Two Level Security using Mindmetrics and ID2S Password Authentication Technique

Snehal Anandkar[1], Ankita Bartakke[2], Pranjali Ganvir[3], Ankita Pawar[4]

[1, 2, 3, 4] *BE student, Department of Computer Engineering, Pimpri Chinchwad College Of Engineering, Pune, Maharashtra, India*

ABSTRACT

Having been a core feature of IT systems for several decades, passwords continue to represent both one of the most familiar and most maligned aspects of security technology. While their potential weaknesses have been well recognized mainly over the past decade, no permanent solution has come up yet as in terms of all-round usage and applicability. Shoulder surfing, simple guessing, external eavesdropping, side channel attacks etc are the common methods which lead to password leakages. The situation gets worse when a user puts a very obvious password which can be easily guessed by anyone knowing the person even vaguely. Most systems propose to improve both identification and verification of user but this method of mind-metrics can augment the current password based systems by strengthening the identification process. Mind-metrics utilizes personal secret data instead of a login id to identify a user uniquely. The proposed system also creates a scenario where two servers cooperate to authenticate a client and if one server is compromised, the attacker still cannot pretend to be the client with the information from the compromised server. The proposed system presents a symmetric solution for two-server key encryption, where the client can establish different cryptographic keys with the two servers, respectively.

Keyword: *Mindmetrics, PAKE protocol, Diffie-Hellman algorithm, ElGamal encryption algorithm, key generation, identification phase, and verification phase.*

1. INTRODUCTION

Traditionally the identification is performed with a username or login ID and the verification is done with a password. Computer systems employ an authentication mechanism to allow access only to legitimate users. The authentication procedure is composed of two parts, identification & verification. The identification is for answering the question, "who am I?", and the verification is for answering, "Am I who I claim I am?"[1].

While passwords are supposed to be random characters, login IDs are not random. They are used for communication or accounting purposes, and must carry a meaningful pattern. It may be part of users' first and/or last names, part of social security number, combination of names and numbers, account number, or email addresses. Thus login IDs are publicly known or can be guessed easily. In other words, obtaining the login ID is generally not a barrier for the attackers, and the success of an attack depends on the difficulty of the password. While a great emphasis was given to the verification, i.e., password system, somewhat less attention was given to the identification, i.e., login ID. By fortifying the identification part, the overall authentication system can be strengthened [1].

The goal is to improve the security of the authentication system as well as verification system by supplementing it with a secure identification process. 8 Instead it uses private information known only to the computer system and the user. This process makes the stolen password files unusable for the attackers. For password security we are using PAKE protocol to acquire secure channel and spilt the password and store it on different servers.

1.1. Goals and Objectives

i. To provide security by means of Mindmetrics and overcome drawbacks of biometrics. This will prevent the ways of exploitation by hackers and result in a strong security system.
ii. This is a product based project to build a better and secure system without hardware dependencies.

2. PROPOSED SYSTEM

2.1. Concept of Mind metrics

- Mindmetrics uses some secret data instead of human characteristics as a token to identify the user.
- It utilizes personal secret data instead of a login ID to identify a user uniquely, hence mindmetrics.

There are two parts in the Mindmetrics-based authentication process:-

i. Mindmetrics token is requested in the login page. A user specifies the token with which a computing system can identify a user account. Then the identification server looks up the registered access tokens to find a matching token and login ID.

ii. The server presents multiple login IDs to the user, with one of the login IDs being the correct login ID for the user account and some more real or fake IDs. To prevent the attackers from recognizing the login IDs, the login IDs are partially obscured. Among these partial login IDs, a legitimate user can still recognize the correct login ID and choose it.

- Above two steps are carried out in the identification phase. Once the server is identified then the conventional password verification method is used for granting the access.
- Mindmetrics-based system allows only the legitimate users to pass the identification stage. Here the password verification server is kept hidden, and users cannot access it unless they pass the identification server.

2.2. ID2S Password Authenticaton Key Exchange Protocol

In two-server password-authenticated key exchange **(**PAKE**)** protocol, a client splits its password and stores two shares of its password in the two servers, respectively, and the two servers then cooperate to authenticate the client without knowing the password of the client[2]**.** Even if one server is compromised, the attacker is still unable to pretend any client to authenticate against another server[2]**.**

3. ALGORITHMS USED

3.1. Diffie–Hellman Algorithm

Diffie–Hellman Key Exchange establishes a shared secret between two parties that can be used for secret communication for exchanging data over a public network. The following conceptual diagram illustrates the general idea of the key exchange by using colors instead of very large numbers.

The process begins by having the two parties, Alice and Bob, agree on an arbitrary starting color that does not need to be kept secret (but should be different every time); in this example the color is yellow. Each of them selects a secret color–red and aqua respectively–that they keep to themselves. The crucial part of the process is that Alice and Bob now mix their secret color together with their mutually shared color, resulting in orange and blue mixtures respectively, then publicly exchange the two mixed colors. Finally, each of the two mix together the color they received from the partner with their own private color. The result is a final color mixture (brown) that is identical to the partner's color mixture.

If another party (usually named Eve in cryptology publications, Eve being a third-party who is considered to be an eavesdropper) had been listening in on the exchange, it would be computationally difficult for that person to determine the common secret color; in fact, when using large numbers rather than colors, this action is likely very difficult for modern supercomputers to do in a reasonable amount of time.

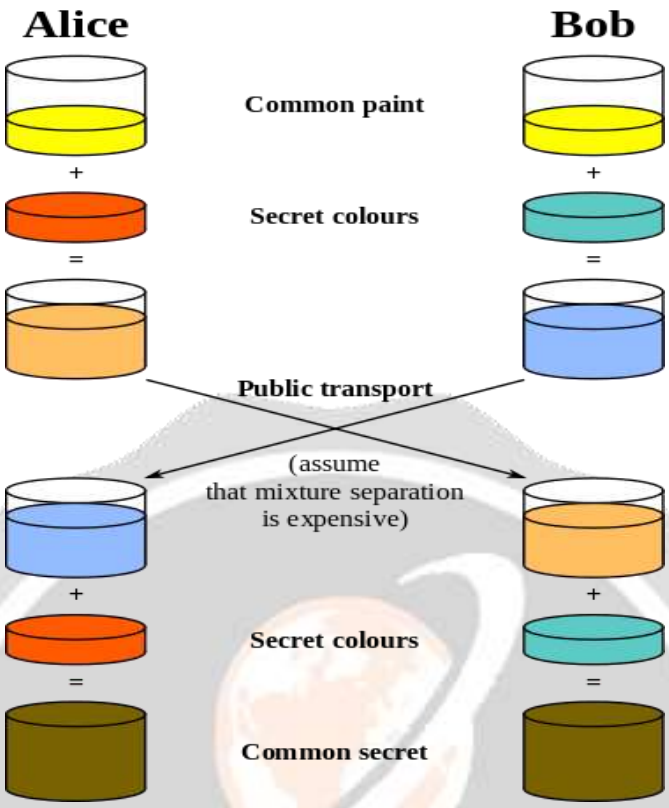

Figure 1: Diffie-Hellman Algorithm

Procedure:

i.
- Participant 1 generates a private random integer a
- Participant 2 generates a private random integer b

ii.
- Participant 1 generates her public value $A = g^a \bmod p$
- Participant 2 generates his public value $B = g^b \bmod p$

iii. Participants 1 and 2 exchange their public values

iv.
- Participant 1 computes $k1 = g^{ab} = (B)^a \bmod p$
- Participant 2 computes $k2 = g^{ba} = (A)^b \bmod p$

v. If k1 = k2, secure channel is established.
Both now have a shared secret k since $k = g^{ab} = g^{ba}$

3.2. ElGamal Encryption Algorithm

ElGamal Enryption algorithm is an assymetric key encryption algorithm for public-key cryptography which is based on Diffie-Hellman key exchange.

- Key Generation
  Participant 1 generates the public/private key pair

i. Generate large prime p and generator g.
ii. Select a random integer a, $1 \leq a \leq p - 2$, and compute $A = g^a \bmod p$.
iii. Participant 1's Public key is (p; g; A);

Participant 1's Private key is "a".

- Encryption Procedure
  Participant 2 encrypts a message m to participant 1

i. Obtain 1's authentic public key (p; g; A).

ii. Represent the message as integer m in the range {0, 1, …., p-1}.

iii. Select a random integer k, $1 \leq k \leq p-2$.

iv. Compute $c1 = (g^k \bmod p)$ and $c2 = m*(A)^k$.

v. Send ciphertext c = (c1; c2) to participant 1.

- Decryption Procedure

Participant A receives encrypted message m from B:

i. Use private key "a" to compute $(\gamma p-1-a) \bmod p$.
   Note: $\gamma p-1-a=\gamma-a=a-ak$.

ii. Recover m by computing $(\gamma-a) * \delta \bmod p$.

4. WORKING

- The System architecture consists of Two phases:
  1. Identification phase
  2. Verification phase
- The identification phase identifies trusted user by matching their data with existing data.
- The verification phase verifies whether the user is trusted user by matching the secret password.
- These two phases are divided into two servers.
- Further, Identification sever is linked with three different databases that contain user information after registration process.
- Verification server is linked with two different servers that contain information about half password each.

i. The system will first ask the user (already registered) for their mind-metrics token, after which, she/he will be presented with a set of fake and authentic ids which are closely similar.

ii. The user has to choose any one id.

iii. At last the password will be asked.

- The system adds a limit to the number of login attempts, failing those, the system will intimidate the user and recommending on changing any of the two passwords.
- If the user feels that the token has got too old or may have been compromised, the system will provide a means to change the token. Thus mind-metrics follows the principle of 'Protection in Abstraction' i.e. it makes a targeted attack on a particular user account nearly impossible.

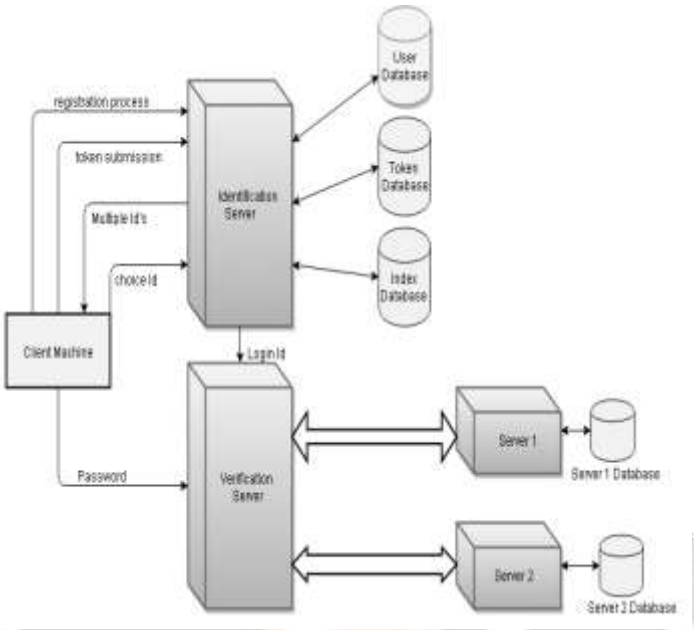

Figure 2: System Architecture

5. ADVANTAGES

5.1. Safety

The system assures safety to the data that user provides.

5.2. Security

The system provides security to trusted users and denies access to illegitimate ones. This is done by increasing the levels of security.

5.3. Scalability

The system is scalable as it does not depend on any hardware for its working.

5.4. Usability

Due to removal of hardware dependency, the system has a wide range of applications. It can be used by any organization.

5.5. Cost effective

As the system does not depend on any hardware for its execution, it is less costly and more effective at the cheaper price.

6. APPLICATIONS

The system has a wide range of applications.

6.1. Banking applications:

Digitization has led to introduction of various banking websites. Different banks allow users to use their banking policies through different websites and applications. Our system can be used by such banking applications.

6.2. E-commerce:

Various different e-commerce applications contain a lot of user information. This information needs to be secured from hacks to safeguard user's personal information, card details, etc. Our system can be effectively used by such applications.

6.3. Mobile applications:

Various high security systems require different hardware to make their systems stronger. Our system does not require any hardware. Hence, mobile applications can use our system to provide access from anywhere and everywhere.

7. EXPERIMENTAL RESULTS

The results obtained from the system developed are displayed in the following figures**.**

7.1. Registration phase:

Fig 3: Registration

Fig 4: Example for Registration phase

7.2. Login phase

i. Token**:**

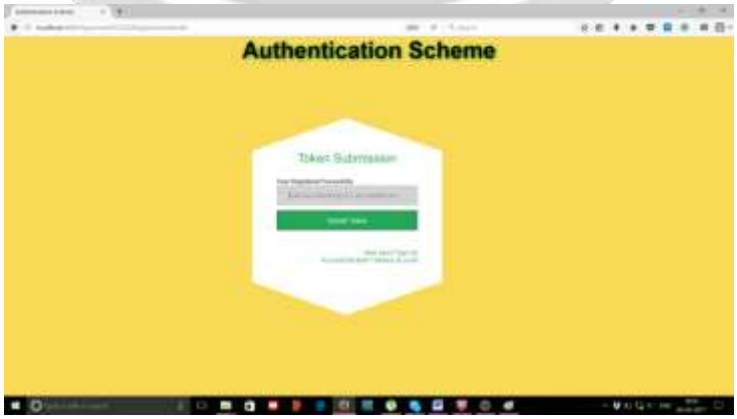

Fig 5: GUI to accept Token

ii. Username**:**

Fig 6: GUI to select Username

iii. Password**:**

Fig 7: GUI to accept password

iv. Successful login**:**

Fig 8*:* Display after successful login

8. CONCLUSION

It proposes a new scheme Mind metrics to strengthen the identification process. In addition, we have provided a rigorous proof of security for our compilers without random oracle.

9. FUTURE SCOPE

9.1. The system requires continuous internet connect for its working. This makes the system dependent of availability of internet.

9.2. The system is linked with various servers. These servers contain user data. Therefore, availability of server is another constraint. It is important for the servers to be available throughout and they should not crash.

6. REFERENCES

[1]. Juyeon Jo, Yoohwan Kim, Sungchul Lee, "Mindmetrics: Identifying users without their login Ids", IEEE, 2014.

[2]. Xun Yi, Fang-Yu Rao, Zahir Tari, Feng Hao, Elisa Bertino, Ibrahim Khalil, Albert Y. Zomaya, "ID2S Password-Authenticated Key Exchange Protocols", IEEE, 2016.

[3]. Michel Abdalla, David Pointcheval, "Simple Password-Based Encrypted Key Exchange Protocols", CT-RSA, 2005.

[4]. Jonathan Katz, Rafail Ostrovsky, Moti Yung, "Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords".