	Admin page is viewed	Success
4.	Trigger back button	Page is redirected to the home page	Redirected to homepage	Success

Table Test case for Server login

6.Conclusion

In this paper, we proposed a password protection scheme called ENP, and presented a password authentication framework based on the ENP. In our framework, the entries in the authentication data table are ENPs. In the end, we analyzed and compared the attack complexity of hashed password, salted password, key stretching and the ENP. The results show that the ENP could resist lookup table attack and provide stronger password protection under dictionary 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 improve password security. Furthermore, other techniques, such as multi-factor authentication and challenge-response authentication, will be introduced into our password authentication framework.

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 password models,” 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*