I. J. Computer Network and Information Security, 2016, 6, 59-66
Published Online June 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.06.07

Verifier-based Password Authenticated 3P-EKE Protocol using PCLA Keys

Archana Raghuvamshi
Adikavi 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—This paper endeavors to present a novel framework for the generic structure of a verifier-based password authenticated Three-Party Encrypted Key Exchange (3P-EKE) protocol which yields more efficient protocol than the ones knew before. A previous framework presented by Archana and Premchand is more secured against all types of attacks like password guessing, replay, pre-play, man-in-the-middle attack etc. But unfortunately, this protocol does not solve the problem of a server compromise. These proofs help as inspiration to search for another framework. The framework we offer produces more efficient 3P-EKE protocol, and, in addition, delivers perceptive clarification about the existing attacks that do not solve in the previous framework. Moreover, it allows direct change from a class of verge private-key encryption to a hybrid (symmetric & Asymmetric) one without significant overhead.

Index Terms—Verifier–based protocols, Password – based Authentication, Three Party Encrypted Key Exchange Protocol (3P-EKE), Public-Key Cryptosystem Based on Logarithmic Approach (PCLA).

I. Introduction

A vital job of cryptography is to guard the confidentiality of messages transferred over the unsecured network. To provide security, messages can be encrypted by using a key (secret information) so that an intruder cannot decode the message. However, encoding the messages may not be only the solution, because an intruder may take the most active role as the network is open and reachable. We may need to change the key according to the session to establish a secure communication through the unsecured open network.

Consecutively, a password-authenticated two-party encrypted key exchange (2P-EKE) protocols are used to exchange a session key based on a low-entropy password. In this network, each party who wants to communicate needs to memorize a low-entropy password, which implies high maintenance of passwords. Due to this drawback, a password-authenticated three-party encrypted key exchange (3P-EKE) protocols has its demand as on date. According to Ding & Hoster [1], many of such 3P-EKE protocols suffer from any one of the three types of password guessing attacks. An intruder can guess the correct password by continuously trying until he succeeds, which is known as password guessing attack.

An Ideal password authenticated key exchange protocol should satisfy the security requirements like, Mutual Authentication, Resistant to password guessing attacks, Session Key (SK) security, Resistant to Trivial Attack, Resistant to Pre-play Attack, Resistant to Replay Attack, Resistant to Man-in-the-middle Attack, Server spoofing security, Perfect forward secrecy, backward secrecy, Known-Key Security, etc.

Based on the low entropy passwords shared between a user and a server, the password-authenticated key exchange (PAKE) protocols are classified into two types. They are:

A. Symmetric model

As the name implies, symmetric (identical) low entropy password shared between a Trusted Party (server) and a user in establishing a secure session key. If a Trusted Party is compromised, an intruder will get succeed in performing an attack on the legitimate user.

B. Asymmetric model

As the name implies, the password distribution is asymmetric in nature; i.e., a user will share the different knowledge (a verifier) about the low entropy password with the Trusted Party in establishing a secure session key. If a Trusted Party is compromised, the password table does not reveal the direct information about the password. In this way, a server spoofing is avoided.

Henceforth, the design of a novel framework which is smart in establishing a secure session key with less computational overhead; which prove to be secure against the attacks like password guessing, a server spoofing and also provide mutual authentication, Backward Secrecy, and Forward Secrecy is the need of the hour. This paper endeavors to propose a novel framework for establishing a secure session key based on an asymmetric model by using PCLA keys. PCLA is a new public key

cryptosystem based on the logarithmic approach proposed by Archana et al. in 2012[2].

Further, the rest of the paper is organized as follows: Related works is discussed in section II. In section III, we listed notations used in the proposed protocol. A framework for the proposed protocol is described in section IV. Security Analysis of the proposed protocol is done in section V. Finally, we made concluding remarks in section VI.

II. Related Work

Diffie-Hellman (1976) [3] key exchange protocol is suffered from a man-in-the-middle attack due to the lack of an authentication. To assure a good access control, many applications require a robust client authentication. In such scenario, password-authenticated key exchange (PAKE) protocols have their own identity.

Bellovin and Merritt (1992)[4] have first proposed password-based authenticated encrypted key exchange protocol for the two-party network. But, due to the server compromise (server hacking: e.g., In 2012, more than million LinkedIn passwords are stolen) this protocol no longer proved to be secure. Hence, to eliminate such a problem he proposed an improvement over it known as Augmented EKE protocol (1993) [5], where a server instead of storing the actual passwords, it stores the verifiers of the passwords which prevents from a server compromise but it does not solve the problem of off-line dictionary attacks.

Subsequently, Gong et al.(1993)[6] proposed a three-party password-based authenticated key exchange protocol using a server's public key, where the clients are given a risk to verify and keep the public key safely. Many improvements proposed by various researchers in terms of a security and computational efficiency [7, 8, 9, 10].

Abdalla et al. (2005) [11] proposed a 'provable secure' one-time password-based authentication and key exchange (OPKeyX) technology for grid computing; where a user changes the password from one session to another session to eliminate the problem of password sniffing. Lin et al. (2008)[12] proposed an efficient verifier-based password-authentication key exchange protocol by using elliptic curve cryptography. Unfortunately, Yang et al. (2011) [13] showed the flaws of Lin et al. protocol and proposed an improvement over the Efficient verifier-based password-authentication key exchange protocol via elliptic curves.

A Novel ECC-3PEKE protocol is proposed by Chang et al. (2004) [14], which proved to be practical, efficient and secure. However, Yoon et al. (2008) [15] notified an undetectable online password guessing attack and proposed an improvement over ECC-3PEKE protocol. Subsequently, PSRJ protocol has been proposed by Padmavathy et al. (2009) [16], which is also an improvement over ECC-3PEKE protocol. They claimed that the proposed protocol achieves better computational complexity and also secure against dictionary attacks.

Later Chang et al. (2009) [17] discussed why Yoon-Yoo's Protocol is still insecure. R. Padmavathy (2010) [18] cryptanalyzed the PSRJ protocol and to overcome an attack she proposed an improvement over the existing one by using reduced modular exponentiation operations. Successively, an impersonation attack has been shown on the ECC-3PEKE protocol by Shirisha Tallapally (2010) [19]. Next, Archana et al. (2012) [20] showed detectable online password guessing the attack on PSRJ protocol.

Also, Kulkarni et al.(2007) [21] proposed a novel key exchange protocol based on verifier-based password authentication for three parties; where each client instead of storing the direct password itself it computes a one-way hash function on each password and stores the corresponding result in a server's password table. Subsequently, Shaban et al.(2008)[22] proposed an improvement over the Kulkarni et al.'s protocol in terms of computational complexity, by showing the reduced rounds from 7 to 4 without using symmetric encryption/decryption. But unfortunately, Archana et al. (2015) [23] cryptanalyzed the Shaban et al.'s protocol by showing the detectable online password guessing attack. Kulkarni et al.'s protocol are proved as secure against the dictionary attacks but it is computationally more expensive than our proposed protocol.

A previous framework presented by Archana et al. "in press" [24] is more secured against all types of attacks like password guessing, replay, pre-play, man-in-the-middle attack etc. But unfortunately, this protocol does not solve the problem of a server compromise. These proofs help as inspiration to search for another framework which eliminates the problems, may occur in the previous framework.

III. Notations

The list of notations along with their descriptions used in this paper is given in Table 1. In fact, Id_a, Id_b, Id_{tp} are the identities of client-A, client-B, Trusted Party-TP respectively, which are known publicly.

Table 1. List of Notations

Client-A/Client-B	Two parties who want to communicate with each other
Eve-E	An Attacker
Trusted Party TP	Trusted third party(a server)
Id_a, Id_b ,Id_{tp}	Identities of Client-A, Client-B and Trusted Party-TP
Pwd_a, Pwd_b	Passwords of Client-A & Client-B used to generate Verifiers
V_A, V_B	Verifiers of Client-A & Client-B
NewPwd	New Password for Backward Secrecy
Key_{pu}, Key_{pr}	PCLA Public & Private keys of Trusted Party
$E_{Pwd}()$	A symmetric Encryption scheme with a password pwd
$D_{Pwd}()$	A symmetric Decryption scheme with a password pwd
$EP_{Kev}()$	A Asymmetric Encryption scheme with a PCLA keys
$DP_{Kev}()$	A Asymmetric Decryption scheme with a PCLA keys
p	A large prime number
G	A generator in GF(Group Field)
r_a, r_b	Random numbers chosen by Client-A, Client-B respectively.
RE_a, RE_b ,RE_{tp}	Random Exponents of Client-A, Client-B and Trusted party respectively
M_a , M_b	$M_a=g^{REa} \bmod p$, $M_b=g^{REb} \bmod p$
K_{atp} , K_{btp}	$K_{atp}=M_a^{ra} \bmod p$, $K_{btp}=M_b^{rb} \bmod p$ are one time strong keys shared by Client-A & Client-B with Trusted Party respectively.
$h_{tp}()$	A one-way trapdoor function, where only trusted party knows the trapdoor tp
$F_K()$	A pseudo random hash function indexed by a key K
SK	Session Key

IV. Framework for Proposed Protocol

This section endeavors to propose a novel verifier-based password-authenticated 3P-EKE protocol using the PCLA keys. PCLA is a new public key cryptosystem based on the logarithmic approach proposed by Archana et al. More details about this algorithm (PCLA) are given in the reference paper [2].The proposed protocol has been divided into three stages. They are:

- Initialization Stage
- Key Agreement Stage
- Key Computation Stage

A. *Initialization Stage*

At this 1st stage, the clients who want to communicate each other have to register with Trusted Party in advance.

The procedure for registration is as follows:

Step 0: Client-A and Client-B compute's the verifiers $V_A=H(Id_a, Id_{tp}, Pwd_a)$ and $V_B=H(Id_b, Id_{tp}, Pwd_b)$ by choosing low random passwords Pwd_a and Pwd_b respectively. Now client-A and client-B sends the verifiers V_A & V_B to Trusted Party respectively through a secure channel.

i.e., Client-A→ Trusted Party: $\{V_A\}$, and **Client-B→ Trusted Party**: $\{V_B\}$.

Then Trusted Party stores the verifiers in its password verifier's table. The detail of the initialization stage is depicted in Fig.1.

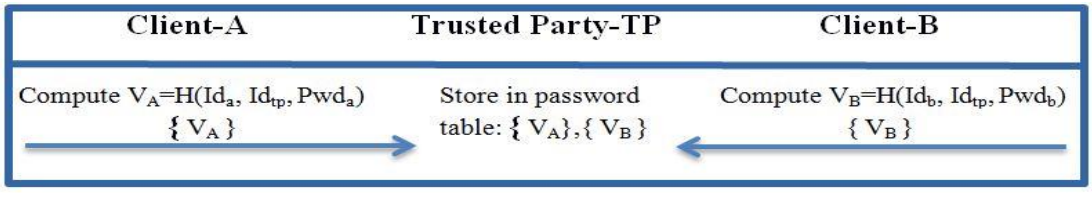

Fig.1. Initialization Stage

B. *Key Agreement Stage*

In this 2nd stage of the protocol, the actual procedure for session key agreement begins. The protocol executes as per the following steps of the protocol.

Step KA1: Client-A generates two random numbers viz., RE_a, $r_a \in Z_R$, and computes $K_{atp}=M_a^{ra} \bmod p$ where $M_a=g^{REa} \bmod p$ and sends the credentials $\{Id_a, Id_b, Id_{tp}, \{EP_{Keypu}(Pwd_a), V_A\}, E_{Pwda}(M_a \oplus r_a), h_{tp}(Pwd_a \oplus M_a), F_{Katp}(M_a)\}$ to Trusted Party.

i.e., Client-A→Trusted Party: $\{Id_a, Id_b, Id_{tp}, \{EP_{Keyp}(Pwd_a), V_A\}, E_{Pwda}(M_a \oplus r_a), h_{tp}(Pwd_a \oplus M_a), F_{Katp}(M_a)\}$.

Similarly, client-B generates two random numbers viz., RE_b, $r_b \in_R Z_P$, and computes $K_{btp}=M_b^{rb} \bmod p$ where $M_b=g^{REb} \bmod p$ and sends the credentials $\{Id_a, Id_b, Id_{tp}, \{EP_{Keypu}(Pwd_b), V_B\}, E_{Pwdb}(M_b \oplus r_b), h_{tp}(Pwd_b \oplus M_b), F_{Kbtp}(M_b)\}$ to Trusted Party.

i.e., Client-B→Trusted Party: $\{Id_a, Id_b, Id_{tp}, \{EP_{Keypu}(Pwd_b), V_B\}, E_{Pwdb}(M_b \oplus r_b), h_{tp}(Pwd_b \oplus M_b), F_{Kbtp}(M_b)\}$.

Step KA2: Upon receiving the credentials from client-

A and client-B, Trusted Party decrypts $EP_{Keypu}(Pwd_a)$ & $EP_{Keypu}(Pwd_b)$ by using its PCLA private key Key_{pr} i.e., $DP_{Keypr}(EP_{Keypu}(Pwd_a))$ & $DP_{Keypr}(EP_{Keypu}(Pwd_b))$ and gets the low entropy passwords Pwd_a, Pwd_b respectively. Now Trusted Party computes $H(Id_a, Id_{tp}, Pwd_a)$ & $H(Id_b, Id_{tp}, Pwd_b)$ and retrieves the verifier V_A & V_B from its table and checks whether both the numbers are equal. If not, then it terminates the protocol at the current session.

If yes, then it implies that client-A & client-B is verified at first level and Trusted Party continues with the residual procedure of the protocol. Trusted party retrieves $Pwd_a \oplus M_a$ & $Pwd_b \oplus M_b$ from $h_{tp}(Pwd_a \oplus M_a)$ & $h_{tp}(Pwd_b \oplus M_b)$ by using trapdoor [25] 'tp' and compute's $M_a=(Pwd_a \oplus M_a) \oplus Pwd_a$ & $M_b=(Pwd_b \oplus M_b) \oplus Pwd_b$ respectively. Now, it gets $r_a=(M_a \oplus r_a) \oplus M_a$ & $r_b=(M_b \oplus r_b) \oplus M_b$ from the credential $E_{Pwda}(M_a \oplus r_a)$ & $E_{Pwdb}(M_b \oplus r_b)$ by decrypting with low entropy password Pwd_a & Pwd_b i.e., $D_{Pwda}(E_{Pwda}(M_a \oplus r_a))$ & $D_{Pwdb}(E_{Pwdb}(M_b \oplus r_b))$. Next, Trusted Party performs the second level of verification by calculating $F_{Katp}(M_a)$ & $F_{Kbtp}(M_b)$ after the computation of $K_{atp}=M_a^{ra} \bmod p$ & $K_{btp}=M_b^{rb} \bmod p$ respectively. That is, it compares the computed value of $F_{Katp}(M_a)$ (or $F_{Kbtp}(M_b)$) with the received value of $F_{Katp}(M_a)$(or $F_{Kbtp}(M_b)$). If not identical, it terminates the protocol at the current session.

If both are identical then verification of client-A & client-B is passed and it continues with the residual procedure of the protocol. Now, a Trusted Party chooses a random exponent $RE_{tp} \in_R Z_P$ to compute $M_b^{REtp} \bmod p$ and $M_a^{REtp} \bmod p$ and encrypts these values with its PCLA private key. Then Trusted Party sends these credentials to client-A and client-B simultaneously.

i.e., Trusted Party→Client-A:$\{EP_{Keypr}(M_b^{REtp} \bmod p)\}$, and **Trusted Party → Client-B**:$\{EP_{Keypr}(M_a^{REtp} \bmod p)\}$.

The detail of key agreement stage is depicted in Fig.2.

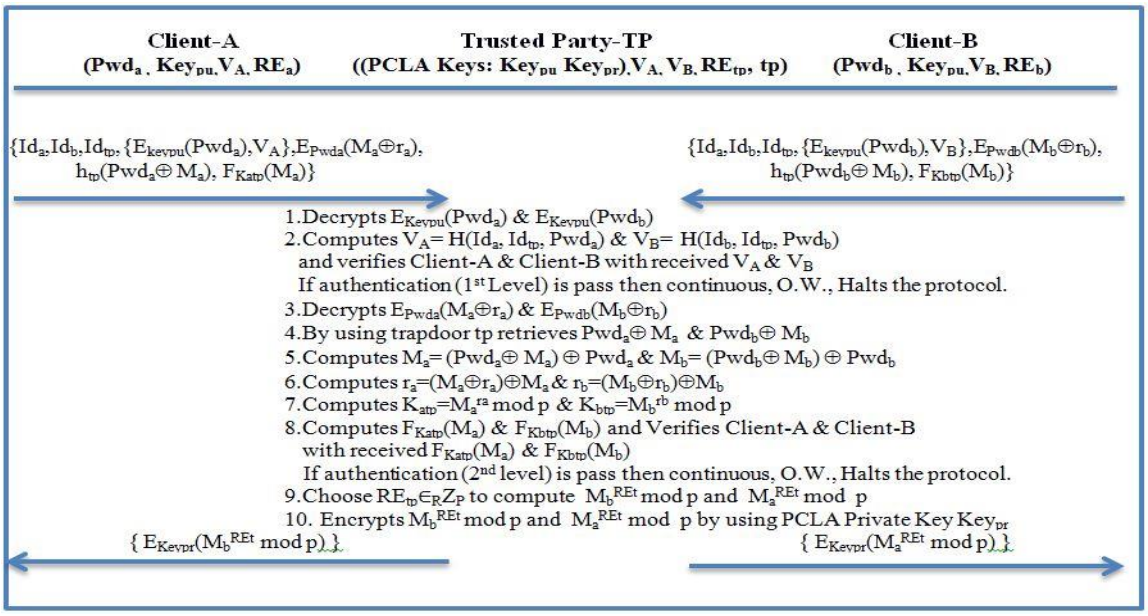

Fig.2. Key Agreement Stage

C. *Key Computation Stage*

In order to compute a secure session key, this stage accomplishes to begin with the verification of the Trusted Party in a smart way.

Step KC1: Upon receiving the credentials from a Trusted Party, client-A decrypts $EP_{Keypr}(M_b^{REtp} \bmod p)$ by using the PCLA public key of Trusted Party i.e., $DP_{Keypu}(EP_{Keypr}(M_b^{REtp} \bmod p))$ to get $M_b^{REtp} \bmod p$. In this way, client-A authenticates the Trusted Party.

Similarly, client-B also authenticates the Trusted Party in the same way.

Now, client-A computes a mutual session key $SK=(M_b^{REtp})^{REa} \bmod p=((g^{REb})^{REtp})^{REa} \bmod p$ & $F_{SK}(Id_a,SK)$ and sends it to client-B. Similarly, client-B computes a mutual session key $SK=(M_a^{REtp})^{REb} \bmod p=((g^{REa})^{REtp})^{REb} \bmod p$ & $F_{SK}(Id_b,SK)$ and sends it to client-A.

i.e., Client-A→ Client-B: $\{F_{SK}(Id_a,SK)\}$, **and Client-B→ Client-A:** $\{F_{SK}(Id_b,SK)\}$.

Step KC2: Upon receiving the incoming credentials $F_{SK}(Id_b, SK)$ and $F_{SK}(Id_a, SK)$ from client-A and client-B respectively, they verify each other and can confirm that the mutual session key is $SK=(M_b^{REtp})^{REa} \pmod p = (M_a^{REtp})^{REb} \pmod p$.

The detail of key computation stage is illustrated in Fig.3.

D. *Password Change Mechanism*

If any one of the client (say Client-A) suspects a 'leak of information', then it invokes a password change mechanism of our proposed protocol, which is helpful in providing backward secrecy. The steps in this mechanism are as follows:

Step PC1: Client-A preserves the session key SK from the previous session.

Step PC2: Client-A, first encrypts the session key SK by using PCLA public key Key_{pu} of Trusted Party. Next,

it computes $H(Id_a, Id_{tp}, Pwd_a) \oplus H(Id_a, Id_{tp}, NewPwd_a) \oplus SK$ & $h_{tp}(H(Id_a, Id_{tp}, NewPwd_a)$ and finally sends the following credentials to Trusted Party to reset the new password verifier.

i.e., Client-A→ Trusted Party: $\{Id_a, Id_{tp}, EP_{Keypu}(SK), H(Id_a, Id_{tp}, Pwd_a) \oplus H(Id_a, Id_{tp}, NewPwd_a) \oplus SK, h_{tp}(H(Id_a, Id_{tp}, NewPwd_a)\}$

Step PC3: Upon receiving the credentials from client-A, Trusted Party compute's SK by decrypting $EP_{Keypu}(SK)$ with the PCLA private key i.e., $DP_{Keypr}(EP_{Keypu}(SK))$ and also retrieves the *old verifier* from the password table and computes the new password verifier as follows: $H(Id_a, Id_{tp}, NewPwd_a) = (H(Id_a, Id_{tp}, Pwd_a) \oplus H(Id_a, Id_{tp}, NewPwd_a) \oplus SK) \oplus H(Id_a, Id_{tp}, Pwd_a) \oplus SK$.

Now, Trusted party check for validation by extracting the *new verifier* from the received credential $h_{tp}(H(Id_a, Id_{tp}, NewPwd_a)$ by using a trapdoor 'tp'; if computed $H(Id_a, Id_{tp}, NewPwd_a)$ is equal to the received one, Trusted Party updates the password table accordingly and sends 'Accepted' message to the client-A. Otherwise, Trusted Party rejects the request by sending the 'Denied' message to the client.

i.e., Trusted Party: Accepted/Denied.

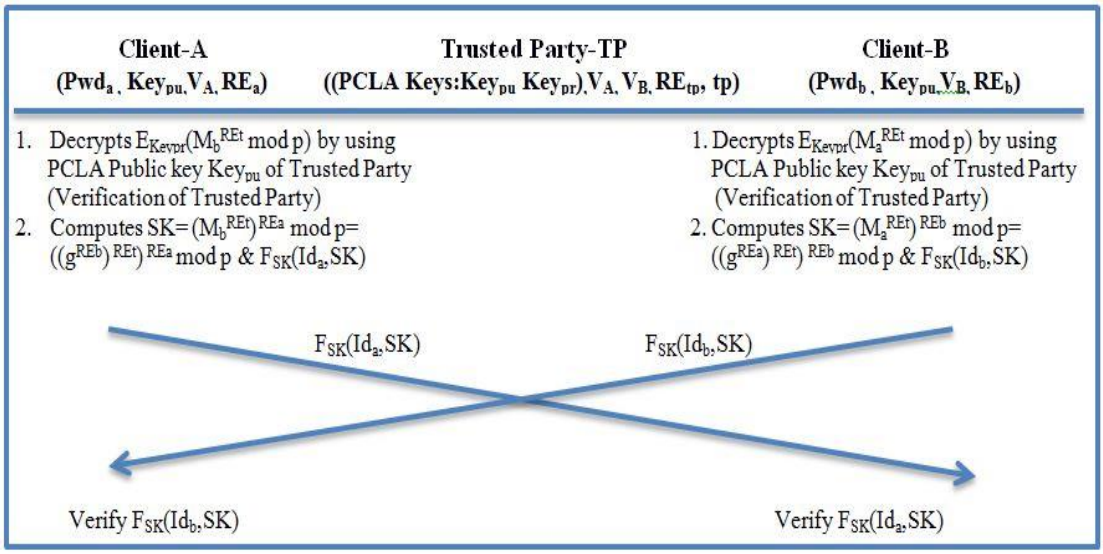

Fig.3. Key Computation Stage

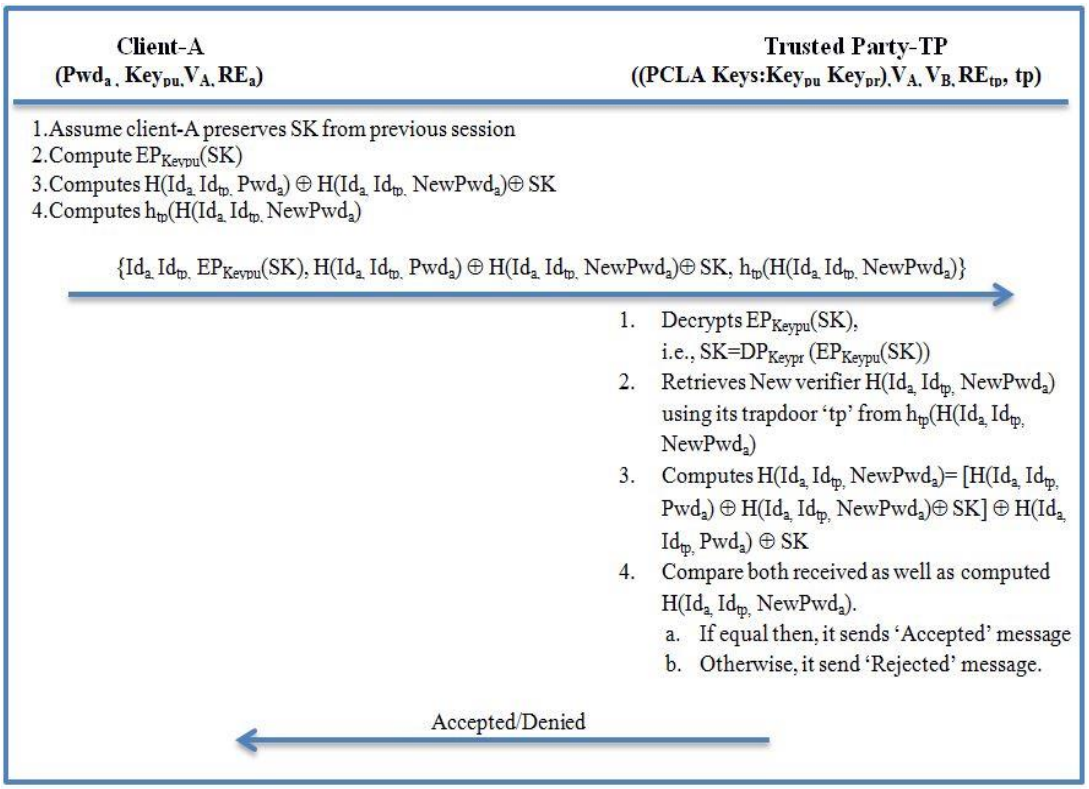

Fig.4. Password Change Mechanism

V. Security Analysis

The following security requirements are satisfied by the proposed protocol; which proves that the proposed protocol is not only efficient but also secure.

A. Resistant to an off-line dictionary attack

An attacker *Eve-E* may try to mount off-line password guessing attack to guess the password. She intercepts $\{Id_a, Id_b, Id_{tp}, \{EP_{Keypu}(Pwd_a),V_A\}, E_{Pwda}(M_a \oplus r_a), h_{tp}(Pwd_a \oplus M_a), F_{Katp}(M_a)\}$ and may guess a password to extract $(M_a \oplus r_a)$, but it is impossible for her to get M_a until trapdoor 'tp' is known, which is known only to Trusted Party. This implies that she cannot verify the hash value $F_{Katp}(M_a)$ which ascertains an offline password guessing attack on the proposed protocol is impossible.

Hence, the proposed protocol is resistant to off-line dictionary attack.

B. Resistant to server spoofing attack

Assume an intruder *Eve-E* succeeds in getting the password table of Trusted Party. Since only the verifier V of clients is stored in password table, *Eve-E* cannot mimic the client and compute SK.

Hence, the proposed protocol is resistant to a server spoofing attack.

C. Provides the mutual authentication

The proposed protocol promotes the mutual authentication and realizes the session key security to a great extent. The following are the scenarios where the mutual authentication can be proved.

- ***First Scenario:*** Client-A and Client-B use the public key Key_{pu} of Trusted Party to hide the corresponding passwords. Only Trusted Party knows the private key Key_{pr} to decrypt it. Hence, for an intruder Eve-E, it is not possible to get the passwords of client-A & client-B.
- ***Second Scenario:*** Client-A and Client-B use the trapdoor 'tp' to hide the random exponents RE_a in M_a & Pwd_a and RE_b in M_b & Pwd_b. Since only Trusted Party knows the trapdoor 'tp' and passwords Pwd_a & Pwd_b he can very well authenticate Client-A and Client-B after receiving the messages sent in *step KA1* of the protocol.
- ***Third Scenario:*** Trusted Party sends $\{EP_{Keypr}(M_b^{REtp} \bmod p)\}$ to client-A & $\{EP_{Keypr}(M_a^{REtp} \bmod p)\}$ to client-B in *step KA2* of the protocol. This message can be used to authenticate Trusted Party.
- ***Fourth Scenario:*** Client-A and Client-B derive a key from M_b^{REtp} and M_a^{REtp} respectively, as mentioned in *step KC1* of the protocol. With the help of $F_{SK}(Id_b, SK)$ & $F_{SK}(Id_a, SK)$ both client-A and client-B can authenticate each other respectively as mentioned in *step KC2* of the protocol.

Hence, the mutual authentication is provided by the proposed protocol.

D. Provides backward secrecy

The proposed protocol can provide backward secrecy, where compromise of Pwd_a will not lead to the compromise of $NewPwd_a$. Assume, a client-A suspects a 'leak of information' to *Eve-E*, then immediately client-A request to Trusted Party to change its password from Pwd_a to $NewPwd_a$. Let us assume, subsequently *Eve-E* intercepted the password change request, i.e., $\{Id_a, Id_{tp}, EP_{Keypu}(SK), H(Id_a, Id_{tp}, Pwd_a) \oplus H(Id_a, Id_{tp}, NewPwd_a) \oplus SK, h_{tp}(H(Id_a, Id_{tp}, NewPwd_a)\}$ sent to Trusted Party by client-A. However, in this process *Eve-E* cannot compute SK by using Pwd_a, hence, he cannot compute $NewPwd_a$ from the intercepted message $\{Id_a, Id_{tp}, EP_{Keypu}(SK), H(Id_a, Id_{tp}, Pwd_a) \oplus H(Id_a, Id_{tp}, NewPwd_a) \oplus SK, h_{tp}(H(Id_a, Id_{tp}, NewPwd_a)\}$.

Hence, the backward secrecy is provided by the proposed protocol.

E. Provides the forward secrecy

The session key is computed as follows: $SK= (M_b^{REtp})^{REa} \pmod p = (M_a^{REtp})^{REb} \pmod p$. If the *Eve-E* gets $\{EP_{Keypr}(M_b^{REtp} \bmod p)\}$ or $\{EP_{Keypr}(M_a^{REtp} \bmod p)\}$, then in order to obtain the session key, she should know the public key of Trusted Party and RE_b or RE_a. The session keys generated in different sessions are independent since RE_a and RE_b are randomly chosen by client-A and client-B respectively. This indicates that *Eve-E* cannot obtain previous session keys even if she obtains the session key used in this run.

Hence, the forward secrecy is provided by the proposed protocol.

VI. Conclusion

In this paper, we proposed a novel verifier-based password authenticated 3P-EKE protocol using PCLA keys, which provides perceptive justification about the existing attacks that do not solve in the previous framework. That is, our proposed protocol is proved to be secure against offline dictionary attacks and server spoofing attack. Further, we have also proved that our protocol provides mutual authentication, backward secrecy and also forward secrecy.

References

[1] Y. Ding and P. Horster. "Undetectable online password guessing attacks," ACM Operating Systems Review vol.29, pp.77-86, 1995.

[2] Archana Raghuvamshi, P.Premchand and P.Venkateswara Rao. "PCLA: A New Public-key Cryptosystem Based on Logarithmic Approach", International Journal of Computer Science Issues(IJCSI), vol.9,no.1, pp.355-359, 2012.

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

[4] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary

attacks", IEEE Symposium on Security and Privacy, IEEE Computer Society Press, pp.72–84 May 1992.
[5] S. M. Bellovin and M. Merritt, "Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise", ACM CCS, ACM Press vol.93, pp.244–250, November 1993.
[6] L. Gong, M. Lomas, R. Needham, and J. Saltzer, "Protecting poorly chosen secrets from guessing attacks", *IEEE Journal on Selected Areas in Communications*, vol.11,no.5,pp. 648-656, 1993.
[7] W.M. Li, and Q.Y. Wen, "Efficient verifier-based password-authentication key exchange protocol via elliptic curves", *Proceedings of 2008 International Conference on Computer Science and Software Engineering*, pp. 1003-1006, 2008.
[8] E.J. Yoon, and K.Y. Yoo, "Robust User Password Change Scheme based on the Elliptic Curve Cryptosystem", *Fundamenta Informaticae*, pp 483-492, 2008.
[9] Zeng, Yong and Ma, Jianfeng, "An improvement on a password authentication scheme over insecure networks" *Journal of Computational Information Systems*, vol.5, no.4, pp.1331-1336, 2009.
[10] Chunling Liu, Yufeng Wanga and Qinxi Bai, "A New Three-party Key Exchange Protocol Based on Diffie-Hellman," I.J. Wireless and Microwave Technologies, vol. 1, no.4, pp. 65-69, 2011.
[11] M. Abdalla, O. Chevassut, and D. Pointcheval. "One-time verifier-based encrypted key exchange", PKC LNCS, Springer, vol. 3386, pp.47–64, January 2005.
[12] W.M. Lin, and Q.Y. Wen, "Efficient verifier-based password-authentication key exchange protocol via elliptic curves", *Proceedings of 2008 International Conference on Computer Science and Software Engineering*, pp.1003-1006, 2008.
[13] Junhan YANG and Tianjie CAO, **"**A Verifier-based Password-Authenticated Key Exchange Protocol via Elliptic Curves", Journal of Computational Information Systems, Binary Information Press, pp.548-553, 2011.
[14] Chin-Chen Chang and Ya-fen Chang, "A novel three-party encrypted key exchange protocol", Elsevier, Computer Standards & Interfaces, vol.26 pp.471 – 476, 2004.
[15] Eun-Jun Yoon, and Kee-Young Yoo, "Improving the novel three-party encrypted key exchange protocol", Elsevier, *Computer Standards and Interfaces*, vol. 30, pp.309-314, 2008.
[16] R.Padmavathy, Tallapally Shirisha, M.Rajkumar, and Jayadev Gyani, "Improved analysis on Chang and Chang Password Key Exchange Protocol", IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp.781-783, 2009.
[17] 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), 2009.
[18] R. Padmavathy, "Improved Three Party Eke Protocol", Information Technology and Control, Vol.39, No.3, pp.220-226, 2010.
[19] Shirisha Tallapally, "Impersonation Attack on EKE Protocol", International Journal of Network Security & Its Applications (IJNSA), vol.2, no. 2, pp. 114-121, 2010.
[20] 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), pp.659-666, March 30, 31, 2012.
[21] S. Kulkarni, D. Jena, and S.K. Jena, "A Novel Secure Key Agreement Protocol using Trusted Third Party", Computer Science and Security Journals (IJCSS), vol.1, no.1, pp. 11 – 18, 2007.
[22] Dina Nabil Shaban, Maged H. Ibrahim, and Zaki B.Nossair, **"**Enhanced Verifier-Based Password Authenticated Key Agreement Protocol For Three-Parties", Journal of Engineering Sciences, vol. 36, no. 6, pp.1513- 1522, 2008.
[23] Archana Raghuvamshi and Premchand Parvataneni. "Cryptanalysis of Verifier-Based Password-Authenticated Key Agreement Protocol for Three Parties", Research Journal of Recent Sciences. Vol. 4, pp. 5-8, Feb 2015.
[24] Archana Raghuvamshi and Premchand Pavataneni, "Design of a Robust, Computation-Efficient and Secure 3P-EKE Protocol using Analogous Message Transmission", International Journal of Computer Network and Information Security (IJCNIS), In Press.
[25] Y. Gertner, T. Malkin, and 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, , pp. 126 – 135, October 2001.

Authors' Profiles

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

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 Ph.D. (CSE) in JNTUK, Kakinada. She published four research papers in IEEE Digital library and another six research papers in various peer reviewed International Journals. Her research interest includes Cryptography and Information Security, Security in Cloud Computing etc.

Ms. Archana Raghuvamshi is a,

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

Prof. Premchand Parvataneni is presently working as a professor in Department of Computer Science and Engineering at University College of Engineering, Osmania University, Hyderabad (Telangana). 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 Ph.D.(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.

Prof.Premchand is having 40+ years of teaching experience

in various Universities. He was as a Director in AICTE, New Delhi. And also, he has been held for the various positions like Head, Chairman of BOS, Additional Controller of Examinations in the Professional wing, Osmania University, Hyderabad.

How to cite this paper: Archana Raghuvamshi, Premchand Parvataneni,"Verifier-based Password Authenticated 3P-EKE Protocol using PCLA keys", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.6, pp.59-66, 2016.DOI: 10.5815/ijcnis.2016.06.07

