Enhanced Security to Login ID's for Authentication Based Applications

Swapnil Shinde[1], Charudatt Rajguru[2], Supriya Shelke[3], Sagar Bhosale[4] & Rupali Tayade[5]
[1,2,3,4,5] Dept. of Information Technology, Pune University
Marathwada Mitra Mandal's College of Engineering
Pune, Maharashtra, India

Abstract— In daily life, maintaining and securing user accounts for different stream user account for different stream of work is a very big task. Authentication of a system is merged into two things, first is identifying the user account and then verifying as well as securing it from outside attacks. In Biometrics, for giving proper authorization there is a need of hardware which is more expensive and the installation is very tedious. But in providing enhanced security to login ID's, there is no need of additional hardware and the installation is very easy.

We propose a method that can elevate current login system by strengthening the authentication process. The login system can be implemented and hosted on cloud based applications of any kind. It provides a password for a username i.e. a docket instead of using username directly to identify user uniquely. Also then it provides a set of multiple choices of randomly generated hazy ID's. Unless the user select correct ID he/she won't be directed to the password section, further attempts can be provide to the user to select the correct ID. The hacker cannot reach to the password section as he/she needs to provide a correct docket for the username. Since the stolen password cannot be used directly by the attackers the user can have extra time to change their password before attackers attain an access. Further the password for the username will be encrypted and split into two parts and stored on different servers. When the existing user enters the password the encrypted password in the split form will get merged, decrypted and will be compared with the entered password. If the password entered matches with the decrypted one then the authentication is provided otherwise it fails.

Enhanced security scheme separates identification and verification server which makes large system scalable. We implement a feasible system and assess it with test users. The survey indicates that the system is not as invasive as other schemes, users feel better secured and they are willing to use the scheme on cloud based applications.

Keywords - Authentication, Identification, Password, Security, Verification.

I. INTRODUCTION

Now-a-days, the problem our country facing is the problem of "Demonetization". Where everyone is planning to go cashless each and every transaction or the bill payment is done by means of the applications. The applications allow the user to store amount of cash called as mobile e-wallet from where they can do the transactions. There is a provided limit of storing the cash of up to 20,000 and more. So the problem arises as these or any other kind of user related accounts are really safe? The applications are so secured that it cannot be hacked by any hacker? The money stored in the e-wallet is safe? There are cryptographic algorithms used in the login section where the password is secured in the encrypted form but the hacker can easily hack the account knowing the username of the user's account. Further if the hacker gets a password in the encrypted form then he can decrypt it. Probably the decryptions of the passwords hackers usually do are by analyzing the hashes, and see which hash type they are, and then brute force them. Brute forcing is when you systematically cycle through each letter in a letterset until it matches a password. Also there are software's used for hacking the passwords like Cain & Abel. Cain is a part of software suite, and is a AIO (All-In-One) Windows hacking tool. So basically hacking a user account is possible. We used to prevent it by making our password stronger so that there are less chances of getting hacked. But as the technology is arising and booming to a next level, hacking cannot be prevented by just making a strong password or making new password once in a quarter year. Also while accessing the app from mobile phones once the user logs in, he/she exits without logging out the app to which the hackers takes benefit by hacking the phone itself sending malwares in the android system.

The problem to be focused is not only in the password section but the whole login section. When the hacker gets the username his work is half way done. We always tend to ignore username and the security regarding it as the username is one of the important aspects for hackers to hack a particular account [8]. As the person is known by the name, his/her account is known by the username. If we focus on securing both username and the password (instead of just securing the password section) the system can be more secured and will prevent the hacker to hack it [1]. Let's consider an example of Gmail. To send and receive mails there's a need of an e- mail id. This e-mail id is known to everyone, whereas for the login system this e-mail id is used as the username to get logged-in. Such usernames are more prone to hackers as there's no security provided in the identification section i.e. the username. Even for the password section, the password is not so feasible to be much secured as if the hacker gets success hacking the database, then they can attain access to all encrypted passwords. The current login systems are not well secured as the cryptographic techniques used are not appropriate and not only the password but also the username needs to be well secured.

Cloud application security requires a comprehensive strategy that balances needs about that application and security risks. Security concerns associated with cloud based applications fall into two broad categories: security issues faced by cloud providers and the security issues faced by the

customers [5]. The customer depends on the cloud providers as they are providing service to the customers by means of the applications. The Customers can be organizations who host their applications or store their valuable data on the cloud. The provider must ensure that they are providing a secured infrastructure and their clients are protected by providing a strong login system. There are faults in the login system as the validation given are not that strong thus make the passwords weaker. Even if there exists a validation system for a password section it is not as strong as the developer has myths that too much validation will make the passwords more complex for the users to remember and changing passwords within a fixed duration will be a tedious job.

The goal of this research is improving the security of the authentication systems on cloud by providing a secure identification as well as verification process. To avoid false login attempts, our method does not use a publicly known login ID for identification. Instead it uses private information i.e. a docket which is just known to the computer and the user. Also to avoid the hacker to hack the database and retrieve the passwords, the password is split and stored in an encrypted form in the databases of two different servers [6]. This process makes the stolen password files of no use for the attackers.

II. LITERATURE SURVEY

The paper presented by Juyeon Jo, Yoohwan Kim and Sungchul Lee in year 2014 describes a scheme, called Mindmetrics, to implement a login based authentication system for identifying a username without their login ID's [1]. It uses a password for knowing a username and providing multiple fake id's to select one among them. The limitations of this system are it does not provide strong validation to password for knowing a username and the password i.e. the verification section is prone to outside attacks.

Also paper presented by Zengqiang Wu, Di Su and Gang Ding in year 2014 describes protecting the passwords in login systems [2]. In this system, one efficient algorithm have been proposed, namely, ElGamal algorithm for encryption of Data. The algorithm converts plain text into 2 divided cipher texts. Splitting and merging of the text can also be done with the help of this algorithm. The paper presented by Joseph Bonneau, Cormac Herley, Paul C. van Oorschot and Frank Stajano in year 2012 describes the passwords i.e. the traditional one and the CAP ones with its advantages, features and its disadvantages [3]. Here they proposed a technique of Biometrics as a security mechanism which can be used in the system. The limitation is Biometrics require specialized hardware which is costly and not suitable for authentication systems.

In 2015, Aqeel Sahi Khader and David Lai presented the paper Preventing Man-In-The-Middle Attack in Diffie-Hellman Key Exchange Protocol [4]. In this paper, we use the concept of Diffie-Hellman for the connection between two servers. It present as how we can prevent the Man-In-The-Middle Attack which is a prone in Diffie-Hellman Key Exchange Protocol.

In 2015, Priyanka Ora and Dr. P.R.Pal presented the paper Data Security and Integrity in Cloud Computing Based on RSA Partial Homomorphic and MD5 Cryptography [5]. In this paper, we propose two algorithms; how they differ by the number of bits they encrypt text into hash key. A cryptographic hash function has an important role to achieve security goals as authenticity, digital signatures, time stamping etc. MD5 gives a fixed-length output of 128 bits whereas RSA uses fixed length output of 32 bit. It gives an idea of which algorithm to be used for storing the docket in the hash key format onto the database. The advantages, disadvantages, features, applications are explained well in this paper.

In 2013, Xun Yi, San Ling, and Huaxiong Wang presented the paper Efficient Two-Server Password-Only Authenticated Key Exchange [6]. In this paper how a password can be split into two parts and stored onto different databases of two different servers is mentioned in the paper. For this various preliminary algorithms can be used such as Diffie Hellman Key Exchange Algorithm, Elgamal Algorithm for Data Encryption, and a strong Hash function.

III. PROPOSED METHODOLOGY

Our Enhanced Security to Login ID's for Cloud Based Applications plays an important role in maintaining and securing the login system from outside attacks. For securing the login system there is a need for using strong cryptographic algorithms**.** For different phases i.e. identification and verification phase, different algorithms have been used.

The algorithms used for the particular phase in the specific module are described in the table as follows:

TABLE I. ALGORITHMS USED IN THE ENHANCED LOGIN SYSTEM

Sr. No.	Algorithms	Keypoints
1	MD-5	Used in the identification section to convert and store docket in hash key Format.
2	ElGamal	Used in the verification section to encrypt and divide the password for storing it in different data bases of two different Servers.
3	Diffie - Hellman Key Exchange	Used for establishing secure connection between two servers to access the distributed encrypted passwords from different databases.

As we discussed in Table I, all the algorithms mentioned, for the system implementation are a must. Let's analyze each algorithm in detail.

A. MD5 (Message Digest Hash Algorithm)

The MD5 algorithm is a widely used hash function used to produce 128-bit hash value. It was initially designed by Ronald Rivest in 1991 to replace an earlier hash function MD4 [5].

Suppose we will take b-bit message as input, and that we need to find its message digest.

Step 1 – Append padded bits:

The message is padded so that its length is Congruent to 448 modulo 512. A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits equals 448 modulo 512.

Step 2 – Append Length:

A 64 bit representation of b is appended to the result of the previous step. The resulting message has a length that is an exact multiple of 512 bits.

Step 3 – Initialize MD Buffer:

A four-word buffer (A, B, C, D) is used to compute the message digest. Here each of A, B, C, D is a 32- bit register. These registers are initialized to the following values in hexadecimal:

Word A: 01 23 45 67
Word B: 89 ab cd ef
Word C: fe dc ba 98
Word D: 76 54 32 10

Step 4 – Process message in 16-words Block:

Four auxiliary functions that take as input three 32-bit words and produce as output as one 32-bit word.

F(X,Y,Z) = XY v not(X) Z
G(X,Y,Z) = XZ v Y not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X v not(Z))

Step 5 – Output:

The message digest produced as output is A, B, C, and D. That is, output begins with the low-order byte of A, and end with the high-order byte of D.

B. ElGamal Cryptographic Algorithm:

In Cryptography, the ElGamal Encryption System is an Asymmetric Key Encryption algorithm for public key cryptography which is based on Diffie-Hellman key exchange. It was described by Taher Elgamal in 1985. The ElGamal system is a public key cryptosystem based on the discrete logarithmic problem. It consists of both encryption and signature algorithm.

Its safety depends on the calculation of the discrete logarithm finite field. The security of ElGamal algorithm is enhanced by elliptic curve cryptography encryption system.

The main characteristic of the ElGamal algorithm is in the process of encryption, cipher text is two times longer than plaintext. When encrypted, it will generate a random K in cipher text, so even if the same plain text is encrypted twice, the result of cipher text is different [2].

ElGamal algorithm encryption steps are :

Step 1 – Get a big prime number randomly (Let discrete logarithm problem be difficult on Z*p)

Step 2 – Calculate in finite domain and get a generating element denoted g, $g \in Z^*_p$

Step 3 –Take one value X, $X \in Z^*_{p-1}$(1< x< p-2) as user A's secret key.

Step 4–Calculate user A's public key denoted $y = g^x \bmod p$.

Step 5–Let y, p and g be public key of A, then take x as private key of A. (plain text space M= Z^*p and Cipher text space $C = Z^*p \times Z^*p$).

Step 6–Encryption transformation: plaintext m, $m \in Z^*p$, a random integer k, $k \in Z^*p-1$, then
1< k < p-2, finally we could get Cipher text (c1= $g^k \bmod p$) through $c2 = my^k \bmod p$.

Step 7–Decryption transformation: text c = (c1, c2), we could get plaintext m by m= $(c2, (c1^x)^{-1}) \bmod p$.

In ElGamal algorithm, encryption is random and cipher text not only depends on plaintext m ,but also depends on random K, therefore, the same plaintext can be encrypted to generate many (p-1 at most) kinds of cipher text.

C. Diffie-Hellman key Exchange Algorithm:

In 1976, Whitfield Diffie and Martin Hellman published a cryptographic protocol called the Diffie Hellman key exchange based on concepts developed by Hellman PHD student Ralph Merkle. The Diffie Hellman key exchange is a secure method for exchanging cryptographic key. It is a key exchanging algorithm. Diffie Hellman is used to secure a variety of internet services. Diffie Hellman is an algorithm used to establish a shared secret key between two parties. It is primarily used as a method of exchanging cryptographic keys for use in symmetric encryption algorithm.

The Diffie Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communication using a symmetric key cipher. This protocol is widely used for secure key exchange [4].

Diffie-Hellman key Exchange Algorithm is explained as follows:

The protocol uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to p-1.

Sender and Receiver agree to use a modulus p=23 and base g=25(which is a primitive root modulo 23).

Sender chooses a secret integer a=6, then sends Receiver A=ga mod p; A=56 mod 23=8

Receiver chooses a secret integer b=15, then sends Sender B=gb mod p; B=515 mod 23=19

Sender computes s=Ba mod p; s=196 mod 23=2

Receiver computes s=Ab mod p; s=815 mod 23=2

Sender and Receiver now can share a secret key (i.e. the number 2). This algorithm is used to establish a secure connection between two different servers with the means of sharing a secret key. The servers include the half part of the encrypted password, which can be merged and decrypted if the connection is successful. If the key matches, then connection is successful, otherwise is fails.

IV. PROPOSED ARCHITECTURE

In our proposed system architecture the Key terms are Identification Server, Verification Server, Cloud Services, Web Server and Database servers. Login Systems are used in Cloud Applications to provide various kinds of services to securely prevent outside attacks. The User is the important case as he/she will be the only person to deal with the login system. If the user is new then he/she needs to get registered by filling the basic information along with the docket he/she wants to give for their username. The existing user can directly login by entering his/her docket for the username. The work of registering, entering docket for the username and the docket get stored in hash format in the database is all done in Identification phase.

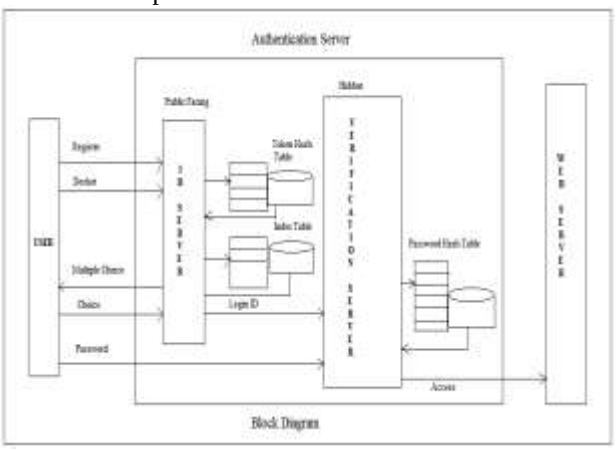

Figure 1: Block Diagram

After entering the correct docket for the username, it will show choices for choosing the correct username among the fake ones. The users need to select the correct username among multiple usernames. Further attempts can be given to select the correct username. After giving the correct username, the users can login by entering his/her password for the username.

The password entered by the user will be stored in the database at the time of registration for the new user. The password entered will be divided in two parts, and then it will be encrypted and stored in the databases of two different servers. That means half password will be stored in a database of a server and the remaining half will be stored in another database of the particular different server. When the user enters the password at the time of login, the connection between the servers is done. After the successful connection, the split passwords in the different databases is merged, decrypted and then compared with the password entered by the user at the time of login. If the entered password matches with the merged, decrypted password then authentication is successful otherwise it fails.

The system can be implemented on Cloud as Cloud is one of the booming and supreme service providers.

V. RESULTS

The implementation of the system is shown accordingly:

The users first register by filling the details and giving appropriate docket for the username. Several validations are applied in the registration phase.

Figure 2: Identification Phase

After getting registered, the next thing to do is to login by entering the docket for the username. If the docket entered is correct then only further choosing of username can be done otherwise the docket entered is invalid.

Figure 3: Login Phase

After entering the correct docket, the user can then select the correct username from the multiple fake usernames. The fake usernames are randomly generated and stored in the database server.

Figure 4: Choosing Correct Username

After selecting the correct username, then the user can move for the password phase.

Figure 5: Verification Phase

The registered data is stored in the database with the appropriate entries along with the unique id of each username.

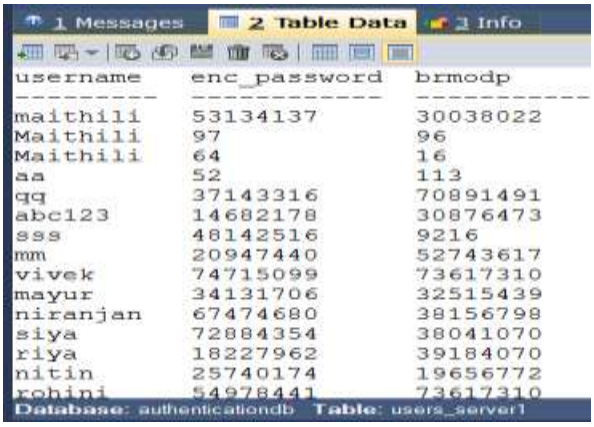

Figure 6: Password Encryption

The password is split and stored in the database of server1 and server2 in the encrypted manner.

VI. CONCLUSION

Authentication of user is done in two steps, identification and verification. The traditional login systems were vulnerable to sophisticated attacks as the only module to be secured was the password phase. Though the verification phase is secured, there's no guarantee that the identification phase is also secured. We proposed a new system called Enhanced Security to Login ID's for Cloud Based Application to strengthen the identification process with personal secret information. In this system, a Login ID is not asked. Instead a user must provide the correct docket to pass the identification phase. In case the password file is stolen, login attempts by hacker are blocked by identification server. By implementing this system, not only the identification phase is secured but also the verification phase. The password is split into two parts depending on the length of the password and stored onto the different databases of two different servers. So even if one database is hacked by the attacker he will get only the half password which is not appropriate to hack the system. The system can simulate biometrics with its high security level where two factor authentication with biometrics is not feasible. It is more practical than other methods and can be easily used on cloud based applications. Thus the system is secured, and can be easily implemented on cloud based applications.

VII. ACKNOWLEDGEMENT

Sincere thanks to our guide Prof. Swapnil S. Shinde and HOD Prof. Rupali Chopade and other faculty members of Information Technology for giving the valuable suggestions and for guiding us.

VIII. REFERENCES

[1] Juyeon Jo, Yoohwan Kim, and Sungchul Lee, "Mindmetrics: Identifying users without their login IDs", *IEEE International Conference on Systems, Man, and Cybernetics,* October 05-08-2014,
San Diego, CA, USA.

[2] Zengqiang Wu, Di Su, Gang Ding,"ElGamal Algorithm for Encryption of Data Transmission", *IEEE International Conference on Mechatronics and Control (ICMC),* July 3-5-2014, Jinzhou, China.

[3] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, Frank Stajano,"The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes", *IEEE Symposium on Security and Privacy,* 2012.

[4] Aqeel Sahi Khader, David Lai,"Preventing Man-In-The-Middle Attack in Diffie-Hellman Key Exchange Protocol", *IEEE* 22nd *International Conference on Telecommunications,* 2015.

[5] Priyanka Ora and Dr. P.R. Pal, "Data Security and Integrity in Cloud Computing Based on RSA Partial Homomorphic and MD5 Cryptography", *IEEE International Conference on Computer, Communication and Control,* (IC4-2015).

[6] Xun Yi, San Ling, and Huaxiong Wang, "Efficient Two-Server Password-Only Authenticated Key Exchange", *IEEE Transactions on Parallel and Distributed Systems, Vol.24, No.9, September* 2013.

[7] Meng-Jiao WANG and Yong-Zhen LI, "Hash Function with Variable Output Length", *IEEE International Conference on Network and Information Systems for Computers,* 2015.

[8] Swapnil Shinde et al., "Survey of Existing Authentication Systems", in *International Journal of Engineering Research and Technology (IJERT), ISSN: 2278-0181, Volume 3, Issue 3, pp.1096-1098, March 2014.*