Cryptanalysis on a Three Party Key Exchange Protocol-STPKE'

Shirisha Tallapally* and R.Padmavathy**

Abstract—In the secure communication areas, three-party authenticated key exchange protocol is an important cryptographic technique. In this protocol, two clients will share a human-memorable password with a trusted server, in which two users can generate a secure session key. On the other hand the protocol should resist all types of password guessing attacks. Recently, STPKE' protocol has been proposed by Kim and Choi. An undetectable online password guessing attack on STPKE' protocol is presented in the current study. An alternative protocol to overcome undetectable online password guessing attacks is proposed. The results show that the proposed protocol can resist undetectable online password guessing attacks. Additionally, it achieves the same security level with reduced random numbers and without XOR operations. The computational efficiency is improved by ≈ 30% for problems of size ≈ 2048 bits. The proposed protocol is achieving better performance efficiency and withstands password guessing attacks. The results show that the proposed protocol is secure, efficient and practical.

Keywords—STPKE' Protocol, The Proposed Protocol, Undetectable Online Password Guessing Attack

1. INTRODUCTION

In the secure communication areas, three party key exchange protocol is one of the most important cryptographic mechanisms. By this, users can communicate over a public unreliable channel and can agree a secure session key. As the password based authenticated key exchange protocols [1, 2] require users only to remember a human memorable (low-entropy) password, it is rather simple and efficient. From this point of view, PAKE provides convenience and mobility. Password-based authenticated key exchange protocols, however, are vulnerable to password guessing attacks [3] since users usually choose easy-to-remember passwords. Unlike typical private keys, the password has limited entropy, and is constrained by the memory of the user. Thus, the password guessing attacks on password-based authenticated key exchange protocols should be considered realistic. 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 [3]:

Detectable online password guessing attacks: An attacker attempts to use a guessed pass-

Manuscript received December 16, 2009; revised February 16, 2010; accepted February 25, 2010.
Corresponding Author: R.Padmavathy
* Vaagdevi College of Engineering, Warangal, Andra Pradesh, India (shirisha27@yahoo.co.in)
** National Institute of Technology, Warangal, Andra Pradesh, India (r_padma@rediffmail.com)

word in an online 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 online password guessing attacks: Similar to the above, an attacker tries to verify a password guess in an online transaction. However, a failed guess cannot be detected and logged by the server, as the server is not able to distinguish an honest request.

Offline password guessing attacks: An attacker guesses a password and verifies his/her guess offline. No participation from the server is required, so the server does not notice the attack as a malicious one.

The popular three-PAKE protocols are designed for the client-client-server architecture, in which each client shares his password with a trusted server and resorts to the server to authenticate the peer for establishing a session key. In 2004, Lee et al. [5] presented two enhanced three-party encrypted key exchange (TPEKE) protocols without using public-key techniques, and showed that their protocols can resist several attacks. In 2005, Wen et al. [6] proposed a three-PAKE protocol using Weil pairing and claimed that their protocol is provably secure against active adversaries in the random oracle model [7-10]. However, Nam et al. [11] showed that Wen et al.'s protocol cannot resist a man-in-the-middle attack.

Recently, Lu and Cao [12] proposed a simple three-party key exchange (STPKE) protocol based on the chosen-basis computational Diffie-Hellman (CCDH) assumption. They claimed that their protocol can resist various attacks and is superior to similar protocols with respect to efficiency. Kim and Choi [13] found that the STPKE protocol is vulnerable to undetectable online password guessing attacks by using formal description and proposed an alternative protocol (STPKE' protocol).

In this paper, it is presented that, STPKE' protocol is vulnerable to an undetectable online password guessing attack, if the identity of the client is exposed, and proposes an alternative protocol by modifying STPKE protocol[12].

Moreover, the proposed protocol is analyzed on a set of experiments.

The paper is organized as follows: Section 2 describes STPKE' protocol, Section 3 presents the undetectable online password guessing attack on STPKE' protocol, Section 4 discusses the proposed protocol, section 5 reports the experimental results, security and efficiency analysis, and the concluding remarks are made in section 6.

2. REVIEW OF THE STPKE' PROTOCOL (KIM AND CHOI PROTOCOL)

Notations and definition (STPKE' protocol and The proposed protocol)

Notation: Definition

A, B: Clients of a protocol run

S: A trusted server

IDx: Identity of participant x

P: A large prime

g: A generator of order p

G: A finite cyclic group of g

M, N: Two elements in G

PW_A: The password shared between A and S

PW_B: The password shared between B and S
H, H': Two distinct one-way hash functions
SK: The session key
Fig.1. illustrates STPKE' protocol (Kim and Choi Protocol).

Step 1a: A chooses $x, a \in Z_p$ and computes $X \leftarrow (g^x \oplus g^a).M^{PW_A}$, $ID_A^{'} \leftarrow ID_A.g^a$ and sends $ID_A^{'} \| X$ to B.

Step 1b: B chooses $y, b \in Z_p$ and computes $Y \leftarrow (g^y \oplus g^b).N^{PW_B}$, $ID_B^{'} \leftarrow ID_B.g^b$ and sends $ID_A^{'} \| X \| ID_B^{'} \| Y$ to S.

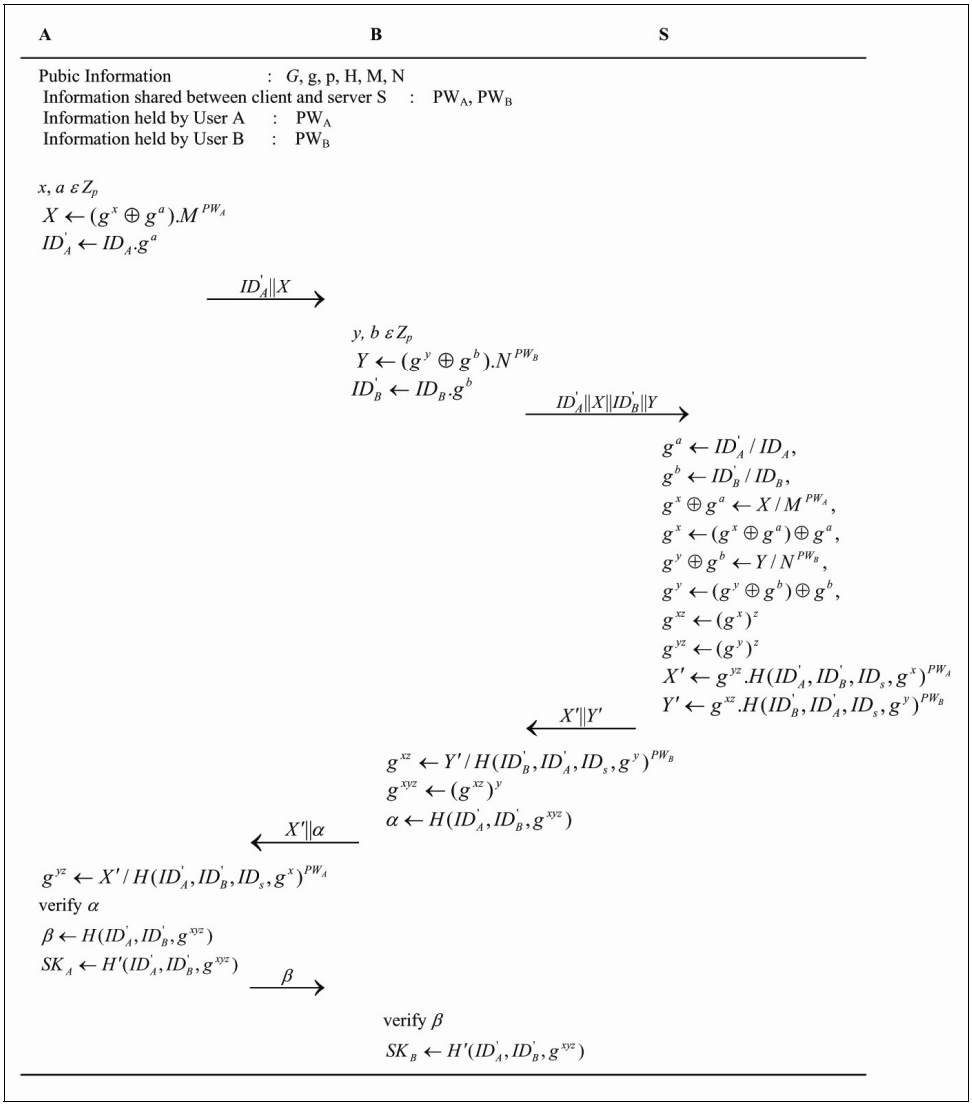

Fig. 1. STPKE' protocol (Kim and Choi protocol)

Step 2a: Upon receiving $ID_A^{'} \| X \| ID_B^{'} \| Y$, 'S' finds $g^a \leftarrow ID_A^{'} / ID_A$, $g^b \leftarrow ID_B^{'} / ID_B$, $g^x \oplus g^a \leftarrow X / M^{PW_A}$, $g^x \leftarrow (g^x \oplus g^a) \oplus g^a$, $g^y \oplus g^b \leftarrow Y / N^{PW_B}$, $g^y \leftarrow (g^y \oplus g^b) \oplus g^b$, then, 'S' chooses a random number $Z \in Z_p$ to compute $g^{xz} \leftarrow (g^x)^z$ and $g^{yz} \leftarrow (g^y)^z$

Step 2b: 'S' computes $X' \leftarrow g^{yz}.H(ID_A^{'}, ID_B^{'}, ID_s, g^x)^{PW_A}$, and $Y' \leftarrow g^{xz}.H(ID_B^{'}, ID_A^{'}, ID_s, g^y)^{PW_B}$ and sends $X' \| Y'$ to B.

Step 3a: Upon receiving $X' \| Y'$, from S, B computes $g^{xz} \leftarrow Y' / H(ID_B^{'}, ID_A^{'}, ID_s, g^y)^{PW_B}$ and uses y to find $g^{xyz} \leftarrow (g^{xz})^y$. Then B computes $\alpha \leftarrow H(ID_A^{'}, ID_B^{'}, g^{xyz})$ and forwards $X' \| \alpha$ to A

Step 3b: Upon receiving $X' \| \alpha$ from B, A computes $g^{yz} \leftarrow X' / H(ID_A^{'}, ID_B^{'}, ID_s, g^x)^{PW_A}$ and uses x to find $g^{xyz} \leftarrow (g^{yz})^x$. Then A computes $\alpha \leftarrow H(ID_A^{'}, ID_B^{'}, g^{xyz})$, then verifies whether computed α is equal to received α. If both are equal, then A authenticates B, finds $\beta \leftarrow H(ID_A^{'}, ID_B^{'}, g^{xyz})$ and the session key $SK_A \leftarrow H'(ID_A^{'}, ID_B^{'}, g^{xyz})$. A forwards β to B.

Step 3c: Upon receiving β, B computes $\beta \leftarrow H(ID_A^{'}, ID_B^{'}, g^{xyz})$ and verifies whether computed β is equal to received β. If both are equal then B authenticates A. Then B computes $SK_B \leftarrow H'(ID_A^{'}, ID_B^{'}, g^{xyz})$ as the session key for securing subsequent communications with A.

3. UNDETECTABLE ONLINE PASSWORD GUESSING ATTACK ON STPKE' PROTOCOL

Since identities of clients are not generally secret, there is a chance that they may get exposed. If ID_A is exposed then STPKE' protocol falls to undetectable online password guessing attacks.

A chooses $x, a \in Z_p$ and computes $X \leftarrow (g^x \oplus g^a).M^{PW_A}$, $ID_A^{'} \leftarrow ID_A.g^a$ and sends $ID_A^{'} \| X$ to B. Now 'B' collects $ID_A^{'}, X$ and performs following calculations:

Step 1: Divide $ID_A^{'}$ with ID_A i.e. nothing but g^a
Step 2: Guess a password $PW_A^{'}$
Step 3: Compute $M^{PW_A^{'}}$
Step 4: Divide X with $M^{PW_A^{'}}$ (i.e. nothing but $(g^x \oplus g^a)$)
Step 5: Now from step (1) g^a is obtained
Step 6: Find $(g^x \leftarrow (g^x \oplus g^a) \oplus g^a)$
Step 7: Let the obtained $g^y \leftarrow g^x$
Step 8: Choose $b \in Z_p$
Step 9: Compute $Y \leftarrow (g^y \oplus g^b).N^{PW_B}$
Step 10: Compute $ID_B^{'} \leftarrow ID_B.g^b$

Now, B sends $ID_A^{'} \| X \| ID_B^{'} \| Y$ to S. Upon receiving $ID_A^{'} \| X \| ID_B^{'} \| Y$, 'S' finds $g^a \leftarrow ID_A^{'} / ID_A$. $g^b \leftarrow ID_B^{'} / ID_B$, $g^x \oplus g^a \leftarrow X / M^{PW_A}$, $g^x \leftarrow (g^x \oplus g^a) \oplus g^a$ $g^y \oplus g^b \leftarrow Y / N^{PW_B}$, $g^y \leftarrow (g^y \oplus g^b) \oplus g^b$. Then 'S' chooses a random number $z \in Zp$ to compute $g^{xz} \leftarrow (g^x)^z$ and $g^{yz} \leftarrow (g^y)^z$. 'S' also computes $X' \leftarrow g^{yz}.H(ID_A^{'}, ID_B^{'}, ID_S, g^x)^{PW_A}$ and $Y' \leftarrow g^{xz}.H(ID_B^{'}, ID_A^{'}, ID_S, g^y)^{PW_B}$ and sends $X' \| Y'$ to B.

Step 11: Upon receiving $X' \| Y'$ from S, B computes $g^{xz} \leftarrow Y' / H(ID_B^{'}, ID_A^{'}, ID_s, g^y)^{PW_B}$
Step 12: Now, take the guessed password $PW_A^{'}$ of step (2)
Step 13: Find $H(ID_A^{'}, ID_B^{'}, ID_s, g^x)^{PW_A^{'}}$ [g^x is obtained from step(6)]
Step 14: B divides X' with $H(ID_A^{'}, ID_B^{'}, ID_s, g^x)^{PW_A^{'}}$ [this is nothing but g^{yz}]
Step 15: Check whether this is equal to the result of step 11.

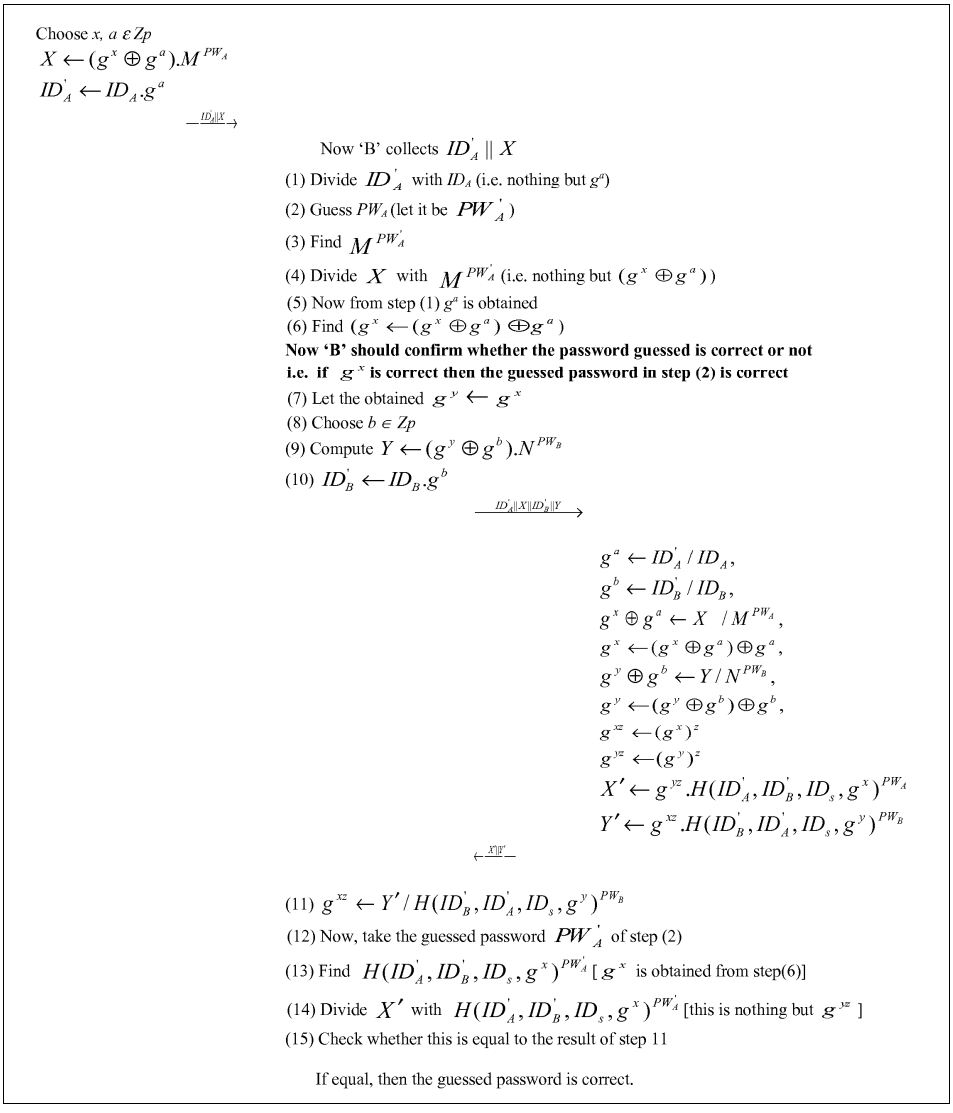

Fig. 2. Undetectable on-line password guessing attack on STPKE' protocol

If equal, then the password guessed is correct.

4. THE PROPOSED PROTOCOL

Step 1a: A chooses a random number $x \in Z_p$ and computes $X \leftarrow g^x.M^{PW_A}$, and then sends $ID_A \| X$ to S.

Step 1b: B chooses a random number $y \in Zp$ and computes $Y \leftarrow g^y.N^{PW_B}$, and then sends $ID_B \| Y$ to S.

Step 2a: Upon receiving $ID_A \| X$ and $ID_B \| Y$, S uses M^{PW_A} and N^{PW_B} to compute

$g^x \leftarrow X / M^{PW_A}$ and $g^y \leftarrow Y / N^{PW_B}$, respectively.

Step 2b: Then, S chooses a random number $z \in Zp$ to compute $g^{xz} \leftarrow (g^x)^z$ and $g^{yz} \leftarrow (g^y)^z$ and then, S computes $X' \leftarrow g^{yz}.H(ID_A, ID_s, g^x)^{PW_A}$ and $Y' \leftarrow g^{xz}.H(ID_B, ID_s, g^y)^{PWB}$, and sends X' to A and Y' to B.

Step 3a: Upon receiving Y', B uses PW_B to compute $g^{xz} \leftarrow Y' / H(ID_B, ID_s, g^y)^{PW_B}$ and then uses y to compute $g^{xyz} \leftarrow (g^{xz})^y$. Then, B computes $\alpha \leftarrow H(ID_A, ID_B, g^{xyz})$ and forwards α to A.

Step 3b: Upon receiving X', A computes $g^{yz} \leftarrow X' / H(ID_A, ID_s, g^x)^{PW_A}$, $g^{xyz} \leftarrow (g^{yz})^x$ and $H(ID_A, ID_B, g^{xyz})$. If the computed $H(ID_A, ID_B, g^{xyz})$ equals the received α, A is convinced that g^{xyz} is valid. Otherwise, A terminates this protocol run. Then, A computes $\beta \leftarrow H(ID_B, ID_A, g^{xyz})$ and then returns β to B. In addition, A computes $SK_A \leftarrow H'(ID_A, ID_B, g^{xyz})$ as the session key for securing subsequent communications with B.

Step 3c: Upon receiving β, B computes $H(ID_B, ID_A, g^{xyz})$. If the computed $H(ID_B, ID_A, g^{xyz})$ equals the received β, B computes $SK_B \leftarrow H'(ID_A, ID_B, g^{xyz})$ as the session key for securing subsequent communications with A. Otherwise, B terminates this protocol run.

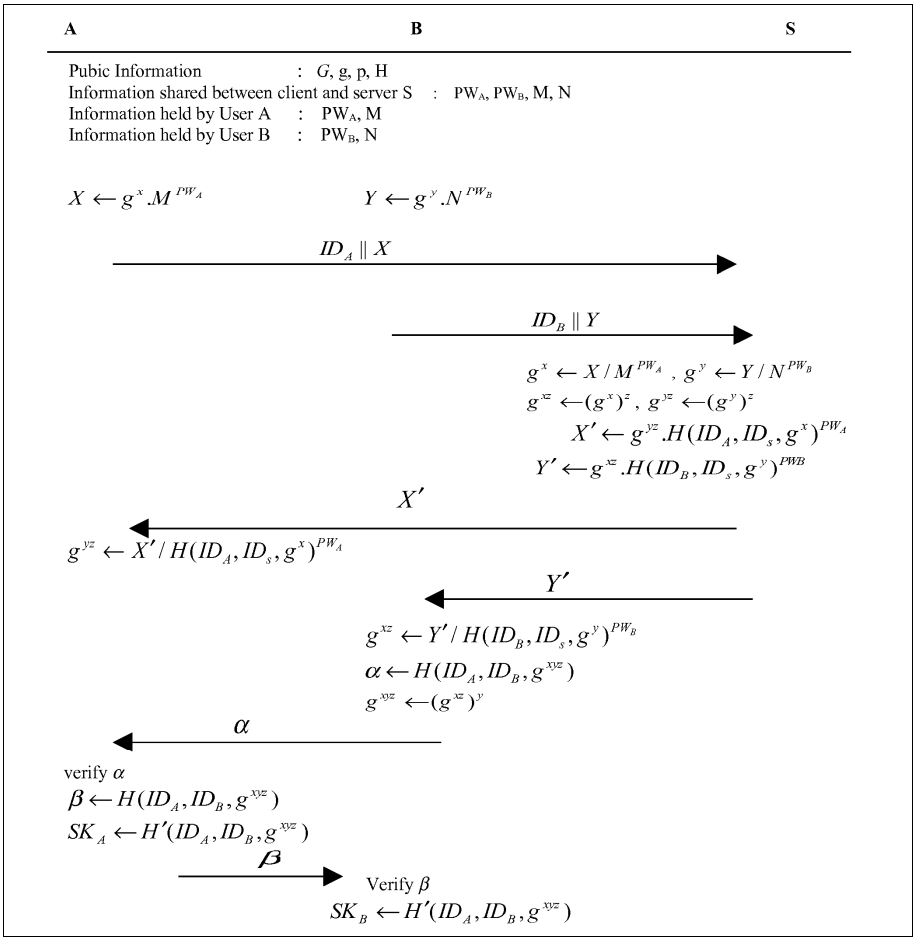

Fig. 3. The proposed protocol

5. SECURITY AND EFFICIENCY ANALYSIS

5.1 Transmission round and computation complexity

In the PAKE research field, the efficiency of a protocol is measured in terms of number of transmission rounds (steps) and the computation complexity. Table 1 shows the comparison analyses of the proposed protocol and the STPKE' protocol. From the view point of the transmission round, our protocol adopts the parallel message transmission mechanism (i.e A→S and B→S) to achieve a higher security level than the protocol proposed by Kim and Choi (i.e. A→B→S). Additionally, it maintains higher security level even after deleting the XOR operations when compared with the STPKE' protocol.

Table 1. Comparison between STPKE' protocol and The proposed protocol

	STPKE' protocol			The Proposed Protocol		
Communication Party	A	B	S	A	B	S
XOR operations required	1	1	2	-	-	-
Random numbers used	2	2	1	1	1	1

5.2 Resistance to the password guessing attacks

First, A, directly sends the message $X \leftarrow g^x.M^{PW_A}$ to server S, and B directly sends the message $Y \leftarrow g^y.N^{PW_B}$ to server S. Hence, there is no way to guess the password of A by B. In STPKE' protocol, the message $X \leftarrow (g^x \oplus g^a).M^{PW_A}$ has been sent through B, hence 'B' is trying to guess the password. In our proposed protocol there is no way for B to guess A' s password or A to guess B' s password.

Second, if the message is $X \leftarrow g^x.M^{PW_A}$ trapped by third party, then he cannot retrieve any information from the obtained message, since M is a random element shared between client A and the server which cannot be guessed by an attacker. Hence, the third party cannot mount any attack.

Third, even if the password PWA is guessed by the attacker, then, in order to get the correct key from the server he should have definite knowledge of 'M', which is impossible.

Perfect forward secrecy: The enhanced protocol has the perfect forward secrecy. The session key is computed as follows: $SK_A \leftarrow H'(ID_A, ID_B, g^{xyz})$ and $SK_B \leftarrow H'(ID_A, ID_B, g^{xyz})$. If the attacker gets X' and Y' in order to obtain the session key, he should know x or y . Since this is not possible he cannot get the key.

The session keys generated in different sessions are independent since x and y are randomly chosen by A & 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 x, y are randomly chosen by A & B, and are independent among protocol executions. This leads to the in-vulnerability of Known-Key security.

Man-in the middle attack: Suppose the attacker frames his own message i.e. $X \leftarrow g^x.M^{PW_A}$ with the correctly guessed password and some 'M', sends to the server. The server will find $g^x \leftarrow X / M^{PW_A}$. Finaly, it computes the hash value with the actual 'M' value and sends back to the client, but since it was calculated with the 'M' value, the key will not be correct, therefore not

allowing man-in the middle to mount any attack.

Trivial attack: An attacker may directly try to compute the session key from α or β. 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 hash function is used, the proposed protocol is in-vulnerable to this attack.

Unknown-key sharing attack: Since M and N are not public, M is shared between client A and the server, N is shared between client B and the server, so there is no chance for an unknown-key sharing attack.

Undetectable online dictionary attack: Since M and N are not public, M is shared between client A and the server, N is shared between client B and the server, so there is no chance for an ousider to mount an undtectable online dictionary attack.

5.3 Experimental results

The proposed protocol is implemented on a data set of problems (p), random numbers of size ranging from 128 bits-2048 bits. Apart from the transmission rounds, the running time for computational steps needed in the protocol is analyzed. Table 2 shows the comparison between

Table 2. Comparison of running time between STPKE' protocol and The proposed protocol

Sl. no.	Prime (p)	Running time of STPKE' protocol (microsec)	Running time of proposed protocol (microsec)
1	3530596818954837594058363570277060586834520000000000 0000000000000000000000000000000000 00000000000229894 811866244249937110343878590800111729769 9	107996	75274
2	4423203904733101730044905437206937176754800000000000 00000000000000000000000000000000 000000000 0000288016 92225669591915187401754372971426438880 3	107980	89881
3	5254302663905376133508798613248963164934800000000000 000000000000000000000000000000000000000 0000003 42133 9179601985669334254217017062364847295 03	73306	59130
4	1651211729563900708030523437487431538182425400000000 00000000000000000000000000000000 0000000000000107518 651770553394603407533631994104608748629921 1	111666	86300
5	3814336501628753502699245308812069671745579552083544 69402441808891292308580580746717399	10545	6664
6	2872187438932867994080957856955856788169335276483966 81212247460911159847224901457171	10281	6200
7	2000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000 0000000000000000000051447	151758	135950.8
8	2000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000501759	602984	483486.6

STPKE' protocol and the proposed protocol with respect to computation time for some selected list of problems.

6. CONCLUSION

An enhanced password-based key exchange protocol which is in-vulnerable to undetectable online password attacks is proposed. The proposed protocol is achieving better performance efficiency by requiring less numbers of random elements. The performance is improved by ≈ 30% for problems of size 2048 bits. The above theoretical and experimental results show that the proposed protocol is secure, efficient and practical.

REFERENCE

[1] Chen TH, Lee WB. A new method for using hash functions to solve remote user authentication, Comput Electr Eng, v34 (1), pp.53-62, 2008.
[2] Yeh HT, Sun HM. Password authenticated key exchange protocols among diverse network domains, Comput Electr Eng, v31(3) pp.175-189, 2005.
[3] Ding Y, Horster P. Undetectable on-line password guessing attacks. ACM Operat Syst Rev v29 (4), pp.77-86, 1995.
[4] Bellovin SM, Merritt M. Encrypted key exchange: password-based protocols secure against dictionary attacks. In: Proceedings of the 1992 IEEE symposium on research in security and privacy, pp.72-84, 1992.
[5] Lee TF, Hwang T, Lin CL. Enhanced three-party encrypted key exchange without server public keys. Comput Secur, v23 (7), pp.571-577, 2004.
[6] Wen HA, Lee TF, Hwang T. Provably secure three-party password-based authenticated key exchange protocol using weil pairing. IEE Proc Commun, v152 (2), pp.138-143, 2005.
[7] Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. ACM annual conference on computer and communications security, pp.62-73, 1993.
[8] Bellare M, Rogaway P. Entity authentication and key distribution. In: Proceedings of the Crypto 93, LNCS, v773, pp.232-249, 1993.
[9] Bellare M, Rogaway P. Provably secure session key distribution: the three party case. In: Proceedings of the 27th ACM symposium on the theory of computing; pp.57-66, 1995.
[10] Bellare M, Pointcheval P, Rogaway P. Authenticated key exchange secure against dictionary attacks. In: Proceedings of the Eurocrypt'00, LNCS, v1807, pp.139-155, 2000.
[11] Nam J, Lee Y, Kim S, Won D. Security weakness in a three-party pairing-based protocol for password authenticated key exchange. Inform Sci, v177(6), pp.1364-1375, 2007.
[12] Lu R, Cao Z. Simple three-party key exchange protocol. Comput Secur , v26(1), pp.94-97, 2007.
[13] Kim, Choi. Enhanced Password-based simple three-party Key exchange protocol. Computers and Electrical Engineering , v35(1), pp.107-114, 2009.

Shirisha Tallapally

She received her B.Tech degree from the National Institute of Technology, Warangal and M.Tech in Computer Science and Engineering from Jayamukhi Institute of Technological sciences affiliated to JNTU. Currently, she is working as Assistant professor at Vaagdevi College of Engineering, Warangal. Her research interests lie in cryptography and network security.

R.Padmavathy

She received her B.E degree from Bharathiar University and M.Tech degree from Andhra University. She is a research scholar and has submitted her Ph.D thesis at the University of Hyderabad, INDIA. At present she is working as an assistant professor at the National Institute of Technology, Warangal. She has published papers in International Conferences and Journals. Her research interest includes Information security, Cryptanalysis and Network Security.