International Journal of Advances in
Soft Computing and Intelligent Systems (IJASCIS)
2023, Vol 02, Issue 01, 14-26

Original Paper

CRYPTANALYSIS ON "PRACTICAL AND PROVABLY SECURE THREE-FACTOR AUTHENTICATION PROTOCOL BASED ON EXTENDED CHAOTIC-MAPS FOR MOBILE LIGHTWEIGHT DEVICES"

Suresh Devanapalli[1] and Kolloju Phaneendra[2]
[1]Department of Mathematics, Rajiv Gandhi University of Knowledge Technologies, Basar 504107, Telangana, India.
dsuresh7799@gmail.com
[2]Department of Mathematics, University College of Science, Osmania University, Hyderabad 500004, India.
kollojuphaneendra@yahoo.co.in

ABSTRACT

Authentication and key agreement (AKA) plays an important role in an open network environment in order to secure communication between two or more participants. Authentication and key agreement (AKA) protocols should protect the sensitive information against a malicious adversary by providing a variety of services, such as authentication, user credentials' privacy, when the smart card is lost/stolen or the private key of a user or a server is revealed. Unfortunately, most of the existing an authentication and key agreement (AKA) protocols proposed in the literature do not safe against smart card loss attacks. Recently, in 2020, Shuming et al. proposed a secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices. In this paper, we analyze the Shuming et al's protocol and show that Shuming et al's scheme is vulnerable to privileged-insider attacks with the help of both offline password guessing attacks, user impersonation, Parallel session attacks and thus, their scheme fails to prevent known session specific temporary information attack. In addition, we show that their scheme does not provide strong user's anonymity. Furthermore, Shuming et al's scheme cannot safe against smart card loss attacks. Apart from these, Shuming et al's scheme has launch DoS attack.

KEYWORDS

Security, Three-factor, Authentication, Chaotic-maps.

1. Introduction

With the booming of various sensitive applications over the Internet, user privacy is becoming more and more important, and has attracted widespread concern. With the rapid development of mobile application technologies, affordable and portable mobile lightweight devices are becoming very popular. Mobile lightweight devices (e.g: laptops, smartphone, smartwatch, personal digital assistants, and wearable devices) are able to access cloud servers for online payment, online voice and video chatting, mobile banking interaction, e-commerce, and so on anytime and anywhere. Key agreement protocols are used to establish common keys between two or more entities. The established key can then be used to assure confidentiality of exchanged messages through encryption. Additionally, authenticated key agreement protocols offer implicit authentication.

A key agreement protocol should possess the following set of desirable properties [1].

– A key agreement protocol is successful if each of the parties accepts the identity of the other party as well as the computed key.

– The key confidentiality property means that unintended parties cannot compute the key.

– A key agreement protocol provides key authentication if only those parties specified to be engaging in the protocol are able to compute the session key. It is clear that key authentication implies key confidentiality. For if only intended parties can compute the key, then unintended parties cannot compute the key.
– The key control property refers to the inability of any of the parties to force the shared key to some value of its own choice.
– A key agreement protocol provides key confirmation if parties provide proof of possession of the session key. Key confirmation is usually achieved via encrypting or hashing a known quantity.
– A key agreement protocol provides forward secrecy if the loss of any long-term secret keying material does not allow the compromise of keys from previous sessions.

In 1976, Diffie and Hellman introduced the first key agreement protocol not requiring a private channel to be established between the two negotiating parties [2]. However, the basic Diffie-Hellman (DH) protocol provided no means for authenticating the identities of the two communicating parties and thus is susceptible to man-in-the-middle attacks. So secure key agreement has to be based on mutual authentication between each two participants in a group. The network communication between mobile users and cloud servers may suffer from various attacks, such as impersonation attack and password guessing attack. Moreover, mobile devices are usually resource constrained and vulnerable to special network attacks. Therefore, it is indispensable to establish an authentication and key agreement (AKA) protocol to protect the conversations between the users with lightweight mobile devices and remote servers in various application environments. Due to the limitations of symmetric-key techniques, authentication and key agreement (AKA) protocols based on public-key techniques have attracted much attention, providing secure access and communication mechanism for various application environments. Among these public-key techniques used for AKA protocols, chaotic-map is more effective than scalar multiplication and modular exponentiation, and it offers a list of desirable cryptographic properties such as un-predictability, un-repeatability, un-certainty and higher efficiency than scalar multiplication and modular exponentiation. Furthermore, it is usually believed that three-factor AKA protocols can achieve higher security level than single- and two-factor protocols.

1.1. Related Work

In 2007, Xiao et al. [3] proposed a chaos-based key agreement protocol based on utilizing chaotic public key cryptosystem. Comparing to the traditional protocols in the area of key agreement, it could reduce computation complexity. However, Guo and Zhang [4] pointed out that Xiao et al.'s [3] scheme could not resist server spoofing attacks and denial-of-service (DoS) attacks. Furthermore, in Guo and Zhang [4] proposed an improved scheme, which claimed that their protocol could resist the security flaws of Xiao et al.'s protocol. Juang et al. [5] proposed a password-authenticated key agreement scheme using smart cards. Later, Sun et al. [6] pointed out the three weaknesses: 1) inability of the password-changing operation; 2) the session-key problem; and 3) inefficiency of the double secret keys, in Juang et al.[5] proposed an enhanced scheme to eliminate these aforementioned weaknesses. However, later Li et al. [7] showed that the scheme of Juang et al. only achieves initiator anonymity rather than initiator untraceability. Next, they proposed a remedy to strengthen the scheme of Juang et al. They claimed that their scheme is efficient and effective, and achieves general security features such as mutual authentication, key agreement, and initiator untraceability. Later, Tsai et al. [8] showed that Li et al. [7] proposed protocol is vulnerable to de-synchronization attack and inefficiency of registration table.

In 2012, Lee et al. [9] proposed chaotic maps-based three-factor key agreement with user anonymity, where hash functions are directly applied to biometrics. He et al. [10] showed that

Lee et al.'s scheme suffers from privileged insider attack and DoS attack, and fails to provide anonymity. Then, they proposed an enhanced key agreement protocol. Later, Lee and Hsu [11] proposed a secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps, which also provides user anonymity. However, this scheme has the same weakness as Lee et al.'s scheme [9]. Guo et al. [12] proposed a chaotic maps-based key agreement protocol which avoided modular exponential computing and scalar multiplication on elliptic curve. Nowadays, with the fast development of Internet, privacy protection of users is a hot issue. In 2014, Liu et al. [13] proposed a multi-function password mutual authentication key agreement scheme with privacy preserving. However, this scheme was based on an elliptic curve. Its efficiency was lower than related scheme [9] based on chaotic maps because of modular exponential computing and scalar multiplication on elliptic curve. Islam [14] proposed a dynamic identity-based three-factor authentication scheme using extended chaotic map. However, Jiang et al. [15] pointed out the major security drawbacks in Islam [14] proposed a robust three-factor authentication scheme. Moreover, in [16], the author has proved that Guo et al.'s [4] scheme cannot resist off-line password guess attack. However, the improved scheme in [16] introduces a traditional asymmetric encryption algorithm to address the issue. Liu et al. [17] showed that Guo et al.'s [4] scheme suffers from replay attack and DoS attack, and it has unnecessary redundancy in protocol design. Then, they proposed an improved secure password and chaos-based two-party key agreement protocol. Tsai and Lo [18] applied an identity-based signature and identity-based encryption to propose an anonymous key distribution scheme for smart grid in which smart meter and service provider mutually authenticate with each other, and then establish a session key between them for secure communication. However, Odelu et al. [19] pointed out that Tsai-Lo's [18] scheme is insecure against the ephemeral secret leakage attack, and it fails to provide the strong credentials' privacy of the smart meter. Furthermore, in Odelu et al. [19] proposed a secure authenticated key agreement scheme for smart grid, which overcomes the security weaknesses of Tsai-Lo's scheme. In 2018, Roy et al. [20] designed a chaotic map-based anonymous authentication protocol with the fuzzy extractor for crowd sourcing Internet of Things. Islam et al. [21] also proposed a provably secure three-factor protocol for multimedia big data communications. Wazid et al. [22] proposed a three-factor user authentication protocol for renewable-energy-based smart grid environment. However, Shuming et al. [23] pointed out that Wazid et al.'s [22] scheme could not resist off-line password guessing attacks and was unable to provide perfect forward secrecy and three-factor security. But, in 2020, Shuming et al. [23] analyzed the security flaws of [20], [21] and showed that they are vulnerable to perform off-line password guessing attacks and unable to provide functionality and security features such as three-factor security. Shuming et al. [23] proposed a secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices in order to withstand these security issues, and claimed that their scheme is secure against all possible known attacks.

1.2. Contributions of the Paper

The contribution of this paper is manyfold:

– We analyze the security limitations of the recently proposed Shuming et al.'s authentication and key agreement (AKA) scheme, and this scheme is, unfortunately, cannot safe against smart card loss attack, and as a result, their scheme cannot prevent the privileged-insider attack with the help of both offline password guessing attack, DoS attack, user impersonation attack, and Parallel session attack.

– In addition, we show that their scheme cannot provide strong user's anonymity.

1.3. Organization of the Paper

The rest of the paper is organized as follows: In Section 2, we briefly discuss the required mathematical background. In Section 3, we review the recently proposed Shuming Qiu's scheme [23]. In Section 4, we present that Shuming Qiu's scheme is vulnerable to various attacks. We also point out some design flaws of Shuming Qiu's scheme in this section. Finally, we wind up the paper in Section 5.

Table 1. The symbols and descriptions.

Symbols	Description
p	a prime number
$T_n(x)$	a Chebyshev polynomial of degree n
Uj	a User
S	Server
A	a malevolent Adversary
$Idj\ , pwj$	the identity, password of User
$T_k(y)$	the Server′s private key, public key
r_s, r_u	a secrete value generated by Server, User
SKi	session key/shared key
$h(.), h_0(.)$	a secure one-way collision avoiding hash function
$Gen(.), Rep(.)$	Bio-metric key extraction, reproduction algorithms
$\oplus, \|$	the bitwise XOR, concatenation operations

2. Mathematical background

2.1. Chebyshev chaotic maps

The polynomial T_n : U → U is said to be Chebyshev polynomial of degree n, if

$$T_n(x) = \begin{cases} \cos(n.\cos^{-1}(x)) & if \quad x \in [-1,1] \\ \cos(n\boldsymbol{\theta}) & if \quad x = cos\boldsymbol{\theta}, \boldsymbol{\theta} \in [-1,1] \end{cases}$$

Where n is a non-negative integer, U = [−1, 1], cos : R → [−1, 1] and cos−1 :[−1, 1] → [0, π].

The recurrence relation of Chebyshev polynomial is defined as [24]

$$T_n(x) = \left(2xT_{n-1}(x) - T_{n-2}(x)\right), \text{ if n > 1 where}$$

$$T_n(x) = \begin{cases} 1 & if \quad n = 0 \\ x & if \quad n = 1 \end{cases}$$

The semi-group properties on improved Chebyshev polynomials are given below [25]:

– $T_n(x) = \left(2xT_{n-1}(x) - T_{n-2}(x)\right) mod\ p$, where x ∈ R, p is large prime number.

– $T_n\left(T_m(x)\right) = T_{nm}(x)$

i.e, $T_n\left(T_m(x)\right) = T_n(\cos(n.\cos^{-1}(x)))$

$= (\cos(n.\cos^{-1}(\cos(m.\cos^{-1}(x)))))$

$= cos(n.m.cos^{-1}(x))$

$= T_{nm}(x)$

It is well known that the following problems are computationally very hard.

Definition 1: CMDLP (Chaotic Map-based Discrete Logarithm Problem). For given (x, y), it is computationally very hard to calculate the discrete logarithm η such that $y = T_\eta(x)\ mod\ p$

Definition 2: CMCDHP (Chaotic Map-based Computational Diffie-Hellman Problem). For given , $T_a(x)$ $and\ T_b(x)$, it is computationally very hard to calculate $T_{ab}(x) mod\ p$ where two scalars a, b $\in$ Z_q^*

Definition 3: (Collision-Resistant Secure One-Way Hash Function)
Let $U = \{0,1\}^*$ and $V = \{0,1\}^l$. A collision-resistant secure one-way hash function H : U → V is considered as a deterministic algorithm. It takes an input $u \in \{0,1\}^*$ and $v \in \{0,1\}^l$, where v and u are binary string of fixed length l and an arbitrary length binary string. If $Adv_A^{HASH}(T)$ is a malevolent adversary A's benefit to detecting conflict, then we have $Adv_A^{HASH}(T)$ = Prb[(u, u′) ⇐= A : u≠ u′, H(u) = H(u′)] where (u, u′) ⇐= A symbolizes the pair (u, u′) is chosen randomly by A , and Prb[F] symbolizes the probability of a random event F. In such circumstances, the malevolent attacker A is permitted to be probabilistic and the probability in the benefit is calculated over the random selections made by the malevolent attacker A with the time period T. A hash function H(˙) is said to be a secure one-way collision-avoiding hash function, if $Adv_A^{HASH}(T)$= ε, for adequate small ε > 0.

2.2. Biometrics-based Fuzzy Extractor

Fuzzy extractor method [26] is useful even when the biometric is noisy and slight variations exist. The fuzzy extractor converts the biometric information into two values, which consists of two procedures, namely, Gen and Rep. More details illustrated as following:
– (.) : The probabilistic algorithm takes a biometric sample FI_j^* and returns a pair of reproduction parameter βj and a secret key αj of a fixed m bits i.e. $Gen(FIj) = (\alpha j, \beta j)$
– (.) : This is a deterministic algorithm which reproduces the secret key αj with the help of βj when an input of noisy biometric sample FIj^* is provided. The hamming distance between FIj^* and FIj , which are the noisy and original biometric sample respectively, should not exceed a specific threshold value. So, $Rep(FIj^*, \beta j) = \alpha j$

The uniqueness property of a biometric allows its applications in authentication protocols. As compared to the low-entropy password, the biometric keys has more advantages [27], [28], [29] such as biometric keys cannot be forgotten or lost, difficult to share or copy, hard to distribute or forge, and as a result, guessing the biometric keys is a hard problem.

3. Review of Shuming et al's. scheme

In this section, we represent the overview of Shuming et al's. scheme. The symbols used in Shuming et al's. scheme are listed in Table 1.

3.1. Initialization phase

The server S randomly selects a number $k \in Z_q^*$ as well as two one-way hash functions h(.) (SHA-160) and $h_0(.)$ (SHA-320). Then, S calculates the public key $T_k(y)$ publicizes these parameters $\{T_k(y)$, y, h(.), $h_0(.)\}$, and keeps a long private key k as a secret.

3.2. User Registration Phase

In this phase, we discuss the detailed steps to register user with server in Shuming et al's. scheme. The summary of the registration phase is shown in Figure 1.
–**Step1**: The user Uj selects an Idj and sends it to the server S.
– **Step2**: Upon getting $\{Idj\}$, S randomly picks $nj, mj \in Z_q^*$ and computes $EIDj = h(Idj \| nj), N_0 = h(Idj \| mj \| k \| EIDj)$. S stores $\{Idj, mj, Honey\ List = Null\}$ in its database, inputs $\{T_k(y),\ y, EIDj\}$ to a new smart card Cj and finally sends $\{Cj, N_0\}$ $to\ Uj$.

– **Step3**: Upon receiving the smart card Cj from the server S, the user Uj inputs his new password pwj and fingerprints FIj into Cj. Then, smart card Cj randomly generates a number $2^4 < z_0 < 2^8$ and calculates some important parameters as follows: $Gen(FIj) = (\alpha j, \beta j)$, $hpwj = h(pwj \parallel Idj \parallel EIDj \parallel \alpha j)$, $Vj = h((h(Idj) \oplus hpwj) mod\ z_0)$, $N_1 = N_0 \oplus hpwj \oplus \alpha j$. Finally, the smart card Cj contains the following parameters: $\{EIDj,\ N_1,\ Vj,\ \beta j,\ T_k(y),\ y,\ z_0\}$ and $\{h(.),\ h_0(.),\ Gen(.), Rep(.)\}$.

User(Uj)	Server(S)
Inputs Idj	
$\{Idj\}$ ⟶ (via secure channel)	Chooses picks $nj, mj \in Z_q^*$ Computes $EIDj = h(Idj \parallel nj)$ $N_0 = h(Idj \| \| mj \| \| k \| \| EIDj)$ Stores: $\{Idj, mj, Honey\ List = Null\}$ In its database New smart card: Cj=$\{T_k(y), y, EIDj, h(.)\}$
Chooses pwj, Inputs FIj Computes $(\alpha j, \beta j) = Gen(()$ $hpwj = h(pwj \parallel Idj \parallel EIDj \parallel \alpha j)$, $Vj = h((h(Idj) \oplus hpwj) mod\ z_0)$, $N_1 = N_0 \oplus hpwj \oplus \alpha j$	$\{Cj,\ N_0\}$ ⟵ (via secure channel)
Update smart card : $\{EIDj,\ N_1,\ Vj,\ \beta j,\ T_k(y),\ y,\ z_0, h(.),\ h_0(.),\ Gen(.), Rep(.)\}$.	

Figure 1. User Registration of Shuming et al. Scheme

3.3. User Login and Mutual Authentication Phase

After the user Uj is registered with the server S successfully, he transmits the login request to when he wishes to obtain some service – see below:. The summary of this is presented in Figure 2.

– **Step1**: Uj inputs the smart card Cj into a card reader, and provides Idj, pwj, and $FIj\ to\ Cj$. Then, Cj computes $\alpha j = Rep(FIj^*, \beta j)$, $hpwj = h(pwj \parallel Idj \parallel EIDj \parallel \alpha j)$ and checks whether $Vj =^? h((h(Idj) \oplus hpwj)\ mod\ z_0)$. If not, Cj rejects the login request. Otherwise, Cj computes $N_0 = N_1 \oplus hpwj \oplus \alpha j$. Subsequently, Cj picks $r_u \in^R Z_q^*$ and computes $M_1 = T_{r_u}(y)$, $M_2 = T_{r_u}(T_k(y))$, $M_3 = (Idj \parallel N_0) \oplus h_0(M_2)$, $M_4 = h(Idj \parallel EIDj \parallel N_0 \parallel M_2 \parallel M_3)$. Finally, Cj sends $\{EIDj, M_1, M_3, M_4\}$ to S.

– **Step2**: After obtaining $\{EIDj,\ M_1, M_3, M_4\}$, S calculates $M_2 = T_k(M1)$, $(Id'j \parallel N_0') = M_3 \oplus h_0(M_2)$. Then, S searches $\{Idj, mj, Honey\ List\}$ in its database. If Id_j' cannot be searched, the session is terminated. Otherwise, S proceeds to the next step. S computes $N_0 = h(Idj \parallel mj \parallel k \parallel EIDj)$ and checks whether $M_4 =^? h(Idj \parallel EIDj \parallel N_0 \parallel M_2 \parallel M_3)$. If they are unequal, S terminates this session. Otherwise, S checks whether the derived N_0' equals the computed N_0. If they are equal, S proceeds to the next step. If they are unequal, S knows that's smart card has been corrupted and the adversary did not get the real password. Accordingly, S inserts the honey word N_0' into Honey List and wraps this login request. Moreover, if $|Honey\ List| >= n_0$ (Such as the threshold n_0 = 5), where n_0 is a threshold value, S suspends the use of Cj until Uj re-registers and requests to restore Cj. Otherwise, S picks $n_j^{new} \in^R Z_q^*$ and calculates $EID_j^{new} = h(Idj \parallel n_j^{new})$, $N_0^{new} = h(Idj \parallel mj \parallel k \parallel EID_j^{new})$.

Subsequently, S updates $\{Idj, Honey\ List = Null\}$ in its back-end database. Moreover, S picks $r_s \in^R Z_q^*$ and computes $M_5 = T_{r_s}(y)$, $M_6 = T_{r_s}(M_1)$, $M_7 = (EID_j^{new} \| N_0^{new}) \oplus h(M_6 \| N_0)$, $SKj = h(Idj \| EIDj \| N_0 \| N_0^{new} \| M_2 \| M_6)$ and $M_8 = h(Idj \| EIDj \| N_0 \| M_5 \| M_2 \| SKj)$. Lastly, S sends $\{M_5, M_7, M_8\}$ *to* Cj.

– **Step3**: On receiving the message $\{M_5, M_7, M_8\}$, Cj computes $M_6 = T_{r_u}(M_5)$, $(EID_j^{new} \| N_0^{new}) = M_7 \oplus h(M_6 \| N_0)$, $SKi = h(Idj \| EIDj \| N_0 \| N_0^{new} \| M_2 \| M_6)$, and verifies $M_8 = h(Idj \| EIDj \| N_0 \| M_5 \| M_2 \| SKi)$. If not, Cj aborts this session. Otherwise, Cj considers a shared key $SK = SKi = SKj$ is being shared with S. Subsequently, Cj randomly generates a number $2^4 < z_0^{new} < 2^8$ and calculates $hpw_j^{new} = h(pwj \| Idj \| EID_j^{new} \| \alpha j)$, $V_j^{new} = h((h(Idj) \oplus hpw_j^{new}) mod z_0^{new})$, $N_1^{new} = N_0^{new} \oplus hpw_j^{new} \oplus \alpha j$. Finally, the smart card Cj replaces $\{EIDj, N_1, V_j, z_0\}$ with $\{EID_j^{new}, N_1^{new}, V_j^{new}, z_0^{new}\}$.

User(Uj)	Server(S)
Insert Cj and inputs Idj, pwj, and FIj Computes $\alpha j = Rep(FIj^*, \beta j)$, $hpwj = h(pwj \| Idj \| EIDj \| \alpha j)$ Checks $Vj =^? h((h(Idj) \oplus hpwj) mod\ z_0)$. Computes $N_0 = N_1 \oplus hpwj \oplus \alpha j$ Chooses $r_u \in^R Z_q^*$ Computes $M_1 = T_{r_u}(y)$ $M_2 = T_{r_u}(T_k(y))$, $M_3 = (Idj \| N_0) \oplus h_0(M_2)$, $M_4 = h(Idj \| EIDj \| N_0 \| M_2 \| M_3)$ $\{EIDj, M_1, M_3, M_4\}$ (via open channel	Calculates $M_2 = T_k(M1)$ $(Id'j \| N_0') = M_3 \oplus h_0(M_2)$ Checks the validity of Idj Checks $N_0 = h(Idj \| mj \| k \| EIDj)$ $M_4 =^? h(Idj \| EIDj \| N_0 \| M_2 \| M_3)$. Picks $n_j^{new} \in^R Z_q^*$ Calculates $EID_j^{new} = h(Idj \| n_j^{new})$ $N_0^{new} = h(Idj \| mj \| k \| EID_j^{new})$ Updates $\{Idj, Honey\ List = Null\}$ Chooses $r_s \in^R Z_q^*$ Computes $M_5 = T_{r_s}(y)$ $M_6 = T_{r_s}(M_1)$ $M_7 = (EID_j^{new} \| N_0^{new}) \oplus h(M_6 \| N_0)$ $SKj = h(Idj \| EIDj \| N_0 \| N_0^{new} \| M_2 \| M_6)$ $M_8 = h(Idj \| EIDj \| N_0 \| M_5 \| M_2 \| SKj)$. $\{M_5, M_7, M_8\}$ (via open channel)
Computes $M_6 = T_{r_u}(M_5)$ $(EID_j^{new} \| N_0^{new}) = M_7 \oplus h(M_6 \| N_0)$ $SKi = h(Idj \| EIDj \| N_0 \| N_0^{new} \| M_2 \| M_6)$ Verifies $M_8 = h(Idj \| EIDj \| N_0 \| M_5 \| M_2 \| SKi)$ Chooses an integer $2^4 < z_0^{new} < 2^8$ $hpw_j^{new} = h(pwj \| Idj \| EID_j^{new} \| \alpha j)$ $V_j^{new} = h((h(Idj) \oplus hpw_j^{new}) mod z_0^{new})$ $N_1^{new} = N_0^{new} \oplus hpw_j^{new} \oplus \alpha j$ Replaces $\{EIDj, N_1, V_j, z_0\}$ with $\{EID_j^{new}, N_1^{new}, V_j^{new}, z_0^{new}\}$.	$\{M_5, M_7, M_8\}$

Figure 2. Login and Authentication of Shuming et al. Scheme

3.4. Updating Credentials-Biometric and Password

Uj Injects his smart-card into the card reader, and provides $Idj, pwj \; and \; FIj^*$. Then, Cj calculates $\alpha j = Rep(FIj^*, \beta j)$, $hpwj = h(pwj \parallel Idj \parallel EIDj \parallel \alpha j)$ and checks whether $Vj =^? h((h(Idj) \oplus hpwj) mod\, z_0)$. If not, Cj declines this update request. Otherwise, Cj acknowledges this update request.

– Case I: If Uj only wants to change the password, then he inputs a new password pw_j^{new}. Cj will then randomly generate $2^4 < z_0^{new} < 2^8$ and calculates $hpw_j^{new} = h(pwj \parallel Idj \parallel EID_j^{new} \parallel \alpha j)$, $V_j^{new} = h((h(Idj) \oplus hpw_j^{new}) mod z_0^{new})$, $N_1^{new} = N_1 \oplus hpw_j \oplus hpw_j^{new} = (N_0 \oplus hpw_j \oplus \alpha j) \oplus hpw_j \oplus hpw_j^{new}$. Finally, the smart card Cj replaces $\{N_1, Vj, z_0\}$ with $\{N_1^{new}, V_j^{new}, z_0^{new}\}$.

– Case II: If Uj only wants to change the biometrics, then he inputs a new biometrics FI_j^{new}. Then, Cj randomly generates a number $2^4 < z_0^{new} < 2^8$ and calculates $(\alpha_j^{new}, \beta_j^{new}) = Gen(FI_j^{new})$, $hpw_j^{new} = h(pwj \parallel Idj \parallel EIDj \parallel \alpha_j^{new})$, $V_j^{new} = h((h(Idj) \oplus hpw_j^{new}) mod z_0^{new})$, $N_1^{new} = N_1 \oplus hpwj \oplus \alpha j \oplus hpw_j^{new} \oplus \alpha_j^{new} = (N_0 \oplus hpwj \oplus \alpha j) \oplus hpwj \oplus \alpha j \oplus hpw_j^{new} \oplus \alpha_j^{new}$. Finally, the smart card Cj replaces $\{N_1, Vj, \beta j, z_0\}$ with $\{N_1^{new}, V_j^{new}, \beta_j^{new}, z_0^{new}\}$.

– Case III: If Uj wants to change both his biometrics and the password simultaneously, then he inputs a new password pw_j^{new} and a new biometrics FI_j^{new}. Subsequently, Cj randomly generates a number $2^4 < z_0^{new} < 2^8$ and calculates $(\alpha_j^{new}, \beta_j^{new}) = Gen(FI_j^{new})$, $hpw_j^{new} = h(pw_j^{new} \parallel Idj \parallel EIDj \parallel \alpha_j^{new})$, $V_j^{new} = h((h(Idj) \oplus hpw_j^{new}) mod z_0^{new})$, $N_1^{new} = N_1 \oplus hpwj \oplus \alpha j \oplus hpw_j^{new} \oplus \alpha_j^{new} = (N_0 \oplus hpwj \oplus \alpha j) \oplus hpwj \oplus \alpha j \oplus hpw_j^{new} \oplus \alpha_j^{new}$. Finally, the smart card Cj replaces $\{N_1, Vj, \beta j, z_0\}$ with $\{N_1^{new}, V_j^{new}, \beta_j^{new}, z_0^{new}\}$.

3.5. Revoking Smart Card

In this subsection, the user to block a stolen or misplaced smart card:

– **Step 1**: Uj verifies the authentication of smart card is similar to the login phase. If Cj authenticates Uj as a legitimate user, Cj sends the revocation request $\{EIDj, M_1, M_3, M_4,$ Revoke request$\}$ to the server.

– **Step 2**: Upon getting the revocation-request, S authenticates Cj by verifying M_4. If it is found to be invalid, S denies this revocation-request. Otherwise, S sets $|Honey\ List| >= n_0$ such that Cj is revoked. Finally, Cj is suspended until Uj re-registers.

3.6. User Re-registration

If a legitimate user's smart-card is revoked or the number of times that the user cannot be authenticated by the server S exceeds the maximum threshold value, then Uj is required to re-register. In this case, Uj will not be able to log into the system even though she inputs the correct values$\{Idj, pwj, FI_j^*\}$. However, Uj can re-register by performing the following steps:

– **Step 1**: Uj submits the re-registration request $\{Idj, \; Re-register\ request\}$ to S through a private channel.

– **Step 2:** Upon receiving the request message from, the server S uniquely identifies the user Uj by checking her identity information. Afterwards, if Idj is found in the database, S confirms whether $|Honey\ List| >= n_0$. And if Cj is found to be revoked, then S accepts this request

and performs Steps 2 and 3 of the Registration phase to complete re-registration. Otherwise, S rejects this request.

4. Cryptanalysis of Shuming et al. scheme

We now cryptanalyze the Shuming et al.′s scheme [23] and prove that it is not secure against the following attacks:

4.1. Known session-specific temporary information attack

According to [30], [31], [32], [33], [34] all the session keys must be secured even if the session random numbers of the user are compromised to an adversary $\boldsymbol{A}$. Assume that the session random number r_u chosen by Uj is unexpectedly revealed to an attacker $\boldsymbol{A}$. Then, Shuming et al. scheme has the following drawback:

– Since Uj and S computes a session key SKi as $SKi = h(Idj \parallel EIDj \parallel N_0 \parallel N_0^{new} \parallel M_2 \parallel M_6)$, an attacker $\boldsymbol{A}$ can compute the session key SKi using known session random number r_u.

Adversary $\boldsymbol{A}$ intercepts the message $\{EIDj, M_1, M_3, M_4\}$ sent to the server S (in Step 1 of User Login and Mutual Authentication Phase), and checks whether $T_{r_u}(y)$ matches with M_1. If it matches, A confirms that r_u corresponds to message $\{EIDj, M_1, M_3, M_4\}$ and computes M_2, Idj and N_0 as $M_2 = T_{r_u}(T_k(y))$ and $(Idj \parallel N_0) = M_3 \oplus h_0(M_2)$ (this may cause user anonymity violation). The adversary $\boldsymbol{A}$ sends reply message $\{EIDj, M_1, M_3, M_4\}$ to S without any modifications. In this case, S cannot identify the message $\{EIDj, M_1, M_3, M_4\}$ as a replied one. From the message $\{M_5, M_7, M_8\}$, the adversary $\boldsymbol{A}$ knows M_5, M_7, and he/she can compute $M_6 = T_{r_u}(M_5)$, $(EID_j^{new} \parallel N_0^{new}) = M_7 \oplus h(M_6 \parallel N_0)$, then compute SKi as $SKi = h(Idj \parallel EIDj \parallel N_0 \parallel N_0^{new} \parallel M_2 \parallel M_6)$, and valid $M_8 = h(Idj \parallel EIDj \parallel N_0 \parallel M_5 \parallel M_2 \parallel SKi)$ for S without knowledge of $Uj's$ authentication parameter k and mj. As a result, $\boldsymbol{A}$ can successfully impersonate the legal user.

4.2. Lost/Stolen smart card attack

As shown in Xie et al. scheme [35], even if one or two of the three factors in a three-factor authentication scheme can be obtained by an attacker, the system should still be secure. Hence in the Shuming et al.'s scheme [23], we assume that an adversary $\boldsymbol{A}$ can get FIj^* Using FIj^*, α_j^* is evaluated as $\alpha_j^* = Rep(FIj^*, \beta_j)$. Also, $\boldsymbol{A}$ can guess all the $Sid\ X\ Spwd$ combinations in polynomial time [36], where Sid and $Spwd$ are the sample space of identity and password respectively. So $\boldsymbol{A}$ guesses Id_j^* and pw_j^* and calculates $hpw_j^* = h(pw_j^* \parallel Id_j^* \parallel EIDj \parallel \alpha_j^*)$, $V_j^* = h((h(Id_j^*) \oplus hpw_j^*) mod z_0)$. If $(V_j^* = V_j)$ then user identity and password have been compromised. Otherwise, $\boldsymbol{A}$ continues guessing Id_j^* and pw_j^* until $(V_j^* = V_j)$. Thus this scheme is not safe against smart card loss attack.

4.3. Privileged-insider attack through offline password guessing attack

Suppose an adversary $\boldsymbol{A}$, who is also a privileged insider user, acts as an adversary, say $\boldsymbol{A}$. In this case, $\boldsymbol{A}$ knows the credentials Idj of a legitimate registered user Uj which are submitted to the Server during the user registration phase (see Section 3.2). Moreover, if $\boldsymbol{A}$ can acquire the lost/stolen smart card Cj of the user Uj, using the "power analysis attacks"[37], [38], the adversary $\boldsymbol{A}$ can extract all the credentials and $\{EIDj, N_1, Vj, \beta j, T_k(y), y, z_0\}$ and $\{h(.), h_0(.), Gen(.), Rep(.)\}$ stored in the memory of Cj, where $EIDj = h(Idj \parallel nj), N_0 = h(Idj \parallel mj \parallel k \parallel EIDj)$, $Gen(FIj) = (\alpha j, \beta j)$, and $N_1 = N_0 \oplus hpwj \oplus \alpha j$. Now, as

$Gen(FIj) = (\alpha j, \beta j)$ and $N_1 = N_0 \oplus hpwj \oplus \alpha j$, $\boldsymbol{A}$ can form the following relation: $hpwj = h(pwj \| Idj \| EIDj \| Rep(FIj, \beta j))$, or $hpwj = N_0 \oplus N_1 \oplus Rep(FIj, \beta j)$. $\boldsymbol{A}$ can then guess a password, say pw_j^*. Using the guessed password pw_j^*, and , $EIDj$ and FIj or $N_0, N_1, FIj\ and\ \beta j$, $\boldsymbol{A}$ further can calculate $hpw_j^* = h(pw_j^* \| Idj \| EIDj \| Rep(FIj, \beta j))$, and verify if the condition $hpw_j^* = hpwj$ is valid or not. If the condition holds, it means that $\boldsymbol{A}$ is successful in guessing the user $Uj's$ correct password. Hence, it is clear that the low-entropy guessed passwords are easily guessed and verified in Shuming et al.'s scheme. As a result, Shuming et al.'s scheme is vulnerable to privileged-insider attack with the help of both offline password guessing.

4.4. User impersonation and Parallel session attacks

A privileged insider adversary $\boldsymbol{A}$ with the knowledge of registration information dj, and $hpwj$, αj and extracted N_1 from the stolen smart card Cj of a valid registered user Uj (discussed in Section 4.2) can easily compute $N_0 = N_1 \oplus hpwj \oplus \alpha j$. Consequently, $\boldsymbol{A}$ can forge the login request message $\{EIDj, M_1, M_3, M_4\}$ to the Server in order to impersonate the user Uj due to the following reason. Since $EIDj$ get from the stolen smart card Cj, the privileged insider adversary $\boldsymbol{A}$ of the Server S also knows it. Now, $\boldsymbol{A}$ can generate random number $r_u^* \in^R Z_q^*$, and computes $M_1^* = T_{r_u}^*(y)$, $M_2^* = T_{r_u}^*(T_k(y))$, $M_3^* = (Idj \| N_0) \oplus h_0(M_2^*)$, $M_4^* = h(Idj \| EIDj \| N_0 \| M_2^* \| M_3^*)$. As a result, the adversary $\boldsymbol{A}$ is able to send a valid login request message $\{EIDj, M_1^*, M_3^*, M_4^*\}$ to the Server S. Thus; a privileged adversary can impersonate a legal registered user Uj in Shuai et al.'s scheme.

We consider another attack, where privileged insider adversary $\boldsymbol{A}$ of the Server S, who has calculated N_0 from Stolen smart card attack, can intercept the message $\{M_5, M_7, M_8\}$ that is sent from the Server to a user Uj. A, having the knowledge of N_0, can calculate $(EID_j^{new} \| N_0^{new}) = M_7 \oplus h(M_6 \| N_0)$ and the session key $SKi = h(Idj \| EIDj \| N_0 \| N_0^{new} \| M_2 \| M_6)$ Thus, A can independently calculate the session key SKi making the scheme of Shuming et al. vulnerable to the parallel session attack.

4.5. No provision of user anonymity

The user anonymity is a desirable property for remote user authentication. Generally, the scheme with user anonymity contains two aspects of content, one is the user's real identity cannot be revealed by the attacker; another is that the user cannot be traced by the attacker. In Shuming et al.'s scheme, server authenticated with the user can recover the identity of the user due to the following reason. Since $T_k(y)$ get from the stolen smart card , server S authenticated with the user can recover the identity of the user through computing $M_2 = T_k(M_1), (Idj \| N_0) = M_3 \oplus h_0(M_2)$ from the message $\{EIDj, M_1, M_3, M_4\}$. Thus, the identity of the user is leaked to the server. Moreover, in each login phase, the user Uj submits the login request message $\{EIDj, M_1, M_3, M_4\}$to the server S. On this message, $EIDj = h(Idj \| nj)$ and $N_0 = h(Idj \| mj \| k \| EIDj)$ are unique for each user. The attacker can distinguish whether two sessions are launched by the same user. Therefore, the attacker can trace the user by stolen smart card Cj. Accordingly, Shuming et al.'s scheme fails to preserve user anonymity.

4.6. Denial of service attack

From the login and authentication phase of Shuming et al.'s scheme, we find that any attacker $\boldsymbol{A}$ who colludes with the malicious server can easily forge a login request message and replay it to the server S. In Shuming et al.'s scheme, the attacker can launch DoS attack as follow: Upon intercepting the message $\{EIDj, M_1, M_3, M_4\}$, the attacker $\boldsymbol{A}$ generates random number $r_u^* \in^R Z_q^*$ and calculates $M_1^* = T_{r_u}^*(y)$, $M_2^* = T_{r_u}^*(T_k(y))$, $M_3^* = (Idj \| N_0) \oplus h_0(M_2^*)$, $M_4^* = h(Idj \| EIDj \| N_0 \| M_2^* \| M_3^*)$. $\boldsymbol{A}$ sends $\{EIDj, M_1^*, M_3^*, M_4^*\}$ to Server. Upon

receiving the message from $\boldsymbol{A}$, server S computes $M_2^* = T_k(M_1^*)$, $(Idj \parallel N_0) = M_3 \oplus h_0(M_2)$. Then, S searches $\{Idj, mj, Honey - List\}$ in its database. If Idj cannot be searched, the session is terminated. Otherwise, S proceeds to the next step. S computes $N_0 = h(Idj \parallel mj \parallel k \parallel EIDj)$ and verifies whether $M_4 =^? h(Idj \parallel EIDj \parallel N_0 \parallel M_2 \parallel M_3)$. Obviously, the verification holds. Server S generates a number $n_j^{new} \in^R Z_q^*$ and computes $EID_j^{new} = h(Idj \parallel n_j^{new})$, $N_0^{new} = h(Idj \parallel mj \parallel k \parallel EID_j^{new})$. Subsequently, S updates $\{Idj, Honey\ List = Null\}$ in its back-end database. Moreover, S picks $r_s^* \in^R Z_q^*$ and computes$M_5^* = T_{r_s}^*(y)$, $M_6^* = T_{r_s}^*(M_1^*)$, $M_7^* = (EID_j^{new} \parallel N_0^{new}) \oplus h(M_6^* \parallel N_0)$, $SK_j^* = h(Idj \parallel EIDj \parallel N_0 \parallel N_0^{new} \parallel M_2^* \parallel M_6^*)$ and $M_8^* = h(Idj \parallel EIDj \parallel N0 \parallel M_5^* \parallel M_2^* \parallel SK_j^*)$. Server S sends message $\{M_5^*, M_7^*, M_8^*\}$ to the user . The attacker $\boldsymbol{A}$ will intercept the message to terminate the communication. By this way, the attacker can launch DoS attack on the server S, which will result in the computing and communication loss of the server.

5. CONCLUSION

In this paper, we have first reviewed the recently proposed Shuming et al. Scheme. We have exhibited that it cannot safe against smart card loss attacks. We have then shown that their scheme is vulnerable to privileged-insider attacks with the help of both offline password guessing attacks, user impersonation, and Parallel session attacks and thus, their scheme fails to prevent known session specific temporary information attack. Further, their scheme cannot provide strong user's anonymity property. Also, we have demonstrated the drawbacks in Shuming et al. Scheme that it can launch a DoS attack. In the future, we aim to design a novel and more secure three-factor authentication protocol using biometric-based smart card and Extended Chaotic-Maps to withstand the security flaws found in Shuming et al. Scheme.

Acknowledgements
The authors would like to thank everyone, just everyone!

References

[1] Ratna Dutta and Rana Barua. Overview of key agreement protocols. *IACR Cryptol. ePrint Arch.*, 2005:289, 2005.

[2] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[3] Di Xiao, Xiaofeng Liao, and Shaojiang Deng. A novel key agreement protocol based on chaotic maps. *Information Sciences*, 177(4):1136–1142, 2007.

[4] Xianfeng Guo and Jiashu Zhang. Secure group key agreement protocol based on chaotic hash. *Information Sciences*, 180(20):4069–4074, 2010.

[5] Wen-Shenq Juang, Sian-Teng Chen, and Horng-Twu Liaw. Robust and efficient password authenticated key agreement using smart cards. *IEEE Transactions on Industrial Electronics*, 55(6):2551–2556, 2008.

[6] Da-Zhi Sun, Jin-Peng Huai, Ji-Zhou Sun, Jian-Xin Li, Jia-Wan Zhang, and Zhi-Yong Feng. Improvements of juang's password-authenticated key agreement scheme using smart cards. *IEEE Transactions on Industrial Electronics*, 56(6):2284–2291, 2009.

[7] Xiangxue Li, Weidong Qiu, Dong Zheng, Kefei Chen, and Jianhua Li. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Transactions on Industrial Electronics*, 57(2):793–800, 2009.

[8] Jia-Lun Tsai, Nai-Wei Lo, and Tzong-Chen Wu. Novel anonymous authentication scheme using smart cards. *IEEE Transactions on Industrial Informatics*, 9(4):2004–2013, 2012.

[9] Cheng-Chi Lee, Chin-Ling Chen, Chia-Ying Wu, and Shiow-Yuan Huang. An extended chaotic maps-based key agreement protocol with user anonymity. *Nonlinear Dynamics*, 69(1):79–87, 2012.

[10] Debiao He, Yitao Chen, and Jianhua Chen. Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol. *Nonlinear Dynamics*, 69(3):1149–1157, 2012.

[11] Cheng-Chi Lee and Che-Wei Hsu. A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. *Nonlinear Dynamics*, 71(1):201–211, 2013.

[12] Cheng Guo and Chin-Chen Chang. Chaotic maps-based password-authenticated key agreement using smart cards. *Communications in Nonlinear Science and Numerical Simulation*,18(6):1433–1440, 2013.

[13] Tian-Hua Liu, Qian Wang, and Hong-Feng Zhu. A multi-function password mutual authentication key agreement scheme with privacy preserving. *J. Inf. Hiding Multim. Signal Process.*, 5(2):165–178, 2014.

[14] SK Hafizul Islam. Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps. *Nonlinear Dynamics*, 78(3):2261–2276, 2014.

[15] Qi Jiang, Fushan Wei, Shuai Fu, Jianfeng Ma, Guangsong Li, and Abdulhameed Alelaiwi. Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. *Nonlinear Dynamics*, 83(4):2085–2101, 2016.

[16] Tian-Fu Lee. Enhancing the security of password authenticated key agreement protocols based on chaotic maps. *Information Sciences*, 290:63–71, 2015.

[17] Yu Liu and Kaiping Xue. An improved secure and efficient password and chaos based two-party key agreement protocol. *Nonlinear Dynamics*, 84(2):549–557, 2016.

[18] Jia-Lun Tsai and Nai-Wei Lo. Secure anonymous key distribution scheme for smart grid. *IEEE transactions on smart grid*, 7(2):906–914, 2015.

[19] Vanga Odelu, Ashok Kumar Das, Mohammad Wazid, and Mauro Conti. Provably secure authenticated key agreement scheme for smart grid. IEEE Transactions on Smart Grid, 9(3):1900–1910, 2016.

[20] Sandip Roy, Santanu Chatterjee, Ashok Kumar Das, Samiran Chattopadhyay, Saru Kumari, and Minho Jo. Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things. *IEEE Internet of Things Journal*, 5(4):2884–2895, 2017.

[21] SK Hafizul Islam, Pandi Vijayakumar, Md Zakirul Alam Bhuiyan, Ruhul Amin, Balamurugan Balusamy, et al. A provably secure three-factor session initiation protocol for multimedia big data communications. *IEEE Internet of Things Journal*, 5(5):3408–3418, 2017.

[22] Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, and Joel JPC Rodrigues. Secure three-factor user authentication scheme for renewable-energy-based smart grid environment. *IEEE Transactions on Industrial Informatics*, 13(6):3144–3153, 2017.

[23] Shuming Qiu, Ding Wang, Guoai Xu, and Saru Kumari. Practical and provably secure three - factor authentication protocol based on extended chaotic-maps for mobile lightweight devices. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[24] Pina Bergamo, Paolo D'Arco, Alfredo De Santis, and Ljupco Kocarev. Security of public-key cryptosystems based on chebyshev polynomials. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(7):1382–1393, 2005.

[25] Linhua Zhang. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos, Solitons & Fractals*, 37(3):669–674, 2008.

[26] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *In International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer, 2004.

[27] Anil Jain, Lin Hong, and Sharath Pankanti. Biometric identification. *Communications of the ACM*, 43(2):90–98, 2000.

[28] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3):614–634, 2001.

[29] Debiao He and Ding Wang. Robust biometrics-based authentication scheme for multiserver environment. *IEEE Systems Journal*, 9(3):816–823, 2014.

[30] SK Hafizul Islam. Design and analysis of an improved smartcard-based remote user password authentication scheme. *International Journal of Communication Systems*, 29(11):1708–1719, 2016.

[31] Debiao He, Neeraj Kumar, Muhammad K Khan, and Jong-Hyouk Lee. Anonymous two-factor authentication for consumer roaming service in global mobility networks. *IEEE Transactions on Consumer Electronics*, 59(4):811–817, 2013.

[32] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. *In International conference on the theory and applications of cryptographic techniques*, pages 453–474. Springer, 2001.

[33] Zhaohui Cheng, Manos Nistazakis, Richard Comley, and Luminita Vasiu. On the indistinguishability-based security model of key agreement protocols-simple cases. *Cryptology ePrint Archive*, 2005.

[34] Dheerendra Mishra, Ashok Kumar Das, and Sourav Mukhopadhyay. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Systems with Applications*, 41(18):8129–8143, 2014.

[35] Qi Xie, Zhixiong Tang, and Kefei Chen. Cryptanalysis and improvement on anonymous three-factor authentication scheme for mobile networks. *Computers & Electrical Engineering*, 59:218–230, 2017.

[36] Ding Wang and Ping Wang. Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE transactions on dependable and secure computing*, 15(4):708–722, 2016.

[37] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. *In Annual international cryptology conference*, pages 388–397. Springer, 1999.

[38] Thomas S Messerges, Ezzat A Dabbish, and Robert H Sloan. Examining smartcard security under the threat of power analysis attacks. *IEEE transactions on computers*, 51(5):541–552, 2002.

Authors

Suresh Devanapalli is currently working towards his Ph.D. degree from the Department of Mathematics, Osmania University, Hyderabad, India. He has received his M.Sc. degree in Mathematics from the Kakatiya University, Warangal, India. He has received his M.Tech. degree in Computer Science and Data Processing from the Indian Institute of Technology, Kharagpur, India. He Secured All India 58th Rank in Council for Scientific and Industrial Research – Junior Research Fellowship (CSIR-JRF-2009) in Mathematical Sciences and Qualified Graduate Aptitude Test in Engineering (GATE-2010) in Mathematics. His current research interests include Cryptographic authentication protocol and network security.

Kolloju Phaneendra is an Associate Professor in the Department of Mathematics at University College of Science, Osmania University, Hyderabad, India. He received his M.Sc and Ph.D. degree from National Institute of Technology Warangal, India. He qualified CSIR-JRF-2003 in Mathematical Sciences. His current research interests include cryptography, network security, Numerical Analysis, Numerical solutions to Singularly perturbed boundary value problems, differential difference equations, multiparameter BVPs. He has authored over 50 papers in international journals and conferences.