Cryptanalysis of Design and Analysis of a Provably Secure Multi-server Authentication Scheme

Naresh Babu Muthu Mohan[1], Ardhani Sathya Narayana Chakravarthy[1], Cherukuri Ravindranath[2]
(Corresponding author: Naresh Babu Muthu Mohan)

Department of Computer Science and Engineers, Jawaharlal Nehru Technological University[1]
Jawaharlal Nehru Technological University Road, Kakinada, Andhra Pradesh 533003, India
Trinity Institute Of Technology & Research[2]
Kokta By-Pass Road, Near Hindustan Petrol Pump,, Raisen Rd, Bhopal, Madhya Pradesh 462021, India
(Email:itsnaresh4u@gmail.com[1], asnchakravarthy@yahoo.com[2], ravindranathc@gmail.com[3])
(Received Mar. 29, 2016; Revised and Accepted June 10, 2016)

Abstract

The rapid growth of inter-networking and communication technologies resulted in an exponential hit rate on commercial service providing websites (servers) like Google, Amazon, Flipkart etc. from remote users connected via Internet. To handle the networking load, the organizations are moving from the traditional two tier client server architecture to multi-server architecture for efficient load balancing. The traditional two-party authentication protocol for remote user authentication are not sufficient to break the ever increasing attacks on open network i.e. Internet. Also, the existing two-party authentication protocols are meant for single server, adopting these protocols for multi-server environment results in the requirement of huge computation cost for separate registration of user at each server. So, researchers started proposing authentication schemes specific to multi-server environment. In 2014, Yeh et al. proposed an improved version over Pippal et al.'s scheme which eliminates all identified weaknesses like susceptible to user impersonation attack, server counterfeit attack, and the man-in-the-middle attack. In 2015, Mishra et al. demonstrated that Yeh et al. scheme is susceptible to off-line password guessing attack, insider attack and user impersonation attack and proposed an improved version. In this manuscript we do a thorough analysis on Mishra et al. scheme and determine that Mishra et al. scheme is liable to 'known session specific temporary information' attack and based on that, the attacker can realize all key attacks. We also demonstrate that Mishra et al. scheme consists of major inconsistencies like 'inefficient login phase' which restrict the protocol to adopt to real time implementation.

Keywords: Authentication; Elliptic Curve Cryptography; Multi-server Authentication; Smart Cards

1 Introduction

The advances in internet, mobile and networking technologies resulted in an exponential access to remote servers using high end mobile devices on the go via Internet (as shown in Figure 1). The traditional authentication schemes are primarily proposed keeping in mind the traditional two-tier client-server architecture and traditional communicating devices like desktop etc. [2, 4, 8, 15, 16]. Due to advances in mobile and communication technologies, users are able to connect to remote servers through mobile devices on the go, which results in an increased hit rate on e-commerce servers. Hence, all the small and medium enterprises are moving to a multi-server environment [3, 7, 14]. Due to this, there is a critical need for robust, efficient and lightweight remote user authentication algorithms. On the one hand, adopting these protocols for multi-server environment results in the users need to register in each server and to store large sets of data, including identities and passwords [1, 5, 17, 26].

Various researchers had proposed authentication protocols for secure authentication of users connecting to remote servers based on various techniques like usage of verification table [13], symmetric key cryptosystem [10], dynamic Identity based [6, 9, 19, 22, 23, 24, 25], modified password based [23, 24], involvement of the registration center in the authentication process [9].etc. Unfortunately, most of the protocols are analyzed insecure shortly, after they were put forward [6, 9, 10, 12, 13, 21, 22, 24]. Meanwhile, identity protection is considered to be important for authentication and key agreement protocol design in single-server and multi-server architectures.

In 2013, Pippal et al. [21] proposed a robust multi-server authentication scheme based on smart cards, with added advantages like elimination of verifier table, registered remote users are allowed to access multiple servers

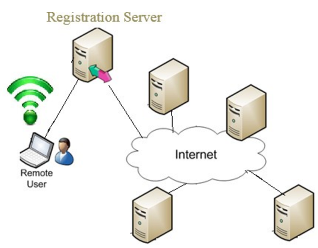

Figure 1: Typical multi server environment

without multiple registration. Also the registered users can alter the password securely without any assistance from the registration center or remote server. In 2014, Yeh et al. [25] demonstrated that the remote user multi server authentication scheme proposed by Pippal et al. [21] is vulnerable to user impersonation attack, server counterfeit attack, and the man-in-the-middle attack and having proved the inconsistencies in Pippal et al. [21] scheme, Yeh et al. proposed an improved version, which eliminates all identified weaknesses with the same order of computation complexity.

In 2015, Mishra et al. [19] did a thorough literature analysis of multi-server authentication schemes and summarized that most of the existing multi-server authentication schemes require all the involved servers to be trusted, involvement of registration center or central authority in mutual authentication [20] or multiple secret keys. In practical scenarios, the servers may be semi-trusted, thus considering all servers to be trusted does not seem to be realistic scenario. Involvement of registration center/central authority in the computation process like mutual authentication may create a bottleneck scenario for a large network, which is a draw back in multi-server authentication scheme proposed by odelu et al. [20]. Also, computation of multiple secret keys may not be suitable for smart card based environment as smart card keeps limited storage space. In sound literature analysis, Mishra et al. [19] demonstrated that recently proposed Yeh et al. [25] multi-server authentication scheme is susceptible to off-line password guessing attack, insider attack and user impersonation attack. Having found the security pitfalls, Mishra et al. [19] proposed an improved multi-server authentication scheme which does not require all servers to be trusted, central authority no longer needed in authentication and smart card need not to store multiple secret keys.

On thorough analysis of Mishra et al. [19] multi-server authentication scheme, we demonstrate that their scheme is susceptible to session specific temporary information attack, on the success of it, Mishra et al. [19] scheme is susceptible to leakage of user identity, password and computation of session key by the attacker. We also established that Mishra et al. [19] scheme includes major inconsistencies in which lack of early detection of wrong credentials by the smart card, which results in excessive computation on the server side, which ultimately results in a Denial of Service attack. In future work, we aim to propose a secure and light weight multi server authentication scheme by eliminating the security pitfalls and inconsistencies found in Mishra and other related schemes.

The rest of the paper is organized as follows. In Section 2, a brief review of Mishra et al. scheme is given. Section 3, describes the security weakness of Mishra et al. scheme. Section 4 provides the conclusion of the paper.

2 Review of Mishra et al. Scheme

In this section, we examine Design and Analysis of a Provably Secure Multi-server Authentication Scheme by Mishra et al. [19] in 2015 and then demonstrate its security pitfalls. The notations used in Mishra et al. [19] are listed in Table 1.

Table 1: The notations used in Mishra et al. [19]

Parameter	***Description***
U_i	$User_i$
R.S	A trustworthy Registration center / Registration server
S_j	j^{th} server in the system
UID_i	Unique identity of $User_i$
UPW_i	Unique password of $User_i$
T_i	Timestamp generated by entity i
SK_{ij}	Session key between $User_i$ and $server_j$
MK	Master key of RS
USK_i, UPK_i	i^{th} user secret/public key
h(.), h1(.), h2(.)	One-way hash functions
p	A large prime number
$E_p(a,b)$	An elliptic curve $y^2 = x^3 + ax + b(mod)p$ over a finite Z_p with $4a^3 + 27b^2 \neq 0((modp))$ based on group G
$\oplus$	Bitwise XOR operation
$\|\|$	Bitwise string concatenation

2.1 Registration Server (R.S)

An additive group G, whose generator is P. G is a set of points over an elliptic curve EP(a,b) of order n.

Select:

$$
\begin{aligned}
h: &\quad \{0,1\}^* \to \{0,1\}^k,\\
h1: &\quad \{0,1\}^* * G \to \{0,1\}^*,\\
h2: &\quad \{0,1\}^* * \{0,1\}^* * \{0,1\}^* \ * G * G * G \to \{0,1\}^k
\end{aligned}
$$

Chooses: A master key MK of 1024 bits.

Registration server makes as public $\{EP(a,b), P, h(.), h1(.), h2(.)\}$ and keeps its master key MK as private.

Figure 2: Server registration phase of Mishra scheme

Server (S_j)		***Registration Server (R.S)***
Selects: Identity SID_j		
	$\{SID_j\}$ $------\rightarrow$	
		Computes: $SSK_j = h(SID_j \| \| MK)$ $SPK_j = SSK_j.P$
	$\{SSK_j, SPK_j\}$ $\leftarrow------$	

Figure 3: User registration phase of Mishra scheme

User (U_i)		***Registration Server*** $(R.S)$
Selects: UID_i, PW_i, N $RPW_i = h(PW_i \| \| UID_i)$.		
	$\{RPW_i \oplus N, UID_i\}$ $----------\rightarrow$	
		$USK_i = h(UID_j \| \| MK)$ $X_i = USK_i \oplus RPW_i \oplus N$ $UPK_i = USK_i.P$
	R.S $\rightarrow (UID_i, UPK_i)$ to all servers	
	S.C $= \{X_i, P, h(.), h1(.), h2(.)\}$ $\leftarrow-------------$	
$Y_i = X_i \oplus N = USK_i \oplus RPW_i$ Replaces $X_i with Y_i$ in S.C. i.e. S.C $= \{Y_i, P, h(.), h1(.), h2(.)\}$		

The registration server(RS) performs the following steps in offline mode before the actual deployment of the servers in deployment field.

Step 1. R.S selects a large odd prime number 'p' of minimum 160 bits, generates a Galois Field G.F (p) and elliptic curve Ep(a,b), which is a set of all points on the curve $y^2 = x^3 + ax + b(modp)$, such that a,b $\epsilon Z_p = \{0, 1, 2, 3....p-1\}$, satisfying the condition $4a^3 + 27b^2 \neq 0$. 'G' represents the base point of elliptic curve 'E' of order 'n', which is of 160 bits such that $n > \sqrt{p}$. R.S chooses three hash functions h, $h1$, $h2$ and opts MK as its master key.

Step 2. Registration server makes as public $\{EP(a,b)$, P, $h(.)$, $h1(.)$, $h2(.)\}$ and keeps its master key MK as private.

2.2 Server Registration Phase

This phase is invoked whenever a server S_j registers with the registration server for the first time. The registration server assigns secret and public keys to the server.

This phase is invoked whenever a server S_j registers with the registration server for the first time (Figure 2).

Step 1. The server S_j selects the identifier SID_j and provides its identity $\{SID_j\}$ to the registration server via a secure channel for registration.

Step 2. On receiving the registration request $\{SID_j\}$, RS computes the secret and public keys for S_j as follows:

$$SSK_j = h(SID_j||MK), SPK_j = SSK_j.P$$

where MK is its secret master key. R.S submits $\{SSK_j, SPK_j\}$ to S_j, through a secure communication channel.

2.3 User Registration Phase

This phase is invoked whenever a user U_i registers with the registration server for the first time (Figure 3).

Step 1. The user U_i selects the identifier UID_i, a random number N, and the password PW_i. U_i then computes $RPW_i = h(PW_i||UID_i)$. U_i submits the registration request $\{RPW_i \oplus N, UID_i\}$ to the registration server via a secure channel for registration.

Step 2. On receiving the login request $\{RPW_i \oplus N, UID_i\}$, the RS performs the following computations to compute the secret and public key for U_i. $USK_i = h(UID_j||MK)$, $UPK_i = USK_i.P, X_i = USK_i \oplus RPW_i \oplus N$. R.S forward the secret and public key pair (USK_i, UPK_i) of U_i to all registered servers. Finally, the RS issues a tamper-proof smart card with the following parameters stored in it $S.C = X_i, P, h(.), h1(.), h2(.)$ to U_i through a secure communication channel.

Figure 4: Login phase of Mishra scheme

$\mathbf{U_i/S.C(UPK_i, USK_i, SPK_j)}$	$\mathbf{Server(S_j)(UPK_i, SSK_j, SPK_j)}$
Submits UID_i, PW_i.	
Computes: $RPW_i = h(PW_i\|\|UID_i)$	
Retrieves $USK_i = Y_i \oplus RPW_i$.	
Selects the targeted server S_j to access the resources.	
Achieves the S_j public key from R.S public directory i.e. (SID_j, SPS_j)	
Generates a session specific arbitrary number $'r_u'$	
Computes: $A_i = r_u.P$	
$B_{ij} = USK_i.SPK_j(= USK_i.SSK_j.P = UPK_i.SSK_j = USK_i.SSK_j.P)$.	
$C_{ij} = r_u.SPK_j = (r_u.SSK_j.P = r_u.P.SSK_j = A_i.SSK_j)$	
$V_i = h(UID_i\|\|SID_j\|\|T1\|\|B_{ij}\|\|C_{ij}\|\|A_i)$	
$DID_i = UID_i \oplus h1(SID_j\|\|C_{ij})$	
$\{DID_i, A_i, V_i, T1\}$ $\dashrightarrow$	

Step 3. On receiving S.C from R.S, U_i computes $Y_i = X_i \oplus N = USK_i \oplus RPW_i$ and replaces X_i with Y_i in its S.C.

Finally the U_i S.C contains the parameters: $\{Y_i, P, h(.), h1(.), h2(.)\}$.

2.4 Login Phase

Whenever the user U_i wants to access data from a server S_j deployed in a multi-server environment, the user U_i needs to perform the following steps (Figure 4).

Step 1. U_i inserts his/her smart card into the card reader of a specific terminal and provides his/her Identity UID_i, password PW_i.

Step 2. The S.C computes $RPW_i = h(PW_i||UID_i)$ and retrieves U_i secret key $USK_i = Y_i \oplus RPW_i$.

Step 3. The S.C achieves the server S_j public key from the R.S public directory, i.e (SID_j, SPS_j). S.C generates a session specific arbitrary number $'r_u'$.

Step 4. The smart card then computes the variables $A_i = r_u.P$, $B_{ij} = USK_i.SPK_j(= USK_i.SSK_j.P = UPK_i.SSK_j = USK_i.SSK_j.P)$. $C_{ij} = r_u.SPK_j = (r_u.SSK_j.P = r_u.P.SSK_j = A_i.SSK_j)$,
$V_i = h(UID_i||SID_j||T1||B_{ij}||C_{ij}||A_i)$, Masked identity $DID_i = UID_i \oplus h1(SID_j||C_{ij})$ where T1 is the current time stamp.

Step 5. The S.C finally forwards the login message $\{DID_i, A_i, V_i, T1\}$ to RS, via a public channel.

2.5 Authentication Phase

On receiving the login request $\{DID_i, A_i, V_i, T1\}$ at time, from S.C, at time $T1^*$, the server S_j validates the login request by checking whether $(T1^* - T1) \leq \Delta t$, then S_j proceeds as follows (Figure 5).

Step 1. Compute $A_i.SSK_j = r_u.P.SSK_j = r_u.SPK_j = {C_{ij}}^*$. A_i is received through login request by U_i. Retrieve ${UID_i}^* = DID_i \oplus h1(SID_j||{C_{ij}}^*)$.

Step 2. Compute: ${B_{ij}}^* = UPK_i.SSK_j$, $V_i^* = h(UID_i||SID_j||T1||{B_{ij}}^*||{C_{ij}}^*||A_i)$. Validate whether $V_i = V_i^*$ if yes, U_i is authenticated.

Step 3. Generates a session specific arbitrary number $'r_s'$, Compute: $D_j = r_s.P$, $E_j = r_s.A_i = r_s.r_u.P.$. $SK_{ij} = h(UID_i||SID_j||T1||{B_{ij}}^*||{C_{ij}}^*||E_j)$, $V_j = h(UID_i||SK_{ij}||T2||{B_{ij}}^*||{C_{ij}}^*||D_j)$ And forwards the login reply message $\{D_j, V_j, T2\}$ to S.C via a public channel.

Step 4. On receiving the login reply message, S.C computes:

$$\begin{aligned} {E_j}^* &= r_u.D_j = r_u.r_s.P, \\ {SK_{ij}}^* &= h(UID_i||SID_j||T1||B_{ij}||C_{ij}||{E_j}^*), \\ {V_j}^* &= h(UID_i||{SK_{ij}}^*||T2||B_{ij}||C_{ij}||D_j). \end{aligned}$$

Validate whether $V_j = {V_j}^*$, if yes, S_j is authenticated. If yes, S.C authenticates the server S_j.

3 Cryptanalysis of Mishra et al. Scheme

In this segment, we will cryptanalyze the Mishra et al. [19] scheme and illustrate that Mishra et al. scheme is vulnerable to Known Session Specific Temporary Information Attack, i.e. if session specific arbitrary numbers, i.e. r_u and r_s are leaked out, then an attacker can achieve the secret key of U_i, password PW_i of U_i, session key SK_{ij}. We describe the detailed steps in attack as follows:

Figure 5: Authentication phase of Mishra scheme

$\mathbf{U_i/S.C\ (UPK_i, USK_i, SPK_j)}$		$\mathbf{Server(S_j)(UPK_i, SSK_j, SPK_j)}$
	$\{DID_i, A_i, V_i, T1\}$ $- - - - - - - \rightarrow$	
		Receive at $T1^*$. Check $(T1^* - T1) \leq \Delta t$ Compute: $A_i.SSK_j = r_u.P.SSK_j = r_u.SPK_j = C_{ij}{}^*$ // A_i is received through login request by U_i. Retrieve: $UID_i{}^* = DID_i \oplus h1(SID_j \vert\vert C_{ij}{}^*)$ $B_{ij}{}^* = UPK_i.SSK_j$ $V_i{}^* = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij}{}^* \vert\vert C_{ij}{}^* \vert\vert A_i)$ Validate whether $V_i = V_i{}^*$ if yes, U_i is authenticated. Genrates a session specific arbitrary number $'r_s'$ Compute: $D_j = r_s.P$, $E_j = r_s.A_i = r_s.r_u.P$. $SK_{ij} = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij}{}^* \vert\vert C_{ij}{}^* \vert\vert E_j)$ $V_j = h(UID_i \vert\vert SK_{ij} \vert\vert T2 \vert\vert B_{ij}{}^* \vert\vert C_{ij}{}^* \vert\vert D_j)$
	$\{D_j, V_j, T2\}$ $\leftarrow - - - - - - -$	
Receive at $T2^*$ Check: $(T2^* - T2) \leq \Delta t$ $E_j{}^* = r_u.D_j = r_u.r_s.P$ $SK_{ij}{}^* = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij} \vert\vert C_{ij} \vert\vert E_j{}^*)$ $V_j{}^* = h(UID_i \vert\vert SK_{ij}{}^* \vert\vert T2 \vert\vert B_{ij} \vert\vert C_{ij} \vert\vert D_j)$ Validate whether $V_j = V_j{}^*$ if yes, S_j is authenticated.		

1) An opponent or an attacker or legal user can extract the information cached in the smart card by several techniques such as power consumption or leaked information [11, 18], etc. i.e. $S.C = \{Y_i, P, h(.), h1(.), h2(.)\}$.

2) An opponent can passive monitor or eavesdrop or alter or replay the login request, login reply messages communicated among U_i and R.S over a public channel which is Internet, i.e. $\{\{DID_i, A_i, V_i, T1\}, \{D_j, V_j, T2\}\}$.

An attacker is supposed to have access to all the values discussed in Table 2, based on these the attacker can accomplish various attacks as discussed below.

3.1 Fails to Resist Known Session Specific Temporary Information Attack

The compromise of session specific arbitrary numbers should not allow the attacker to compute any unknown value of the communication participants and should not compromise the computed the session key.

Case 1: Offline Identity computation by an attacker. In Mishra et al. scheme assume that the session specific arbitrary number r_u, r_s are compromised and an attacker got hold of it. As discussed above, the attacker is having access to the values as discussed in Table 2, can perform following steps:

Step 1: Compute $C_{ij}{}^* = r_u.SPK_j$.SPK_j is a server public key which is known to all the participants.

Step 2: From the intercepted login message $\{DID_i, A_i, V_i, T1\}$, retrieve UID_i from DID_i using $C_{ij}{}^*$ computed in Step 1, i.e. $UID_i = DID_i \oplus h1(SID_j \mid\mid C_{ij}{}^*)$. Hence Mishra et al. scheme failed to preserve user anonymity.

Case 2: Offline Password guessing by an attacker.

Step 1: 'E' can retrieve Y_i from the U_i S.C and can frame:

$$\begin{aligned} Y_i &= X_i \oplus N = USK_i \oplus RPW_i \\ &= USK_i \oplus h(PW_i \vert\vert UID_i). \\ USK_i &= Y_i \oplus h(PW_i \vert\vert UID_i). \qquad (1) \end{aligned}$$

Step 2: Compute

$$C_{ij} = r_u.SPK_j. \qquad (2)$$

We are assuming r_u is compromised and SPK_j is the server public key.

Step 3: Replace Equation (1) in place of USK_i.

$$\begin{aligned} B_{ij} &= USK_i.SPK_j \\ &= (Y_i \oplus h(PW_i \vert\vert UID_i)).SPK_j. \qquad (3) \end{aligned}$$

On intercepting $V_i, A_i, T1$ from the login request message sent by U_i to S_j, the attacker 'E' can

Table 2: Values known and unknown to an attacker

Values Known to the Attacker	Values Known to the Attacker	Values doesn't known to the Attacker
A legal adversary 'E' is assumed to know: 1. The smart card values of legal user U_i. 2. The intermediate communication messages exchanged between U_i and S. 3. All public values of U_i and S_j	1. $\{Y_i, P, h(.), h1(.), h2(.)\}$ $Y_i = USK_i \oplus RPW_i$ 2. $\{\{DID_i, A_i, V_i, T1\}, \{D_j, V_j, T2\}\}$ 3. SPK_j, SID_j, UPK_j	1. SSK_j, UID_i, USK_i, MK

proceed as follows to compute the U_i password from Equation (3).

Step 3.1: $V_i = h(UID_i \mid\mid SID_j \mid\mid T1 \mid\mid B_{ij} \mid\mid C_{ij} \mid\mid A_i)$. In V_i, 'E' knows UID_i, SID_j, $T1$, A_i, C_{ij} and B_{ij} are computed in step 2 and Step 3 above.

Step 3.2: Substitute B_{ij} in V_i. i.e $V_i = h(UID_i \mid\mid SID_j \mid\mid T1 \mid\mid B_{ij} \mid\mid C_{ij} \mid\mid A_i) = h(UID_i \mid\mid SID_j \mid\mid T1 \mid\mid (Y_i \oplus h(PW_i \mid\mid UID_i)).\ SPK_j \mid\mid C_{ij} \mid\mid A_i)$ using Equation (3).

Step 3.3: In V_i of Step 3.2, the only unknown parameter to an attacker is PW_i. As discussed in [6, 12, 23], if in Mishra et al. [19] scheme, if the user U_i opts for a password, which is a weak (low entropy), the attacker can perform password guessing attack as follows similar to [6, 24]:

Step 3.3.1: Guesses the value of PW_i to be $PW_i{}^*$ from a dictionary space ∂.

Step 3.3.2: Compute: $V_i{}^* = h(UID_i||SID_j||T1||(Y_i \oplus h(PW_i{}^*||UID_i)))$. Check computed V_i* equal to V_i in the intercepted login request. If yes, the U_i original password is $PW_i{}^*$ else 'E' proceeds to Step 3.3.1.

Hence, as discussed above in Mishra et al. scheme, the attacker succeeds to guess the low-entropy password PW_i.

Step 4: On getting the password PW_i of U_i, the attacker 'E' can compute the U_i secret key as follows:

$$\begin{aligned} Y_i &= X_i \oplus N = USK_i \oplus RPW_i \\ &= USK_i \oplus h(PW_i||UID_i). \\ USK_i &= Y_i \oplus h(PW_i||UID_i). \end{aligned}$$

As 'E' knows Y_i from U_i smart card and UID_i as discussed in Case 1.

Hence, we can confirm that Mishra et al. suffers from the biggest drawback that, on compromise f session specific arbitrary numbers, all the secret parameters of protocol participants can be find out.

Case 3: Computing the session key by an attacker. In Mishra et al. [19] scheme, the session key $SK_{ij} = h(UID_i||SID_j||T1||B_{ij}{}^*||C_{ij}{}^*||E_j)$. As discussed above, in SK_{ij}, 'E' knows all the values except E_j. As discussed above, if r_u, r_s are compromised, 'E' can compute $E_j = r_u.r_s.P$. Hence, based on above discussion, we can confirm that in Mishra et al. scheme, if r_u, r_s are compromised, the attacker 'E' can compute U_i identity i.e. UID_i, password PW_i and session key SK_{ij}.

3.2 Fails to Resists Denial of Service Attack (Inefficient Login Phase)

Assume that the legal user provides a wrong password PW_R. Instead of PW_i during login stage.

Mishra et al. [19] scheme is not secured against computation exhaustive attacks like denial of service attack as there is no verification of user data by S.C during the login phase. Thus if a legal user U_i submits a wrong password PW_r instead of PW_i, as discussed in [6, 23], SC performs all calculations to compute the login request without verifying the correctness of inserted identity ID and password PW. This loophole endangers the security of the scheme in following ways (Figure 6).

Offline and online password guessing attack, user impersonation attack, and denial of service attack. Network Flooding with wrong login request above, the smart card still proceeds further to compute the login message which is a fake login request messages to the server which leads to the excessive computation on the server side. Similarly to guess the password correctly, an adversary sends the guessed password online a number of times till she will not succeed which leads to excessive computation on server as smart card lacks any verification mechanism. Thus proto-

Figure 6: Denial of service attack

$\mathbf{U_i/S.C\ (UPK_i, USK_i, SPK_j)}$ $\mathbf{Server(S_j)(UPK_i, SSK_j, SPK_j)}$

U_i provides UID_i, PW_R.
PW_R is a wrong password instead of his correct password PW_i
Computes: $RPW_R = h(PW_R||UID_i)$
Retrieves $USK_R = Y_i \oplus RPW_R$.
Selects the targeted server S_j to access the resources.
Achieves the S_j public key from R.S public directory i.e. (SID_j, SPS_j)
Generates a session specific arbitrary number $'r_u'$
Computes: $A_i = r_u.P$
$B_{R_{ij}} = USK_R.SPK_j$
$C_{ij} = r_u.SPK_j$
$V_{R_i} = h(UID_i||SID_j||T1||BR_{ij}||C_{ij}||A_i)$
$DID_i = UID_i \oplus h1(SID_j||C_{ij})$

$\{DID_i, A_i, V_i, T1\}$
$------\rightarrow$

Compute: $A_i.SSK_j = r_u.P.SSK_j = r_u.SPK_j = C_{ij}*$
// A_i is received through login request by U_i.
Retrieve: $UID_i* = DID_i \oplus h1(SID_j||C_{ij}^*)$
$B_{ij}^* = UPK_i.SSK_j$
$V_i^* = h(UID_i||SID_j||T1||B_{ij}^*||C_{ij}^*||A_i)$
Verifies V_i^* equal V_i
As the verification fails, R.C rejects the message.

col is not secure against denial of service attack. Due to inefficient login phase, it costs, 3Hash operations +3Elliptic Point Multiplication operations.

4 Conclusion

Recently Mishra et al. proposed an ECC-based multi-server authentication scheme. Even though it is a novel attempt, after thorough analysis of Mishra et al. paper, we demonstrated that their scheme is vulnerable to known session specific temporary information attack which results in leakage of user identity, password and computation of session key by the attacker. We also established that Mishra et al. scheme include major inconsistencies in which lack of early detection of wrong credentials by S.C, which results in excessive computation on the server side, which ultimately results in Denial of Service attack. In future work, we aim to propose a secure and light weight multi server authentication scheme by eliminating the security pitfalls and inconsistencies found in Mishra and other related schemes.

References

[1] R. Amin, "Cryptanalysis and efficient dynamic id based remote user authentication scheme in multi-server environment using smart card," *International Journal of Network Security*, vol. 18, no. 1, pp. 172–181, 2016.

[2] N. Anwar, I. Riadi, A. Luthfi, "Forensic SIM card cloning using authentication algorithm," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 71–81, 2016.

[3] C. C. Chang, W. Y. Hsueh, T. F. Cheng, "An advanced anonymous and biometrics-based multi-server authentication scheme using smart cards," *International Journal of Network Security*, vol. 18, no. 6, pp. 1010–1021, 2016.

[4] T. Y. Chang, W. P. Yang, M. S. Hwang, "Simple authenticated key agreement and protected password change protocol", *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

[5] T. Y. Chen, C. C. Lee, M. S. Hwang, J. K. Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, Nov. 2013.

[6] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks", *Peer-to Peer Networking and Applications*, 2014.

[7] T. H. Feng, C. H. Ling, and M. S. Hwang, "Cryptanalysis of Tan's Improvement on a Password Authentication Scheme for Multi-server Environments,"

International Journal of Network Security, vol. 16, no. 4, pp. 318–321, 2014.

[8] M. S. Hwang, C. C. Lee, Y. L. Tang, "An improvement of SPLICE/AS in WIDE against guessing attack", *International Journal of Informatica*, vol. 12, no. 2, pp.297-302, Apr. 2001.

[9] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment", *IEEE System Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[10] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards", *IEEE Transactions on Consumer Electronics*, vol. 12, no. 6, pp. 251–255, 2004.

[11] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis", in *Advances in Cryptology*, pp. 388–397, Springer, 1999.

[12] S. Kumari, M. K. Khan and R. Kumar, "Cryptanalysis and improvement of a privacy enhancedscheme for telecare medical information systems", *Journal of Medical Systems*, vol. 37, pp. 37, 2013.

[13] L. H. Li, L. C. Lin and M. S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks", *IEEE Transactions on Neural Networks*, vol. 50, no. 1, pp. 1498–1504, 2001.

[14] I. C. Lin, M. S. Hwang, L. H. Li, "A new remote user authentication scheme for multi-server architecture", *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[15] C. W. Lin, C. S. Tsai, M. S. Hwang, "A new strong-password authentication scheme using one-way hash functions", *International Journal of Computer and Systems Sciences*, vol. 45, no. 4, pp. 623–626, July 2006.

[16] Y. Liu, C. C. Chang and S. C. Chang, "An efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 1, pp. 1–10, 2017.

[17] Y. Liu, C. C. Chang, C. Y. Sun, "Notes on 'An anonymous multi-server authenticated key agreement scheme based on trust computing using smart card and biometrics'," *International Journal of Network Security*, vol. 18, no. 5, pp. 997–1000, 2016.

[18] T. S. Messerges, E. A. Dabbish, R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks", *IEEE Transactions on Computers*, vol. 51, pp. 541–552, 2002.

[19] D. Mishra, "Design and analysis of a provably secure multi-server authentication scheme", *Wireless Personal Communications*, vol. 86, pp. 1095–1119, 2016.

[20] V. Odelu, A. K. Das and A. Goswam, "A secure biometrics-based multi-server authentication protocol using smart cards", *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[21] R. S. Pippal, C. D. Jaidhar, and S. Tapasw, "Robust smart card authentication scheme for multi-server architecture", *Wireless Personal Communications*, vol. 72, pp. 72, 2013.

[22] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function withoutverification table", *Computers and Security*, vol. 27, no. 3, pp. 115–121, 2008.

[23] J. L. Tsai and N. W. Lo, "A new password-based multi-server authentication scheme robust to password guessing attacks", *Wireless Personal Communications*, vol. 71, pp. 1977–1988, 2013.

[24] B. Wang and M. Ma, "A smart card based efficient and secured multi-server authentication scheme", *Journal of Wireless Personal Communication*, vol. 68, no. 2, pp. 361–378, 2013.

[25] K. H. Yeh, "A provably secure multi-server based authentication scheme", *Wireless Personal Communications*, vol. 79, pp. 1621–1634, 2014.

[26] H. Zhu, Y. Zhang, and Y. Sun, "Provably secure multi-server privacy-protection system based on chebyshev chaotic maps without using symmetric cryptography," *International Journal of Network Security*, vol. 18, no. 5, pp. 803–815, 2016.

Biography

Naresh Babu Muthu Mohan received his M.Tech degree from VIT University, Vellore, India. Currently he is pursuing Ph. D. in the Department of C.S.E., Jawaharlal Nehru Technological University Kakinada, Kakinada, A.P., India. He has published papers in various International journals and conferences. His areas of current research include Networks, Mobile Security & Cryptography.

Ardhani Sathya Narayana Chakravarthy is currently working as Professor, Dept. of Computer Science & Engineering, Coordinator MOOCs & Skill Development Centre, Jawaharlal Nehru Technological University Kakinada,Kakinada, A.P., India. He has 62 papers published in various International journals and conferences. His research areas include Networks, Security & Cryptography, Biometrics, and Digital Forensics. He is Editorial board member for various International Journals.

Cherukuri Ravindranath received the Ph.D degree in Electrical and Computer Engineering from University of Texas at San Antonio(USA). He is currently the principal of Trinity Institute of technology & Research, Bhopal, India. He is reviewer of IEEE, SPIE, Elsevier. He was post research fellow at University of Texas at San Antonio(USA). He has published large number of research papers in the reputed International/National journals and conference proceedings. He has two patents in Digital Image Security. His areas of current research interest include System Security, Multimedia Processing, Information Assurance and Applied Statics.