I. J. Computer Network and Information Security, **2016, 5, 9-17**
Published Online May 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.05.02

Design of a Robust, Computation-Efficient and Secure 3P-EKE Protocol using Analogous Message Transmission

Archana Raghuvamshi
Adkavi Nannaya University /CSE Department, Rajahmundry, 533296, India
E-mail: archana_anur@yahoo.in

Premchand Parvataneni
Osmania University/CSE Department, Hyderabad, 500007, India
E-mail: profpremchand.p@gmail.com

Abstract—In this modern era of digital communication even a trivial task needs to be performed over internet which is not secure. Many cryptographic algorithms existed to provide security which facilitates secure communication through internet. As these algorithms need a secret session key, it is required to interchange this key in a secure way. In two-party communication, two clients initially share a low random (entropy) password through a secure channel to establish a secret session key. But this paradigm necessitates high maintenance of passwords, since each communicating pair requires separate passwords to establish a secure session key. In three-party communication network, each communication party shares a password with the trusted third-party (server) to exchange a secret session key. The beauty of this setting is that, even a server does not know the session key. The Password Authenticated Encrypted Key Exchange (PA-EKE) protocols have attracted a lot of curiosity to authors to propose various two-party and three-party PA-EKE protocols. Security flaws in various protocols proposed by Chang-Chang, Yoon-Yoo, PSRJ and Raj et al. inspired to design a robust, computationally efficient and highly secure protocol. This paper is an attempt to propose a secure and novel Password Authenticated 3P-EKE protocol using XOR operations and analogous (parallel) message transmission. The proposed protocol is easy to design and more secured against all types of attacks like password guessing, replay, pre-play, server spoofing etc. which made this protocol special.

Index Terms—Password Authenticated Key Exchange Protocols, Three-Party Encrypted Key Exchange Protocols, Types of Attacks.

I. Introduction

In order to facilitate the communication over internet cryptographic algorithms are needed to establish communication in a highly secure way. These algorithms need a secret key to perform encryption and decryption. The two parties who want to communicate securely, has to interchange the secret session key in a smart way. Nowadays, PA-EKE protocols are extensively used in communication networks. Design of any new communication protocol which is smart and helps in establishing a secret session key with fewer computations is need of the hour. Diffie-Hellman (D-H) (1976) [1] proposed a key agreement protocol which is suffered from man-in-the-middle attack. This flaw inspired many authors to propose different two-party PA-EKE protocols. In two-party communication network, each communicating pair of clients shares a low random password over secure channel to establish a Secret Session Key. But this paradigm requires high maintenance of passwords due to each communication pair needs a separate password to establish a secure session key. This flaw motivates the authors to propose three-party PA-EKE protocols. In password authenticated Three-Party Encrypted Key Exchange (3P-EKE) protocols, each communication party shares a low random (entropy) password in advance with the trusted third-party through a secure channel. Once this happens, any two parties who anticipate in establishing a strong session key, has to route to the trusted third-party with their shared passwords to verify each other. Subsequently, only the valid parties can be endorsed to develop the secure session key. The novelty of this setting is even a trusted third-party is not knowing the session key. This particular mechanism (3P-EKE) is extensively set up on oodles of distant user verification systems due to its easiness and suitability of maintaining a low random password at user side. Such 3P-EKE protocols can be used in applications which are light-weight and allow the users to communicate securely. But unfortunately, many of such mechanisms suffered from any one of the three types of password guessing attacks as explained by Ding & Horster [2]. A password guessing attack is nothing but learning the correct secret password by continuously trying until it is discovered by an intruder. Hence, any good PA-EKE protocol should satisfy the following security requirements:

- *Mutual Authentication:* This requirement enables the users to authenticate each other mutually. (through their identity).
- *Resistance to password guessing attacks* proposed by Ding & Hoster (D&H) [2].
- *Session Key (SK) security*: This enables that session key cannot be obtained without persistent secrets.
- *Resistant to Trivial Attack*: This ensures that one cannot compute the SK directly.
- *Resistant to Pre-play Attack:* This requirement ensures that playing on simulated system should not succeed in practice on original system.
- *Resistant to Replay Attack*: This ensures that an intruder should not succeed in replaying with the intercepted & stored messages.
- *Resistant to Man-in-the-middle Attack*: By interrupting the message, an intruder establishes his own communication with either party. This requirement ensures that the proposed protocol should be resistant to such type of attack.
- *Server spoofing security:* If server is compromised, it should not affect the existing protocol.
- *Perfect forward secrecy:* This requirement ensures that compromise of preserved long-life keys should not lead to the compromise of previous session keys.
- *Known-Key Security:* This requirement ensures that the compromise of one session key should not lead to the compromise of later session keys.

This paper is an attempt to propose a novel method for computing the session key based on XOR operations with analogous (parallel) message transmission which satisfies the above security requirements. The following sections provide a detailed summary of the procedures followed to design a novel protocol which facilitates a robust, secure and computationally efficient way of communication over internet.

II. Related Work

Bellovin and Merrit (1992) have proved that password-based authenticated protocols are secure against password guessing attacks [3]. Later, various authors have proposed many two-party password authenticated key exchange (2PAKE) protocols. Due to shortfalls, 2PAKE protocols are found only suitable for client-server architecture, which motivates research community to encompass 2PAKE protocols into 3PAKE schemes for three-party communication environment.

Chang et al. (2004) [4] proposed an efficient three-party encrypted key exchange protocol (ECC-3PEKE) with both round and computation efficiencies. However, unfortunately, an undetectable online password guessing attack has been notified on ECC-3PEKE protocol by Yoon et al. (2008) [5]. At the same time they have also proposed an improvement over ECC-3PEKE protocol and claimed that the proposed protocol defends against undetectable online dictionary attacks. Also in the same year Chung et al. (2008) [6] has shown three weaknesses in a simple three-party key exchange protocol. Later, Padmavathy et al. (2009) [7] have proposed some improvement over the ECC-3PEKE protocol and claimed that the proposed protocol (PSRJ Protocol) is secure against undetectable online dictionary attacks and also achieves better performance, which requires only 4 message transmission rounds. Later, Chang et.al (2009) [8] specified the insecurity of Yoon-Yoo's Protocol and R. Padmavathy (2010) [9] has notified an undetectable online dictionary attack on PSRJ protocol. Additionally, to overcome an attack, she proposed an enhancement over the existing protocol with reduced modular exponentiation operations. In connection with this, Shirisha Tallapally (2010) [10] has proposed an impersonation attack on ECC-3PEKE protocol. Later, Archana et al. (2012) [11] has cryptanalyzed the PSRJ protocol which forms a basis for this paper. In the same year, Raj et al. [12] proposed Security Enhancement for 3-PEKE protocol using parallel Message transmission. Later, Raj et al. (2013) [13] has discussed the performance analysis of 3P-EKE protocol using parallel message transmission technique. But unfortunately, Archana et.al (2013) [14] have notified the weakness of 3P-EKE Protocol proposed by Raj et.al (2013).

The list of notations and their descriptions used throughout this paper are illustrated in Table 1. The protocols discussed in this paper assume that the passwords Pwd_a (Alice) and Pwd_b (Bob) should initially be shared with trusted party through a secured channel.

Table 1. List of Notations

Alice/Bob	Two parties who want to communicate with each other
Carol	An Attacker
Trusted Party	Trusted third party(Server)
Id_a, Id_b ,Id_t	Identities of Alice, Bob and Trusted Party
Pwd_a, Pwd_b	Passwords secretly shared by Alice and Bob with Trusted Party respectively
K_t	Trusted Party's Public key
$E_{pwd}()$	A symmetric Encryption scheme with a password pwd
$D_{pwd}()$	A symmetric Decryption scheme with a password pwd
p	A large prime number
g	A generator in GF(Group Field)
r_a, r_b	Random numbers chosen by Alice & Bob respectively.
RE_a, RE_b ,RE_t	Random Exponents of Alice, Bob and Trusted party respectively
M_a ,M_b	$M_a=g^{RE_a} \bmod p$, $M_b=g^{RE_b} \bmod p$
K_{at} , K_{bt}	$K_{at}=M_a^{r_a} \bmod p$, $K_{bt}=M_a^{r_b} \bmod p$ are one time strong keys shared by Alice & Bob with Trusted Party respectively
$h_t()$	A one-way trapdoor function , where only trusted party knows the trapdoor
$f_k()$	A pseudo random hash function indexed by a key k
sk	Session Key

III. Analysis Of PSRJ And Raj Et Al. 3P-EKE Protocols

The aforementioned sections have reviewed and highlighted the PSRJ and Raj et al. 3P-EKE protocols and its weaknesses in detail.

A. Analysis of PSRJ Protocol

As required, the PSRJ protocol has been reviewed and then cryptanalyzed by showing its vulnerability to online detectable dictionary attacks.

Review

In this section, we review the PSRJ protocol. The details of the protocol can be described as follows:

Step 1: Alice computes $M_a=g^{REa} \bmod p$ & $K_{at}=M_a^{ra} \bmod p$ by generating the random numbers, RE_a, $r_a \in_R Z_p$ and sends the credentials $\{Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_a), f_{Kat}(M_a)\}$ to Trusted Party.

i.e., Alice → Trusted Party: $\{Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_a), f_{Kat}(M_a)\}$

Similarly, Bob computes $M_b=g^{REb} \bmod p$ & $K_{bt}=M_b^{rb} \bmod p$ by generating the random numbers, RE_b, $r_b \in_R Z_p$ and sends the credentials $\{Id_a, Id_b, Id_t, E_{pwdb}(M_b), h_t(r_b), f_{Kbt}(M_b)\}$ to Trusted Party.

i.e., Bob → Trusted Party: $\{Id_a, Id_b, Id_t, E_{pwdb}(M_b), h_t(r_b), f_{Kbt}(M_b)\}$

Step 2: After receiving the credentials from Alice & Bob, Trusted Party verifies the credentials. Then Trusted Party generates a random exponent, $RE_t \in_R Z_p$ to compute $M_b^{REt} \bmod p$ & $M_a^{REt} \bmod p$ and sends the credentials $\{M_b^{REt} \bmod p, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ to Alice and $\{M_a^{REt} \bmod p, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$ to Bob.

i.e., Trusted Party → Alice: $\{M_b^{REt} \bmod p, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ and

Trusted Party → Bob: $\{M_a^{REt} \bmod p, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$

Step 3: Alice computes the session key $SK=(M_b^{REt})^{REa} \bmod p$ and sends $f_{SK}(Id_a, SK)$ to Bob.

i.e., Alice → Bob: $f_{SK}(Id_a, SK)$

Step 4: Bob computes the session key $SK=(M_a^{REt})^{REb} \bmod p$ and sends $f_{SK}(Id_b, SK)$ to Alice.

i.e., Bob → Alice: $f_{SK}(Id_b, SK)$.

After Alice and Bob successfully examine the validation of the incoming messages $f_{SK}(Id_b, SK)$ and $f_{SK}(Id_a, SK)$, both of them can ensure that they actually share the secret session key $SK=(M_b^{REt})^{REa} \pmod p = (M_a^{REt})^{REb} \pmod p$ at present. If validation fails, the protocol will be terminated. Fig 1 illustrates 3P-EKE PSRJ Protocol.

Attack

This section endeavors to demonstrate the detectable online password guessing attack on PSRJ 3P-EKE protocol.

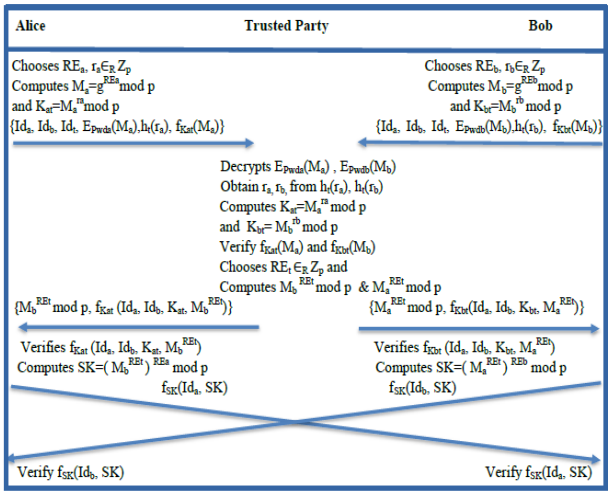

Fig.1. PSRJ 3P-EKE Protocol

The details of attack are shown below.

Step 1: User Alice generates two random numbers viz., RE_a, $r_a \in_R Z_p$, and calculates $M_a=g^{REa} \bmod p$ & $K_{at}=M_a^{ra} \bmod p$ to compute $E_{pwda}(M_a)$, $h_t(r_a)$ & $f_{Kat}(M_a)$ and sends $\{(Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_a), f_{Kat}(M_a))\}$ to Trusted Party.

i.e., Alice → Trusted Party: $\{(Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_a), f_{Kat}(M_a))\}$.

Step 2: An invader Carol guesses Alice's password as $Pwd_{a'}$ by intercepting the message i.e.$\{(Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_a), f_{Kat}(M_a))\}$ and decrypts $E_{pwda'}(M_a)$ & to get $M_{a'}$. Now attacker computes $K_{at'}=M_{a'}^{rc'} \bmod p$ by generating her random number $r_{c'}$ and sends $\{Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_{c'}), f_{Kat'}(M_{a'})\}$ to Trusted Party.

i.e., Carol **sends** Trusted Party: $\{Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_{c'}), f_{Kat'}(M_{a'})\}$

Step 3: The Trusted Party decrypts $E_{pwda}(M_a)$ to get M_a and retrieves $r_{c'}$ from $h_t(r_{c'})$ by using trapdoor[15], after receiving the credentials $\{Id_a, Id_b, Id_t, E_{pwda}(M_a), h_t(r_{c'}), f_{Kat'}(M_{a'})\}$ from the attacker Carol. Now Trusted Party computes $K_{at'}=M_{a'}^{rc'} \bmod p$ to authenticate the received $f_{Kat'}(M_{a'})$. If both $f_{Kat}(M_a)$ and $f_{Kat'}(M_{a'})$ are equal then the guessed password by Carol is correct and Trusted Party will continue with the remaining procedure of the protocol by assuming that he is in a secured zone.

If both $f_{Kat}(M_a)$ and $f_{Kat'}(M_{a'})$ are not equal, it means that the attack is detected by Trusted Party and Trusted Party terminates this protocol at current session. An intruder never sits indolent. After some time Carol repeats the same process. She will continue the same process until she hits with the successful password.

In this way a malevolent user can copycat the actual user by getting the secrete session key successfully. Fig 2 shows the detectable on-line password guessing attack.

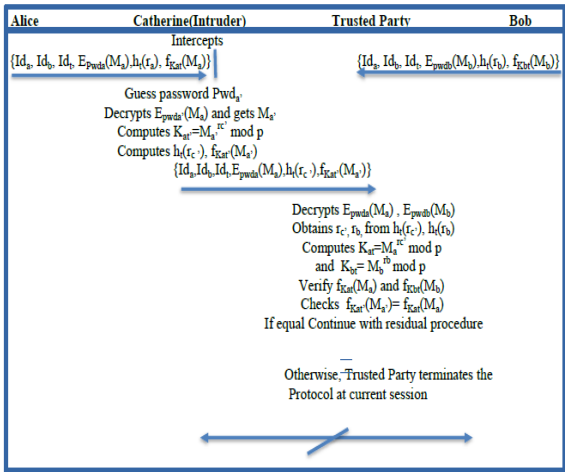

Fig.2. Detectable Online Password Guessing Attack on PSRJ 3P-EKE Protocol

B. Analysis of Raj et al. Protocol

This section devotes to review the Raj et al. protocol and then cryptanalyze the protocol by showing the detectable online dictionary attack.

Review

The detailed explanation is given below:

Step 1: Alice generates two random numbers viz., r_a, RE_a $\in_R$ Z_p and computes $M_a=g^{REa}$ (mod p) & $K_{at}=M_a^{ra}$(mod p). Then she calculates $E_{pwda}(K_{at}\oplus Ma)$, $h_t(M_a\oplus Id_a)$ & $f_{Kat}(M_a)$ and sends these credentials To Trusted Party.

i.e., Alice → Trusted Party**:** $\{Id_a, Id_b, Id_t, E_{pwda}(K_{at}\oplus M_a), h_t(M_a\oplus Id_a), f_{Kat}(M_a)\}$

Simultaneously, Bob generates two random numbers viz., r_b, RE_b $\in_R$ Z_p to compute $M_b=g^{REb}$ (mod p) & $K_{at}=M_a^{ra}$(mod p) and then calculates $E_{pwdb}(K_{bt}\oplus M_b)$, $h_t(M_b\oplus Id_b)$ & $f_{Kat}(M_b)$ and sends these credentials To Trusted Party.

Here users Alice and Bob communicate with the Trusted Party in a parallel way.

i.e., Bob → Trusted Party**:** $\{Id_a, Id_b, Id_t, E_{pwdb}(K_{bt}\oplus M_b), h_t(M_b\oplus Id_b), f_{Kbt}(M_b)\}$

Step 2: After receiving the messages sent from users Alice and Bob, Trusted Party utilizes a trapdoor to obtain $M_a\oplus Id_a$ & $M_b\oplus Id_b$ from $h_t(M_a\oplus Id_a)$ & $h_t(M_b\oplus Id_b)$ and retrieves M_a $=(M_a\oplus Id_a)\oplus$ Id_a & M_b $=(M_b\oplus Id_b)\oplus$ Id_b respectively.

Next, it gets $K_{at}\oplus M_a$ & $K_{bt}\oplus M_b$ by decrypting $E_{pwda}(K_{at}\oplus M_a)$ & $E_{pwdb}(K_{bt}\oplus M_b)$ with the passwords Pwd_a and Pwd_b respectively. Now, $f_{Kat}(M_a)$ & $f_{Kbt}(M_b)$ are computed by retrieving the values K_{at} = $K_{at}\oplus M_a\oplus$ M_a & K_{bt} = $K_{bt}\oplus M_b\oplus$ M_b. Trusted Party verifies whether computed value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$) and received value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$)) are identical or not. If this verification is true then Trusted Party continues with the residual procedure of this protocol by computing M_a^{REt} mod p & M_b^{REt} mod p to calculate the corresponding hashed credentials $f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})$ & $f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})$ respectively.

Finally, Trusted Party sends$\{M_b^{REt}, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ to Alice and $\{M_a^{REt}, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$ to Bob simultaneously. If computed value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$) and received value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$)) are not identical then Trusted Party terminates the protocol.

i.e., Trusted Party → Alice**:** $\{M_b^{REt}, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$,

Trusted Party → Bob**:** $\{M_a^{REt}, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$

Step 3: Bob verifies the credentials $f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})$ to authenticate, after receiving the transmitted messages sent from Trusted Party. If this verification is passed, Bob believes the received M_a^{REt} is valid and then computes the session key $SK=(M_a^{REt})^{REb}$ (mod p) and sends $f_{SK}(Id_b,SK)$ to Alice. Otherwise, Bob terminates the protocol.

i.e., Bob → Alice**:** $f_{SK}(Id_b, SK)$.

Step 4: Similarly, Alice verifies $f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})$ to authenticate, after receiving the transmitted messages sent from Trusted Party. If this verification is passed, Alice believes the received M_b^{REt} is valid and then computes the session key $SK=(M_b^{REt})^{REa}$ (mod p) and sends $f_{SK}(Id_a,SK)$ to Bob. And also $f_{SK}(Id_b, SK)$ will be used by the user Alice to verify the authenticity of user Bob. If this verification does not hold, Alice terminates the protocol.

i.e., Alice → Bob**:** $f_{SK}(Id_a, SK)$.

Step 5: After the verification of incoming messages $f_{SK}(Id_b, SK)$ & $f_{SK}(Id_a, SK)$ from the other side, both Alice and Bob ensures that they actually shares the secret session key $SK= (M_b^{REt})^{REa}$ (mod p)$= (M_a^{REt})^{REb}$ (mod p).

Otherwise, the protocol will be terminated. Fig 3 illustrates the Raj et al. 3P-EKE Protocol.

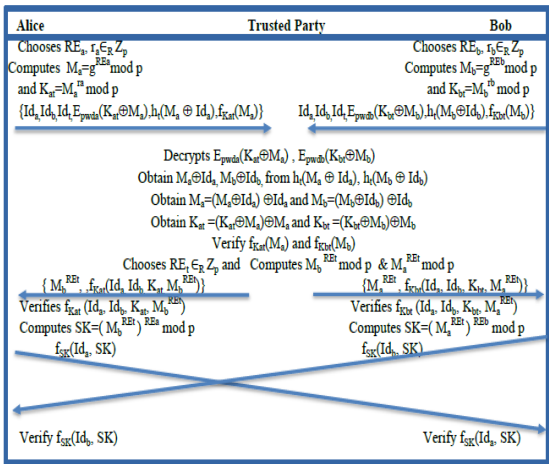

Fig.3. Raj et al. 3P-EKE protocol

Attack

This section exhibits the detectable online password guessing attack on Raj et al. 3P-EKE Protocol. An

invader Carol can copycat Alice and connect with Bob. While Bob is thinking that it is talking with Alice but actually it is talking with the invader Carol.

The details of attack are as follows:

Step 1: Alice generates two random numbers viz., RE_a, $r_a \in_R Z_p$ to compute $M_a=g^{REa} \pmod p$ & $K_{at}=M_a^{ra} \pmod p$ and then calculates $E_{Pwda}(K_{at} \oplus M_a)$, $h_t(M_a \oplus Id_a)$ & $f_{Kat}(M_a)$ and sends these credentials to Trusted Party.

i.e., Alice → Bob: $\{Id_a, Id_b, Id_t, E_{Pwda}(K_{at} \oplus M_a), h_t(M_a \oplus Id_a), f_{Kat}(M_a)\}$

Step 2: Let an attacker Carol interrupts and intercepts the message $\{Id_a, Id_b, Id_t, E_{Pwda}(K_{at} \oplus M_a), h_t(M_a \oplus Id_a), f_{Kat}(M_a)\}$ and generates her own two random numbers viz., $RE_{a'}, r_{a'} \in_R Z_p$ to compute $M_{a'}=g^{REa'} \pmod p$ & $K_{at'}=M_{a'}^{ra} \pmod p$. Now attacker Carol guess Alice's password as $Pwd_{a'}$ to encrypt $E_{Pwda'}(K_{at'} \oplus M_{a'})$.Again she also computes the another two credentials $h_t(M_a \oplus Id_a)$ & $f_{Kat'}(M_{a'})$ by its own because the Id's are not secret and sends $\{Id_a, Id_b, Id_t, E_{Pwda'}(K_{at'} \oplus M_{a'}), h_t(M_{a'} \oplus Id_a), f_{Kat'}(M_{a'})\}$ to Trusted Party.

i.e., Carol → Trusted Party: $\{Id_a, Id_b, Id_t, E_{Pwda'}(K_{at'} \oplus M_{a'}), h_t(M_{a'} \oplus Id_a), f_{Kat'}(M_{a'})\}$

Step 3: Trusted Party decrypts $E_{Pwda'}(K_{at'} \oplus M_{a'})$ to get $(K_{at'} \oplus M_{a'})$ and also retrieves $(M_{a'} \oplus Id_a)$ from $h_t(M_{a'} \oplus Id_a)$ by using trapdoor after receiving the credentials $\{Id_a, Id_b, Id_t, E_{Pwda'}(K_{at'} \oplus M_{a'}), h_t(M_{a'} \oplus Id_a), f_{Kat'}(M_{a'})\}$ from an attacker Carol and computes $M_{a''}=(M_{a'} \oplus Id_a) \oplus Id_a$ to obtain $K_{at''}= K_{at'} \oplus M_{a'} \oplus M_{a''}$. Now Trusted Party verifies whether computed $f_{Kat''}(M_{a''})$ and received $f_{Kat'}(M_{a'})$ are equal or not. If both $f_{Kat''}(M_{a''})$ & $f_{Kat'}(M_{a'})$ are equal then the guessed password is right and Trusted Party will continue with the remaining procedure of the protocol by assuming that he is in secured zone. In this way an attacker Carol may succeed in her attempt to attack.

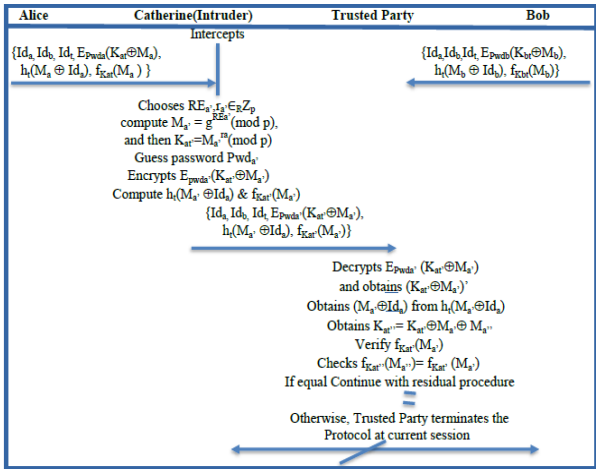

Fig.4. Detectable Online Password Guessing Attack on Raj et al.3P-EKE Protocol

If both $f_{Kat''}(M_{a''})$ and $f_{Kat'}(M_{a'})$ are not equal, then the attack can be detected by Trusted Party. And the Trusted Party terminates the protocol for current session. An intruder never sits idle. After some time she repeats the same process. She will continue this until she hits the successful password. In this way a malicious user can impersonate the actual user by getting the secrete session key successfully. Fig 4 illustrates the detectable online password guessing attack on Raj et al. protocol.

IV. Proposed Password Authenticated 3P-EKE Protocol

In order to disregard the detectable online password guessing attacks, this paper endeavors to propose a secure Three-Party Encrypted Key Exchange (3P-EKE) Protocol based on XOR operation with analogous (parallel) message transmission. By using XOR operator and analogous message transmission, the proposed protocol makes it highly secure and improve the efficiency of transmission round. The detailed procedure of the proposed protocol is described in different steps as follows:

Step 1: Alice generates two random numbers viz., r_a, $RE_a \in_R Z_p$ to compute $M_a=g^{REa} \pmod p$ & $K_{at}=M_a^{ra} \pmod p$ and then calculates $E_{Pwda}(M_a \oplus r_a)$, $h_t(Pwd_a \oplus M_a)$ & $f_{Kat}(M_a)$.Now she sends these credentials To Trusted Party.

i.e., Alice → Trusted Party: $\{Id_a, Id_b, Id_t, E_{Pwda}(M_a \oplus r_a), h_t(Pwd_a \oplus M_a), f_{Kat}(M_a)\}$.

Step 2: Simultaneously, Bob also generates two random numbers viz., RE_b $r_b \in_R Z_p$ to compute $M_b=g^{REb} \bmod p$ & $K_{bt}=M_b^{rb} \bmod p$. Finally, he calculates $E_{Pwdb}(M_b \oplus r_b)$, $h_t(Pwd_b \oplus M_b)$ & $f_{Kbt}(M_b)$ and transfers $\{Id_a, Id_b, Id_t, E_{pwdb}(M_b \oplus r_b), h_t(Pwd_b \oplus M_b), f_{Kbt}(M_b)\}$ to Trusted Party.

i.e., Bob → Trusted Party: $\{Id_a, Id_b, Id_t, E_{Pwdb}(M_b \oplus r_b), h_t(Pwd_b \oplus M_b), f_{Kbt}(M_b)\}$.

Step 3: After receiving the messages sent from Alice and Bob, Trusted Party utilizes a trapdoor [15] to obtain $Pwd_a \oplus M_a$ & $Pwd_b \oplus M_b$ from $h_t(Pwd_a \oplus M_a)$ & $h_t(Pwd_b \oplus M_b)$ and computes $M_a=Pwd_a \oplus M_a \oplus Pwd_a$ & $M_b= Pwd_b \oplus M_b \oplus Pwd_b$ respectively. Now, Trusted Party decrypts $E_{Pwda}(M_a \oplus r_a)$ & $E_{Pwdb}(M_b \oplus r_b)$ and obtains $M_a \oplus r_a$ and $M_b \oplus r_b$. Then it retrieves the values $r_a=M_a \oplus r_a \oplus M_a$ & $r_b=M_b \oplus r_b \oplus M_b$ and computes $K_{at}=M_a^{ra} \bmod p$ & $K_{bt}=M_b^{rb} \bmod p$ accordingly and verifies whether computed value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$) and received value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$) are identical or not. If this verification is passed, then it continues with the residual procedure of the protocol. Otherwise, Trusted Party terminates the protocol.

Finally, Trusted Party generates random exponent $RE_t \in_R Z_p$ to compute $M_b^{REt} \bmod p$ & $M_a^{REt} \bmod p$ and calculates the hashed credentials $f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})$ & $f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})$. Finally, Trusted Party sends these credentials to Alice and Bob simultaneously.

i.e., Trusted Party → Alice: M_b^{REt}, $f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ and

Trusted Party → Bob: $\{M_a^{REt}, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$

Step 4: Now to authenticate the Trusted Party, Alice & Bob verifies the received credentials $f_{Kat}(Id_a, d_b, K_{at}, M_b^{REt})$ & $f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})$ respectively from it. If this verification does not hold, Alice and Bob terminate the protocol accordingly.

If this verification holds, Alice computes the session key $SK=(M_b^{REt})^{REa} \pmod p$ & $f_{SK}(Id_a,SK)$ and sends it to Bob. Similarly Bob computes the session key $SK=(M_a^{REt})^{REb} \pmod p$ & $f_{SK}(Id_b, SK)$ and sends it to Alice.

i.e., Alice → Bob: $f_{SK}(Id_a, SK)$ and
Bob → Alice: $f_{SK}(Id_b, SK)$.

Step 5: After verification of the incoming messages $f_{SK}(Id_b, SK)$ & $f_{SK}(Id_a, SK)$ from the other side, both Alice and Bob can ensure that they actually share the secret session key $SK=(M_b^{REt})^{REa} \pmod p = (M_a^{REt})^{REb} \pmod p$ for the current session. Otherwise, the protocol will be terminated. Fig 5 illustrates the proposed 3P-EKE Protocol.

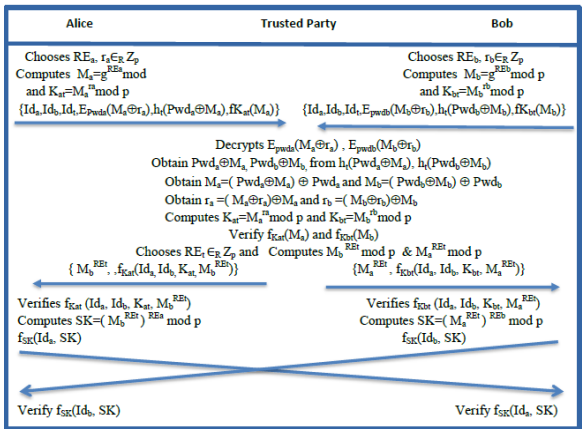

Fig.5. Proposed Password Authenticated 3P-EKE Protocol

V. Security and Performance Analysis

After careful revision and implementation of the proposed protocol, it is understood that the protocol is not only secure but highly efficient and also meets the given security requirements. The detailed analysis is presented in the following sections.

A. *Security Analysis*

The security analysis has shown that the proposed protocol is secure and robust to face any kind of possible password guessing attacks. The proposed protocol has successfully demonstrated its capabilities by meeting the necessary security requirements as follows:

i. Mutual Authentication

First: Alice & Bob use the trapdoor function h_t to hide the random number & password i.e., $(REa \oplus Pwd_a)$ and $(RE_b \oplus Pwd_b)$ respectively. Trusted Party needs a corresponding password to decrypt $M_a \oplus r_a$ & $M_b \oplus r_b$ respectively. Since, only Trusted Party knows the trapdoor t and passwords Pwd_a & Pwd_b they can very well authenticate Alice & Bob after receiving the messages sent in step1 & step2 of the proposed protocol.

Second: Trusted Party sends $\{M_b^{REt}, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ to Alice & $\{ M_a^{REt}, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$ to Bob in step 3 of proposed protocol. These messages can be used to authenticate the Trusted Party by Alice & Bob respectively.

Third: Alice & Bob derive key from M_a^{REt} & M_b^{REt} respectively, as mentioned in step 4 of the proposed protocol. With the help of $f_{SK}(Id_b, SK)$ & $f_{SK}(Id_a, SK)$ both Alice & Bob can authenticate each other respectively.

ii. Resistance to the Password Guessing Attacks

Attack Situation 1: A malevolent user Bob wants to mount undetectable online password guessing attacks on the proposed protocol.

A malicious attacker may try to guess the password with undetectable online password guessing attacks. If that is the case, the mutual authentication step is not possible. If Bob tries to guess Alice's password, then Bob should perform the following procedure to mount an undetectable online password guessing attack. Bob obtains $(M_a \oplus r_a)^*$ by decrypting $E_{Pwda}(M_a \oplus r_a)$ with a guessed password Pwd_a^*.

Next, he selects his random exponent RE_b and computes $M_b=g^{REb} \bmod p$ and $K_{bt}=M_b^{rb} \bmod p$ to find $h_t(Pwd_b \oplus M_b)$, $f_{Kbt}((M_a \oplus r_a)^*)$ and sends $\{E_{Pwda}((M_a \oplus r_a)^* \oplus M_b), h_t(Pwd_b \oplus M_b), f_{Kbt}((M_a \oplus r_a)^*)\}$ to Trusted Party. Trusted Party authenticates the clients and sends $\{M_b^{REt}, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ to Alice and $\{M_a^{REt}, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$ to Bob. Now client Bob intercepts $\{M_b^{REt}, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ but cannot compare any two terms and verify whether the guessed password is correct or not. Hence, Bob cannot mount an undetectable online password guessing attack on the proposed 3P-EKE protocol.

Attack Situation 2: A malicious user Carol wants to mount Detectable online password guessing attacks on the proposed protocol.

An attacker may try to guess the password with detectable online password guessing attack. She guesses password Pwd_a^* or Pwd_b^* to impersonate Alice or Bob. She chooses her own random exponent RE_a or RE_b and computes $M=g^{REb} \bmod p$ or $M_b=g^{REb} \bmod p$. Now she selects a random number $r_a$or r_b to compute $K_{at}=M_a^{ra} \bmod p$ or $K_{bt}=M_b^{rb} \bmod p$ and then sends the credentials $\{E_{Pwda}^*(M_a \oplus r_a), h_t(Pwd_a \oplus M_a), f_{Kat}(M_a)\}$ or $\{E_{Pwdb}^*(M_b \oplus r_b), h_t(Pwd_b \oplus M_b), f_{Kbt}(M_b)\}$ to Trusted Party. Trusted Party will decrypt $E_{Pwda}^*(M_a \oplus r_a)$ or $E_{Pwdb}^*(M_b \oplus r_b)$ and gets $(M_a \oplus r_a)^*$or $(M_b \oplus r_b)^*$. Now M_a or M_b will get from $h_t(Pwd_a \oplus M_a)$ & Pwd_a or $h_t(Pwd_b \oplus M_b)$ & Pwd_b. Trusted Party computes $r_a= (M_a \oplus r_a)^* \oplus M_a$ or $r_b= (M_b \oplus r_b)^* \oplus M_b$ and finds $K_{at}=M_a^{ra} \bmod p$ or $K_{bt}=M_b^{rb} \bmod p$ to determine $f_{Kat}(M_a)$ or $f_{Kbt}(M_b)$. But the computed hash values will not be equal to the received hash values. Hence Trusted Party can detect this attack and take the counter measure. Hence, it is impossible for an attacker to mount detectable online password guessing attack.

Attack Situation 3: A malicious user Carol wants to mount off-line password-guessing attack on the proposed protocol.

An attacker may try to mount off-line password guessing attack to guess the password. She intercepts $\{E_{Pwda}(M_a \oplus r_a), h_t(Pwd_a \oplus M_a), f_{Kat}(M_a)\}$ and may guess a password, extracts $M_a \oplus r_a$, but it is impossible for her to get M_a until trapdoor is known, which is known only to Trusted Party. This implies that she cannot verify the hash value $f_{Kat}(M_a)$. Hence, offline password guessing attack on the proposed protocol is impossible.

Attack Situation 4: Carol wants to get the session key SK.

Carol may want to get the session key SK agreeing to either $f_{SK}(Id_a, SK)$ or $f_{SK}(Id_b, SK)$. However, this approach cannot work, since it is very hard to retrieve a number according to the hashed value returned by the hash function.

Similarly, there might be a case that, an intruder Carol still cannot get the session key SK by impersonating the Alice/Bob or analyzing the transmitted data because of the same reasons mentioned already.

Trivial Attack

An attacker Carol may directly try to compute the session key from M_a^{REt} or M_b^{REt}. However, due to the one-way hash function and the intractability of the Discrete Logarithmic Problem (DLP), the trivial attack is not possible on the proposed protocol.

Pre-play attacks

An intruder Carol may impersonate Alice to cheat Bob to get the essential information for making Bob convinced that Carol is Alice. However, Carol cannot succeed, since Pwd_a & Pwd_b are unknown to it. Hence, she will encounter the same difficulties as already mentioned.

Replay Attacks

Suppose that Carol intercepts the message $\{Id_a, Id_b, Id_t, E_{Pwda}(M_a \oplus r_a), h_t(Pwd_a \oplus M_a), f_{Kat}(M_a)\}$ send by Alice to cheat Trusted Party. Carol cannot cheat trusted party, because Alice chooses a random number r_a whenever Alice wants to communicate with Bob to perform the password authenticated key exchange protocol. If an attacker Carol just sends $\{Id_a, Id_b, Id_t, E_{Pwda}(M_a \oplus r_a), h_t(Pwd_a \oplus M_a), f_{Kat}(M_a)\}$, then Trusted Party will detect the attack easily, since Pwd_a's & r_a's are different all the time. Consequently, an intruder Carol cannot perform replay attacks successfully.

Man-in-the-middle Attack

Suppose the attacker Carol frames her own message i.e. $\{E_{Pwdc}(M_c \oplus r_c), h_t(Pwd_c \oplus M_c), f_{Kct}(M_c)\}$ with the guessed password Pwd_c and sends it to trusted party. The trusted party will decrypt $E_{Pwdc}(M_c \oplus r_c)$ to get $(M_c \oplus r_c)$' and obtains '$Pwd_c \oplus M_c$' from $h_t(Pwd_c \oplus M_c)$. Hence carol in the middle cannot obtain the correct M_c until its guessed password is correct. Finally, Trusted Party computes hash value which will not match with the received hash value. Hence the protocol gets terminated by not allowing man-in-the-middle to mount any attack.

Server Spoofing

The trusted party computes $f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ & $f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})$ and sends to Alice and Bob, respectively. Now, Alice and Bob can authenticate the trusted party by computing $f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ & $f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})$ respectively. Thus, the Carol cannot imporsonate the trusted party to deceive the client.

Perfect Forward Secrecy

The proposed protocol has the perfect forward secrecy.The session key is computed asfollows: $SK= (M_b^{REt})^{REa} \pmod p = (M_a^{REt})^{REb} \pmod p$. If the Carol gets $\{M_b^{REt}, f_{Kat}(Id_a, Id_b, K_{at}, M_b^{REt})\}$ or $\{M_a^{REt}, f_{Kbt}(Id_a, Id_b, K_{bt}, M_a^{REt})\}$, then in order to obtain the session key, she should know RE_b or RE_a. The session keys generated in different sessions are independent, since RE_a and RE_b are randomly chosen by Alice and Bob, respectively. This indicates that Carol cannot obtain previous session keys even if she obtains the session key used in this run.

Known-Key Security

The proposed protocol assumes that, RE_a and RE_b are randomly chosen by Alice and Bob and are independent among protocol executions. This leads to the invulnerability of Known-Key security.

B. Performance Analysis

The development of an efficient protocol should take the number of transmission rounds and the computation complexity into account. The proposed protocol requires four message transmission rounds.

Transmission Round and Computation Complexity

Table 2 depicts the comparison of performance among Chang-Chang's, Yoon-Yoo's, PSRJ, Raj et.al and the proposed password authenticated 3P-EKE protocols. The invention or design of a proficient protocol should take the number of transmissions (though communication channel) and the complexity of computation into account. The proposed protocol requires (2, 2, 2) number of transmissions from Alice, Bob and Trusted Party respectively. From the table, it is quite obvious that there is no difference at all in complexity of computation among the well-known protocols and the proposed protocol.

The modular exponential operations are not increased due to the inclusion of XOR operation, since client Alice sends $E_{Pwda}(M_a \oplus r_a)$, $h_t(Pwd_a \oplus M_a)$ & $f_{Kat}(M_a)$ to Trusted Party and client Bob sends $\{E_{Pwdb}(M_b \oplus r_b), h_t(Pwd_b \oplus M_b), f_{Kbt}(M_b)\}$ to Trusted Party. By imposing XOR operation between Pwd_a & M_a client Alice can enforce perfect security on the protocol and is the same case with Bob too. Trusted Party first utilizes a trapdoor to obtain $(Pwd_a \oplus M_a)$ & $(Pwd_b \oplus M_b)$ from $h_t(Pwd_a \oplus M_a)$ & $h_t(Pwd_b \oplus M_b)$ and then retrieves $M_a = Pwd_a \oplus M_a \oplus Pwd_a$ & $M_b = Pwd_b \oplus M_b \oplus Pwd_b$ respectively.

Now Trusted Party utilizes Pwd_a & Pwd_b to decrypt $E_{Pwda}(M_a \oplus r_a)$ & $E_{Pwdb}(M_b \oplus r_b)$ and gets $(M_a \oplus r_a)$ & $(M_b \oplus r_b)$. Then it retrieves the values $r_{a=}$ $(M_a \oplus r_a) \oplus M_a$ & $r_{b=}$ $(M_b \oplus r_b) \oplus M_b$ and then computes $K_{at}=(M_a^{ra} \bmod p)$ & $K_{bt}=(M_b^{rb} \bmod p)$ accordingly and verifies whether computed value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$) and received value $f_{Kat}(M_a)$ (or $f_{Kbt}(M_b)$) are identical or not. A total four modular exponential operations are required for the execution of the proposed protocol, which also preserves the computation complexity. The first two modular exponential operations are required for authentication of the clients, while the remaining two modular exponential operations for sending the message to the corresponding clients (Alice and Bob).

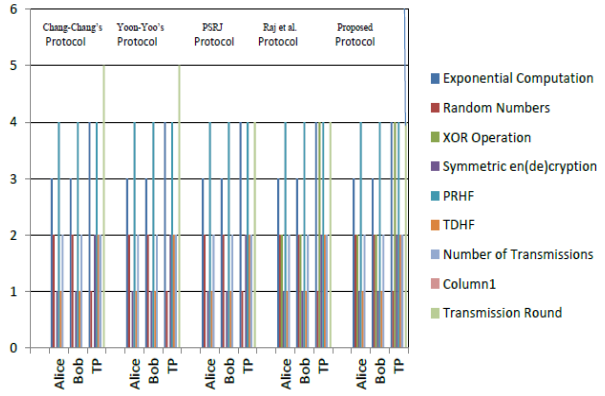

Fig.6. Performance Comparison

The comparison of computation complexities among four 3P-EKE Protocols viz., Chang-Chang's, Yoon-Yoo's, PSRJ, Raj et.al protocols against the proposed password authenticated 3P-EKE protocol is illustrated in Fig 6.Similarly, Table 2 demonstrates the comparison between different types of attacks such as Chang-Chang's, Yoon-Yoo's, PSRJ, Raj et.al protocols as against the proposed password authenticated 3P-EKE protocol. From the table it can be noticed that the proposed protocol is invulnerable to various attacks as mentioned earlier.

Table 2. Performance Analysis of Various 3P-EKE Protocols

3P-EKE Protocols Participants Computation Type & Attack Type	Chang-Chang's protocol			Yoon-Yoo;s protocol			PSRJ protocol			Raj et al. protocol			The Proposed protocol		
	Alice	Bob	TP	Alice	Bob	TP	Alice	Bob	TP	Alice	Bob	TP	Alice	Bob	TP
Exponential Computations	3	3	4	3	3	4	3	3	4	3	3	4	3	3	4
Random numbers	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1
Exclusive OR Operation	0	0	0	0	0	0	0	0	0	2	2	4	2	2	4
Symmetric en(de)cryption	1	1	2	1	1	2	1	1	2	1	1	2	1	1	2
Pseudo Random hash Function(PRHF)Operations	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Trapdoor hash functions (TDHF) Operations	1	1	2	1	1	2	1	1	2	1	1	2	1	1	2
Number of Transmissions	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Transmission Round	5			5			4			4			4		
Undetectable Online Password Guessing Attack	YES			YES			YES			YES			NO		
Detectable Online password Guessing Attack	YES			YES			YES			YES			NO		
Off-line Password Guessing Attack	YES			YES			YES			YES			NO		

VI. Conclusion

Keeping in view of the weaknesses and shortfalls of the existing protocols proposed by various authors like Chang-Chang, Yoon-Yoo, PSRJ and Raj et.al, this paper endeavors to propose a robust, computationally efficient and highly secure password authenticated 3P-EKE Protocol using XOR operations with analogous message transmission. Subsequent review and successful implementation of the proposed protocol has demonstrated its invulnerability to all three types of attacks proposed by Ding & Horster[1]. The proposed protocol also proved that it is not only secured against the attacks proposed by D&H but also secure against all types of attacks such as offline attacks, Replay attacks, Pre-play attacks, Server spoofing, Man-in-middle attack and Trivial attacks. The results show that the proposed protocol is robust, highly secure, efficient and practical for implementation.

References

[1] W. Diffie and M. E. Hellman. "New directions in cryptography". IEEE Transactions on Information Theory, vol.22, no.6, pp.644– 654, 1976.

[2] Y. Ding and P. Horster, "Undetectable on-line password guessing attacks," *ACM Operating Systems Review*, vol.29, no.4, pp.77-86, 1995.

[3] S.M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against password guessing attacks," *in Proc. of 1992 IEEE Symposium on Research in Security and Privacy*, pp.72–84, 1992.

[4] Chin-Chen Chang*, Ya-fen Chang, "A novel three-party encrypted key exchange protocol", Elsevier, Computer Standards & Interfaces 26 (2004) pp.471 – 476.

[5] Eun-Jun Yoon, Kee-Young Yoo, "Improving the novel three-party encrypted key exchange protocol", Elsevier, *Computer Standards and Interfaces*, 30:309-314 , (2008).

[6] H.R. Chung and W.C. Ku, "Three weaknesses in a simple three-party key exchange protocol," *Information Science*, vol.178, no.1, pp.220-229, 2008.

[7] R.Padmavathy, Tallapally Shirisha, M.Rajkumar, Jayadev Gyani, "Improved analysis on Chang and Chang Password Key Exchange Protocol", IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009, pp.781-783.

[8] Ya-Fen Chang, Wei-Cheng Shiao, and Chung-Yi Lin, "Comments on Yoon and Yoo's Three-party Encrypted Key Exchange Protocol", International Conference on Advanced Information Technologies (AIT) in 2009.

[9] R.Parvathy, "IMPROVED THREE PARTY EKE PROTOCOL", ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2010, Vol.39, No.3, pp.220-226.

[10] Shirisha Tallapally, "IMPERSONATION ATTACK ON EKE PROTOCOL", International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010, pp. 114-121.

[11] Archana Raghuvamshi, P.Venkateshwara Rao, and Prof.P.Premchand, "Cryptanalysis of Authenticated Key Exchange 3P-EKE Protocol and its Enhancement", IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012) March 30, 31, 2012, pp.659-666.

[12] P.Rajkumar and C.Manoharan, "Security Enhancement for 3-PEKE protocol using parallel Message transmission", International Journal of Engineering Science and Technology (IJEST),Vol.4, No.8, pp.3767-3772

[13] P.Rajkumar, C.Manoharan, M.Ananthi, "Performance Analysis Of 3pek Exchange Protocol Using Parallel Message Transmission Technique", International Journal

of Engineering Research & Technology (IJERT), Vol. 1 Issue 7, September – 2012, ISSN: 2278-0181, pp.1-5.

[14] Archana Raghuvamshi , Prof.P.Premchand ,"A Weakness in 3pek Exchange Protocol using Parallel Message Transmission Technique", International Journal of Advanced Research in Computer Science(IJARCS), ISSN No. 0976-5697, Volume 4, No. 11, Nov-Dec 2013,pp.104-108.

[15] Y. Gertner, T. Malkin, O. Reingold, "On the impossibility of basing trapdoor functions on trapdoor predicates", Proceedings of the 42nd IEEE Symposium on foundations of Computer Science, Las Vegas, Nevada, October, 2001, pp. 126 – 135.

Authors' Profiles

Archana Raghuvamshi she received her Bachelor's Degree BSc (M.S.Cs), Master's Degrees M.C.A and M.Tech(CSE) from Osmania University, Hyderabad. She did course work in ADS and WMN in IITM (Indian Institute of Technology, Madras). She is perusing PhD (CSE) in JNTUK, Kakinada. She has published four research papers in IEEE Digital library and another four research papers in various peer reviewed International Journals. Her research interest includes Cryptography and Information Security, Security in Cloud Computing etc.

Professional Bodies:

She is a,

1. Professional Member of ACM
2. Member of Professional Body IAENG
3. Member of IACSIT
4. Associate Member of theIRED

Work Experience:

She is having 13+ year of teaching experience. She is working as an Assistant Professor in Dept. of CSE, UCOE, Adikavi Nannaya University, Rajahmundry, India.

Co-Author:

Name: PROF. PREMCHAND PARVATANENI

Academic Achievements:

He received his Bachelor's Degree B.Sc (Engg.) from RIT, Jamshedpur. He received his Master's M.E (CE) from AU (Andhra University), Visakhapatnam. He received his PhD(CSSE) from AU. He has published more than 50 publications in various International Journals and Conference proceedings. His research Interest includes Cryptography and Network Security, Image Processing, Software Engineering etc.

Work Experience:

He is having 40+ years of teaching experience in various Universities. He is working as a Professor in Dept. of CSE, University College of Engineering, Osmania University. He was as a Director in AICTE, New Delhi. And also he has been held with the various positions like Head, Chairman of BOS, Additional Controller of Examinations in Professional wing, Osmania University, Hyderabad.

How to cite this paper: Archana Raghuvamshi, Premchand Parvataneni,"Design of a Robust, Computation-Efficient and Secure 3P-EKE Protocol using Analogous Message Transmission", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.5, pp.9-17, 2016.DOI: 10.5815/ijcnis.2016.05.02