Research Journal of Recent Sciences _____ ISSN 2277-2502
Vol. 4**(ISC-2014),** 5-8 **(2015)** *Res. J. Recent. Sci.*

Cryptanalysis of Verifier-Based Password Authenticated Key Agreement Protocol for Three Parties

Archana Raghuvamshi[1] and P. Premchand[2]
[1]Department of Computer Science and Engineering, Adikavi Nannaya University, Rajahmundry, Andhra Pradesh, INDIA
[2]Department of Computer Science and Engineering, Osmania University, Hyderabad, Andhra Pradesh, INDIA

Available online at: www.isca.in, www.isca.me
Received 5th November 2014, revised 22nd January 2015, accepted 30th January 2015

Abstract

In this modern era of communication world even minor task requires to perform through internet which is not trustable. It is required to interchange secret session keys securely through insecure network for establishing communication securely. In two-party network, two communication clients share a low entropy password secretly to communicate in later sessions securely. But this paradigm requires high maintenance of passwords due to each new communicating pair requires separate passwords to establish communication securely. In three-party network, each communicating party shares a password with the third-party (server) to interchange a secret session key securely. The beauty of this setting is even server is not knowing the session key. Many authors have proposed various two-party and three-party protocols which are having their won pros and cons. In this paper we have cryptanalyzed verifier-based password authenticated key agreement protocol for three-party setting proposed by shaban et al.

Keywords: Cryptanalysis, authentication, key agreement protocol, Three-Party.

Introduction

Secret session keys has to be exchanged securely through insecure public channel to establish a secure communication. Principles of cryptography are achieved when shared session keys are used. In an asymmetric and symmetric key cryptography's longer keys will be used to achieve goal of cryptography. Memorizing the longer keys require additional challenging task. Consecutively, a password-authenticated key agreement protocols are used to allow the 2 or more communicating parties by memorizing low entropy passwords to share a secret session key for further communication securely. In a two-party setting, an authenticated secret session key will be established by using the pre-shared password. The major difficulty in two-party setting is maintenance low entropy passwords is exponential, as each communicating pair requires individual password for establishing a secret session key. This flaw enthuses the researcher to come with the invention of three-party setting. In three-party setting, a secret session key can be established between two communicating parties by using a authenticated low entropy passwords individually shared with a trusted party know as server. Consequently, each communicating party need to memorize only low-entropy password, by giving the burden to the server, where the server participates only in establishing a session key.

Based on the knowledge of shared passwords between users and server the password-authenticated key agreement protocols are classified into two categories: They are i. Symmetric model, ii. Asymmetric model (verifier-based)

In a symmetric model, the same knowledge related with passwords will be used by user and server to authenticate users. That is, the low entropy passwords (initially shared between users and server) used as it is in establishing a session. In this model, if server is compromised, the intruder can use the password from the password file to impersonate the legitimate user.

In an Asymmetric model, the different knowledge related with passwords will be used by user and server to authenticate users. That is, each user holds the low entropy password as it is and server holds the verifier of the password which is computed by performing one-way hash function on password. Here, server and users hold different images of a password by giving a justification to asymmetric model. In this model, if server is compromised, the password file cannot reveal any useful information to the adversary[1]; as the password file contains only the verifiers of the passwords not directly the passwords.

Password-based authenticated key exchange protocols, however are vulnerable to password guessing attacks because of low entropy.

In general, the password guessing attacks can be divided into three classes[2]: i. Detectable on-line password guessing attacks. ii. Undetectable on-line password guessing attacks. iii. Off-line password guessing attacks

Background

In 1976, Diffie-Hellman proposed a new directions in cryptography, which allows two users to agree on a single shared secret key to exchange the messages over an untrusted channel without having a prior communication between users[3]. However, unfortunately this protocol suffers from man-in-the middle attack. Later many password authenticated key agreement protocols have been probe up over the past years to overcome this problem. But due to the high maintenance of the low entropy passwords for each pair of users in two-party system[4-7]. Steiner et al. proposed the first 3PEKE protocol in 1995[8]. Lin et al. proposed a LSSH-3PEKE protocol without servers public key in 2001, and claimed that it is secure against password guessing attacks[9].

Recently, Sun et al. proposed two improved 3PEKE protocols, called SCH-3PEKE, respectively based on the password and the verifier[10]. A security weakness of the SCH-3PEKE scheme was recently revealed by Nam et al.[11]. Also Kulkarni et al. has proposed a three-party key agreement protocol in which each user stores one way hash function of the password at the server rather than storing the password itself[12]. Kulkarni et al.'s protocol is secure against online and dictionary attacks.

The rest of the paper is organized as follows. In section 2, we describe the review of Shaban et al. protocol. In section 3, we explain the cryptanalysis of Shaban et al. protocol. Finally, Section 4 concludes the paper.

Review of Shaban et al. Protocol

Notations: Notations used in this protocol are as listed below:

Table-1
Notations

A,B and TS	The identifiers of Alice, Bob and Trusted Server
C	Catherine(An Intruder)
V	A verifier that is computed from a password PW
n	Large Prime Number
g	Generator in the cyclic group Z^*_n
h( )	One-way hash function

Review: This protocol has been divided into three phasesl. They are: i. Initial Phase, ii. Session Key Agreement Phase, iii. Key Computation Phase

Initial Phase: Registering with Trusted Server, Alice and Bob computes H_A= h(A, TS, PW_A) and H_B= h(B, TS, PW_B) by choosing passwords PW_A and PW_B respectively. Now Alice and Bob calculate the verifiers $V_A=g^{h(A, TS, PWA)}$and $V_B= g^{h(B, TS, PWB)}$ sends to Trusted Server over a protected channel. Then Trusted Server stores the verifiers V_A and V_B in its password's table. Figure-1 Illustrates the Initial Phase of the protocol i.e., Alice→Trusted Server: $V_A=g^{h(A, TS, PWA)}$ and Alice→Bob: $V_B= g^{h(B, TS, PWB)}$

Alice	Trusted Server	Bob
H_A= h(A, TS, PW_A)	$V_A=g^{h(A, TP, PW_A)}$ $V_B=g^{h(B, TP, PW_B)}$	H_B= h(A, TS, PW_B)

Figure-1
Initial Phase

Session Key Agreement Phase: Assume Alice wants to establish a session key with Bob.

Initially, Alice by choosing a $\in_R Z^*_n$ computes $X_A = g^a \bmod n$ and sends {A, B and X_A} to Trusted Server.
i.e., Alice→Trusted Server: {A, B, X_A}

Similarly Bob by choosing b $\in_R Z^*_n$ computes $X_B = g^b \bmod n$ and sends {B, A and X_B} to Trusted Server.
i.e., Bob→Trusted Server: {B, A, X_A}

Upon receiving the messages from Alice and Bob, Trusted Server checks whether X_A or X_B equal to V_A or V_B by retrieving verifiers V_A and V_B from a password table. If they are equal, Trusted Server ends the protocol. Otherwise moves to the next step by choosing c, d $\in Z^*_n$ to compute $X_{TSA}= (V_A)^c \bmod n$ and $X_{TSB}= (V_B)^d \bmod n$ and sends {X_{TSA}, X_B} to Alice and {X_{TSB}, X_A} to Bob, respectively.
i.e., Trusted Server→Alice:{X_{TSA}, X_B} and Trusted Server→Bob:{X_{TSB}, X_A}

Trusted Server also computes $K_{TSA}= (X_A)^c = g^{ac} \bmod n$ and $K_{TSB}= (X_B)^d = g^{bd} \bmod n$ for further continuity of the protocol.

Next, Alice calculates $K_{ATS} = ((X_{TSA})^{H_A^{-1}})^a = (g^c)^a = g^{ca} \bmod n$ and V_{ATS} = h(A, B, TS, X_A, X_B, K_{ATS}, 0) and sends {V_{ATS}} to Trusted Server .
Alice→Trusted Server:{V_{ATS}}

Similarly, after reception of the message from Trusted Server , Bob calculates $K_{BTS} = ((X_{TSB})^{H_B^{-1}})^b = (g^d)^b = g^{db} \bmod n$ and V_{BTS} = h(A, B, TS, X_A , X_B , K_{BTS}, 0) and sends {V_{BTS}} to Trusted Server .
Bob→Trusted Server:{V_{BTS}}

In this step, Trusted Server computes V_{TSA} = h(A, B, TS, X_A , X_B , K_{TSA}, 0) and V_{TSB} = h(A, B, TS, X_A , X_B , K_{TSB}, 0) and checks whether V_{ATS} or V_{BTS} equal to V_{TSA} or V_{TSB}. If they are equal, Trusted Server calculates V_{ATS} = h(A, B, TS, X_A , X_B , K_{TSA}, 1) and V_{BTS} = h(A, B, TS, X_A , X_B , K_{TSB}, 1) and sends V_{TSA} and V_{TSB} to Alice and Bob, respectively.
i.e., Trusted Server→Alice: {V_{TSA}} and Trusted Server→Bob :{V_{TSB}}

This complete process of session key agreement phase is depicted in figure-2.

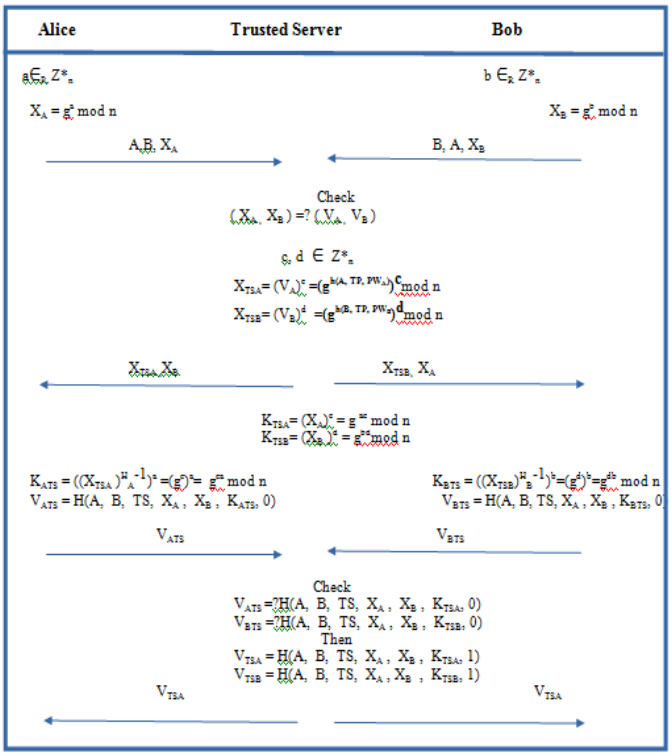

Figure-2
Session Key Agreement Phase

Key Computation Phase: Upon accepting the message V_{TSA} from TS, Alice computes V_{ATS} = h(A, B, TS, X_A , X_B , K_{ATS}, 1) and checks whether it is equal to V_{TSA} or not. If they are equal then TS and Bob are validated.

Similarly after accepting the message V_{TSB} from TS, Bob computes V_{BTS} = h(A, B, TS, X_A , X_B , K_{BTS}, 1) and checks whether it is equal to V_{TSB} or not. If they are equal then TS and Bob are validated.

Now, Alice computes a common session Key SK=h(A,B,K_{AB}) where K_{AB}= $(X_B)^a$=g^{ab} mod n. Similarly Bob computes a common Session Key SK=h(A,B,K_{BA}) where K_{BA} =$(X_A)^b$=g^{ab} mod n.

H_A, $H_A^{-1}$and V_A can be pre-computed by each client A before the protocol starts executing to enhance the efficiency of the etiquette. Figure-3 Illustrates key computation phase.

Alice	Trusted Party	Bob
Check V_{TSA}=? H(A, B, TS, X_A , X_B , K_{ATS}, 1)		**Check** V_{TSB} =?H(A, B, TS, X_A , X_B , K_{BTS}, 1)
Then K_{AB}= $(X_B)^a$=g^{ab} mod n K=H(A,B,K_{AB})		**Then** K_{BA} =$(X_A)^b$=g^{ab} mod n K=H(A,B,K_{BA})

Figure-3
Key Computation Phase

Cryptanalysis of Shaban et al. Protocol

This section demonstrates the Detectable online password guessing attack on Shaban et al. protocol. An intruder Catherine can mimic Alice and communicate with Bob. But Bob is thinking that it is interacting with Alice but actually Bob is communicating an intruder Catherine. Detectable on-line password guessing attacks on Shaban et al. protocol is explained in detail below. An intruder Catherine mimic Alice to guess Alice's password as the password is low entropy.

Initial Phase: Registering with Trusted Server, Alice and Bob computes H_A= h(A, TS, PW_A) and H_B= h(B, TS, PW_B) by choosing passwords PW_A and PW_B and respectively. Now Alice and Bob calculate the verifiers V_A=$g^{h(A, TS, PWA)}$ and V_B= $g^{h(B, TS, PWB)}$ respectively and sends to Trusted Server over a safe channel. This is shown in figure-4. Then Trusted Server stores the verifiers V_A and V_B in its password's table.

Alice	Trusted Server	Bob
H_A= h(A, TS, PW_A)	V_A=$g^{h(A, TP, PW_A)}$ V_B=$g^{h(B, TP, PW_B)}$	H_B= h(A, TS, PW_B)

Figure-4
Initial Phase

Session Key Agreement Phase: Initially, Alice by choosing a $\in_R$ Z^*_n computes X_A = g^a mod n and sends {A, B and X_A}to Trusted Server.
i.e., Alice→Trusted Server: {A, B, X_A}

Similarly Bob by choosing b $\in_R$ Z^*_n computes X_B = g^b mod n and sends {B, A and X_B} to Trusted Server.
i.e., Bob→Trusted Server: {B, A, X_A}

A user Catherine an intruder intercepts this message i.e., {A, B, X_A}. Now Catherine guess Alice's password PW_A' and finds $H_{A'}$ =h(A, TS, PWA'). Now Catherine computes $X_{A'}$ = $g^{a'}$ mod n by choosing a' $\in_R$ Z^*_n and then sends {A, B, $X_{A'}$}to Trusted Server .

After receiving the messages from Catherine and Bob, Trusted Server retrieves V_A and V_B from a password table, checks whether $X_{A'}$ or X_B equal to V_A or V_B, if they hold, Trusted Server terminates otherwise moves to the next step where it computes X_{TSA}= $(V_A)^c$ mod n and X_{TSB}= $(V_B)^d$ mod n by choosing c, d ∈ Z^*_n and sends {X_{TSA}, X_B} to Alice (Catherine intercepts) and {X_{TSB}, $X_{A'}$} to Bob, respectively. While waiting for messages from Alice and Bob, Trusted Server computes K_{TSA}= $(X_{A'})^c$ = $g^{a'c}$ mod n and K_{TSB}= $(X_B)^d$ = g^{bd}mod n.

After receiving the messages from Trusted Server , Catherine computes $K_{A'TS}$ = $((X_{TSA})^{H_{A'}^{-1}})^{a'}$=$(g^{c'})^{a'}$= $g^{c'a'}$ mod n and $V_{A'TS}$ = h(A, B, TS, $X_{A'}$, X_B , $K_{A'TS}$, 0) and sends $V_{A'TS}$ to

Trusted Server . Similarly, after receiving the message from Trusted Server, Bob computes $K_{BTS} = ((X_{TSB})^{H_B^{-1}})^b = (g^d)^b = g^{db}$ mod n and V_{BTS} = h(A, B, TS, $X_{A'}$, X_B , K_{BTS}, 0) and sends V_{BTS} to Trusted Server .

Trusted Server checks whether received $V_{A'TS}$from Alice is equal to computed V_{TSA}= H (A, B, TS, X_A, X_B, K_{TSA}, 0) or not. If they are equal, then the guessed password by an intruder Catherine is correct and trusted server will continue the remaining procedure of the protocol by computing the key going thru the key computation phase.

If it is not equal, then the attack can be detectable by the trusted server. Hence trusted server terminates the protocol at 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. The online detectable attack has been shown in figure-5.

In this way a malicious client can impersonate the actual client by successfully getting the secrete session key.

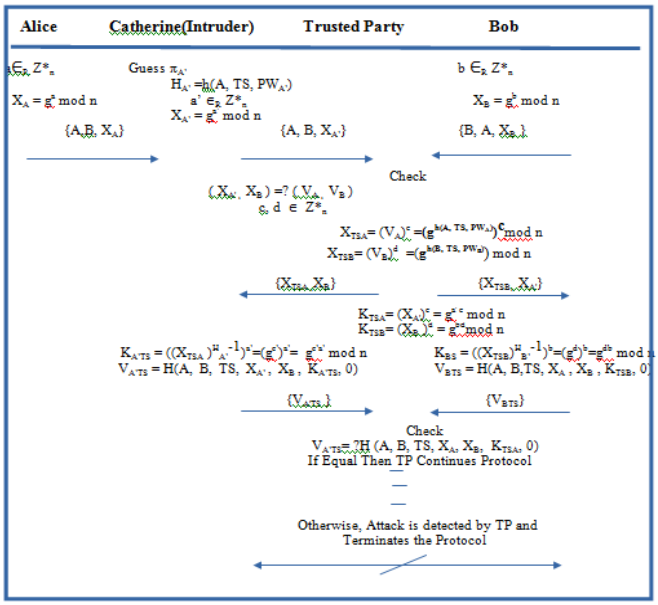

Figure-5
Detectable Online Password Guessing Attack on Shaban et al. Protocol

Conclusion

In this paper, initially we review the Shaban et al. verifier-based password authenticated key agreement protocol for three parties. Next, we have demonstrated that a verifier-based password authenticated key agreement protocol for three parties proposed by Shaban et al. is insecure against the detectable on-line password guessing attack.

References

1. S. Bellovin and M. Merritt, Augmented encrypted key exchange: a password based protocol secure against dictionary attacks and password-file compromise, ACM Conference on Computer and Communications Security, 244-250, **(1993)**

2. Y. Ding and P. Horster, Undetectable on-line password guessing attacks, ACM Operat-ing Systems Review, 29(4), 77-86, **(1995)**

3. W. Diffie and M. Hellman, New directions in cryptography, IEEE Trans, On Information Theory, 22(6), 644-654, **(1976)**

4. W. Diffie, P.C. Van Oorschot and M.J. Wiener, Authentication and authenticated key exchanges, *Design, Codes and Cryptography,* **2,** 107–125, **(1992)**

5. V. Boyko, P.D. MacKenzie and S. Patel. Provably secure password-authenticated key exchange using Diffe-Hellman// Proceedings of the 2000 Advances in Cryptology (EUROCRYPT'2000), 156-171, *Springer-Verlag,* **(2000)**

6. E. Bresson, O. Chevassut and D. Pointcheval, New security results on encrypted key exchange// Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography (PKC'2004), 145-158, *Springer-Verlag,* **(2004)**

7. R. Gennaro and Y. Lindell, A framework for password-based authenticated key exchange// Proceedings of the 2003 Advances in Cryptology (EUROCRYPT'2003), 524-543, *Springer-Verlag,* **(2003)**

8. M. Steiner, G. Tsudik and M. Waidner, Refinement and extension of encrypted key exchange, *ACM Operating Systems Review,* **29(3),** 22-30, **(1995)**

9. C.L. Lin, M. Steiner and T. Hwang, Three-party Encrypted Key Exchange without Server Public-keys, *IEEE Communications Letters,* **5(12),** 497-499 **(2001)**

10. H.M. Sun, B.C. Chen and T. Hwang, Secure key agreement protocols for three-party against guessing attacks, *The Journal of Systems and Software,* **75,** 63–68, **(2005)**

11. J. Nam, S. Kim, and D. Won, A weakness in Sun-Chen-Hwang's three-party key agreement protocols using passwords, Cryptology e Print Archive, Report 2004/348, **(2004)** http://eprint.iacr.org/2004/348.pdf **(2014)**

12. S. Kulkarni, D. Jena and S.K. Jena, A Novel Secure Key Agreement Protocol using Trusted Third Party, *Computer Science and Security Journals, IJCSS,* **1(1),** 11–18, **(2007)**