Smart Card based Secure Authentication and Key Agreement Protocol

Sandeep K. Sood, Anil K.Sarje and Kuldip Singh
Department of Electronics & Computer Engineering
Indian Institute of Technology
Roorkee, India
{ssooddec, sarjefec, ksconfcn}@iitr.ernet.in; san1198@yahoo.co.in

Abstract: **In 2005, Yoon-Ryu-Yoo proposed a simple remote user authentication scheme which is an improvement on Hwang-Lee-Tang's scheme. However, we found that Yoon-Ryu-Yoo's scheme easily reveals a user's password and is vulnerable to impersonation attack using stolen smart card. This scheme is also found to be vulnerable to parallel session attack and man-in-the-middle attack. This paper proposes a new remote user authentication scheme that resolves the aforementioned problems, while keeping the merits of Yoon-Ryu-Yoo's scheme**.

Keywords: Cryptography; Password; Authentication protocol; Smart card; Hash function.

I. INTRODUCTION

Smart cards have been widely used in many e-commerce applications and network security protocols due to their low cost, portability, efficiency and the cryptographic properties. Smart card stores some sensitive data corresponding to the user that assist in user authentication. The user (card holder) inserts his smart card into a card reader machine and submits his identity and password. Then smart card and card reader machine perform some cryptographic operations using submitted arguments and the data stored inside the memory of smart card to verify the authenticity of the user.

In 1981, Lamport [1] proposed a password based authentication scheme that authenticates the remote users over an insecure communication channel. Lamport's scheme removes the problems of password table disclosure and communication eavesdropping. Since then, a number of remote user authentication schemes have been proposed to improve security, efficiency and cost. In 2000, Hwang and Li [2] found that Lamport's scheme [1] is vulnerable to the risk of a modified password table and the cost of protecting and maintaining the password table is large. Therefore, they proposed a cost effective remote user authentication scheme using smart card that is free from the mentioned risk. Hwang and Li's scheme [2] can withstand replay attack and also authenticate the remote users without maintaining a password table. In 2000, Sun [3] proposed a smart card based remote user authentication scheme to improve the efficiency of Hwang and Li's scheme [2]. In 2002, Hwang-Lee-Tang [4] proposed a remote user authentication scheme that does not require any password verification table on the remote server and legitimate users are free to choose and change their password freely without the help of a remote server. They claimed that their scheme provides effective authentication and requires less computation as compared to other schemes proposed by Wu [5] in 1995, Jan and Chen [6] in 1998, Yang and Shieh [7] in 1999, Hwang and Li [2] in 2000 and Chien-Jan-Tseng [8] in 2002. In 2005, Yoon-Ryu-Yoo [9] found that Hwang-Lee-Tang's scheme [4] is vulnerable to stolen verifier attack and denial of service attack using the stolen smart card. They proposed an improved scheme to preclude the weaknesses of Hwang-Lee-Tang's scheme [4].

In 2009, Hsiang and Shih [11] found that Yoon-Ryu-Yoo's scheme [10] is vulnerable to parallel session attack, masquerading attack and password guessing attack using stolen smart card and proposed an improved scheme free from these flaws. In 2009, Kim and Chung [13] found that Yoon and Yoo's scheme [12] is vulnerable to leak of password using stolen smart card, masquerading user attack, masquerading server attack, stolen verifier attack and proposed an improved scheme free from these flaws.

In this paper, we show that the Yoon-Ryu-Yoo's scheme [9] easily reveals a user's password and is vulnerable to impersonation attack using stolen smart card. This scheme is also found to be vulnerable to parallel sessions attack and man-in-the-middle attack. The remedy of these pitfalls, this paper presents an efficient scheme free from these attacks. The proposed scheme inherits the merits of Yoon-Ryu-Yoo's scheme with improved security.

The rest of this paper is organized as follows. In Section 2, a brief review of Yoon-Ryu-Yoo's scheme [9] is given. Section 3 describes the cryptanalysis of Yoon-Ryu-Yoo's scheme [9] to different attacks. In Section 4, the improved scheme is proposed. The security analysis of the proposed improved scheme is presented in Section 5. The comparison of the cost and functionality of the proposed scheme with the other related schemes is shown in Section 6. Section 7 concludes the paper.

II. REVIEW OF YOON-RYU-YOO SCHEME **[9]**

In this section, we examine a simple remote user authentication scheme proposed by Yoon-Ryu-Yoo [9] in 2005. Yoon-Ryu-Yoo's scheme consists of four phases (i.e., registration, login, authentication and password change) as summarized in Fig. 1. The notations used in this section are listed in Table 1.

978-1-4244-9034-/10/$26.00©2010 IEEE

A. Registration Phase

The user U_i registers with the server S by submitting his identity ID_i and password P_i over a secure communication channel. The server S computes $V_i = H(ID_i, T_{TSA}, x)$ and $A_i = H(ID_i, T_{TSA}, x) \oplus P_i$, where T_{TSA} is the trusted time stamp provided by a trusted time stamping authority and x is the secret key of the server. Then the server S issues the smart card containing secret parameters $(ID_i, V_i, A_i, H(\,))$ to the user U_i through a secure communication channel.

Table 1
Notations

U_i	i^{th} User
S	Server
ID_i	Unique Identification of User U_i
P_i	Password of User U_i
T_{TSA}	Trusted Time Stamp Provided by TSA Authority
x	Master Secret of Registration Server S
H()	One-Way Hash Function
$\oplus$	XOR Operation
|	Concatenation

B. Login phase

The user U_i inserts his smart card into a card reader to login on to the server S and submits his identity ID_i^* and password P_i^*. The smart card verifies the submitted identity ID_i^* with the stored value of ID_i in its memory. Then smart card computes $B_i = A_i \oplus P_i^*$ and verifies the computed value of B_i with the stored value of V_i in its memory. If both values match, the legitimacy of the user is assured and the smart card proceeds to the next step. Otherwise the login request from the user U_i is rejected. Afterwards, the smart card computes $C_1 = H(B_i, T)$, where T is current date and time of input device and sends the login request message (ID_i, C_1, T) to the service provider server S.

C. Authentication phase

The service provider server S checks the format of ID_i and the validity of timestamp T by checking $(T' - T) <= \delta T$, where T' denotes the server's current timestamp and δT is permissible time interval for a transmission delay. Afterwards, the server S computes $B_i^* = H(ID_i, T_{TSA}, x)$, $C_1^* = H(B_i^*, T)$ and compares C_1^* with the received value of C_1. If they are not equal, the server S rejects the login request and terminates this session. Otherwise the server S acquires the current time stamp T'' and computes $C_2 = H(B_i^*, C_1^*, T'')$ and sends the message (C_2, T'') back to the smart card of user U_i. On receiving the message (C_2, T''), smart card checks the validity of timestamp T'' by checking $(T''' - T'') <= \delta T$, where T''' denotes the client's smart card current timestamp. Then the user U_i's smart card computes $C_2^* = H(B_i, C_1, T'')$ and compares it with received value of C_2. This equivalency authenticates the legitimacy of the service provider server S and the login request is accepted else the connection is interrupted.

D. Password change phase

A user U_i inserts his smart card into the card reader and enters his identity ID_i^* and password P_i^* corresponding to his smart card. The smart card verifies the submitted identity ID_i^* with the stored value of ID_i in its memory. Then smart card computes $B_i = A_i \oplus P_i^* = H(ID_i, T_{TSA}, x)$ and compares the computed value of B_i with stored value of V_i in its memory to verify the legitimacy of the user U_i. Once the authenticity of cardholder is verified then the user U_i can instruct the smart card to change his password. Afterwards, the smart card asks the cardholder to resubmit a new password P_i^{new} and then smart card updates the value of $A_i = H(ID_i, T_{TSA}, x) \oplus P_i$ stored in its memory with $A_i^{new} = H(ID_i, T_{TSA}, x) \oplus P_i^{new}$. Finally, the password of the user U_i gets changed. Afterwards, the user U_i can login on to the server S with his old identity ID_i and using new password P_i^{new}.

III. CRYPTANALYSIS OF YOON-RYU-YOO'S SCHEME

Yoon-Ryu-Yoo [9] claimed that their protocol can resist various known attacks. However, we found that their protocol is flawed for stolen smart card attack, impersonation attack, parallel sessions attack and man-in-the-middle attack.

A. Stolen smart card attack

A user may lose his smart card, which is found by the attacker or the attacker steals the user's smart card. The attacker can extract the stored information through some technique such as by monitoring their power consumption and reverse engineering techniques as pointed by Kocher et al. [14] and Messerges et al. [15]. He can extract ID_i, $V_i = H(ID_i, T_{TSA}, x)$ and $A_i = H(ID_i, T_{TSA}, x) \oplus P_i$ from the memory of smart card because smart card contains $(ID_i, V_i, A_i, H(\,))$. Then the attacker can find out the password P_i of the user U_i as $P_i = V_i \oplus A_i$. Now the attacker has the smart card of user U_i, knows the identity ID_i and password P_i corresponding to the user U_i and hence can login on to the server S.

B. Impersonation attack

In login phase of Yoon-Ryu-Yoo's scheme [9], B_i should be equal to the stored value of V_i in the smart card. This means that the attacker needs not to know password P_i corresponding to the user U_i to compute C_1, if the attacker had known V_i from the stolen smart card attack.

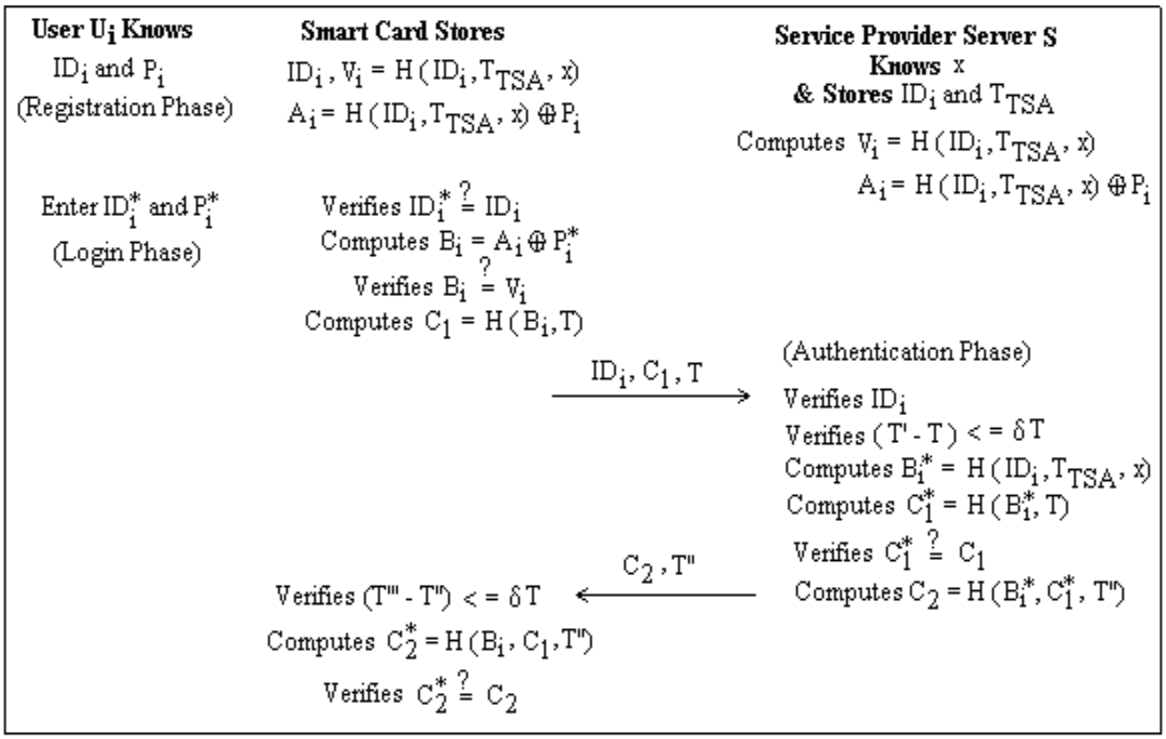

Fig. 1 Yoon-Ryu-Yoo's scheme

Now the attacker can easily go through the steps in the login phase to forge a valid login request message as {ID_i, C_1, T}, where T is a current timestamp and C_1 = H (B_i, T) = H (V_i, T). Therefore, the attacker can successfully make a valid login request and impersonate as a legitimate user U_i.

C. Parallel sessions attack

An attacker can masquerade as a legitimate user U_i by creating a valid login message from the eavesdropped communication between the client and the server without knowing the user's password. He can intercept the login request message (ID_i, C_1, T) from the user U_i to the server S. Then he starts a new session with the server S by sending a login request by replaying the login request message (ID_i, C_1, T) within the valid time frame window. After receiving the login request, the server S checks the validity of ID_i and the validity of timestamp T by checking (T' – T) <= δT, where T' denotes the server's current timestamp. The server S computes B_i^* = H (ID_i, T_{TSA}, x), C_1^* = H (B_i^*, T) and compares C_1^* with received value of C_1 to check the legitimacy of the user U_i. This equivalency authenticates the masquerading user.

D. Man-in-the-middle attack

In this type of attack, the attacker can intercept the messages sent between the client and the server and replay these intercepted messages within the valid time frame window. The attacker can act as a client to the server or vice-versa with recorded messages. He can intercept the login request (ID_i, C_1, T) from the user U_i to the server S. Then he starts a new session with the server S by sending a login request by replaying the login request message (ID_i, C_1, T) within the valid time frame window. After receiving the login request, the server S check the format of ID_i and the validity of timestamp T by checking (T' – T) <= δT, where T' denotes the server's current timestamp. The server S computes B_i^* = H (ID_i, T_{TSA}, x), C_1^* = H (B_i^*, T) and compares C_1^* with the received value of C_1 to check the legitimacy of the user U_i. This equivalency authenticates the masquerading user. The attacker can also intercept the response message (C_2, T''), which is sent by the server S to the user U_i. Using this message, the attacker can masquerade as legitimate server S to the legitimate user U_i by replaying this message with in the valid time frame window. Now the attacker can act as middle man and masquerade as legitimate client to legitimate server S and vice-versa.

IV. PROPOSED PROTOCOL

In this section, we describe a new remote user authentication scheme which resolves the above security flaws of Yoon-Ryu-Yoo's [9] scheme. Fig. 2 shows the

entire protocol structure of the new authentication scheme. Legitimate client C can easily login on to the service provider server using his smart card, identity and password. The notations used in this section are listed in Table 1.

1) Registration Phase: The user U_i has to submit his identity ID_i and password P_i to the server S to become a legitimate client via a secure communication channel. The server S computes some security parameters and stores them on the smart card of the user U_i. Then the server S issues the smart card to user U_i.

2) Login Phase: A user U_i inserts his smart card into a card reader to login on to server S and submits his identity ID_i and password P_i. Smart card verifies authenticity of the user U_i and sends user U_i's verification information to the destination server S.

3) Authentication Phase: Service provider server S verifies the authenticity of the user U_i. Once the user U_i authenticates itself to the server S then the user U_i and the server S agree on the common session key.
4) Password change Phase: The user U_i has to authenticate itself to smart card before requesting the password change.

A. Registration phase

A user U_i has to submit his identity ID_i and password P_i to the server S via a secure communication
channel to register itself to the server S.

Step 1: $U_i \rightarrow S$: ID_i, P_i

The server S computes the security parameters $V_i = H(ID_i \mid T_{TSA} \mid x) \oplus H(P_i)$, $A_i = H(ID_i \mid P_i) \oplus P_i$, $B_i = H(P_i) \oplus H(T_{TSA})$ and issues the smart card containing security parameters (V_i, A_i, B_i, $H(\ )$) to the user U_i through a secure communication channel.

Step 2: $S \rightarrow U_i$: Smart card

B. Login phase

A user U_i inserts his smart card into a card reader to login on to the server S and submits his identity ID_i^* and password P_i^*. The smart card computes $A_i^* = H(ID_i^* \mid P_i^*) \oplus P_i^*$ and compares it with the stored value of A_i in its memory to verify the legitimacy of the user U_i.

Step 1: Smart card checks $A_i^* ?= A_i$

After verification, smart card computes $C_1 = V_i \oplus H(P_i)$ and $C_2 = H(C_1 \mid T)$, where T is current date and time of input device. Then smart card sends the login request message (ID_i^*, C_2, T) to the service provider server S.

Step 2: Smart card $\rightarrow$ S: ID_i^*, C_2, T

The user U_i's smart card extracts the value of $H(T_{TSA})$ as $H(T_{TSA}) = B_i \oplus H(P_i)$, which is used by the user U_i's smart card for the computation of the agreed session key between the user U_i and the server S.

C Authentication phase

After receiving the login request from the user U_i, the service provider server S verifies the received ID_i^* with stored value of ID_i in its database. The server S checks the validity of timestamp T by checking $(T' - T) <= \delta T$, where T' is current date and time of the server S and δT is permissible time interval for a transmission delay. The server S extracts the value of T_{TSA} corresponding to the client's identity ID_i. Then server S computes $C_1^* = H(ID_i \mid T_{TSA} \mid x)$, $C_2^* = H(C_1^* \mid T)$ and compares C_2^* with the received value of C_2.

Step 1: Server S checks $C_2^* ?= C_2$

This equivalency authenticates the legitimacy of the user U_i and the login request is accepted else the connection is interrupted. Finally, the user U_i and the server S agree on the common session key as $SK = H(C_1 \mid H(T_{TSA}) \mid T)$. Afterwards, all the subsequent messages between the user U_i and server S are encrypted with this session key.

D. Password change phase

The user U_i can change his password without the help of the server S. The user U_i inserts his smart card into a card reader and enters his identity ID_i^* and password P_i^* corresponding to his smart card. The smart card computes $A_i^* = H(ID_i^* \mid P_i^*) \oplus P_i^*$ and compares it with the stored value of A_i in its memory to verify the legitimacy of the user U_i. Once the authenticity of cardholder is verified then the user U_i can instruct the smart card to change his password. Afterwards, the smart card asks the cardholder to resubmit a new password P_i^{new} and then $V_i = H(ID_i \mid T_{TSA} \mid x) \oplus H(P_i)$, $A_i = H(ID_i \mid P_i) \oplus P_i$ and $B_i = H(P_i) \oplus H(T_{TSA})$ stored in the smart card can be updated with $V_i^{new} = H(ID_i \mid T_{TSA} \mid x) \oplus H(P_i^{new})$, $A_i^{new} = H(ID_i \mid P_i^{new}) \oplus P_i^{new}$ and $B_i^{new} = H(P_i^{new}) \oplus H(T_{TSA})$.

V. SECURITY ANALYSIS

Smart card is a memory card that uses an embedded micro-processor from smart card reader machine to perform the required operations specified in the protocol. Kocher et al. [14] and Messerges et al. [15] pointed out that all existing smart cards can not prevent the information stored in them from being extracted such as by monitoring their power consumption. Some other reverse engineering techniques are also available

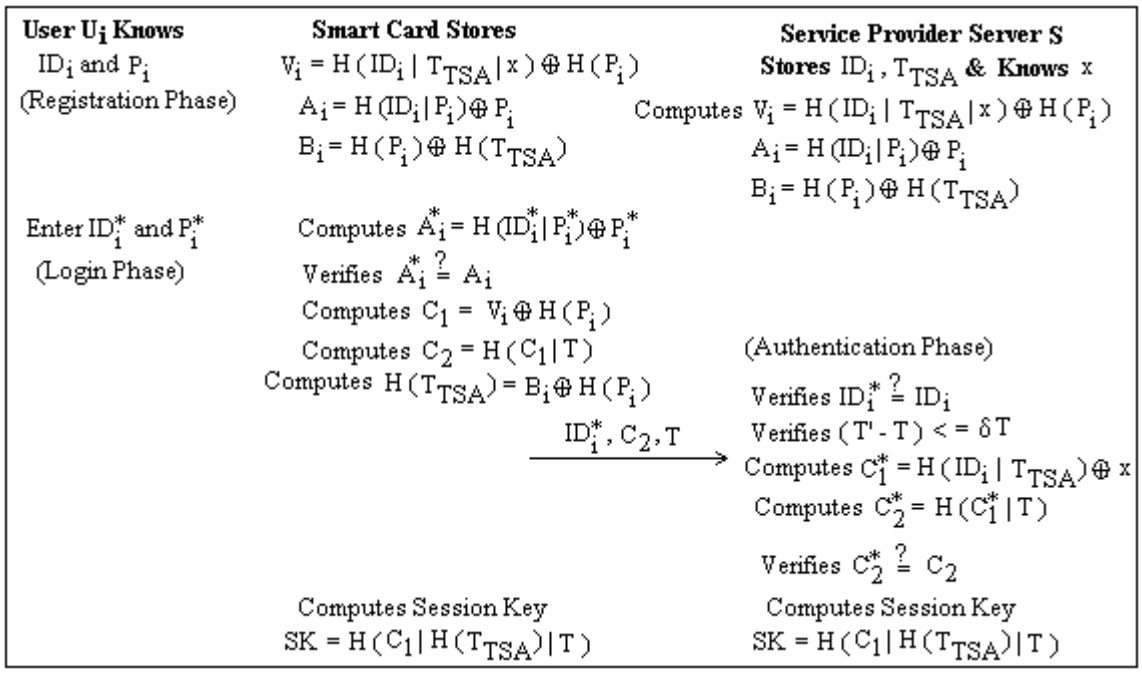

Fig. 2 Proposed improvements in Yoon-Ryu-Yoo's scheme

for extracting information from smart cards. This means that a good password authentication scheme should provide protection from different feasible attacks.

A. Stolen smart card attack:

In case a user's smart card is stolen by the attacker, he can extract the information stored in its memory. The attacker can extract $V_i = H(ID_i \mid T_{TSA} \mid x) \oplus H(P_i)$, $A_i = H(ID_i \mid P_i) \oplus P_i$ and $B_i = H(P_i) \oplus H(T_{TSA})$ from the memory of smart card. Even after gathering this information, the attacker has to guess ID_i and P_i correctly at the same time. It is not possible to guess out two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against stolen smart card attack.

B. Impersonation attack:

In this type of attack, the attacker impersonates as a legitimate client and forges the authentication messages using the information obtained from the authentication protocol. The attacker can attempt to modify a login request message (ID_i^*, C_2, T) into (ID_i^*, C_2^*, T^*) so as to succeed in the authentication phase, where T^* is the attacker's current date and time. However, such a modification will fail in Step 1 of the authentication phase because the attacker has no way of obtaining the value of $C_1 = H(ID_i^* \mid T_{TSA} \mid x)$ to compute the valid parameter C_2^*. Moreover, the attacker can not compute the agreed session key $SK = H(C_1 \mid H(T_{TSA}) \mid T)$ between the user U_i and the

server S. Therefore, the proposed protocol is secure against impersonation attack.

C. Parallel session attack:

In this type of attack, the attacker first listens to communication between the client and the server. After that, he initiates a parallel session to imitate legitimate user to login on to the server by resending the captured messages transmitted between the client and the server with in the valid time frame window. He can masquerade as legitimate user U_i by replaying a login request message (ID_i^*, C_2, T) with in the valid time frame window. The attacker can not compute the agreed session key $SK = H(C_1 \mid H(T_{TSA}) \mid T)$ between the user U_i and the server S because the attacker does not know the values of C_1 and $H(T_{TSA})$. Therefore, the proposed protocol is secure against parallel session attack.

D. Man-in-the-middle attack:

In this type of attack, the attacker intercepts the messages sent between the client and the server and replay these intercepted messages with in the valid time frame window. The attacker can act as the client to the server or vice-versa with recorded messages. In the proposed protocol, the attacker can intercept the login request message (ID_i^*, C_2, T) from the user U_i to the server S. Then he starts a new session with the server S by sending a login request by replaying the login request message (ID_i^*, C_2, T) with in the valid time frame window. The attacker can authenticate itself to the server S but can not compute the agreed session key $SK = H(C_1 \mid H(T_{TSA}) \mid T)$ between the user U_i and the server S because the attacker does not know the values of C_1 and $H(T_{TSA})$. Therefore, the proposed protocol is secure against man-in-the-middle attack.

E. Replay attack:

In this type of attack, the attacker first listen to communication between the client and the server and then tries to imitate user to login on to the server by resending the captured messages transmitted between the client and

the server. Replaying a login request message (ID_i^*, C_2, T) of one session into another session is useless because the client's smart card uses current time stamp value T in each new session, which makes all the messages dynamic and valid for small interval of time. Old messages can not be replayed successfully in any other session and hence the proposed protocol is secure against message replay attack.

F. Leak of verifier attack:

In this type of attack, the attacker may be able to steal verification table from the server. If the attacker steals the verification table from the server, he can use the stolen verifiers to impersonate a participant of the authentication protocol. In the proposed protocol, the service provider server S knows secret x and only stores T_{TSA} corresponding to user identity ID_i in its database. The attacker does not have any technique to find out the value of x. In case verifier is stolen by breaking into smart card database, the attacker does not have sufficient information to calculate user's identity ID_i and password P_i. Therefore, the proposed protocol is secure against leak of verifier attack.

G. Server spoofing attack:

In server spoofing attack, the attacker can manipulate the sensitive data of legitimate users via setting up fake servers. In the proposed protocol, malicious server can not compute the session key $SK = H(C_1 \mid H(T_{TSA}) \mid T)$ between the user U_i and server S because the malicious server does not know the value of C_1 and $H(T_{TSA})$. Moreover, the session key is different for the same user in different login sessions. Therefore, the proposed protocol is secure against server spoofing attack.

H. Malicious user attack:

A malicious privileged user U_i having his own smart card can gather information like $V_i = H(ID_i \mid T_{TSA} \mid x) \oplus H(P_i)$, $A_i = H(ID_i \mid P_i) \oplus P_i$ and $B_i = H(P_i) \oplus H(T_{TSA})$ from the memory of smart card. This malicious user can not compute the value of x or T_{TSA} from these parameters even if he knows the values of ID_i and P_i. Moreover, the value of T_{TSA} is unique corresponding to different users. Also the malicious user can not generate smart card specific value of $C_2 = H(C_1 \mid T)$ to masquerade as other legitimate user to service provider server S because the value of C_1 is smart card specific and depends upon the values of ID_i, x and T_{TSA}. Therefore, the proposed protocol is secure against malicious user attack.

I. Message modification or insertion attack:

In this type of attack, the attacker modifies or inserts some messages on the communication channel with the hope of discovering the client's password or gaining unauthorized access. Modifying or inserting messages in the proposed protocol can only cause authentication between the client and the server to fail but can not allow the attacker to gain any information about the client's identity ID_i and password P_i or gain unauthorized access. Therefore, the proposed protocol is secure against message modification or insertion attack.

J. Online dictionary attack:

In this type of attack, the attacker pretends to be the legitimate client and attempts to login on to the server by guessing different words as password from a dictionary. In the proposed protocol, the attacker has to get the valid smart card of the user U_i and then has to guess the identity ID_i and password P_i. Even after getting the valid smart card by any mean, the attacker gets a very few chances (maximum 3) to guess the identity ID_i and password P_i because smart card gets locked after certain number of unsuccessful attempts. Moreover, it is not possible to guess identity ID_i and password P_i correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against online dictionary attack.

K. Offline dictionary attack:

In offline dictionary attack, the attacker can record messages and attempt to guess the user's identity ID_i and password P_i from recorded messages. The attacker first tries to obtains the user U_i's verification information T, $C_2 = H((H(ID_i \mid T_{TSA} \mid x) \mid T)$ and then tries to guess the ID_i, x and T_{TSA} by offline guessing. Even after gathering this information, the attacker has to guess all three parameters ID_i, x and T_{TSA} correctly at the same time. It is not possible to guess all three parameters correctly at the same time in real polynomial time. In another option, the attacker requires smart card of user U_i and then has to guess the identity ID_i and password P_i correctly at the same time. It is not possible to guess two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against offline dictionary attack.

Table 2
Cost comparison among related smart card based authentication schemes

	Proposed Protocol	Kim-Chung [13]	Hsiang-Shih [11]	Yoon-Ryu-Yoo [9]	Chein et al. [8]	Hwang-Lee-Tang [4]
E1	384 bits	384 bits	384 bits	384 bits	128 bits	256 bits
E2	3*128 bits	6 *128 bits	5 *128 bits	5 *128 bits	5 * 128 bits	3*128 bits
E3	$4T_H + 3T_X$	$4T_H + 6T_X$	$4T_H + 4T_X$	$1T_H + 1T_X$	$1T_H + 2T_X$	$2T_H + 2T_X$
E4	$4T_H + 3T_X$	$4T_H + 6T_X$	$4T_H + 4T_X$	$2T_H + 1T_X$	$2T_H + 3T_X$	$2T_H + 2T_X$
E5	$4T_H + 1T_X$	$4T_H + 5T_X$	$4T_H + 3T_X$	$3T_H + 0T_X$	$3T_H + 3T_X$	$2T_H + 2T_X$

E1: Memory needed in the smart card.
E2: Communication cost of the authentication.
E3: Computation cost of the registration.
E4: Computation cost of the user.
E5: Computation cost of the service provider server.

Table 3
Functionality comparison among related smart card based authentication schemes

	Proposed Protocol	Kim-Chung [13]	Hsiang-Shih [11]	Yoon-Ryu-Yoo [9]	Chein et al. [8]	Hwang-Lee-Tang [4]
Stolen smart card attack	No	Yes	No	Yes	Yes	Yes
Impersonation attack	No	Yes	Yes	Yes	Yes	No
Parallel session attack	No	Yes	Yes	Yes	Yes	Yes
Man-in-the-middle attack	No	Yes	Yes	Yes	Yes	Yes
Session key agreement	Yes	No	No	No	No	No

VI. COST AND FUNCTIONALITY ANALYSIS

An efficient authentication scheme must take communication and computation cost into consideration during user's authentication. The cost and functionality comparison of the proposed scheme with the relevant smart card based authentication schemes is summarized in Table 2 and Table 3. Assume that the identity ID_i, password P_i and timestamp value are all 128-bit long. Moreover, we assume that the output of the secure one-way hash function is 128-bit. Let T_H and T_X denote the time complexity for hash function and XOR operation respectively. Typically, time complexity associated with these operations can be roughly expressed as $T_H >> T_X$. In the proposed protocol, the parameters stored in the smart card are V_i, A_i, B_i and the memory needed in the smart card (E1) is 384 (= 3*128) bits. The communication cost of authentication (E2) includes the capacity of transmitting message involved in the authentication scheme. The capacity of transmitting message $\{ID_i, C_2, T\}$ is 384 (= 3*128) bits. The computation cost of registration (E3) is the total time of all operations executed in the registration phase. The computation cost of registration is $4T_H + 3T_X$. The computation cost of the user (E4) and the service provider server (E5) is the time spent by the user and the service provider server during the process of authentication. Therefore, both the computation cost of the user and that of the service provider server are $4T_H + 3T_X$ and $4T_H + 1T_X$ respectively. The proposed scheme requires less computation than that of latest schemes proposed by Kim-Chung [13] and Hsiang-Shih [11] and is secure against different possible attacks launched by the attacker.

VII. CONCLUSION

Corporate network and e-commerce applications require secure and practical remote user authentication solutions. Smart card based password authentication is one of the most convenient ways to provide multi-factor authentication for the communication between a client and a server. In this paper, we present a cryptanalysis of Yoon-Ryu-Yoo's scheme by showing that their scheme is vulnerable to stolen smart card attack, impersonation attack, parallel session attack and man-in-the-middle attack. An improvement to Yoon-Ryu-Yoo's scheme is proposed that inherits the merits

of Yoon-Ryu-Yoo's scheme and enhances the security of their scheme. The proposed protocol is simple and fast if the user possesses valid smart card, knows the correct identity and correct password for its authentication. The proposed protocol is practical and efficient because only one-way hash functions and XOR operations are used in its implementation. Security analysis proved that the improved scheme is more secure and practical.

REFERENCES

[1] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, November 1981.

[2] M.S. Hwang and L.H. Li, "A New Remote User Authentication Scheme using Smart Cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, February 2000.

[3] H.M. Sun, "An Efficient Remote User Authentication Scheme using Smart Cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958-961, November 2000.

[4] M.S. Hwang, C.C. Lee and Y.L. Tang, "A Simple Remote User Authentication," *Mathematical and Computer Modelling*, vol. 36, pp. 103-107, October 2002.

[5] T.C. Wu, "Remote Login Authentication Scheme Based on a Geometric Approach," *Computer Communications*, vol. 18, no. 12, pp. 959-963, December 1995.

[6] J.K. Jan and Y.Y. Chen, "'Paramita Wisdom' Password Authentication Scheme without Verification Tables," *The Journal of Systems and Software*, vol. 42, no. 1, pp. 45-57, July 1998.

[7] W.H. Yang and S.P. Shieh, "Password Authentication Schemes with Smart Card," *Computers & Security*, vol. 18, no. 8, pp. 727-733, February 1999.

[8] H.Y. Chien, J.K. Jan and Y.M. Tseng, "An Efficient and Practical Solution to Remote Authentication: Smart Card," *Computers & Security*, vol. 21, no. 4, pp. 372-375, August 2002.

[9] E.J. Yoon, E.K. Ryu and K.Y. Yoo, "An Improvement of Hwang-Lee-Tang's Simple Remote User Authentication Scheme," *Computers & Security*, vol. 24, no. 1, pp. 50-56, February 2005.

[10] E.J. Yoon, E.K. Ryu and K.Y. Yoo, "Further Improvement of an Efficient Password Based Remote User Authentication Scheme using Smart Cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 612-614, August 2004.

[11] H.C. Hsiang and W.K. Shih, "Weaknesses and Improvements of the Yoon-Ryu-Yoo Remote User Authentication Scheme using Smart Cards," *Computer Communications*, vol. 32, no. 4, pp. 649-652, March 2009.

[12] E.J. Yoon and K.Y. Yoo, "More Efficient and Secure Remote User Authentication Scheme using Smart Cards," *Proc. of 11th International Conference on Parallel and Distributed System*, vol. 2, pp. 73-77, July 2005.

[13] S.K. Kim and M.G. Chung, "More Secure Remote User Authentication Scheme," *Computer Communications*, vol. 32, no. 6, pp. 1018-1021, April 2009.

[14] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," *Proc. CRYPTO 99, Springer-Verlag*, pp. 388-397, August 1999.

[15] T.S. Messerges, E.A. Dabbish and R.H. Sloan, "Examining Smart-Card Security under the Threat of Power Analysis Attacks," *IEEE Transactions on Computers* , vol. 51, no. 5, pp. 541-552, May 2002.