International Journal of Security and Its Applications
Vol.9, No.1 (2015), pp.397-408
http://dx.doi.org/10.14257/ijsia.2015.9.1.38

An Improved Biometric-based Multi-server Authentication Scheme Using Smart Card

Khanjan Ch. Baruah, Subhasish Banerjee, Manash P. Dutta and Chandan T. Bhunia

Department of Computer Science and Engineering
National Institute of Technology, Arunachal Pradesh, India - 791112
khanjan099@yahoo.com, subhasish.cse@nitap.in, manash.cse@nitap.in, ctbhunia@vsnl.com

Abstract

To protect the resources from unauthorized users, the remote user authentication have become an essential part in the communication network. Currently, smart card-based remote user authentication for multi-server environment is a widely used and researched method. Remote user authentication for multi-server environment has resolved the problem of users to manage the different identities and passwords. Recently, Mishra et al. proposed a multi-server authenticated key agreement scheme using smart cards, where they claim that their scheme is secure enough and could resist the various well known attacks. However, in this paper, we have shown that their scheme is not secure as they have claimed and can suffer from impersonation attacks and stolen smart card attack. Later in the paper, we propose an improved multi-server authentication scheme using smart cards, which not only overcomes the mentioned weaknesses but also can provide more functionality features.

Keywords: *Mutual Authentication, Biometric, Smart card, Multi-server*

1. Introduction

The development of Internet has revolutionized the lifestyle of the people. In the recent time, more and more traditional day-to-day affairs, like access to information, entertainment, financial services, and product purchase are carried out through the Internet. The exchange of personal information through insecure channel is forcing people to be concerned with Internet security. Authentication of any remote user, based on their identity, is a part of essence in Internet security. This process can be categorized based on either single server or multi server environment. Multi-server authentication schemes are superior to single-server authentication schemes, since in single-server authentication scheme, the user has to remember different identities and passwords for getting access into different remote servers. Moreover, the multi-server authentication provides the user the ease of login into different servers with a single registration. The smart card based multi-server authentication scheme is quite feasible for communication in insecure networks.

The authentication schemes, to be secure and efficient, should satisfy the following criteria [1, 2]:

a. Single user registration process: Users, to consume service from any application server, must first register themselves with the registration center. Moreover, the scheme should require the user to register once and still can communicate with multiple servers and thereby reducing the overhead of the network as well as the registration center.

b. Anonymity: The authentication process should not directly exchange the actual identity of the user.

ISSN: 1738-9976 IJSIA

c. No verification table and password table: Storage of password in the registration center increases the risk of various attacks. The process must avoid maintaining such tables, which increases the overhead of the registration center and also increases the vulnerability for attacks.
d. Efficiency: The smart cards have limited amount of computational power. Hence minimum amount of computation should be done in the smart cards during the different phases in the authentication process.
e. Ease of password selection: The user should have the freedom to select any password he/she wants and can easily change the password without informing the registration center.
f. Mutual authentication: The computation of the session key must ensure the use of information from both the user and the participating server.
g. Synchronization: Global clock synchronization is complicated in most of the network topology. So the authentication process should not use time stamp or clock related information.
h. Resistant to various kinds of attacks.

Remote user authentication through insecure channel was first introduced by Lamport [3], in 1981. However, this scheme was proven to be prone to various attacks. Since then different researchers put forward different schemes to enhance security or to lower computational cost. In 2001, Li *et al.* [4] used neural network for remote user authentication. In their scheme, each user must have large memory to store the public parameters used for authentication, thus resulting in higher computational cost. The traditional identity-based authentication schemes are totally based on the use of passwords. But passwords are simple and can be easily broken or forgotten. Researcher then introduced some biological characteristics of persons such as fingerprint, iris, palm prints etc. as keys. The main feature of using biometric is its uniqueness. Lee *et al.* [5], in 2002, proposed a remote user authentication scheme based on fingerprint. In their scheme, a user for login inserts his/her smart card, inputs their identity and password, and imprints their fingerprint into the fingerprint input device. However, Lin and Lai [6] and Chang and Lin [7], in 2004, pointed out that the scheme could not resist masquerade attacks. In 2010, Li and Hwang [8] proposed a remote user authentication scheme which was based on biometrics verification, smart card, one-way hash function and nonce for authentication. The use of one-way hash function and random nonce made it more efficient than the other schemes, however, Li *et al.* [9], in 2011, found that Li and Hwang's scheme does not provide proper authentication and cannot resist man-in-the-middle attack. In 2014, Chuang and Chen [2] proposed an anonymous multi-server authentication scheme based on trust computing. However, Mishra *et al.* [10] identified that their scheme did not resist stolen smart card attack and impersonation attacks, so they proposed an improved multi-server based authentication scheme using smart cards. We find that their scheme cannot withstand stolen smart card attack and impersonation attacks as well. To tackle their weaknesses, we have proposed an improved biometric-based remote user authentication scheme in multi-server environment. In section 3 of this paper, we provide a brief review of Mishra *et al.'s* scheme [10] and also provide its cryptanalysis in section 4. The proposed remote user authentication scheme and corresponding security analysis are presented in section 5 and section 6 of this paper respectively.

2. Threat Model

The following assumptions are made during the analysis and design of the scheme:

i. An adversary can be either a user or a server. A registered user as well as a registered server can act as an adversary.
ii. An adversary can eavesdrop every communication in public channels. He/ she can capture any message exchanged between user and server.

iii. An adversary has the ability to alter, delete or reroute the captured message.
iv. Information can be extracted from the smart card by examining the power consumption of the card.

Table 1. Notations Used in This Paper

ID_i	Identity of the i^{th} user
SID_j	Identity of the j^{th} server
PW_i	Password of the i^{th} user
BIO_i	Biometric of the i^{th} user
PSK	Pre-shared key of the servers
x	Master secret maintained by the registration center
T_r	Time of registration of the user
h(.)	A one-way hash function
N_i , n_1	Random nonce of the i^{th} user
N_j , n_2	Random nonce of the j^{th} server
$\oplus$	Exclusive-OR operation
||	Message concatenation operation

3. Mishra *et al.'s* Scheme

Many researchers have put forward their ideas in literature. Mishra et al. also proposed a remote user authentication scheme for multi-server environment. Their scheme comprised of five phases: server registration phase, user registration phase, login phase, authentication phase and password change phase. In this section, we briefly discuss their scheme.

3.1. Server Registration Phase

When the application server wants to provide its services to the public, it sends a join request to the registration center. The registration center authorizes the server and provides it a secret PSK through the Internet Key Exchange Protocol (IKEv2) [11]. Only the legitimate servers has the knowledge of PSK.

3.2. User Registration Phase

A user wishes to register themselves with the registration center so that they can have access to different services provided by the servers. He/she first selects an identity ID_i and password PW_i.

i. The user generates a random number N_i. Using their identity, password and the random number, the user computes $W_1 = h(PW_i||N_i)$ and $W_2 = h(ID_i \oplus N_i)$, and sends it to the registration center via a secure channel.
ii. The registration center receives W_1 and W_2, and computes $A_i = h(ID_i||x||T_r)$, $B_i = h(A_i)$, $X_i = B_i \oplus W_1$, $Y_i = h(PSK) \oplus W_2$, and $Z_i = PSK \oplus A_i$.
iii. The registration center personalizes the user's smart card SC_i with $\{X_i, Y_i, Z_i, h(.)\}$ and provides it to the user via a secure channel.
iv. The user, upon receiving the smart card, imprints the biometric BIO_i and computes $N = N_i \oplus h(BIO_i)$ and $V = h(ID_i||N_i||PW_i)$. Now, the smart card is updated with the new information as $SC_i = \{X_i, Y_i, Z_i, h(.), N, V\}$.

3.3. Login Phase:

The user inserts the smart card SC_i into the smart card reader and inputs the identity ID_i, password PW_i and biometric information BIO_i. The following operations are performed to generate the login messages:

i. SC_i computes $N_i = N \oplus h(BIO_i)$ and $V = h(ID_i||N_i||PW_i)$, and verifies whether the computed value of V matches with the stored value of V or not. If the verification fails, the session is simply terminated.

ii. If successful, W_1 and W_2 are computed using the required information, and further computes $B_i = X_i \oplus W_1$, $h(PSK) = Y_i \oplus W_2$.

iii. After computing B_i and h(PSK), the smart card SC_i generates a random nonce n_1 and computes $M_1 = h(PSK) \oplus n_1$, $M_2 = ID_i \oplus h(n_1||B_i)$ and $M_3 = h(ID_i||n_1||B_i)$. These messages are then transmitted to the server via a public channel.

3.4. Authentication Phase

The following sequences of operations are performed in this phase:

i. The server upon receiving the messages, $<M_1, M_2, M_3, Z_i>$, uses its secret PSK to retrieve A_i and n_1 from Z_i and M_1 respectively: $PSK = Z_i \oplus A_i$, $h(PSK) = M_1 \oplus n_1$. These two parameters are used to extract the identity ID_i from M_2: $ID_i = M_2 \oplus h(n_1||B_i)$.

ii. Before proceeding further the server checks whether the received M_3 is equal to $h(ID_i||n_1||B_i)$ or not. If the verification holds, the phase proceeds to the next step and upon failure the process is terminated.

iii. The server selects a nonce n_2 and computes $SK_{ji} = h(ID_i||SID_j||B_i||n_1||n_2)$ as the session key for future communication.

iv. The server computes $M_4 = n_2 \oplus h(ID_i||n_1)$, $M_5 = h(SK_{ji}||n_1||n_2)$ and responds to the user's request with M_4, M_5 and SID_j

v. The smart card receives the messages and retrieves n_2 from M_4: $n_2 = M_4 \oplus h(ID_i||n_1)$ and computes the session key as $SK_{ij} = h(ID_i||SID_j||B_i||n_1||n_2)$. It verifies M_5 with the computed value of $h(SK_{ji}||n_1||n_2)$. If the verification holds, it computes $M_6 = h(SK_{ij}||n_2||n_1)$ and sends it via a public channel.

vi. The server does the verification of M_6. On success the server confirms that the session key is legitimate and the user is authentic.

3.5. Password Change Phase

The password change is done locally without the involvement of the registration center.

i. The user insert the smart card SC_i and inputs ID_i, PW_i and BIO_i. The smart card computes $N_i = N \oplus h(BIO_i)$ and verifies whether $h(ID_i||N_i||PW_i)$ matches with the stored V or not. The success of verification prompts the user to input the new password PW_i^{new}.

ii. The smart card generates $W_1 = h(PW_i||N_i)$, $W_1^{new} = h(PW_i^{new}||N_i)$, $X_i^{new} = X_i \oplus W_1 \oplus W_1^{new}$ and $V^{new} = h(ID_i||N_i||PW_i^{new})$.

After this computation, the smart card is updated with the computed value of X_i^{new}, W_1^{new} and V^{new} by replacing X_i, W_i and V.

4. Cryptanalysis of the Model

The strength of any scheme can be determined by thorough analysis of the scheme. This section expresses the vulnerability of Mishra *et al's* scheme [10] in various communication scenarios.

4.1. Impersonation Attack

In this kind of attack, a registered but malicious server can masquerades as another server or legal user by using the common shared secret key PSK. The possibility of such attacks are defined as follow:

Act As another server: A registered server with identity SID_x, may spoof as another server. During server registration, all the server receives a common secret key PSK from the registration center. Thus, it enables all the servers to read any message meant for another server (say, SID_y). The adversary can intercepts all the messages intended for SID_y and authenticate the user as shown below:

Adversary computes:

$A_i = PSK \oplus Z_i$

$n_1 = h(PSK) \oplus M_1$

$ID_i = M_2 \oplus h(n_1||B_i)$

The adversary selects a random nonce, say N, and computes the session key but replaces its identity SID_x with SID_y. The session key as well as the other messages can be generated as:

$SK_{ji} = h(ID_i||SID_y||B_i||n_1||N)$

$M_4 = N \oplus h(ID_i||n_1)$

$M_5 = h(SK_{ji}||n_1||N)$

The user receives these messages, agrees upon the same session key but is unaware of the impersonation by the adversary.

Act As a user: Say, a registered server (say, identity SID_x) may be an adversary and may try to masquerade the identity of any user (identity ID_i). The server, during its communication with the user, computes the user's identity. It can use the user's identity ID_i to authenticate with another server (say, SID_y). The server can generates the following for creating a valid login request message:

$M_1 = h(PSK) \oplus N$

$M_2 = ID_i \oplus h(N||B_i)$

$M_3 = h(ID_i||N||B_i)$

These messages, $\{M_1, M_2, M_3\}$, along with Z_i is transmitted to the server (SID_y) via a public channel for authentication.

4.2. Stolen Smart Card Attack

In this attack, an adversary uses the stolen smart card to masquerade as a legitimate user. Therefore, the authentication process should be secure enough that an adversary must not be able to misuse a stolen smart card. In the analysis of Mishra et al.'s scheme, they said that the adversary cannot generate valid login messages using the parameters extracted from the stolen smart card. However, after careful analysis we find that this is not the case, any legal but malicious server can use the extracted smart card parameters to authenticate as a legal user to another server. The stolen smart card is used for authentication given as follow:

i. The adversary, a registered server SID_i, extracts the parameters stored in the smart card, $\{ X_i, Y_i, Z_i, h(.), N, V\}$, and uses a previously communicated message to obtain the user's identity ID_i: $ID_i = M_2^{old} \oplus h(n_1||B_i)$.

ii. The adversary replays the previous login message to the server SID_y as: $M_1^{old} = h(PSK) \oplus n_1$, $M_2^{old} = ID_i \oplus h(n_1||B_i)$ and $M_3^{old} = h(ID_i||n_1||B_i)$.

iii. The server SID_y considers the message as legitimate and selects a random nonce n to generate the reply messages: $M_4 = n \oplus h(ID_i||n_1)$, $M_5 = h(SK_{ji}||n_1||n)$.

iv. The adversary receives the message and extracts n from M_4: $n = M_4 \oplus h(ID_i||n_1)$. Now, it uses the parameters to generate the session key $SK_{jy} = h(ID_i||SID_y||B_i||n_1||n)$.

v. The malicious server uses the session key and the nonce, n and n_1, to generate the verification message $M_6 = h(SK_{ij}||n||n_1)$ and transmits it to the server SID_y.

vi. The server SID_y verifies M_6 with its computed M_6 and finds it to be equal.

Thus, the adversary can be authenticated, as a valid user, to the server SID_y using the stolen smart card.

4.3. Man-in-the-middle Attack

In this attack, the adversary eavesdrop the communication between the user and the server. Mishra *et al.*, showed that their scheme was secure against this attack. However, we analyzed it to be vulnerable to such attack. A registered but malicious server may eavesdrop any communication between a user and a server. The justification is as follow:

i. When the smart card sends the message < M_1, M_2, M_3, Z_i> to the server for authentication, the adversary may intercept this message.

ii. Since the adversary is a registered server, it has the common shared secret PSK. Thus the adversary, using this secret extracts the identity as well as the random number n_1 of the user.

$A_i = PSK \oplus Z_i$
$n_1 = h(PSK) \oplus M_1$
$ID_i = M_2 \oplus h(n_1 || B_i)$

iii. The adversary simultaneously forwards the message < M_1, M_2, M_3, Z_i> to the corresponding server.

iv. The respective server, unaware of the intermediate malicious server, proceeds with the required operation of the authentication process. It generates a random number n_2 and communicate it to the user through the message <SID_j, M_4, M_5 >.

v. The adversary intercepts this message and extracts the server generated random number n_2 as follow:

$n_2 = M_4 \oplus h(ID_i || n_1)$

vi. Using the information, the adversary computes the session key as:

$SK_{ji} = h(ID_i || SID_j || B_i || n_1 || N)$.

This session key can be used to read or modify the messages exchanged between the user and the server.

5. Proposed Scheme

The proposed remote user authentication scheme for multi-server environment has four phases: registration phase, login phase, authentication phase and password change phase. The detail description of each phase are given as follow:

5.1. Registration Phase

The registration phase is the initial phase of the scheme. In this phase, the registration center provide secrets to the user as well as the server. Basically, it can be sub-categorized into the server registration phase and the user registration phase.

Server Registration Phase: When a server wants to provide some service to the public, then it has to first register itself to the registration center. The server sends a join request along with its identity (say, SID_j) to the registration center. In return, the registration center replies with $h(SID_j || h(PSK))$ and $h(PSK || x)$ through the Internet Key Exchange Protocol version 2 (IKEv2) [11]. The server uses these secret to authenticate any registered user.

User Registration Phase: The users must first register themselves if they want to access any services provided by the set of registered servers. Therefore, the user submits his/her identity ID_i and $R_1 = h(PW_i || BIO_i)$ through a secure channel. The registration center then computes the following:

$A_i = h(ID_i || x)$
$B_i = h(PSK || x) \oplus A_i$
$C_i = h(R_1 || ID_i) \oplus h(A_i)$
$D_i = h(PSK) \oplus h(ID_i)$
$E_i = R_1 \oplus ID_i$

The registration center creates a smart card SC_i with the following information $SC_i = \{B_i, C_i, D_i, E_i, h(.)\}$. This personalized smart card is then provided to the user via a secure channel.

5.2. Login Phase

To start any conversation, the user must first login to a specific terminal using its smart card. The user inserts the smart card and inputs his/her identity ID_i, password PW_i and biometric information BIO_i. The smart card executes the following sequence of operations:

i. The smart card before sending any information to the server first checks whether the user is authorized to gain access or not. Therefore, it computes $R_1 = h(PW_i||BIO_i)$ and then verifies whether the entered identity ID_i is equal to stored identity $ID_i = R_1 \oplus E_i$ or not. If failure occurs, the login phase is immediately aborted. Otherwise, proceeds for the succeeding steps.

ii. The smart card extracts $h(PSK) = h(ID_i) \oplus D_i$ and $h(A_i) = C_i \oplus h(R_1||ID_i)$ from the stored data.

iii. It then randomly generates a nonce N_i and computes the messages:

$M_1 = h(SID_j||h(PSK)) \oplus h(ID_i||N_i)$

$M_2 = N_i \oplus h(A_i)$

$V_1 = h(N_i \oplus B_i)$

iv. The smart card transmits the message $<B_i, M_1, M_2, V_1>$ to the server SID_j via a public channel for authentication.

5.3. Authentication Phase

The server SID_j, upon receiving the authentication messages, performs the following set of operations to agree on the same session key.

i. The server uses its secrets, obtained during registration, to compute $A_i = B_i \oplus h(PSK||x)$ and $h(ID_i||N_i) = M_1 \oplus h(SID_j||h(PSK))$. Using $h(A_i)$, it gets N_i from $M_2 : N_i = M_2 \oplus h(A_i)$.

ii. Before generating any messages, the server must verify the user’s authenticity. So, it uses the above derived information and verifies whether V_1 is equal to the computed value $h(N_i \oplus B_i)$ or not. If this holds, then the server generates a random nonce N_j. On failure, the phase is simply exited.

iii. The server uses the user’s information and its nonce N_j and identity SID_j to generate the session key as $SK_{ji} = h(h(ID_i||N_i)||SID_j||B_i||N_j)$.

iv. Now, the server sends its randomly selected nonce to the user as $M_3 = N_j \oplus h(ID_i||N_i)$ and also $V_2 = N_i \oplus h(SK_{ji}||N_j)$ via a public channel.

v. Once the message is received, the user computes N_j from M_3. It then uses the information to compute the session key as $SK_{ij} = h(h(ID_i||N_i)||SID_j||B_i||N_j)$. It is to be noted that both session keys are the same.

vi. Now, the user verifies whether the server is the actual one or not with whom he wants to communicate with. It is done by checking N_i with the computed value $V_2 \oplus h(SK_{ij}||N_j)$.

5.4. Password Change Phase

The mechanism is simple enough that if the user wants to change his/her password, it can be done without informing the registration center. The user inserts his/her smart card into the machine and inputs his/her identity ID_i, password PW_i and biometric BIO_i. The card checks the entered information. If the user is the authentic one, then the card prompts the user for new password PW_i^* and computes:

$R_1^* = h(PW_i^*||BIO_i)$

$E_i^* = E_i \oplus R_1 \oplus R_1^*$

$C_i^* = h(R_1^*||ID_i) \oplus h(R_1||ID_i) \oplus C_i$

Lastly, the smart card updates E_i^* and C_i^* in the place of E_i and C_i. Now, the updated smart card has $SC_i = \{B_i, C_i^*, D_i, E_i^*, h(.)\}$.

6. Security Analysis

In this section, we analyze the common security features of our proposed authentication scheme:

6.1. Resist against Impersonation Attack

In this attack, an adversary can masquerade as a legitimate user or a server. The following are the analysis for the different scenarios of this attack:

Server Side: A legal but malicious server, with identity SID_x, may masquerade the identity ID_i of a user. The malicious server uses the messages from previous conversation:

$B_i = h(PSK||x)\oplus A_i$,

$M_1 = h(SID_x||h(PSK))\oplus h(ID_i||N_i)$,

$M_2 = N_i\oplus h(A_i)$

and from these messages the server gets the following parameters: $h(ID_i||N_i)$, N_i, B_i. The server may use these parameters to generate a new message $M_1^* = h(SID_y||h(PSK))\oplus h(ID_i||N_i)$ to authenticate another server (say, SID_y), but it is not possible to generate the message M_1^* as $h(SID_y||h(PSK))$ is unknown to the server SID_x. Moreover, the server cannot compute $h(SID_y||h(PSK))$ from the information it possesses.

Secondly a malicious server, with identity SID_x, may intercept the login messages meant for the server with identity SID_y. The adversary will not correctly extract $h(ID_i||N_i)$ from the login message M_1, since the information $h(SID_y||h(PSK))$ is unknown. It tries to generate a valid authentication message $\{M_3, V_2\}$ for a random nonce N_j, where $M_3 = N_j\oplus h(ID_i||N_i)$ and also $V_2 = N_i\oplus h(SK_{ji}||N_j)$. This attempt will not succeed, since the adversary SID_x cannot compute SK_{ij} and M_3 correctly, as $h(ID_i||N_i)$ is required.

User Side: A user, either registered or unregistered, may try to impersonate as another entity. However, our scheme can resist such impersonations. An adversary, in this case is an unregistered user, will not be able to generate the message without knowing the important parameters like h(PSK) and $h(A_i)$. Moreover, if the adversary has access to a smart card he/she will not be able to get these parameters, since these are stored along with the password as well as the biometrics of the user.

In case of a registered user, the adversary may eavesdrops the communication between a user and the server and then tries to extract the parameter $h(A_i)$, which is required for login. In our proposed scheme, this attempt will not succeed, since $h(A_i)$ is protected by the $h(R_1||ID_i)$, where $R_1 = h(PW_i||BIO_i)$.

6.2. Resist Against Stolen Smart Card Attack

The user's smart card may get lost or stolen. An adversary may extract all the information, $SC_i = \{B_i, C_i, D_i, E_i, h(.)\}$, stored in the smart card and use them to login into the system. However, this attempt will fail, since for generating message $h(A_i)$ is needed, but it is protected by the user's password, biometric and identity.

Besides, if a server is the adversary and it somehow retrieved the information $\{B_i, C_i, D_i, E_i, h(.)\}$ from the user's smart card, can try to generate the valid login message. However, the server would not be able to generate the message since the adversary cannot construct $h(SID_y||h(PSK))$ required for login into the server SID_y.

6.3. Forward Secrecy

User's long term secret A_i is protected by $h(PSK||x)$ and only an authentic server can extract the user's long term secret. Even though, an adversary gets hold of this secret still it will not be able to compute the session key as it depends on four parameters: $h(ID_i||N_i)$, SID_j, N_j and B_i. Also the session key keeps on varying with each session.

6.4. Anonymity

Anonymity of user is to prevent the disclosure of the user's identity to any unauthorized personals. In our proposed scheme, during the login phase, the user sends a login request, $\{M_1, M_2, V_1, B_i\}$, to the desired server. The user's identity ID_i is well protected by N_i as well as $h(SID_j||h(PSK))$. Thus, an adversary will not be able to retrieve the user's identity ID_i, since $h(SID_j||h(PSK))$ is present only with the server SID_j.

6.5. Resist against Offline Password Guessing Attack

An adversary may try to guess the user's password offline by extracting the information, $\{B_i, C_i, D_i, E_i, h(.)\}$, from the user's lost/stolen smart card. However, the adversary will not be able to verify the password PW^* using the extracted information. The verification of the guessed password PW^* requires the adversary to compute $R_1=h(PW_i^*||BIO_i\ )$, which is not possible, since the adversary does not have any knowledge about the biometric BIO_i of the user.

6.6. Resist against Man-in-the-middle Attack

In this attack, the adversary intercepts all the messages between the user and the server and selectively modifies the data. In our proposed scheme, if an adversary, either actively or passively, eavesdrops the communication, it will not succeed in retrieving any useful information. This attempt is shown below:

The adversary intercepts the login request from the user $\{B_i, M_1, M_2, V_1\}$ and tries to extract the parameters:

$A_i = B_i \oplus h(PSK||x)$

$h(ID_i||N_i) = M_1 \oplus h(SID_j||h(PSK))$

$N_i = M_2 \oplus h(A_i)$

However, the adversary will not be able to extract these information, since the adversary does not have the knowledge of $h(PSK||x)$ and $h(SID_j||h(PSK))$. Thus, the verification of the computed $V_1 = h(N_i \oplus B_i)$, where $B_i = h(PSK||x) \oplus A_i$, will fail.

6.7. Resist against Replay Attack

An adversary eavesdrop a communication between a user and the server and then may try to use these messages for opening a communication to a server in future. However, our proposed authentication scheme is resistant to such attempts. Adversary may eavesdrop a communication and store the login messages, $\{M_1, M_2, V_1, B_i\}$, for performing replay attack in future, where $M_1 = h(SID_j||h(PSK)) \oplus h(ID_i||N_i)$, $M_2 = N_i \oplus h(A_i)$, $V_1 = h(N_i \oplus B_i)$ and $B_i = h(PSK||x) \oplus A_i$. The adversary transmits these stored messages, $\{M_1, M_2, V_1, B_i\}$, to a registered server SID_j. The server SID_j, upon receiving the messages retrieves $A_i = h(PSK||x) \oplus B_i$, $h(ID_i||N_i) = h(SID_j||h(PSK)) \oplus M_1$, $N_i = M_2 \oplus h(A_i)$ and also verifies these using V_1. This verification holds, since the messages has not been modified by the adversary. Upon verification, the server SID_j selects a random nonce N_j^* and generates the session key as $SK_{ij}^* = h(h(ID_i||N_i)||SID_j||B_i||N_j^*)$. It then uses this session key for computing the reply messages $M_3^* = N_j^* \oplus h(ID_i||N_i)$ and also $V_2^* = N_1 \oplus h(SK_{ji}||N_j^*)$, and transmits to the adversary. The adversary tries to compute N_j^* but this attempt will fail, since he/she does not know $h(ID_i||N_i)$.

6.8. Mutual Authentication and Freshness

In our proposed authentication scheme, the server verifies the authenticity of the user by comparing V_1 with the computed value $h(N_i \oplus B_i)$, where $B_i = h(PSK||x) \oplus A_i$. The user gives a challenge to the server to operate on the user's nonce N_i with the hash of the session key and the server's nonce N_j: $V_2 = N_i \oplus h(SK_{ji}||N_j)$. The server sends the message V_2 and its own nonce N_j as: $M_3 = N_2 \oplus h(ID_i||N_i)$ to the user. The user checks the server's authenticity using the message V_2. Moreover, the computation of the session key depends on the user's and server's identity as well as their nonce: $SK_{ij} = h(h(ID_i||N_i)||SID_j||B_i||N_j)$. The use of random nonce, N_i and N_j, verifies the freshness to the process. Thus decreasing the probability of generation of the same session key.

7. Performance Comparison

In this section, we compare the security properties of our scheme with other related biometrics–based authentication schemes which is shown in the Table 2. We have compared our scheme with Mishra et al.'s scheme [10] and other schemes.

Table 2. Comparison of Security Attributes of our Scheme with Other Schemes

Security Attributes	Proposed scheme	Mishra et al. [10]	Chuang and Chen [2]	Li and Hwang [8]
User anonymity	Yes	Yes	Yes	No
Biometric template	Yes	Yes	Yes	Yes
Simple password change	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	No
Resist impersonation attack	Yes	No	No	No
Resist Server Spoofing	Yes	No	No	Yes
Resist Stolen smart card attack	Yes	No	No	No
Resist Offline guessing attack	Yes	Yes	Yes	Yes
Resist man-in-the-middle attack	Yes	Yes	No	No
Time synchronization	No	No	No	No
Resist Insider attack	Yes	Yes	Yes	No
Session key verification	Yes	Yes	Yes	No

The computational cost of the authentication process depends on the hash function, exclusive – OR and other mathematical operations. We define the following notations used for computing the computational complexity of the proposed scheme:

T_h: time for executing a one-way hash function h(.).

T_x: time for executing exclusive – OR operation.

T_c: time required for executing comparison operation.

The Table 3 gives the comparison of the computational cost with the other schemes.

Table 3. Performance Comparison with Other Multi-server Schemes

Phase	Proposed scheme	Mishra et al. [10]	Chuang and Chen [2]	Li and Hwang [8]
Registration (P1)	$8T_h+4T_x$	$7T_h+5T_x$	$3T_h+3T_x$	$3T_h+T_x$
Login (P2)	$6T_h+6T_x+T_c$	$6T_h+6T_x+T_c$	$4T_h+3T_x+T_c$	$T_h+2T_x+T_c$
Authentication (P3)	$7T_h+6T_x+2T_c$	$12T_h+5T_x+3T_c$	$13T_h+6T_x+3T_c$	$5T_h+4T_x+2T_c$
Password Change (P4)	$2T_h+3T_x+T_c$	$5T_h+3T_x+T_c$	$2T_h+5T_x+T_c$	$3T_h+2T_x+ T_c$
Total	$23T_h+19T_x+4T_c$	$30T_h+19T_x+5T_c$	$22T_h+17T_x+5T_c$	$12T_h+9T_x+4T_c$

8. Conclusion

In this paper, we have discussed the existing authentication scheme proposed by Mishra *et al.* and shown how their scheme can suffer from stolen smart card, and impersonation attacks. In order to remedy their weaknesses, we have proposed an efficient and secure authentication scheme. The proposed scheme satisfies all the required security attributes for a secure authentication, which are demonstrated in security analysis. Finally, we have shown the computational complexity comparison of our proposed scheme with other related schemes.

References

[1] X. Li, J. Ma, W. Wang, Y. Xiong and J. Zhang, “A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments”, Mathematical and Computer Modelling, vol. 58, no.1-2, **(2013)**, pp. 85-95.

[2] M. C. Chuang and M. C. Chen, “An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics”, Experts Systems with Applications, vol. 41, no. 4, **(2014)**, pp. 1411-1418.

[3] L. Lamport, “Password authentication with insecure communication”, Communications of the ACM, vol. 24, no. 11, **(1981)**, pp. 770–772.

[4] L. H. Li, I. C. Lin and M. S. Hwang, “A remote password authentication scheme for multi-server architecture using neural networks”, IEEE Transactions on Neural Networks, vol. 12, no. 6, **(2001)**, pp. 1498–1504.

[5] J-K. Lee, S-R. Ryu and K-Y. Yoo, “Fingerprint-based remote user authentication scheme using smart cards”, Electronics Letters, vol. 38, no. 12, **(2002)**, pp. 554–555.

[6] C-H. Lin and Y-Y. Lai, “A flexible biometrics remote user authentication scheme”, Computer Standards & Interfaces, vol. 27, no. 1, **(2004)**, pp. 19–23.

[7] C-C. Chang and I-C. Lin, “Remarks on fingerprint-based remote user authentication scheme using smart cards”, ACM SIGOPS Operating Systems Review, vol. 38, no. 4, **(2004)**, pp. 91–96.

[8] C-T Li and M-S. Hwang, “An efficient biometrics-based remote user authentication scheme using smart cards”, Journal of Network and Computer Applications, vol. 33, no. 1, **(2010)**, pp. 1–5.

[9] X. Li, J-W. Niu, J. Ma, W-D. Wang and C-L. Liu, “Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards.”, Journal of Network and Computer Applications, vol. 34, no. 1, **(2011)**, pp. 73-79.

[10] D. Mishra, A.K. Das and S. Mukhopadhyay, “A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards”, Expert Systems with Applications, vol. 41, no. 18, **(2014)**, pp. 8129-8143.

[11] C. Kaufman, Internet Key Exchange (ikev2) protocol, (**2005**).

Authors

Khanjan Changmai Baruah received his B.Tech in Computer Science and Engineering from Tezpur University, Assam, India in 2013. Currently, he is pursuing his M.Tech degree in Mobile Communication and Computing from National Institute of Technology, Arunachal Pradesh. His research interests are Network Security, Wireless Ad hoc Network and Green

Subhasish Banerjee received his M.Tech degree in Computer Application from Indian School of Mines, Dhanbad, India in 2012. Currently he is pursuing his Ph.D. and also working as Assistant Professor in the Department of Computer Science and Engineering in National Institute of Technology, Arunachal Pradesh. His research activities are mainly focused on cryptography and information security.

Manash Pratim Dutta received his M.Tech degree in Information Technology from Sikkim Manipal University, Sikkim, India in 2012. Currently, he is working as Assistant Professor in the Department of Computer Science and Engineering in National Institute of Technology, Arunachal Pradesh. His research activities are mainly focused on cryptography.

Chandan Tilak Bhunia did his B. Tech. in Radiophysics and Electronics in 1983 from Calcutta University. He received his M. Tech. in Radiophysics and Electronics in 1985 and then joined North Bengal University as a lecturer of Computer Science & Applications in 1988. He became Assistant Professor of ECE at NERIST, Govt. of India in 1990. He got P. hd. in Computer Science & Engineering from Jadavpur University. He became a full Professor in 1997 at NERIST. Currently, he is working as a Director of National Institute of Technology, Arunachal Pradesh. He has published around 150 research papers in various national and international journals of repute. Under his supervision, five Ph.D. scholars got awarded and nine scholars are currently working in various fields.