AUTHENTICATION BY ENCRYPTED NEGATIVE PASSWORD

Mrs. R. Bavithra, Assistant Professor, Dhanalakshmi Srinivasan College of Engineering and Technology
Mr. P. S. Jayakumar, Assistant Professor, Dhanalakshmi Srinivasan College of Engineering and Technology
Ajay Kumar:, Student, Dhanalakshmi Srinivasan College of Engineering and Technology

1. ABSTRACT Its about securing the passwords and making the system more secured from intruders. 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. Most e-commerce sites, consumers have the responsibility of creating their own passwords and often do so without guidance from the web site or system administrator. One fact is well known about password creation—consumers do not create long or complicated passwords because they cannot remember them. Most of the e-commerce sites uses the method where in the passwords are just encrypted and are not secured properly. In our project, we propose a password authentication framework that is designed for secure password. In our framework, first the received plain password from a client is hashed through a cryptographic hash function (Script and PBKDF2.)Then, the hashed password is converted into a negative password. Finally, the negative password is encrypted into an Encrypted Negative Password(ENP)(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 pre computation attacks (e.g., Brute Force Attack , Rainbow table attack and Dictionary attack) infeasible ENP) using a symmetric-key algorithm (e.g., AES), and multi-iteration encryption could be employed to further improve security.

1. INTRODUCTION

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 users' careless behaviors . 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. 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.

We propose a Encrypted negative password method based on Negative database (NDB generation), which further improve the security of password. The users can use this application without any fear of security flaws. The existing system actually uses the simplest mechanism of all the other techniques. The plain password is just encrypted and stored in the database. 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.

The existing system actually uses the simplest mechanism of all the other techniques. The plain password is just encrypted and stored in the database. 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

The Hashing mechanism provides more security and also it doesn't provide actual password but the hashed value. Encryption mechanism is highly insecure. The plain password can be from the hashed value from the Rainbow Table Attack.

To protect passwords in an authentication data table, the system designer must first select a cryptographic hash function and a symmetric-key algorithm, where the condition that must be satisfied is that the size of the hash value of the selected cryptographic hash function is equal to the key size of the selected symmetric-key algorithm. For convenience, some matches of cryptographic hash functions and symmetric-key algorithms. In addition, cryptographic hash functions and symmetric-key algorithms that are not listed here could also be used in the ENP, which adequately indicates the flexibility of our framework. The proposed framework is based

on the ENP; hence, for better understanding, the data flow diagram of the generation procedure of the ENP.

2. MATERILAS AND METHODS

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate.

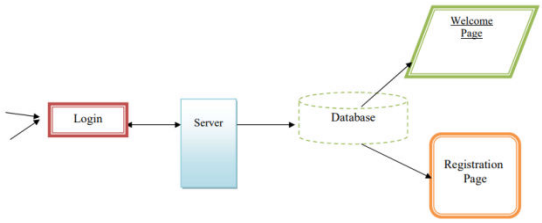

Fig 2. 1. User Interface Design

Once you login there is book list page. What the book you like and comfortable with price also means click buy now. After you have to once again see the title of what the you have to be purchased and click make payment option it will go for bank login page.

Once you click make payment option there is bank login page. You have to enter your account number, username and password then only they can able to connect the server. There is transfer account option just enter your account number and transfer account number and enter the amount click submit button the enter amount will be updated in to owner account.

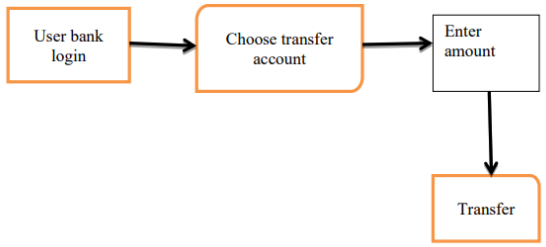

Fig 2.2 Payment Process

There is a separate login for owner once login the owner and click the book request page. There is the detail of whom to buy and which book to be buy and also having the user detail. Once he paid the amount means click choose file option and upload the book click submit button in that book will send only for particular user.

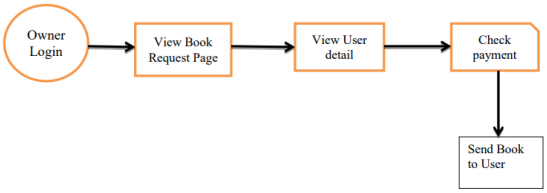

Fig 2.3. Owner Info

Once owner accept the request and send file means user to be login them account. To connect with server user must give their username and password then only they can able to connect the server. See the page and select the download option your will get the file for what you have to be ordered.

In the payment page user have the responsibility to enter user name and password. While entering the password the password is secured with different new algorithms. First, that password is hashed using Script and PBKDF2 algorithm, then the hashed value is converted as negative using NDB generation algorithm. After, that the negative value is encrypted using AES algorithm, then it will stored into the database.

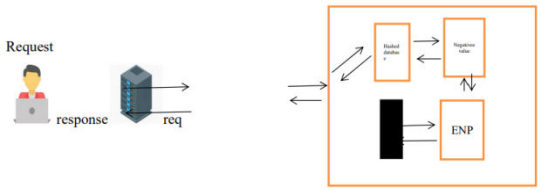

Fig 2.4. Encrypt Negative Passwor

A. THREAD

A thread is a sequential path of code execution within a program. And each thread has its own local variables, program counter and lifetime. Like creation of a single thread, we can also create more than one thread (multithreads) in a program using class Thread or implementing interface Runnable to make our project efficient and dynamic. In our project we are using request process with the helpof multi threading concepts.

B. SWING

Swing, which is an extension library to the AWT, includes new and improved components that enhance the look and functionality of GUIs. Swing can be used to build Standalone swing GUI apps as well as Servlets and Applets. It employs a model/view design architecture. Swing is more portable.

3 .RESULT AND DISCUSION

We develop achievability protocols and outer bounds for the secure network coding setting, where the edges are subject to packet erasures, and public feedback of the channel state is available to both Eve and the legitimate network nodes. Secure network coding assumes that the underlying network channels are error-free; thus, if our channels introduce errors, we need to first apply a channel code to correct them, and then build security on top of the resulting error-free network. We show that by leveraging erasures and feedback, we can achieve secrecy rates that are in some cases multiple times higher than the alternative of separate channel-error-correction followed by secure network coding; moreover, we develop outer bounds and prove optimality of our proposed schemes in some special cases.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the

application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Functional tests provide systematic emonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

A. Valid Input

Iidentified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised. Output : identified classes of application outputs must be exercised. Systems/Procedures : interfacing systems or procedures must be invoked. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

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.

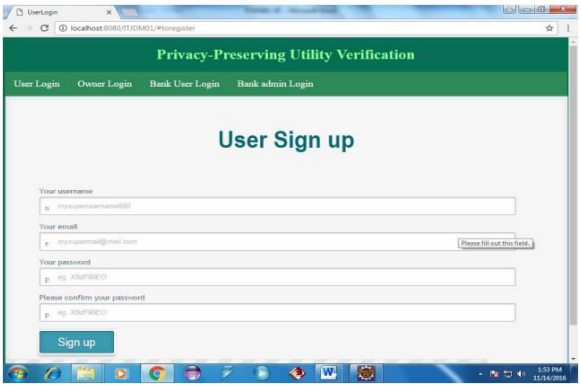

Fig 3.1. User Signup Page

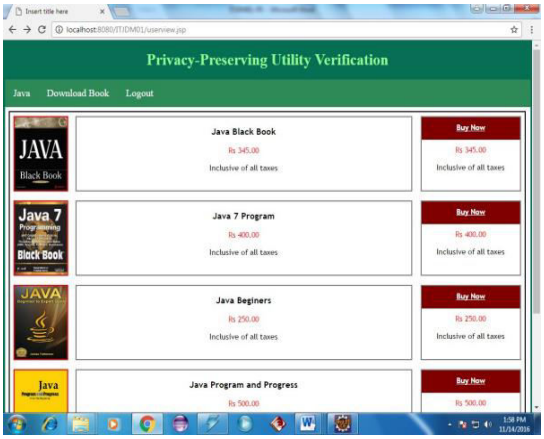

Fig 3.2. Purchaging Page

Fig 3.3. Owner Book Publishing Page

4. CONCLUSIONAND FUTURE REFERENCE

We develop achievability protocols and outer bounds for the secure network coding setting, where the edges are subject to packet erasures, and public feedback of the channel state is available to both Eve and the legitimate network nodes. Secure network coding assumes that the underlying network channels are error-free; thus, if our channels introduce errors, we need to first apply a channel code to correct them, and then build security on top of the resulting error-free network

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.

6. REFERENCES

1. 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.

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. 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.

4. J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwordsand the evolution of imperfect authentication," Communications of the ACM,vol. 58, no. 7, pp. 78–87, Jun. 2015.

5. 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.

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

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