ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2010, Vol. 39, No. 3

IMPROVED THREE PARTY EKE PROTOCOL

R.Padmavathy

National Institute of Technology, Warangal
Andra Pradesh, India
e-mail: r_padma3@rediffmail.com

Abstract. The key exchange protocol using passwords achieved great attention due to its simplicity and efficiency. On the other hand, the protocol should resist all types of password guessing attacks, since the password is of low entropy. Recently, Chang and Chang proposed a novel three party simple key exchange protocol. They claimed the protocol was secure, efficient and practical. Overriding their claims Yoon and Yoo presented an Undetectable online password guessing attack on the above protocol. Recently, a password key exchange protocol PSRJ was proposed and claimed to be in-vulnerable to Undetectable online password guessing attack proposed by Yoon and Yoo. This paper presents an Undetectable on-line password guessing attack on PSRJ protocol. Additionally, to overcome the attack, an enhancement over the existing protocol with reduced modular exponentiation operations is proposed.

Keywords: Chang-Chang password key exchange protocol, Undetectable online password guessing attack, PSRJ protocol.

1. Introduction

The key exchange protocol is one of the most elegant ways of establishing secure communication between pair of users by using a session key. The session key, which is exchanged between two users, assures the secure communication for later sessions. The first practical key exchange protocol was proposed by Diffie and Hellman [1]. Since the introduction of key exchange protocol by Diffie and Hellman, various versions and improvements in key exchange protocol have been developed. In the line of key exchange protocol development, password based key exchange mechanism achieved attention due to its simplicity and wide range of applicability, as it requires the users to remember the password. Even though the protocol is simple and efficient, according to Ding and Horster [2], it should not be vulnerable to any type of off- line, undetectable or detectable on-line password guessing attacks, since the passwords are of low entropy.

In general, the password guessing attacks can be divided into three classes and they are listed below:

- **Detectable on-line password guessing attacks:** An attacker attempts to use a guessed Password in an on-line transaction. He/She verifies the correctness of his/her guess using the response from server. A failed guess can be detected and logged by the server.
- **Undetectable on-line password guessing attacks:** Similar to Detectable on-line password guessing attack, an attacker tries to verify a password guess in an on-line transaction. However, a failed guess cannot be detected and logged by server, as server is not able to distinguish an honest request from a malicious one.
- **Off-line password guessing attacks**: An attacker guesses a password and verifies his/her guess online. No participation of server is required, so the server does not notice the attack.

Since the first proposal of Bellovin and Merrit (PAKE) [3], many efficient key exchange protocols based on password have been developed. Recently these two Party key exchange protocols were extended to three party, in which, the two parties initially communicate the passwords with the trusted server securely. Later the server authenticates the clients when they want to agree upon a session key. The 3-party protocol is introduced by Steiner et al [4]. Subsequently Ding and Hoster published on-line and offline guessing attacks on Stener's protocol [2]. Later Lin et al. proposed two versions of improved three party protocol [5], one with server's public key and another without.

Recently Chang and Chang [6] proposed a novel three party encrypted key exchange protocol without server public key and claimed the protocol is secure, efficient and practical. Unlike their claims, Yoon and Yoo [7] pointed out an Undetectable on-line password guessing attack on their protocol, in which one party is able to know the other party's password and furthermore they presented an improved version of it to avoid the above attack. Lo and Yeh [8] pointed out undetectable password guessing attack on Yoon and Yoo protocol and proposed an enhanced protocol. Most recently, an enhanced protocol (PSRJ protocol) was proposed without XOR operation [9].

But the enhanced protocol (PSRJ protocol) falls to Undetectable on-line password guessing attack, if client 'B' intercepts the message coming from client 'A'. To eliminate the undetectable on-line password guessing attack, an extension is done on the existing protocol.

The paper is organized as follows: Section 2 briefly reviews the PSRJ protocol. Section 3 shows the undetectable on-line password guessing attack. Section 4 describes the proposed protocol. Section 5 discusses the security and efficiency analyses and the concluding remarks are made in Section 6.

2. Review of PSRJ protocol

This section briefly reviews the enhanced protocol [9]. The notations used in this protocol are listed below:

- A,B : two communication parties
- S: the trusted server
- IDA, IDB, IDS: the identities of A,B and S, respectively
- pwA, pwB: the passwords securely shared by A with S and B
- EPWA (.), EPWB (.): a symmetric encryption scheme with a password PWA and password PWB respectively.
- r_A, r_B: the random numbers chosen by A and B, respectively
- p: a large prime
- g: a generator of order p – 1
- RA, RB, RS: the random exponents chosen by A, B and S, respectively
- N_A,N_B: N_A=g^{RA}(mod p)and N_B=g^{RB}(mod p)
- F_S(.): the one-way trapdoor hash function(TDF) where only S knows the trapdoor
- f_K(.): the pseudo-random hash function (PRF) indexed by a key K
- KAS, KBS: a one time strong keys shared by A with S and B with S respectively

The detailed procedures of the protocol can be described as follows.

1. A→S: ID_A, ID_B, ID_S, $E_{pwA}(N_A)$, $F_S(r_A)$, $f_{KAS}(N_A)$,
   B→S: ID_A,ID_B,ID_S, $E_{pwB}(N_B)$, $F_S(r_B)$, $f_{KBS}(N_B)$.

Client A generates two random numbers RA and r_A, and calculates $E_{pwA}(N_A)$, $F_S(r_A)$ and $f_{KAS}(N_A)$, where N_A=g^{RA} (mod p) and K_{AS}=N_A^{rA} (mod p). Next, A sends these three messages to S via his/her own private communication channel.

Meanwhile, client B calculates N_B=g^{RB} (mod p), K_{BS}=N_B^{rB} (mod p), $E_{PWB}(N_B)$, $F_S(r_B)$ and $f_{KBS}(N_B)$ with two newly generated random numbers RB and r_B. Then, B transmits $E_{pwB}(N_B)$, $F_S(r_B)$ and $f_{KBS}(N_B)$ to S via his/her own private communication channel.

2. S → A: N_B^{RS}, $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$,
   S→ B: N_A^{RS}, $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$.

Once receiving the message sent from A and B , S first utilizes pw_A, pw_B and decrypts $E_{pwA}(N_A)$, $E_{pwB}(N_B)$ and gets N_A, N_B, then it utilizes a trapdoor to obtain r_A and r_B from $F_S(r_A)$ and $F_S(r_B)$, verifies whether computed value $f_{KAS}(N_A)$(or $f_{KBS}(N_B)$) and received value $f_{KAS}(N_A)$ (or $f_{KBS}(N_B)$) are identical or not. If this verification holds, S continues the residual procedures of this protocol. Otherwise, S terminates this protocol at current session. Next, S computes N_B^{RS}, N_A^{RS} and corresponding hashed credential $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ and $f_{KAS}(ID_A, ID_B, K_{BS}, N_A^{RS})$. Finally, S sends these messages to A and B simultaneously.

3. B → A: $f_K(ID_B, K)$,
4. A → B: $f_K(ID_A, K)$.

Upon obtaining the transmitted messages sent from S, B first verifies $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$ to authenticate S.

If this verification is passed, B believes the received N_A^{RS} is valid and then computes the session key K=$(N_A^{RS})^{RB}$ (mod p) and $f_K(ID_B, K)$. Otherwise, B terminates this protocol. Finally, B sends the $f_K(ID_B, K)$ to A. Note that $f_K(ID_B, K)$ will be used by client A to verify the legality of client B and the established session key K. At the same time, A verifies $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ to authenticate S. If this verification does not hold, A terminates this protocol. Otherwise, A computes the session key K=$(N_B^{RS})^{RA}$ (mod p) and $f_K(ID_A, K)$. Finally, A sends the $f_K(ID_A, K)$ to B.

After A and B successfully examine the validation of the incoming messages $f_K(ID_B, K)$ and $f_K(ID_A, K)$, both of them can ensure that they actually share the secret session key K=$(N_B^{RS})^{RA}$ (mod p)=$(N_A^{RS})^{RB}$(mod p) at present. Otherwise, the protocol will be terminated**.** Figure 1 illustrates PSRJ protocol.

3. Undetectable on-line password guessing attack on PSRJ protocol

1. A→S:ID_A, ID_B, ID_S, $E_{pwA}(N_A)$, $F_S(r_A)$, $f_{KAS}(N_A)$.

Client A generates two random numbers RA and r_A, and calculates $E_{pwA}(N_A)$, $F_S(r_A)$ and $f_{KAS}(N_A)$, where N_A=g^{RA} (mod p) and K_{AS}=N_A^{rA} (mod p). Next, A sends these three messages to S via his/her own private communication channel.

2. B→ S:ID_A, ID_B,ID_S, $E_{pwB}(N_B)$, $F_S(r_B)$, $f_{KBS}(N_B)$.

Client 'B' intercepts this message i.e. ID_A, ID_B, ID_S, $E_{pwA}(N_A)$, $F_S(r_A)$, $f_{KAS}(N_A)$.

Now, he/she will guess a password pwA*, decrypts $E_{pwA*}(N_A)$ and gets N_A*. Let N_A*= N_B. Client B generates a random number r_B, and calculates $E_{pwB}(N_B)$, $F_S(r_B)$ and $f_{KBS}(N_B)$, where $K_{BS}=N_B^{rB}$ (mod p).

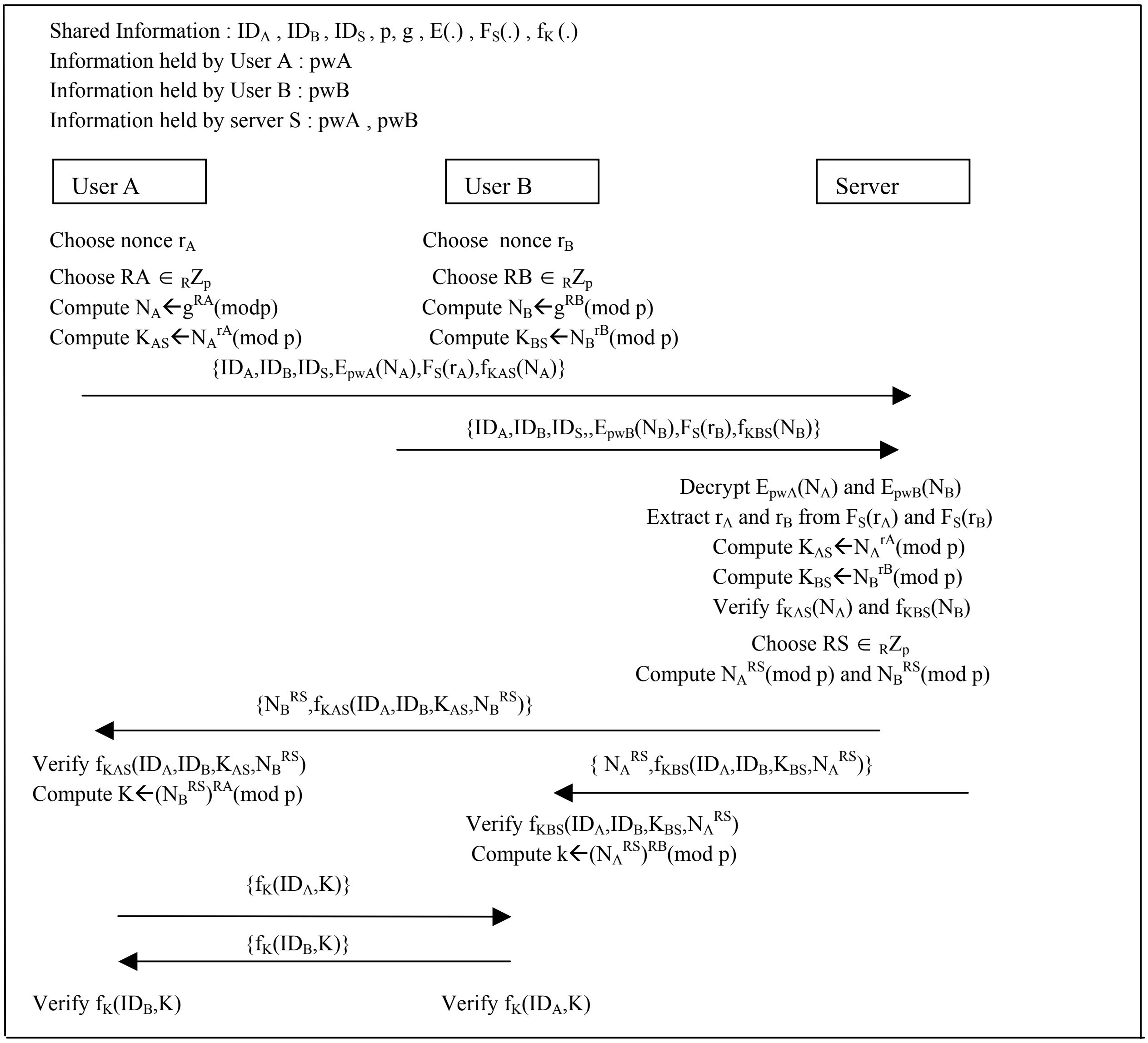

Shared Information : ID_A , ID_B , ID_S , p, g , E(.) , F_S(.) , f_K (.)
Information held by User A : pwA
Information held by User B : pwB
Information held by server S : pwA , pwB
User A
User B
Server
Choose nonce r_A
Choose nonce r_B
Choose RA $\in {}_R Z_p$
Choose RB $\in {}_R Z_p$
Compute $N_A \leftarrow g^{RA}$(modp)
Compute $N_B \leftarrow g^{RB}$(mod p)
Compute $K_{AS} \leftarrow N_A^{rA}$(mod p)
Compute $K_{BS} \leftarrow N_B^{rB}$(mod p)
$\{ID_A,ID_B,ID_S,E_{pwA}(N_A),F_S(r_A),f_{KAS}(N_A)\}$
$\{ID_A,ID_B,ID_{S,},E_{pwB}(N_B),F_S(r_B),f_{KBS}(N_B)\}$
Decrypt $E_{pwA}(N_A)$ and $E_{pwB}(N_B)$
Extract r_A and r_B from $F_S(r_A)$ and $F_S(r_B)$
Compute $K_{AS} \leftarrow N_A^{rA}$(mod p)
Compute $K_{BS} \leftarrow N_B^{rB}$(mod p)
Verify $f_{KAS}(N_A)$ and $f_{KBS}(N_B)$
Choose RS $\in {}_R Z_p$
Compute N_A^{RS}(mod p) and N_B^{RS}(mod p)
$\{N_B^{RS},f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})\}$
Verify $f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})$
Compute $K \leftarrow (N_B^{RS})^{RA}$(mod p)
$\{ N_A^{RS},f_{KBS}(ID_A,ID_B,K_{BS},N_A^{RS})\}$
Verify $f_{KBS}(ID_A,ID_B,K_{BS},N_A^{RS})$
Compute $k \leftarrow (N_A^{RS})^{RB}$(mod p)
$\{f_K(ID_A,K)\}$
$\{f_K(ID_B,K)\}$
Verify $f_K(ID_B,K)$
Verify $f_K(ID_A,K)$

Figure 1. The PSRJ protocol

Then, B transmits $E_{pwB}(N_B)$, $F_S(r_B)$ and $f_{KBS}(N_B)$ to S via his/her own private communication channel.

3. S → A: N_B^{RS}, $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$
   S→ B: N_A^{RS}, $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$.

Once receiving the message sent from A and B , S first utilizes pw_A, pw_B and decrypts $E_{PWA}(N_A)$, $E_{PWB}(N_B)$ and gets N_A, N_B, then it utilizes a trapdoor to obtain r_A and r_B from $F_S(r_A)$ and $F_S(r_B)$, verifies whether computed value $f_{KAS}(N_A)$(or $f_{KBS}(N_B)$) and received value $f_{KAS}(N_A)$ (or $f_{KBS}(N_B)$) are identical or not. If this verification holds, S continues the residual procedures of this protocol. Otherwise, S terminates this protocol at current session. Next, S computes N_B^{RS}, N_A^{RS}, and corresponding hashed credential $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ and $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$. Finally, S sends $\{N_B^{RS}$, $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ } to A and $\{N_A^{RS}$, $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})\}$ to B simultaneously.

'B' intercepts the message N_B^{RS}, $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ again.

It verifies whether N_A^{RS} is equal to N_B^{RS}. If both are equal then the guessed password is correct.

Figure 2 illustrates the Undetectable on-line password guessing attack on the enhanced protocol.

4. The proposed protocol

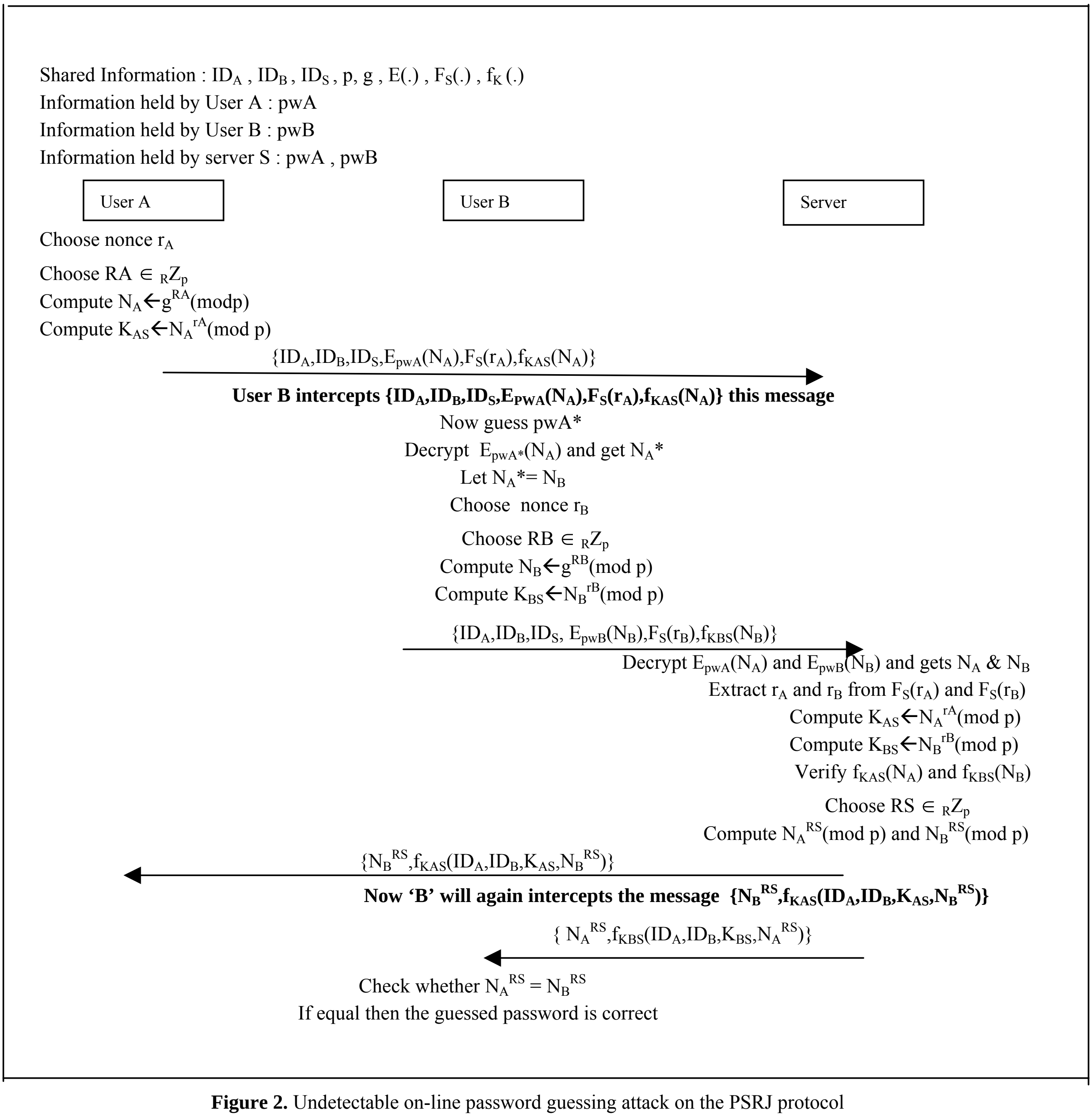
To overcome the Undetectable on-line password guessing attack, an extension is made on the PSRJ protocol. The detailed procedures of the proposed protocol can be described as follows.

1. A→S: ID_A, ID_B, ID_S, $E_{pwA}(K_{As} \oplus N_A)$, $F_S(N_A \oplus ID_A)$, $f_{KAS}(N_A)$.

B→.S: ID_A,ID_B,ID_S, $E_{pwB}(K_{Bs}\oplus N_B)$, $F_S(N_B\oplus ID_B)$, $f_{KBS}(N_B)$.

Client A generates two random numbers R_A and r_A, and calculates $E_{pwA}(K_{As}\oplus N_A)$, $F_S(N_A\oplus ID_A)$ and $f_{KAS}(N_A)$, where $N_A=g^{RA}$ (mod p) and $K_{AS}=N_A^{rA}$ (mod p). Next, A sends these three messages to S via his/her own private communication channel.

Meanwhile, client B calculates $N_B=g^{RB}$ (mod p), $K_{BS}=N_B^{rB}$ (mod p), $E_{pwB}(K_{BS}\oplus N_B)$, $F_S(N_B\oplus ID_B)$ and $f_{KBS}(N_B)$ with two newly generated random numbers R_B and r_B. Then, B transmits $E_{pwB}(K_{BS}\oplus N_B)$, $F_S(N_B\oplus ID_B)$ and $f_{KBS}(N_B)$ to S via his/her own private communication channel.



Shared Information : ID_A , ID_B , ID_S , p, g , E(.) , F_S(.) , f_K (.)
Information held by User A : pwA
Information held by User B : pwB
Information held by server S : pwA , pwB

Choose nonce r_A
Choose RA $\in {}_R Z_p$
Compute $N_A \leftarrow g^{RA}$(modp)
Compute $K_{AS} \leftarrow N_A^{rA}$(mod p)

$\{ID_A,ID_B,ID_S,E_{pwA}(N_A),F_S(r_A),f_{KAS}(N_A)\}$

User B intercepts $\{ID_A,ID_B,ID_S,E_{PWA}(N_A),F_S(r_A),f_{KAS}(N_A)\}$ this message

Now guess pwA*
Decrypt $E_{pwA*}(N_A)$ and get N_A*
Let N_A*= N_B
Choose nonce r_B
Choose RB $\in {}_R Z_p$
Compute $N_B \leftarrow g^{RB}$(mod p)
Compute $K_{BS} \leftarrow N_B^{rB}$(mod p)

$\{ID_A,ID_B,ID_S, E_{pwB}(N_B),F_S(r_B),f_{KBS}(N_B)\}$

Decrypt $E_{pwA}(N_A)$ and $E_{pwB}(N_B)$ and gets N_A & N_B
Extract r_A and r_B from $F_S(r_A)$ and $F_S(r_B)$
Compute $K_{AS} \leftarrow N_A^{rA}$(mod p)
Compute $K_{BS} \leftarrow N_B^{rB}$(mod p)
Verify $f_{KAS}(N_A)$ and $f_{KBS}(N_B)$
Choose RS $\in {}_R Z_p$
Compute N_A^{RS}(mod p) and N_B^{RS}(mod p)

$\{N_B^{RS},f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})\}$

Now 'B' will again intercepts the message $\{N_B^{RS},f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})\}$

$\{ N_A^{RS},f_{KBS}(ID_A,ID_B,K_{BS},N_A^{RS})\}$

Check whether $N_A^{RS} = N_B^{RS}$
If equal then the guessed password is correct

Figure 2. Undetectable on-line password guessing attack on the PSRJ protocol

2. S → A: N_B^{RS}, $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$,
   S→ B: N_A^{RS}, $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$.

Once receiving the message sent from A and B , S first utilizes a trapdoor to obtain $N_A\oplus ID_A$ and $N_B\oplus ID_B$ from $F_S(N_A\oplus ID_A)$ and $F_S(N_B\oplus ID_B)$ then retrieves N_A= $N_A\oplus ID_A\oplus ID_A$ and N_B= $N_B\oplus ID_B\oplus ID_B$, respectively. Next it uses the passwords pw_A and pw_B and decrypts $E_{pwA}(K_{As}\oplus N_A)$ and $E_{pwB}(K_{Bs}\oplus N_B)$, respectively, and gets $K_{AS}\oplus N_A$ and $K_{BS}\oplus N_B$. Now, $K_{AS}=K_{AS}\oplus N_A\oplus N_A$ and $K_{BS}=K_{BS}\oplus N_B\oplus N_B$ will be determined. $f_{KAS}(N_A)$ and $f_{KBS}(N_B)$ are computed. S verifies whether computed value $f_{KAS}(N_A)$(or $f_{KBS}(N_B)$) and received value $f_{KAS}(N_A)$ (or $f_{KBS}(N_B)$) are identical or not. If this verification holds, S continues the residual procedures of this protocol. Otherwise, S terminates this protocol at current session. Next, S computes N_B^{RS}, N_A^{RS}, and corresponding hashed credential $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ and $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$. Finally, S sends $\{N_B^{RS}, f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})\}$ to A and $\{N_A^{RS}, f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})\}$ to B simultaneously.

3. B → A: $f_K(ID_B, K)$.
4. A → B: $f_K(ID_A, K)$.

Upon obtaining the transmitted messages sent from S, B first verifies $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$ to authenticate S. If this verification is passed, B believes the received N_A^{RS} is valid and then computes the session key $K=(N_A^{RS})^{RB}$ (mod p) and $f_K(ID_B, K)$. Otherwise, B terminates this protocol. Finally, B sends the $f_K(ID_B, K)$ to A. Note that $f_K(ID_B, K)$ will be used by client A to verify the legality of client B and the established session key K. At the same time, A verifies $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$ to authenticate S. If this verification does not hold, A terminates this protocol. Otherwise, A computes the session key $K=(N_B^{RS})^{RA}$ (mod p) and $f_K(ID_A, K)$. Finally, A sends the $f_K(ID_A, K)$ to B.

After A and B successfully examine the validation of the incoming messages $f_K(ID_B, K)$ and $f_K(ID_A, K)$, both of them can ensure that they actually share the secret session key $K=(N_B^{RS})^{RA}$ (mod p)$=(N_A^{RS})^{RB}$(mod p) at present. Otherwise, the protocol will be terminated. Figure 3 illustrates the proposed protocol.

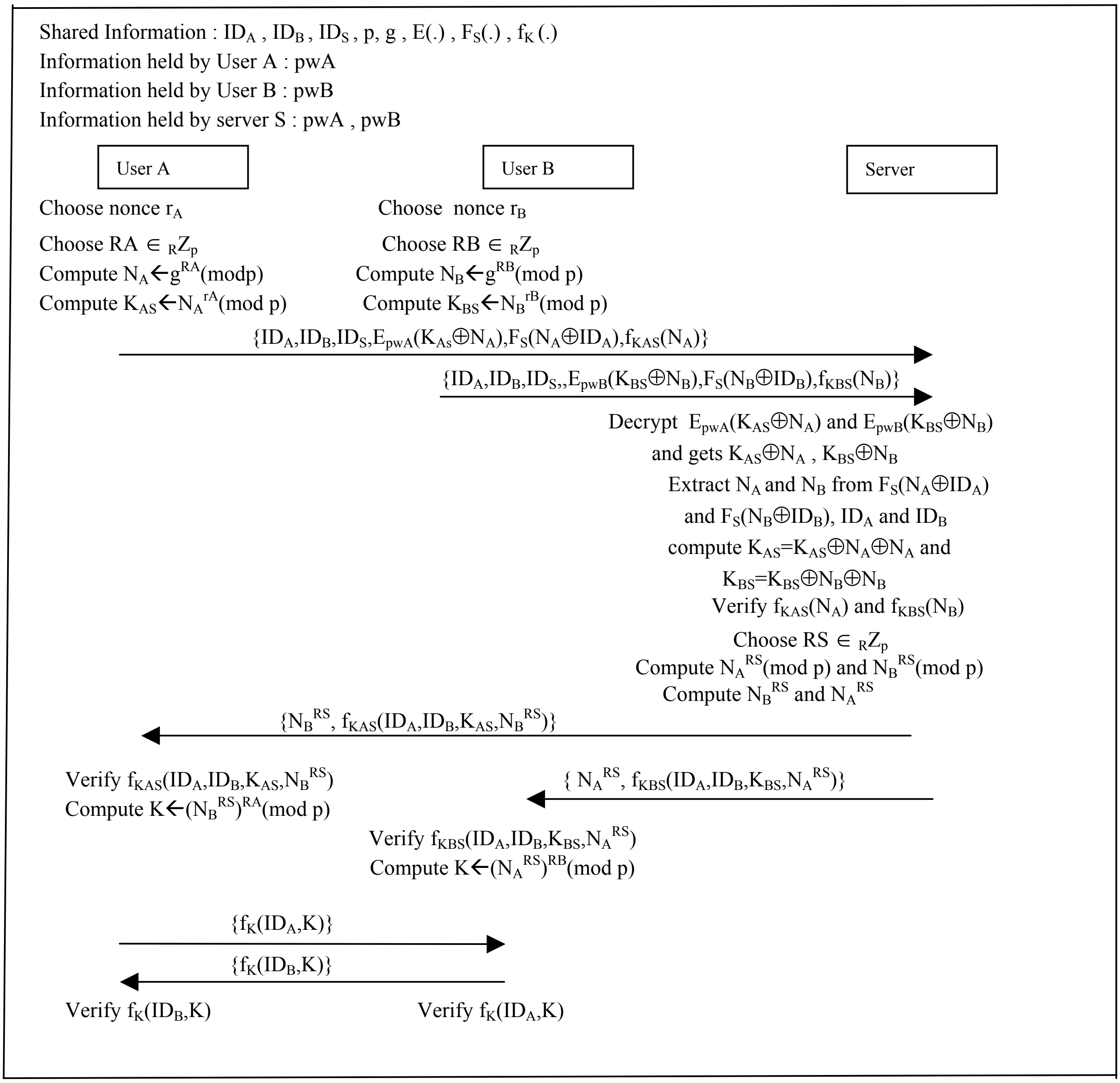

Shared Information : ID_A , ID_B , ID_S , p, g , E(.) , F_S(.) , f_K (.)
Information held by User A : pwA
Information held by User B : pwB
Information held by server S : pwA , pwB
User A
User B
Server
Choose nonce r_A
Choose nonce r_B
Choose RA $\in {}_RZ_p$
Choose RB $\in {}_RZ_p$
Compute $N_A \leftarrow g^{RA}$(modp)
Compute $N_B \leftarrow g^{RB}$(mod p)
Compute $K_{AS} \leftarrow N_A^{rA}$(mod p)
Compute $K_{BS} \leftarrow N_B^{rB}$(mod p)
$\{ID_A,ID_B,ID_S,E_{pwA}(K_{As}\oplus N_A),F_S(N_A\oplus ID_A),f_{KAS}(N_A)\}$
$\{ID_A,ID_B,ID_{S,},E_{pwB}(K_{BS}\oplus N_B),F_S(N_B\oplus ID_B),f_{KBS}(N_B)\}$
Decrypt $E_{pwA}(K_{AS}\oplus N_A)$ and $E_{pwB}(K_{BS}\oplus N_B)$
and gets $K_{AS}\oplus N_A$, $K_{BS}\oplus N_B$
Extract N_A and N_B from $F_S(N_A\oplus ID_A)$
and $F_S(N_B\oplus ID_B)$, ID_A and ID_B
compute $K_{AS}=K_{AS}\oplus N_A\oplus N_A$ and
$K_{BS}=K_{BS}\oplus N_B\oplus N_B$
Verify $f_{KAS}(N_A)$ and $f_{KBS}(N_B)$
Choose RS $\in {}_RZ_p$
Compute N_A^{RS}(mod p) and N_B^{RS}(mod p)
Compute N_B^{RS} and N_A^{RS}
$\{N_B^{RS}, f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})\}$
Verify $f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})$
Compute $K \leftarrow (N_B^{RS})^{RA}$(mod p)
$\{ N_A^{RS}, f_{KBS}(ID_A,ID_B,K_{BS},N_A^{RS})\}$
Verify $f_{KBS}(ID_A,ID_B,K_{BS},N_A^{RS})$
Compute $K \leftarrow (N_A^{RS})^{RB}$(mod p)
$\{f_K(ID_A,K)\}$
$\{f_K(ID_B,K)\}$
Verify $f_K(ID_B,K)$
Verify $f_K(ID_A,K)$

Figure 3. The proposed protocol

5. Security and Efficiency Analyses

The following are the security requirements to be met by a password key exchange protocol[6].

- Mutual authentication
- Resistance to the password guessing attacks.
- Transmission round and computation complexity.

The proposed protocol is satisfying the above requirements.The following section presents the brief report on the security analyses of the protocol with respect to requirements.

5.1. Mutual authentication

First, A and B use the trapdoor function F_S to hide the random number r_A & r_B and pwA & pwB to encrypt N_A & N_B in step 1, as described in section 4 . since only S knows the trap door , pwA & pwB , only S can authenticate A/B after receiving the message sent in step 1.

Second, S sends $\{N_B^{RS}, f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})\}$ to A,$\{ N_A^{RS}, f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS}\}$ to B in step 2. This message can be used to authenticate 'S' as mentioned in step 2 in section 4.

Third, A and B derive key from N_B^{RS} and N_A^{RS} respectively, as mentioned in step 2 in section 4. With the help of $f_K(ID_B, K)$, $f_K(ID_A, K)$ A and B can authenticate each other.

5.2. Resistance to the password guessing attacks

First, a malicious attacker may try to guess the password with Undetectable on-line password guessing attacks. If that is the case, the mutual authentication step is not possible. If B tries to guess A's password, then B should perform the following procedure to mount an Undetectable on-line password guessing attack. B obtains $(K_{AS}\oplus N_A)^*$ by decrypting E_{PWA} $(K_{As}\oplus N_A)$ with a guessed password pwA*. Next he selects his random exponent RB and computes $N_B=g^{RB}$ mod p and finds $Fs(N_B\oplus ID_B)$, $f_{(KAS\oplus NA)*}(N_B)$ and sends $E_{pwA}((K_{AS}\oplus N_A)^*\oplus N_B)$, $Fs(N_B\oplus ID_B)$, $f_{(KAS\oplus NA)*}(N_B)$ to server. Server authenticates the clients and sends $\{N_B^{RS}, f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})\}$ to A and $\{ N_A^{RS}, f_{KBS}(ID_A,ID_B,K_{BS},N_A^{RS})\}$ to B. Now client B intercepts $\{N_B^{RS}, f_{KAS}(ID_A,ID_B,K_{AS},N_B^{RS})\}$ but cannot compare any two terms and verify whether the guessed password is correct or not. Hence B cannot mount an Undetectable on-line password guessing attack on the proposed protocol.

Second, an attacker may try to guess the password with detectable on-line password guessing attack. He guesses pwA* or pwB* to impersonate A or B, chooses RA or RB, computes $N_A=g^{RA}$ mod p or $N_B=g^{RB}$ mod p and selects r_A or r_B, computes $K_{AS}=N_A^{rA}$ mod p or $K_{BS}=N_B^{rB}$ mod p and then sends $E_{pwA*}(K_{As}\oplus N_A)$, $F_S(N_A\oplus ID_A)$, $f_{KAS}(N_A)$ or $E_{pwB*}(K_{Bs}\oplus N_B)$, $F_S(N_B\oplus ID_B)$, $f_{KBS}(N_B)$. Server will decrypt $E_{pwA*}(K_{As}\oplus N_A)$ or $Epw_{B*}(K_{Bs}\oplus N_B)$ and gets $(K_{As}\oplus N_A\ )^*$or $(K_{Bs}\oplus N_B)^*$. Now N_A or N_B will be extracted from $F_S(N_A\oplus ID_A)$, ID_A or $Fs(N_B\oplus ID_B)$, ID_B. S computes $K_{AS}=\ (K_{AS}\oplus N_A\ )^*\oplus K_{AS}$ or $K_{BS}=(K_{BS}\oplus N_B)^*\oplus K_{BS}$ and $f_{KAS}(N_A)$ or $f_{KBS}(N_B)$ will be determined. But the computed hash values will not be equal to the received hash values. Hence S can detect this attack and take the counter measure. Hence, it is impossible for an attacker to mount detectable on-line password guessing attack.

Third, an attacker may try to mount off-line password guessing attack to guess the password. He intercepts $E_{pwA}(K_{AS}\oplus N_A)$, $F_S(N_A\oplus ID_A)$, $f_{KAS}(N_A)$ and may guess a password, extracts $K_{AS}\oplus N_A$, but it is impossible for him to get N_A until trapdoor is known, which is known only to server. This implies that he cannot verify the hash value $f_{KAS}(N_A)$. Hence off-line password guessing attack on the proposed protocol is impossible.

Perfect forward secrecy: The enhanced protocol has the perfect forward secrecy. The session key is computed as follows: $K=(N_B^{RS})^{RA}$ (mod p)$=(N_A^{RS})^{RB}$ (mod p). If the attacker gets $\{N_B^{RS}, f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})\}$ or $\{N_A^{RS}, f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})\}$, then in order to obtain the session key, he should know RB or RA. Since this is not possible he cannot get the key.

The session keys generated in different sessions are independent since RA and RB are randomly chosen by A and B respectively. This indicates that the attacker cannot obtain previous session keys even if he obtains the session key used in this run.

Known-Key Security: In the enhanced protocol as RA, RB are randomly chosen by A and B, and are independent among protocol executions. This leads to the in-vulnerability of Known-Key security.

Server spoofing: The server computes $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$, $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$ and sends to A and B, respectively. A and B can verify the identity of server or authenticate the server by computing $f_{KAS}(ID_A, ID_B, K_{AS}, N_B^{RS})$, $f_{KBS}(ID_A, ID_B, K_{BS}, N_A^{RS})$, respectively. Thus, the attacker cannot impersonate the server to deceive the client.

Man-in the middle attack: Suppose the attacker frames his own message i.e. $E_{PWc}(K_{CS}\oplus N_C)$, $F_S(N_c\oplus ID_C)$, $f_{KCS}(N_c)$ with the correct guesses password and sends to server. The server will decrpt $E_{PWC}(K_{CS}\oplus N_C)$ and gets '$K_{CS}\oplus N_C$' and obtains '$N_C\oplus ID_C$' from $F_S(N_C\oplus ID_C)$. Finaly, S computes hash value which will not match with the received hash value. Hence the protocol gets terminated and not allowing man-in the middle to mount any attack.

Trivial attack: An attacker may directly try to compute the session key from N_A^{RS} or N_B^{RS}. However, due to the intractability of DLP and the one-wayness of hash function, the trivial attack is not possible in the proposed protocol.

Replay attack: Since one way trapdoor hash function is used, the proposed protocol is invulnerable of this attack.

5.3. Transmission round and computation complexity

The development of an efficient protocol should take the number of transmission rounds (and steps) and the computation complexity into account. The proposed protocol requires four message transmission rounds. Table 1 shows the performance comparison analyses of the proposed protocol, PSRJ protocol and Lo and Yeh protocol [8]. The modular exponential operations are reduced since client A sends E_{pwA} $(K_{AS}\oplus N_A)$, $F_S(N_A\oplus ID_A)$, $f_{KAS}(N_A)$ to S and client B sends $E_{pwB}(K_{BS}\oplus N_B)$, $F_S(N_B\oplus ID_B)$, $f_{KBS}(N_B)$ to S. S decrypts $E_{pwA}(K_{AS}\oplus N_A)$ and $E_{pwB}(K_{BS}\oplus N_B)$ and gets $K_{AS}\oplus N_A$ and $K_{BS}\oplus N_B$ respectively. Next S extracts N_A and N_B from $F_S(N_A\oplus ID_A)$, $F_S(N_B\oplus ID_B)$ and ID_A,ID_B. Now K_{AS} and K_{BS} are computed by $K_{AS}= K_{AS}\oplus N_A\oplus N_A$ and $K_{BS}=K_{BS}\oplus N_B\oplus N_B$. Since $E_{pwA}(K_{AS}\oplus N_A)$, $F_S(N_A\oplus ID_A)$, $E_{pwB}(K_{BS}\oplus N_B)$, $F_S(N_B\oplus ID_B)$ are arranged in a proper sequence two modular exponential operations are reduced on the server side hence computation complexity is reduced.

Table 1. Performance comparison between the proposed protocol, PSRJ protocol and Lo and Yeh protocol

	The Proposed protocol	PSRJ protocol	Lo-Yeh protocol
communication party	A B S	A B S	A B S
Modular exponential operation	**3 3 2**	**3 3 4**	**3 3 4**
Symmetric encryption/ decryption	1 1 2	1 1 2	1 1 2
PRF operation	4 4 4	4 4 4	4 4 4
TDF operation	1 1 2	1 1 2	1 1 2
Random number		2 2 1	2 2 1
XOR operation	2 2 4	0 0 0	2 2 4
Transmission round	4	4	4

6. Conclusion

An enhanced password-key exchange protocol which is in-vulnerable to undetectable on-line password attacks is proposed. The modular exponential operations are expensive. The designed protocol is developed with reduced modular exponential operation on server side. The above results show that the proposed protocol is secure , efficient and practical.

References

[1] **W. Diffie, M. Hellman.** New Directions in cryptography. *IEEE Transactions on Information theory, Vol*. 22, *No*. 6 , 1976, 644-654.

[2] **Y. Ding, P. Horster.** Undetectable Online password guessing attacks. *ACM operating systems Review*, *Vol*. 29, *No*. 4,pp 77-86 (1995)

[3] **S.M. Bellovin, M. Merrit.** Encrypted key exchange: password-based protocols secure against dictionary attacks. *Proceedings of IEEE sysmposium on research in security and privacy, IEEE Computer society press*, 1992, 72-84.

[4] **M. Steiner, G. Tsudik, M. Waidner.** Refinement and extention of encrypted key exchange. *ACM Operating Systems Review*, *Vol*. 29, *No*. 3, 1995, 22-30.

[5] **C.L. Lin, H.M. Sun, M. Steiner, T. Hwang.** Three-party excrypted key exchange without server public-Keys. *IEEE Communication letters*, *Vol*. 5, *No*. 12, 2001, 497- 499.

[6] **C.C. Chang, Y.F. Chang.** A novel three party encrypted key exchange protocol. *Computer Standards and Interfaces*, *Vol*. 26 , *No*. 5, 2004, 471-476.

[7] **E.J. Yoon, K.Y. Yoo.** Improving the novel three-party encrypted key exchange protocol. *Computer Standards and Interfaces*, 30, 2008, 309-314.

[8] **N.W. Lo, K.-H. Yeh.** Cryptanalysis of two three-party encrypted key exchange protocols. *Computer Standards & Interfaces*, *Vol*. 31, *issue* 6, *Nov*. 2009, 1167-1174.

[9] **R. Padmavathy, S. Tallapally, R. JayadevGyani.** Improved Analysis of Chang and Chang password key exchange protocol, *ACT* 2009, 781-783, Doi 10.1109 / ACT.2009.197.

Received January 2010.