I J C T A, 9(17) 2016, pp. 8445-8454

An Improved Remote User Password Authentication Scheme Using Smart Card with Session Key Agreement

Ajay Kumar Sahu* **and Ashish Kumar****

ABSTRACT

Remote user authentication is a mechanism, in which the remote server verifies the legitimacy of a user over an insecure communication. Password authentication based on smart cards is one of the simplest and most efficient authentication methods and is a commonly deployed to authenticate the legitimacy of remote users. Based on cryptographic techniques, several password authentication schemes have previously been implemented have its own merits and demerits. Recently, Chang Y F *et al.* have pointed out the weaknesses of Wang et. al. scheme and proposed an untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. As per their claims, scheme ensures security because its security is based on the secret number and the password at the same time and providing privacy preservation even with low-computation-ability devices and user's in these applications can choose and change password freely. However, we find that Chang et. al.'s scheme violates the purpose of dynamic-identity as claimed by the author. In this paper we found that once the smart card of an arbitrary user is lost or stolen, passwords of all registered user's are at risk. Using the information from an arbitrary smart card, an adversary can impersonate any user of the system. Its password change phase has some drawback and even not secure. There is no any provision for the smart card verification mechanism and even session key agreement. In this paper, we propose a scheme to overcome the aforementioned weaknesses and shows that our scheme is user friendly and more secure than other related schemes.

Keywords: Authentication, Dynamic ID, Smart card, Security, Cryptanalysis, Password, Session Key

I. INTRODUCTION

The Internet has become an integral part of everyday life. With rapid growth of the internet, we can access any service from any place and at any time. However, with the increase of network attacks, information security becomes an important issue in network based application systems. Authentication of identity is a process to authenticate the identity of user before access a service. The authentication of password is the simplest and the most convenient authentication mechanism to deal with secret data and privacy over insecure networks.

In this paper, our main aim is to present an efficient scheme for a remote user authentication which can not only survive in smart card loss situation but also it can withstand the aforementioned threats. In this paper we analyze Chang *et al's* scheme [9] and found some weaknesses. In Chang at al's scheme, we show that the secret number used, which is the basis of the security of this scheme provide the loopholes to break the security of his scheme. Once an adversary steals or obtains the smartcard anyhow of an arbitrary user, adversary can obtain the identity of any legal user of the system by using that smart card; hence the Chang's scheme well secured as they claimed [9] user un-traceability fails to do justice with the concept of dynamic-identity. After studying and analyzing the Chang's *et al.*'s scheme and various others related schemes we

* Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology, Ghaziabad, Uttar Pradesh, India, *E-mail: ajay4989@gmail.com*

** Department of Computer Science and Engineering, ITS, Greater Noida, Uttar Pradesh, India, *E-mail: ashishcse29@gmail.com*

identified various security problems. As a remedy to overcome these identified security problems, we propose a more efficient and secure scheme. In this scheme, our main aim is to overcome the various security flaws identified in Chang *et al.*'s scheme. This scheme avoids impersonation attacks by imposing proper mutual authentication between user and server, and very truly this scheme also provides user anonymity along with user un-traceability in real sense. The proposed scheme's in this paper deal with many security threats identified in various previous proposals and free from those security threats. On comparing the efficiency and performance of the proposed scheme with some other related schemes, including Wang *et al.*'s[6] scheme and Chang *et al.*'s[9] scheme, we show that our scheme will becomes adopted as a better option for real applications because our scheme is most efficient and highly secure yet lightweight at communication and computation.

The rest of the paper is organized as follows. In section 2, review of the Chang *et al.*'s [9] scheme is given. In Section 3, presents Cryptanalysis of Chang *et al.*'s scheme. In Section 4, the proposed scheme is given. In Section 5, security analysis of the proposed scheme is given. In Section 6, we have done performance analysis and security requirement comparison. Finally, Conclusion will be drawn in Section 7.

II. REVIEW OF CHANG *et al.*'S SCHEME

In the following section, the scheme of Chang *et al.*'s [9], untraceable dynamic-identity-based remote user authentication scheme with verifiable password update is reviewed. The scheme of Chang *et al.* has four phases: registration phase, login phase, authentication phase and password change phase.

2.1. Registration Phase

In registration phase, when a new user (U_i) wants to access server's (S) services, he registers itself at the server side (S).

Step 1: U_i chooses his identity (ID_i), password (Pw_i) and send to server (S) via a secure channel.

Step 2: After receiving the registration request ($ID_{i,} Pw_i$), server (S) computes $N_i = h(ID_i||x) \oplus h(Pw_i)$ using its secret key x.

Step 3: Server (S) stores parameters $\{N_i, y, h(.)\}$ into user's smart card (SC). Here y is unique random number assigned to user (U) by remote server (S).

Step 4: Server (S) issue the smart card (SC) to user (U_i) via a secure channel.

2.2. Login Phase

When a registered user (U_i) wants to access the services of the server (S), U_i inserts his smart card into a terminal device and input ID_i, Pw_i.

Step 1: The smart card computes $CID_i = ID_i \oplus h(N_i||y||T)$, N_i '$= N_i \oplus h(y||T)$, $B= N_i \oplus h(Pw_i) = h(ID_i||x)$ and $C = h(N_i||y||B||T)$, Where T is the current time-stamp.

Step 2: The smart card sends this login request $\{CID_i, N_i', C, T\}$ to server (S) through a common channel.

2.3. Authentication Phase

Upon receiving the login request $\{CID_i, N_i', C, T\}$ from the smart card at time T ', server (S) and user (U_i) authenticate each other. The details are as follows:

Step 1: S checks whether (T '-T) $\leq \Delta T$ and if there is no login request with the same parameters $\{CID_i, N_i', C, T\}$ within the time period from $(T - \Delta T)$ to $(T + \Delta T)$. If both the conditions hold, this phase continues; otherwise, S aborts all login requests with the same parameters $\{CID_i, N_i', C, T\}$ immediately and terminates this phase.

Step 2: S retrieves $N_i = N_i' \oplus h(y||T)$, $ID_i = CID_i \oplus h(N_i||y||T)$, then computes $B^* = h(ID_i||x)$, and $C^* = h(N_i||y||B^*||T)$.

Step 3: S checks if C* is equal to C received previously. If they are equal, User (U_i) successfully authenticates by S, and S computes $a = h(B^*||y||T'')$, where T '' is the current timestamp. Otherwise, S rejects U_i's login request and records ID_i and the number of cumulative failed requests for security issues. If three continuous requests from ID_i, fail in a short interval, S will ignore U_i's following request within a guard interval.

Step 4: Server sends {a, T ''} to the smart card via a common channel.

Step 5: After receiving {a, T ''} from Server, the smart card checks the freshness of T''. If T '' is fresh in an expected time interval, the smart card computes $a' = h(B||y||T'')$ and compares a' with a received previously. If they are equal, the smart card /User can ensure that S is legal.

2.4. Password change phase

Password change phase will be executed whenever user (U_i) wants to change password Pw_i to Pw_{new}.

Step 1: U_i inserts his smart card into a terminal device, keys in ID_i and Pw_i and sends a password change request.

Step 2: The smart card computes $CID_i = ID_i \oplus h(N_i||y||T)$, $N_i' = N_i \oplus h(y||T)$, $B = N_i \oplus h(Pw_i) = h(ID_i||x)$ and $C = h(N_i||y||B||T)$, Where T is the current time-stamp.

Step 3: The SC sends {CID_i, N_i', C, T, password change request} to S through a common channel.

Step 4: After getting the password change request {CID_i, N_i', C, T, Password change request} from the smart card at time T ', S checks whether $(T' - T) \leq \Delta T$ and if there is neither login nor password change request with the same parameters {CID_i, N_i', C, T} at time from $(T - \Delta T)$ to $(T + \Delta T)$. If they both hold, this phase continues; otherwise, S aborts all request with the same parameters {CID_i, N_i', C, T} immediately and terminates this phase.

Step 5: S retrieves $N_i = N_i' \oplus h(y||T)$, $ID_i = CID_i \oplus h(N_i||y||T)$, then computes $B^* = h(ID_i||x)$ and $C^* = h(N_i||y||B^*||T)$.

Step 6: S checks if C* is equal to C received previously. If they are equal, U_i is successfully authenticated by Server, and S computes $a = h(B^*||y||m||T'')$, where T '' is the current timestamp and m is the reply **yes/no** to the password change request. Otherwise, S rejects U_i's password change request and records ID_i and the number of cumulative failed requests for security issues. If three continuous requests from ID_i fail in a short interval, S will ignore U_i's following request within a guard interval.

Step 7: Server (S) sends {a, m, T ''} to smart card via a common channel.

Step 8: After receiving {a, m, T ''} from server, the smart card checks the freshness of T ''. If T '' is fresh in an expected time interval, the smart card computes $a' = h(B||y||m||T'')$ and compares a' with a received previously. If they are equal, the smart card ensures that S is legal and the password change request is verified.

Step 9: If the password change request is verified, the smart card will ask user to input the new password Pw_{new} twice for confirmation. Note that if the inputted passwords are not same, the smart card will ask user to key in the new password Pw_{new} twice again. If the inputted passwords are same, the smart card computes $(N_i)_{new} = N_i \oplus h(Pw_i) \oplus h((Pw)_{new})$ and replaces N_i with $(N_i)_{new}$.

III. CRYPTANALYSIS OF CHANG *ET AL.'*S SCHEME

In this section our concern mainly on security problems found in Chang *et al.*'s [9] scheme. After analyzing the Chang's *et al.*'s scheme we found various security drawbacks exists in their proposed scheme.

3.1. Off-line password guessing attack

In Chang *et al.*'s scheme an adversary U_a can guess a user's password is as follows: If U_a obtains the lost/ stolen smart card of an arbitrary user of the system then as per the researches by Kocher *et al.* [1] and Messerges *et al.*[3], an adversary U_a can extracts the secret number y from it. In this scheme, the server uses a secret number y which is common for all users of the system and is stores in plaintext form in each user's smart card. Then U_a intercepts the login request $\{CID_i, N_i', C, T\}$ of user$_{,}$ then guess user's password, by the following manner:

1. Retrieves N_i from N_i' by computing $N_i = N_i' \oplus h(y||T)$. Then retrieves ID_i from CID_i by computing $ID_i = CID_i \oplus h(N_i||y||T)$.
2. Guesses Pw_i* as user's possible password and computes $C^* = h(N_i||y||N_i \oplus h(Pw_i^*)||T)$.
3. Compares the computed C* with C available in login request. If $C^* \neq C$, then U_a repeats from step 2 with some another guess and so on until he gets success. If C* = C, it implies that U_a has successfully guesses U_i's password.

In this way, U_a can guess the password of any user after extracting the secret number y from lost/stolen smart card of an arbitrary user.

3.2. User impersonation attack

In this scheme, the adversary U_a can extract [1, 3] the secret number y from lost/stolen smart card and since this secret number y is common for all users, therefore U_a can intercept any login request from the network and can impersonate the corresponding user. Suppose U_a intercepts the login request $\{CID_i, N_i', C, T\}$ of user U_i. When U_a successfully guesses the password of $U_{i,}$ he possesses N_i, ID_i, and Pw_i corresponding to U_i and he can impersonate U_i at any time in the following manner:

1. U_a acquires the current timestamp T_a and computes $CID_a = ID_i \oplus h(N_i||y||T_a)$, $N_a' = N_i \oplus h(y||T_a)$ and $C_a = h(N_i||y||N_i \oplus h(Pw_i)||T_a)$.
2. U_a transmits the login request $\{CID_a, N_a', C_a, T_a\}$ to sever S. Clearly, this login request will be accepted by the server S because it is computed using valid identity ID_i and fresh timestamp T_a. By virtue of correctly guessed password $Pw_{i,}$ the equivalence of $C_a^* = C_a$ will hold at server side S as $C_a^* = h(N_a||y||h(ID_i||x)||T_a)$ computed by server will be equal to $C_a = h(N_a||y||N_a \oplus h(Pw_i)||T_a)$.

Thus, U_a can impersonate a legal user of the system by only intercepting the login request of the user.

3.3. Server masquerading attack

In order to masquerade as the legal server S, the adversary U_a proceeds in the following manner:

1. U_a intercepts the login request $\{CID_i, N_i', C, T\}$ of a user U_i from open network and blocks it from reaching to the server S. Then immediately U_a computes $h(y||T_i)$ to retrieve N_i by computing $N_i = N_i' \oplus h(y||T)$, and checks if there exist some field containing N_i in his record or not. If not so, then U_a replays the blocked login request. But if N_i exists in U_a's record, he continues further.
2. Acquires current timestamp T_a and computes the response message $a_a = h(N_i \oplus h(Pw_i)||y||T_a)$. Then U_a transmits the response message $\{a_a, T_a\}$ to the user U_i.
3. The response message $\{a_a, T_a\}$ will pass the authentication test at the user side because timestamp T_a is fresh and B = B*, as $N_i \oplus h(Pw_i) = h(ID_i||x)$, hence $a_a' = a_a$.

In this way, the user believes that the response message $\{a_a, T_a\}$ is from the legal server S whereas it is the adversary (U_a) in place of legal server (S).

3.4. Lacks proper mutual authentication

In this scheme both user and the server authenticate to each other. For this there is a provision in the scheme to verify the legitimacy of the server by user as well as for the server by the user. Server (S) accepts the login request of user, only after checking the current timestamp freshness and also it will check the equivalence $C^*= C$ holds or not. If condition fails then the login request fake and server deny for providing service. In the similar manner, the server is also authenticated by the user by checking the timestamp freshness followed by checking the equivalence $a'= a$. But the adversary U_a can impersonate as a user (as in Section 3.2) and masquerade as legal server (as in Section 3.3). This scattered the mutual authentication between the user and the server and hence mutual- authentication is not properly achieved in this scheme.

3.5. Insider attack

When U_i wants the services from S, he must register himself. In this scheme for registration phase, a user submits his identity ID_i and password Pw_i in plaintext form to server. This facilitates direct access of user's password to the privileged insider of the system at server. After getting user's password, insider attacker plays as a valid user of the system for other servers where the same password applied by the user for his own convenience. Hence the insider may break the privacy and security of any user in this scheme. So, in the registration phase, transmitting the password in plaintext form can be very dangerous for the security of the user.

IV. THE PROPOSED SCHEME

Here we propose an improved untraceable dynamic-ID based password authentication scheme using smart card with Session key agreement to overcome the security problems of Chang *et al.*'s [9] scheme. The proposed scheme composed of four phases: registration phase, login phase, authentication phase and password change phase.

4.1. Registration phase

In this phase, when a new user (U_i) wants to access some services from the server S, he registers himself at the S.

Step 1: User (U_i) chooses his or her identity (ID_i), password (Pw_i) and an arbitrary number (α). Then compute $MPw_i = h\,(\alpha||Pw_i)$ and sends this registration request $\{ID_i, MPw_i\}$ to server S via a secure channel.

Step 2: After receiving the registration request $\{ID_i, MPw_i)$ from user U_i, the server (S) chooses an arbitrary number (β_i). This number (β_i) is unique for each user that means no any two users having the same arbitrary number (β_i).

Step 3: The server (S) computes: $N_i = h\,(ID_i||x) \oplus MPw_i$, $X_i = \beta_i \oplus h\,(ID_i||x)$, $Y_i = h\,(ID_i|| \beta_i||MPw_i)$ and $Z_i = \beta_i \oplus h\,(\beta||x)$, where x is the secret key maintained by server S.

Step 4: Server (S) stores the parameters $\{X_i, Y_i, Z_i, h\,(.)\}$ into smart card (SC_i) and delivers {Smart card, N_i} to user via a secure channel.

Step 5: Upon receiving {Smart card and N_i} from the server, user computes $L_i= (ID_i||Pw_i) \oplus \alpha$ and $M_i= N_i \oplus \alpha$. User inserts L_i and M_i in Smart card and he need not remember this random number α anymore, finally the contents of Smart card = $\{L_i, M_i, X_i, Y_i, Z_i, h\,(.)\}$.

4.2. Login Phase

When a registered user (U_i) wants to access the services from server (S), U_i inserts his smart card into a terminal device and keys in ID_i and Pw_i.

Step 1: Compute $\alpha = L_i \oplus (ID_i||Pw_i)$ and $MPw_i= h(\alpha||Pw_i)$. Then retrieves $h(ID_i||x) = M_i \oplus MPw_i \oplus \alpha$ and $\beta_i= X_i \oplus h(ID_i||x)$ and computes $Y_i^* = h(ID_i||\beta_i||MPw_i)$.

Step 2: Smart card verifies if the computed Y_i^* and stored Y_i are equal or not. If $Y_i^* \neq Y_i$, SC_i drops the session. If it repeats thrice then SC_i gets blocked and U_i is required to enter PUK (Private Unblocking Key) to re-activate his Smart card.

Step 3: Only if $Y_i^*= Y_i$, the accuracy of inputted ID_i and Pw_i is verified and Smart card proceeds further. Smart card retrieves $h(\beta||x)=\beta_i \oplus Z_i$ and $N_i=M_i \oplus \alpha$. Acquires current timestamp T and computes $CID_i = ID_i \oplus h(N_i||\beta_i||T)$, $N_i' = N_i \oplus h(\beta_i||T)$, $U_i = N_i \oplus MPw_i = h(ID_i||x)$, $V_i = h(N_i||\beta_i||U_i||T)$ and $W_i = \beta_i \oplus (h(\beta||x)||T)$.

Step 4: SC_i transmits the login request = $\{CID_i, N_i', V_i, W_i, T\}$ to the server via a public channel.

4.3. Authentication Phase

Upon receiving the login request = $\{CID_i, N_i', V_i, W_i, T\}$ from the smart card at time T’, server and user authenticate each other. The details are as follows:

Step 1: Server checks whether $(T‘-T) \leq \Delta T$ and if there is no login request with the same parameters $\{CID_i, N_i', V_i, W_i, T\}$ within the time period from $(T - \Delta T)$ to $(T + \Delta T)$. If both the conditions hold, this phase continues; otherwise, server aborts all login requests with the same parameters $\{CID_i, N_i', V_i, W_i, T\}$ immediately and terminates this phase directly.

Step 2: S computes $\beta_i = W_i \oplus (h(\beta||x)||T)$, $N_i = N_i' \oplus h(\beta_i||T)$ and $ID_i= CID_i \oplus h(N_i||\beta_i||T)$. Then computes $U_i^* = h(ID_i||x)$ and $V_i^* = h(N_i||\beta_i||U_i^*||T)$.

Step 3: S checks if V_i^* is equal to V received previously. If they are equal, User successfully authenticates by server at current time stamp T “and S computes $a = h(U_i^*||\beta_i||T”)$. If $V_i^* \neq V$, S rejects U_i’s login request $\{CID_i, N_i', V_i, W_i, T\}$, records ID_i and the number of cumulative failed requests for security issues. If three continuous requests from ID_i fail in a short interval, S will ignore U_i’s following request within a guard interval.

Step 4: Server (S) sends {a, T ”} to the smart card (SC) via a common channel.

Step 5: After receiving {a, T ”} from S, the smart card checks the freshness of T ”. If T ” is fresh in an expected time interval, smart card computes $a' = h(U_i||\beta_i||T”)$ and compares a’ with a received a previously. If a’ = a, the smart card /User can ensure that S is legal.

Step 6: After successful mutual authentication, U_i and S independently compute the common session key as $S_{session}= h(U_i||\beta_i||T||T”||h(\beta||x))$ and $(S_{session})^* = h(U_i^*||\beta_i||T||T”||h(\beta||x))$ respectively.

4.4. Password change phase

Password change phase will be executed whenever user wants to change his password Pw_i to Pw_{new}. The procedures of Password change phase are as follows:

Step 1: User inserts his smart card into a terminal device, keys in ID_i and Pw_i and sends a password change request.

Step 2: Smart card computes $\alpha = L_i \oplus (ID_i||Pw_i)$ and computes $MPw_i = h(\alpha||Pw_i)$. Then retrieves $h(ID_i||x) = M_i \oplus MPw_i \oplus \alpha$ and $\beta_i= X_i \oplus h(ID_i||x)$, and computes $Y_i^* = h(ID_i||\beta_i||MPw_i)$.

Step 3: The smart card compare computed Y_i^* and stored Y_i. If $Y_i^* \neq Y_i$, SC rejects this request. If it fails three times then SC gets blocked and user required entering PUK (Private Unblocking Key) to re-activate his SC.

Step 4: If $Y_i^*= Y_i$, the inputted ID_i and Pw_i is verified by Smart card.

Step 5: Smart card allows the user to enter the new password $(Pw)_{new}$ two times. If the entered passwords are not equal, the smart card asks user to re-enter the new password $(Pw)_{new}$ again two times. If they are same, Smart card computes $(MPw_i)_{new} = h(\alpha||(Pw)_{new})$, $(L_i)_{new} = (ID_i||(Pw)_{new}) \oplus\alpha$, $(M_i)_{new}= M_i \oplus(MPw_i) \oplus(MPw_i)_{new}$ and $(Y_i)_{new}= h(ID_i|| \beta_i||(MPw_i)_{new})$. Stores $(L_i)_{new}$, $(M_i)_{new}$ and $(Y_i)_{new}$ in place of L_i, M_i, and Y_i respectively**.**

V. SECURITY ANALYSIS OF PROPOSED SCHEME

Here, we analyze the security of proposed scheme and found that our proposed scheme is much better than Chang *et al.*'[9] scheme. There are various major points regarding the security analysis of the proposed scheme as given below:

5.1. Prevent Insider attack

Here the user does not submit the password in plaintext form to the server. Here, the user concatenate the password Pw_i with random number α and submits hashed value in modified form as $MPw_i = h(\alpha||Pw_i)$. So, the insider cannot obtain a user's password. As in the form of (MPw_i), both values of $\{\alpha, Pw_i\}$ are not known by the insider, and for him it is not possible to guess both value randomly and verify it. Hence, this scheme is free from insider attack.

5.2. Prevent offline password guessing attack

In section 5.2, we show that adversary can extract the values $\{L_i, M_i, X_i, Y_i, Z_i, h(.)\}$ from lost/stealing smart card but cannot obtain the values $\{ID_i, \beta_i, h(ID_i||x), \alpha\}$. In the proposed scheme it is also not possible for U_a to guess the users password arbitrary as Pw_a as he cannot verify his guess using $L_i= (ID_i||Pw_i) \oplus \alpha$, $M_i= N_i \oplus \alpha = h(ID_i||x) \oplus MPw_i \oplus \alpha$, $X_i = \beta_i \oplus h(ID_i||x)$, $Y_i= h(ID_i|| \beta_i||MPw_i)$, $Z_i = \beta_i \oplus h(\beta||x)$. Because at one time U_a needs to guess at least two unknown values which is not possible. If we assume that U_a intercepts login request of the user U_i as $\{CID_i, N_i', V_i, W_i, T\}$ but U_a cannot guess the password using $V_i = h(N_i||\beta_i||U_i||T)$ as $\{N_i, \beta_i, \alpha\}$ are unknown here as well as without the values of $\{N_i, \beta_i, \alpha\}$, U_a cannot obtain the value of $V_i^* = h(N_i||\beta_i||U_i^*||T)= h(N_i||\beta_i||h(ID_i||x)||T)=h(N_i||\beta_i||N_i \oplus h(\alpha||Pw_a)||T)$. Hence, cannot verify the guessed password by comparing V_i* and V_i. In this scheme, the values of stored smart card and the values of login request are not matched, so the adversary cannot correlate these values either stolen SC or intercept login request. Hence, the proposed scheme resists the offline password guessing attack.

5.3. Provides forward secrecy

In our scheme forward secrecy plays very crucial security features for authentication mechanism, it guarantees that our data will secure during transmission once session between user and server has established. In our proposed scheme, user and server compute the session key $S_{session}=h(U_i||\beta_i||T||T''||h(\beta||x))= h(h(ID_i||x)||\beta_i||T||T''||h(\beta||x))$. Suppose server's secret key x is disclosed, the secret number β still not known by U_a. Suppose the secret key x and secret number β are disclosed anyhow, $S_{session}$ still remains secure because U_a not know the values of ID_i and unique secret number β_i assigned by server to user. Hence, disclosing the server's secret key x, secret number β or password Pw_i does not facilitate an attacker to compute the established session key.

5.4. Smart card having inbuilt verification mechanism

When a registered user wants to access the services of the server, U_i inserts his smart card into a terminal device and keys in $ID_{i,}$ Pw_i and before computing the login request, the smart card performs some computations as: $\alpha = L_i \oplus(ID_i||Pw_i)$ and $MPw_i=h(\alpha||Pw_i)$. Then retrieves $h(ID_i||x) = M_i \oplus MPw_i \oplus\alpha$ and $\beta_i= X_i \oplus h(ID_i||x)$ and finally, computes $Y_i^* = h(ID_i||\beta_i||MPw_i)$. If $Y_i^*= Y_i$ holds, then it confirms that inputted

ID_i and Pw_i are correct then smart card proceeds to compute the login request. If $Y_i^* \neq Y_i$, smart card drops the session. If inputted ID_i or Pw_i is wrong continuously thrice then smart card gets blocked and U_i is required to enter PUK (Private Unblocking Key) to re-activate his smart card. In this way, the smart card verifies the correctness of ID_i or Pw_i, there is no need of server involvement in this mechanism.

5.5. Provides proper mutual authentication

In our proposed scheme, after accepting login request depending upon the result of timestamp freshness test by the user, the server checks the equivalence $V_i^* = V_i$, if found correct then authenticate the user. In response, the Server sends {a, T ”} to the smart card via a common channel. After receiving {a, T ”} from Server, the smart card checks the freshness of T ”. If T ” is fresh in an expected time interval, then SC will check the equivalence of a’ = a, if found correct, the smart card /User can ensure that server is legal. In our scheme, the U_a can neither impersonate as a user nor can act as a legal server. Hence, in our scheme, we can achieve proper mutual authentication.

VI. PERFORMANCE ANALYSIS AND SECURITY REQUIREMENT COMPARISON

In this section, we analyze and compare the performance and efficiency of the proposed scheme with other related schemes in terms of storage capacity, communication cost, efficiency/security characteristics and achievements/goals. In the proposed scheme, we use the lightweight hash function h (.) and exclusive OR ⊕ operation. It is usually take very low computation cost as well as storage capacity. Here we assume that $\{ID_{i,}\ Pw_i\}$, random numbers $\{\alpha, \beta_i\}$, timestamps {T, T”} and output of one way hash function $\{h\ (ID_i||x)$, $h(ID_i||\beta_i||MPw_i)$, etc} are 128-bit long. In table 1, we have compare the storage capacities required by the smart card/sever.

Table 1
Storage Capacity Comparison

Storage/ Scheme	*W B Hsieh et al.[15]*	*Y C Lee [11]*	*Wang Y Y et al. [6]*	*F Wen et al. [10]*	*Y F Chang et al [9]*	*Our Scheme*
Smart Card	3*128=384 bits	3*128=384 bits	4*128=512 bits	3*128=384 bits	3*128=384 bits	6*128=768 bits
Server	2*128=256 bits	1*128=128 bits	2*128=256 bits	2*128=256 bits	1*128=128 bits	2*128=256 bits

Moreover, the security comparison of the proposed scheme with the relevant authentication schemes is summarized in

Table 2. it is clear that our scheme is more secure and achieves more features than other relevant studies. Our scheme achieves almost all features that are essentially required in implementing a practical and universal remote user authentication scheme using smart cards.

Table 2
Efficiency/Security Characteristics Comparison

Security Characteristics/Scheme	*W B Hsieh et al.[15]*	*Y C Lee [11]*	*Wang Y Y et al. [6]*	*F Wen et al. [10]*	*Y F Chang et al [9]*	*Our Scheme*
Prevent User impersonation attack	Yes	Yes	No	No	No	Yes
Prevent server masquerading attack	Yes	No	Yes	No	No	Yes
Prevent Offline Password attack	Yes	Yes	No	No	No	Yes
Prevent Insider attack	Yes	No	Yes	No	No	Yes
Prevent secret key x	No	No	No	Yes	No	Yes

Table 3
Comparison of achievements/goals

Goals/Schemes	*W B Hsieh et al.[15]*	*Y C Lee [11]*	*Wang Y Y et al. [6]*	*F Wen et al. [10]*	*Y F Chang et al [9]*	*Our Scheme*
Provide Proper Mutual authentication	Yes	No	Yes	Yes	Yes	Yes
Provide perfect forward secrecy	No	No	No	No	N/A	Yes
Establish Session Key Agreement	No	No	No	Yes	Yes	Yes
Provides freely Pw choosing facility	Yes	Yes	Yes	Yes	No	Yes
Provides user un-traceability	No	Yes	No	No	No	Yes
Provides verification mechanism in Smart Card	No	No	No	Yes	No	Yes
No need to maintain the verifier table by Server	Yes	Yes	Yes	Yes	Yes	Yes
Preserve User anonymity	No	No	No	Yes	No	Yes
Storage, Communication and Computation cost should be low	Yes	Yes	No	No	Yes	Yes

VII. CONCLUSION

We have proposed an improved untraceable remote user password authentication scheme using smart card with Session key agreement to overcome the drawbacks present in Chang *et al.*'s scheme. In the proposed scheme, we have prevented various security vulnerabilities and the need to interact with server for password is removed. The proposed scheme provide proper mutual authentication and session key agreement between the user and the server for communication at the end of each session, which was not present in Chang *et al.*'s scheme. Also the provision of verification mechanism needed in smart card which avoids denial of service is present. The proposed scheme outperformed the related existing schemes based on comparison of results. Based on the security and performance analysis, it is observed that the increased computational complexity cost of the proposed scheme pays off in preventing more attacks against the Chang *et.als.'*. The scheme is equipped with feature of user anonymity with un-traceability and does not require any database to be maintained on the server. The scheme is involving only hash/ XOR, to keep the computational load minimum. The proposed scheme can be utilized for the applications requiring privacy preservation, enhance security and having low computation ability.

REFERENCES

[1] Kocher P, Jaffe J, Jun B. Differential power analysis. In: Proceedings of advances in cryptology CRYPTO'99'; 1999. P. 388-97.

[2] Hung-Min Sun. An efficient remote user authentication scheme using smart cards. IEEE Transaction on consumer Electronics, Vol. 46, No. 4, November 2000.

[3] Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comp 2002: 51 (5): 541-52.

[4] Min-Shiang Hwang. A Simple Remote User Authentication Scheme. Mathematical and Computer Modelling 36; 2002, p. 103-107.Password Authentication scheme over insecure networks. J. of Computer and System Sciences 72 (2006), p. 727-740. doi:10.1016/j.jcss.2005.10.001.

[5] Da-ZhiSun, Jin-Peng Huai, Ji-Zhou Sun, Jian-Xin Li, Jia-Wan Zhang. Improvements of Juang *et al.*'s Password-Authenticated Key Agreement Scheme Using Smart Cards. IEEE Transactions on Industrial Electronics, Vol. 56, No. 6. June 2009.

[6] Wang Y Y, Liu J Y, Xiao F X, Dan J. A more efficient and secure dynamic ID- based remote user authentication scheme. Journal of computer communication 32 (2009), pp 583-585. *http://dx.doi.org/10.1016/j.comcom.2008.11.008.*

[7] Manoj Kumar, Mridul Kumar Gupta, Saru Kumari. An Improved Efficient Remote Password Authentication Scheme with Smart Card over Insecure Networks. International J. of Network Security, Vol. 13, No. 3, PP. 167-177, Nov. 2011.

[8] Khan M K, Kim S K, Alghathbar K. Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme. Journal of computer communication 34 (2011), pp 305-309. .http://dx.doi.org/10.1016/j.comcom.2010.02.011.

[9] Chang Y F, Tai W L, Chang H C. Untraceable dynamic identity based remote user authentication scheme with verifiable password update. International Journal communication system 2013. *http://dx.doi.org/10.1002/dac.2552.*

[10] Wen F, Li X. An improved dynamic ID-based remote user authentication with key agreement scheme. Journal of computers and electrical engineering 38 (2012), 381-387. Doi:10.1016/j.compeleceng.2011.11.010.

[11] Lee Y C. A new dynamic ID- based user authentication scheme to resist smart-card-theft attack. An international journal of applied mathematics & Information Sciences 6 No. 2S pp. 355S-361S (2012).

[12] Deepchand Ahirwal, Sandeep Raghuwanshi. A Noble Remote User Authentication Protocol Based on smart card using Hash Function. IJETTCS: Volume 1, Issue 4, Nov. Dec. 2012.

[13] Ram Ratan Ahiewal, Swarn Sanjay Sonwanshi. An efficient and secure ID-based remote user authentication scheme using smart card. IJAIS: Volume 1, Issue 6, Feb. 2012.

[14] Zhen-Yu Wu, Dai-Lun Chiang, Tzu-Ching Lin, Yu-Fang Chung. A reliable Dynamic User-Remote password Authentication scheme over Insecure Network. 26th Int. Conference on Advanced Information Networking and Applications, 2012.

[15] Hsieh W B, Leu J S. Exploiting hash functions to intensify the remote user authentication scheme. Computer & Security 31 (2012), pp 791-798. *http://dx.doi.org/10.1016/j.cose.2012.06.001.*

[16] Saru Kumari, Muhammad Khurram Khan, XiingLi. An improved remote user authentication scheme with key agreement. Computer and Electrical Engineering 40, 2014; pp. 1997-2012. http://dx.doi/org/10.1016/j.compeleceng.2014.05.007.

[17] Marimuthu Karuppiah, R. Saravanan. A secure remote user mutual authentcation scheme using smart cards. J of information security and applications 19; 2014, p. 282-294. http://dx.doi.org/10.1016/j.jisa.2014.09.006.