# *Efficient biometric and password based mutual authentication for consumer USB mass storage devices*

Article

Accepted Version

**Full-Text version**

Title: **Efficient Biometric and Password Based Mutual Authentication for Consumer USB Mass Storage Devices**

Authors: Debasis Giri,
Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia-721657, India (e-mail: debasis_giri@hotmail.com)

R. Simon Sherratt, *Fellow*, *IEEE*
School of Systems Engineering, the University of Reading, RG6 6AY, UK
(e-mail: sherratt@ieee.org)

Tanmoy Maitra,
Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India
(e-mail: tanmoy.maitra.in@ieee.org)

Ruhul Amin,
Department of Computer Science and Engineering, Indian Schools of Mines University, Dhanbad-826004, India (e-mail: amin_ruhul@live.com)

Publication: **IEEE Transactions on Consumer Electronics**
Volume: 61
Issue: 4
pp.: 491 - 499
Date: November 2015
DOI: 10.1109/TCE.2015.7389804

## Abstract

A Universal Serial Bus (USB) Mass Storage Device (MSD), often termed a USB flash drive, is ubiquitously used to store important information in unencrypted binary format. This low cost consumer device is incredibly popular due to its size, large storage capacity and relatively high transfer speed. However, if the device is lost or stolen an unauthorized person can easily retrieve all the information. Therefore, it is advantageous in many applications to provide security protection so that only authorized users can access the stored information. In order to provide security protection for a USB MSD, this paper proposes a session key agreement protocol after secure user authentication. The main aim of this protocol is to establish session key negotiation through which all the information retrieved, stored and transferred to the USB MSD is encrypted. This paper not only contributes an efficient protocol, but also does not suffer from the forgery attack and the password guessing attack as compared to other protocols in the literature. This paper analyses the security of the proposed protocol through a formal analysis which proves that the information is stored confidentially and is protected offering strong resilience to relevant security attacks. The computational cost and communication cost of the proposed scheme is analyzed and compared to related work to show that the proposed scheme has an improved tradeoff for computational cost, communication cost and security.

## Index Terms

Authentication, Security, Biometric, Consumer Storage, Password, MSD, USB.

# I. INTRODUCTION

Universal Serial Bus (USB) is a well-accepted ubiquitous serial interface, typically used for connecting peripherals such as keyboards, cell phones, printers, Mass Storage Devices (MSD), etc. to a host PC or though USB on-the-go to peer devices, primarily due to high availability and ease of connectivity. This ease of connectivity offers many advantages, but it does suffer from significant weaknesses such as (1) an unauthorized user could read or steal confidential information easily as all the information is stored in a ‘plaintext’ format, specifically unencrypted binary, and (2) an attacker could intercept all the information sent over the bus as the channel can be open to the attacker (e.g. physical, virus or malware) between the device and the computer.

User authentication and session key agreement is an efficient way to resolve the aforementioned difficulties. It is worth noting that all the confidential files and data would then be stored in an encrypted form. The established session key from the authentication protocol would be used as an encryption key. In this regard, a user authentication and session key agreement protocol should be implemented in order to negotiate a session key. Avoiding a verification table at the server end is the most desirable property in this regard. Simple Password Exponential Key Exchange (SPEKE) method [1], proposed by Jablon is a well known Password-Authentication Key Exchange (PAKE) protocol which is based on the Diffie-Hellman key exchange protocol [2]. Subsequently, Hao and Shahandashti [3] showed that SPEKE [1] suffered from the impersonation attack and the session key negotiation attack. Simultaneous Authentication of Equals (SAE) [4] is well known PAKE protocol that was proposed by Harkins. In 2000, Hwang and Li [5], and Sun [6] proposed user authentication schemes, however Chan and Cheng [7] calculated that Hwang and Li’s scheme [5] suffered from the user impersonation attack. Furthermore, Shen, Lin and Hwang [8] subsequently proposed an enhanced scheme compared to Hwang and Li’s scheme [5]. In 2004, Ku and Chen [9] proposed a password based authentication scheme, but Yoon, Ryu and Yoo [10] showed that Ku and Chen’s scheme [9] was vulnerable to the parallel session attack so they proposed a counter measure scheme.

In 2010, Yang, Wu and Chiu [11] proposed an authentication protocol for USB MSDs. However, in order to avoid the insider attack and the off-line password guessing attack [12], research focused on using user biometrics [13] as a further factor to the user authentication protocol. Li and Hwang [14] proposed a biometrics-based remote user authentication scheme in 2010. However, in 2011, Das [15] derived that Li and Hwang’s scheme [14] had flaws in the login phase, authentication phase and password change phase. To overcome these flaws, Das [15] also proposed an authentication scheme. An [16] showed that Das’s scheme [15] cannot resist the server masquerading attack, user impersonation attack, password guessing attack and insider attack, and so proposed an improved scheme. Li *et al.* [17] found that An’s scheme [16] suffered from the denial-of-service (DoS) attack, the forgery attack and also did not provide forward secrecy. In 2014, He *et al.* [18] also proposed a biometric scheme based on a three-factor security protocol for USB MSDs. In the same year, Jiping *et al.* [19] also proposed a biometric and password based authentication scheme to overcome the weaknesses of Das’s scheme [16].

This paper, shows that the Jiping *et al.* scheme [19] does suffer from the forgery and password guessing attacks. This paper then proposes an efficient Biometric and Password based Secure User Authentication Scheme (BPSUAS) for consumer USB MSDs to overcome the weaknesses in the current literature.

The rest of the paper is organized as follows: in Section II a discussion is presented on the basic concepts of the cryptographic one-way hash function and biometric extraction mechanism as the background for this work. Section III briefly addresses the scheme presented by Jiping *et al.* [19] and the security weaknesses of their scheme is presented in Section IV. Section V presents the authentication scheme proposed in this work and the security of the proposed scheme is analyzed in

Section VI. Section VII presents the performance evaluation and Section VIII finally concludes the paper. TABLE I shows the nomenclature that is used throughout the paper.

**TABLE I**
**NOMENCLATURE**

| **Term** | **Usage** |
|---|---|
| $U_i$ | $i^{th}$ user |
| $S$ | remote server |
| $\mathcal{A}$ | adversary or attacker |
| $G$ | multiplicative cyclic group of order $n$ |
| $g$ | generator of group $G$ |
| $d(\cdot)$ *and* $des(\cdot)$ | distance measurement function |
| $x$ | secret key of server $S$ |
| $pw_i$ | password of user $U_i$ |
| $B_i$ | biometric parameter of user $U_i$ |
| $ID_i$ | identity of user $U_i$ |
| $ENC_k[\cdot]$ | symmetric key encryption by key $k$ |
| $DEC_k[\cdot]$ | symmetric key decryption by key $k$ |
| $SKu$ or $SK_s$ | shared secret session key between $U_i$ and $S$ |
| $T$ | current time-stamp |
| $\Delta T$ | estimated time delay |
| $h(\cdot)$ | cryptographic one-way hash function |
| $\oplus$ | bitwise xor operation |
| $\|\|$ | concatenation operation |
| $\times$ | multiplication operation |

## II. BACKGROUND

This section defines the collision resistant cryptographic one-way hash function [20] and the collision resistant fuzzy extractor [21], [22] in order to analyze the security of this proposed scheme.

*Definition 1*: A collision resistant cryptographic one-way hash function $h(\cdot)$ maps a binary string of an arbitrary length to a binary string of fixed length called the hashed value. It can be symbolized as: $h: H_1 \rightarrow H_2$, where $H_1 = \{0,1\}^*$, $H_2 = \{0,1\}^n$ and $n$ is a positive integer. $H_1$ is a binary string of an arbitrary length and $H_2$ is a binary string of fixed length $n$. If $Adv_A^H(t_1)$ is the advantage to an adversary $\mathcal{A}$ to choose a pair $(m, m') \in_R H_1 \times H_1$ randomly such that $h(m) = h(m')$ where $m \neq m'$ for the time duration $t_1$, it can be considered that $Adv_A^H(t_1)$ is the probability of the advantage computed over the random choices made by $\mathcal{A}$ for the time duration $t_1$. Then, $h(\cdot)$ is termed collision-resistant, if $Adv_A^H(t_1) \leq \xi_1$, for any small $\xi_1 > 0$. Thus:

$$Adv_A^H(t_1) = Pr\left[(m, m') \in_R H_1 \times H_1 \mid (m \neq m') \wedge h(m) = h(m')\right] \quad (1)$$

where $Pr[E]$ denotes the random event $E$.

*Definition 2:* A collision resistant fuzzy extractor can be modeled as a procedure which takes a binary string say, *b* of metric space *M* as an input, where $M \in \{0,1\}^n$, for some *n* bits and produces a random string say, $\psi \in_R \{0,1\}^l$, for some *l* bits and an auxiliary string say, $\theta \in \{0,1\}^r$, for some *r* bits, where *r=l* or *n* bits. This mapping procedure is known as *GEN* and it can be represented by $GEN : M \to \psi \times \theta$. Another procedure which takes a binary string say, $b^{'}$ of the metric space $M \in \{0,1\}^n$, where $b \neq b^{'}$ and a uniform distributed binary string say, $\theta \in \{0,1\}^r$, and produces the random string $\psi \in_R \{0,1\}^l$ is known as *REP* and symbolized as $REP : M \times \theta^{'} \to \psi$. If $Adv_A^{FE}(t_2)$ is the advantage to $\mathcal{A}$ to choose a pair $\left(b,b^{'}\right) \in_R M \times M$ randomly such that $des\left(b,b^{'}\right) \leq d$, $GEN(b) = GEN\left(b^{'}\right)$ and $REP(b,\theta) = REP\left(b^{'},\theta^{'}\right)$, where *d* is the difference tolerance level and $b \neq b^{'}$ for the time duration $t_2$, it can be considered that $Adv_A^{FE}(t_2)$ is the probability that the advantage is computed over the random choices made by $\mathcal{A}$ over time duration $t_2$. Then, the Fuzzy Extractor, *FE*, is called collision-resistant, if $Adv_A^{FE}(t_2) \leq \xi_2$, for any small $\xi_2 > 0$. Thus:

$$Adv_A^{FE}(t_2) = Pr\begin{bmatrix} \left(b,b^{'}\right) \in_R M \times M \mid \left(b \neq b^{'}\right) \wedge des\left(b,b^{'}\right) \leq d \wedge \\ GEN(b) = GEN\left(b^{'}\right) \wedge REP(b,\theta) = REP\left(b^{'},\theta^{'}\right) \end{bmatrix} \tag{2}$$

for all probabilistic polynomial-time algorithms *GEN* and *REP*.

## III. Brief Review of the Jiping et al. Scheme

This section briefly describes the user authentication scheme presented by Jiping *et al* [19]. Their scheme consisted of three main phases being, registration, login and authentication.

### A. Registration Phase

*i*<sup></sup>th user $U_i$ inputs their personal biometric parameter $B_i$ (e.g. fingerprint), password $pw_i$ and the identity $ID_i$ to server *S* securely. *S* computes $F_i = h(B_i)$, $G_i = h(ID_i)$, $R_i = h(pw_i) \oplus F_i$ and $E_i = h(G_i \,\|\, x) \oplus R_i$. *S* stores parameters $\langle F_i, h(\cdot), G_i, R_i, E_i, \tau, d(.)\rangle$ into the memory of $U_i$ 's MSD, where $\tau$ is a threshold value and $d(\cdot)$ is the difference measurement function. *S* then sends the USB MSD to $U_i$ securely or in person.

### B. Login Phase

In the login phase, the user inserts their USB MSD into the client terminal and provides their biometric parameter $B_i^{'}$ to the terminal which subsequently checks condition $d\left(B_i, B_i^{'}\right) \leq \tau$. If the condition holds, the terminal gives permission for $U_i$ to provide their password $pw_i$; otherwise, *S* terminates the session. After receiving $pw_i$, the device computes $R^{'}_i = h(pw_i) \oplus F_i$. If $d\left(R_i, R_i^{'}\right) \geq \tau$ then the password verification fails, and the terminal aborts the session; otherwise, the MSD computes $M_{i_1} = E_i \oplus R_i^{'}$, $M_{i_2} = h(r_i \,\|\, T_i)$ and $M_{i_3} = M_{i_1} \oplus M_{i_2}$ where $r_i$ is a random number chosen by the MSD and $T_i$ is the current login timestamp of $U_i$ . Finally, $U_i$ sends a login message $\left\langle G_i, M_{i_2}, M_{i_3}, T_i \right\rangle$ to *S* over a public channel.

### *C. Authentication Phase*

After receiving the login request message at $T_s$ , *S* first tests the condition $\left(T_s - T_i\right) \leq \Delta T$ , where $\Delta T$ is the estimated time delay. If true, *S* proceeds to the next stage; otherwise, terminates the session. *S* computes $M_{s_1} = h\left(G_i \| x\right)$ , $M_{s_2} = M_{s_1} \oplus M_{i_3}$ and tests equality $M_{i_2} = M_{s_2}$ . If true then the login message is correct; otherwise *S* aborts the session.

*S* generates random number $r_s$ and computes $M_{s_3} = h\left(r_s \| T_s\right)$ , $M_{s_4} = M_{s_1} \oplus M_{s_3}$ . *S* sends the message $\left\langle M_{s_1}, M_{s_3}, M_{s_4}, T_s \right\rangle$ to $U_i$ over a public channel.

After receiving the message from *S* at time $T_i^1$ , the MSD tests condition $\left(T_i^1 - T_s\right) \leq \Delta T$ . If false, the session is terminated; otherwise *S* computes $M_{i_4} = M_{s_1} \oplus M_{s_4}$ and tests $M_{i_4} = M_{s_3}$ . If true, the device computes $M_{s_5} = M_{s_1} \oplus M_{i_3}$ and tests $M_{i_5} = M_{s_4}$ . If true, the device computes $M_{i_6} = h\left(r_i \| T_i^2\right)$ and $M_{i_7} = M_{i_6} \oplus M_{s_4}$ , where $T_i^2$ is the current timestamp and sends $\left\langle M_{i_7}, r_i, T_i^2 \right\rangle$ to *S*.

After receiving the message at $T_s^1$ , *S* checks the condition $\left(T_s^1 - T_i^2\right) \leq \Delta T$ . If true, *S* computes $M_{s_5} = h\left(r_i \| T_i^2\right)$ , $M_{s_6} = M_{s_5} \oplus M_{s_4}$ and tests condition $M_{s_6} = M_{i_7}$ . If true, *S* accepts the login message from $U_i$ ; otherwise rejects the current session.

## IV. CRYPTANALYSIS OF THE JIPING ET AL. SCHEME

This section considers the forgery attack and off-line password guessing attack using the scheme described by Jiping *et al.* [19].

### *A. Forgery Attack*

Consider adversary $\mathcal{A}$ trapping all the communication messages between $U_i$ and *S* during execution of the protocol. Thus, $\mathcal{A}$ will know all the parameters $\left\langle G_i, M_{i_2}, M_{i_3}, T_i, M_{s_1}, M_{s_3}, M_{s_4}, T_s, M_{i_7}, r_i, T_i^2 \right\rangle$ from the messages. $\mathcal{A}$ can create a forged login message by performing the following steps for any timestamp $T_i^{[a]}$ , where $\left(T_s - T_i^{[a]}\right) \leq \Delta T$ :

1. $\mathcal{A}$ computes $M_{i_2}^{[forge]} = h\left(r_i^{[a]} \| T_i^{[a]}\right)$ and $M_{i_3}^{[forge]} = M_{s_1}^{[trap]} \oplus M_{i_2}^{[forge]}$ where $\mathcal{A}$ chooses random number $r_i^{[a]}$ . $M_{s_1}^{[trap]} = h\left(G_i \| x\right)$ is the trapped parameter from the communicating messages between $U_i$ and *S*, and $T_i^{[a]}$ is the current timestamp of $\mathcal{A}$. Then, $\mathcal{A}$ sends the login message $\left\langle G_i, M_{i_2}^{[forge]}, M_{i_3}^{[forge]}, T_i^{[a]} \right\rangle$ to *S*.

2. After receiving the login message at timestamp $T_s$ from $\mathcal{A}$, *S* tests $\left(T_s - T_i^{[a]}\right) \leq \Delta T$ (which will always be true.) *S* computes $M_{s_1} = h(G_i \| x)$ and $M_{s_2} = M_{s_1} \oplus M_{i_3}^{[forge]}$. Then, *S* tests equality $M_{i_2}^{[forge]} = M_{s_2}$. However, it will be always equal because $\mathcal{A}$ already knows the correct parameter $M_{s_1} = h(G_i \| x)$ from listening to the messages between $U_i$ and *S*.

Therefore, the forge login request message satisfies the validity of the authentication of *S*. Hence, the adversary can impersonate a valid user to login to *S*.

### *B. Off-line Password Guessing Attack*

Adversary $\mathcal{A}$ can extract information from the USB MSD by monitoring power analysis [23], [24]. Thus, if the USB MSD of $U_i$ is lost or stolen, $\mathcal{A}$ can obtain the parameters $\langle F_i, h(\cdot), G_i, R_i, E_i, \tau, d(\cdot)\rangle$. Then, $\mathcal{A}$ can perform the following steps to guess the password of $U_i$:

*Step 1:* $\mathcal{A}$ chooses a random password $pw_i^{[a]}$ and computes $R_i^{[a]} = h\left(pw_i^{[a]}\right) \oplus F_i$.

*Step 2:* Then, $\mathcal{A}$ tests the equality of $R_i^{[a]}$ and stored $R_i$. If true, the passwords $pw_i^{[a]}$ and $pw_i$ are the same and $\mathcal{A}$ has successfully guessed the password of $U_i$; otherwise, $\mathcal{A}$ can repeat *Step 1* until correct password has been obtained.

After guessing, $\mathcal{A}$ can obtain the correct password $pw_i$ of $U_i$ due to the low entropy property of the password [18]. Therefore, given time, the scheme from Jiping *et al.* [19] cannot resist the off-line password guessing attack.

## V. THE PROPOSED SCHEME

This section presents the proposed scheme derived from this research. The proposed scheme consists of a registration phase, a login phase, an authentication and session key agreement phase, a data retrieval phase and then password update phase.

Server *S* chooses a cryptographic one-way hash function $h(\cdot)$ such that when using an arbitrary input binary string then a fixed length binary string is created. This process can be symbolized as $h:\{0,1\}^* \rightarrow \{0,1\}^l$, where *l* is a fixed length (say, 128 bits) integer. *S* also chooses a multiplicative cyclic group *G* of order *n*, a generator *g* of the group *G* and a secret key *x*. Then, *S* publishes $\langle g, n, h(\cdot)\rangle$ as the public parameters and keeps *x* secret.

### *A. Registration Phase*

Whenever $U_i$ wants to access data on their USB MSD through *S*, then the registration phase is invoked. $U_i$ can register to *S* by:

1. $U_i$ provides their identity $ID_i$ to *S* over the public channel.

2. After receiving $ID_i$ from $U_i$, *S* then computes $D_i = h(ID_i \| x)$. Then, *S* creates a personalized USB MSD for user $U_i$ after storing $\langle ID_i, D_i \rangle$ into the memory of the MSD. *S* sends the USB MSD to $U_i$ securely or in person.

3. A biometric sensor generates a unique biometric parameter $B_i$ (i.e. fingerprint) being a unique biometric feature of $U_i$ . Password $pw_i$ is obtained. The MSD locally generates $(\psi_i, \theta_i) = GEN(B_i)$ , computes $F_i = h(pw_i \| ID_i) \oplus \theta_i$ , $C_i = h(pw_i \| \psi_i)$ , $K_i = ENC_{C_i}[D_i]$ and $G_i = h(C_i \| D_i)$ . The MSD stores $\langle F_i, K_i, G_i \rangle$ into the memory of the MSD instead of $D_i$ . Finally, parameters $\langle ID_i, F_i, K_i, G_i \rangle$ are stored into the memory of the MSD.

### *B. Login Phase*

When $U_i$ wants to access the memory of the MSD, $U_i$ inserts the MSD to the client terminal. The password $pw_i$ and biometric parameter $B_i$ are also entered to the terminal. After receiving $pw_i$ and $B_i$ for $U_i$, the terminal then performs the following steps:

1. The terminal computes $\theta_i^{'} = F_i \oplus h(pw_i \| ID_i)$ , $REP(B_i, \theta_i^{'}) = \psi_i^{'}$ , $C_i^{'} = h(pw_i \| \psi_i^{'})$ , $D_i^{'} = DEC_{C_i^{'}}[K_i]$ and $G_i^{'} = h(C_i^{'} \| D_i^{'})$.

2. The terminal tests $G_i^{'} = G_i$ . If the equality fails then the terminal rejects the login from $U_i$.

3. The MSD computes $M_{i_1} = h(D_i^{'} \| T_i) \oplus \alpha$ and $M_{i_2} = h(ID_i \| \alpha \| D_i^{'} \| T_i)$, where $T_i$ is the current login time of $U_i$ and $\alpha$ is a random number selected by the MSD.

4. The terminal sends $LM_i = \langle ID_i, M_{i_1}, M_{i_2}, T_i \rangle$ as a login request message to *S* over the public channel.

### *C. Authentication and Session Key Agreement Phase*

In this phase, *S* and the MSD perform the following steps:

1. After receiving the login request message $LM_i = \langle ID_i, M_{i_1}, M_{i_2}, T_i \rangle$ at time $T_s$ , *S* checks the format of $ID_i$ . If valid, *S* tests $(T_s - T_i) \le \Delta T$ . If false, *S* rejects the login message.

2. *S* then computes $D_i^* = h(ID_i \| x)$, $\alpha^* = M_{i_1} \oplus h(D_i^* \| T_i)$ and $M_{i_2}^* = h(ID_i \| \alpha^* \| D_i^* \| T_i)$.

3. Then, *S* checks the equivalency of $M_{i_2}^*$ and $M_{i_2}$ . If true, $U_i$ can then be authenticated to use *S*; otherwise, *S* terminates the session.

4. *S* generates random number $\beta$ and computes $R_i = D_i^* \oplus \beta$ , $M_{i_s} = h(ID_i \| \beta \| \alpha^* \| D_i^* \| T_{s1})$, where $T_{s1}$ is the current time of *S*. It computes shared secret session key $SK_s = g^{\alpha^* \times \beta} \bmod n$. Then, it sends a reply message $RM_i = \langle M_{i_s}, R_i, T_{s1} \rangle$ to the MSD of $U_i$.

5. After receiving the reply message at time $T_{i1}$, the terminal checks condition $\left(T_{i1}-T_{s1}\right)\leq \Delta T$. If false, the terminal rejects the reply message of *S*.

6. The MSD computes $\beta^{'}=R_i \oplus D_i^{'}$ and $M_{i_s}^{'}=h\left(ID_i \,||\, \beta^{'} \,||\, \alpha \,||\, D_i^{'} \,||\, T_{s1}\right)$. Then, the device checks equality $M_{i_s}^{'}=M_{i_s}$. If true, *S* can be authenticated to $U_i$. Then, the device computes the shared secret session key $SK_u = g^{\alpha\times\beta^{'}} \bmod n$

7. $U_i$ uses $SK_u$ to encrypt the data as $\left\langle DATA_{Id}, ENC_{SK_u}\left[DATA\right]\right\rangle$ to ensure the security of the data in the MSD memory, computes $W_i = h\left(ID_i \,||\, \psi_i^{'}\right)$ and also stores the encrypted data identity plus session key combinations as $ENC_{W_i}\left[DATA_{Id}, SK_u\right]$.

### D. Data Retrieval Phase

When $U_i$ wants to access the data in the MSD, $U_i$ inserts their MSD into the client terminal, provides their password $pw_i$ and biometric parameter $B_i$ to the terminal. Then the MSD performs steps 1 and 2 of the login phase. If the submitted $B_i$ and $pw_i$ are deemed correct, the MSD computes $W_i = h\left(ID_i \,||\, \psi_i^{'}\right)$ and decrypts the encrypted data identity plus session key combinations as $DEC_{W_i}\left(ENC_{W_i}\left[DATA_{Id}, SK_u\right]\right)$ on the MSD to obtain session key $SK_u$. After obtaining $SK_u$, the MSD decrypts the encrypted data as $DEC_{SK_u}\left(ENC_{SK_u}\left[DATA\right]\right)$.

### E. Password Update Phase

The password update phase is invoked when $U_i$ wants to change their password.

$U_i$ inserts their MSD into the terminal and submits their old password $pw_i$, new password $pw_i^{[new]}$ and $B_i$ to the terminal. The terminal performs following steps to change $U_i$'s password:

1. The terminal computes $\theta_i^{'} = F_i \oplus h\left(pw_i \,||\, ID_i\right)$, $REP\left(B_i, \theta_i^{'}\right)=\psi_i^{'}$, $C_i^{'} = h\left(pw_i \,||\, \psi_i^{'}\right)$, $D_i^{'} = DEC_{C_i}\left[K_i\right]$ and $G_i^{'} = h\left(C_i^{'} \,||\, D_i^{'}\right)$.

2. The terminal tests for $G_i^{'} = G_i$. If false, the terminal rejects $U_i$.

3. Then, the terminal computes $F_i^{[new]} = h\left(pw_i^{[new]} \,||\, ID_i\right)\oplus \theta_i^{'}$, $C_i^{[new]} = h\left(pw_i^{[new]} \,||\, \psi_i^{'}\right)$, $K_i^{[new]} = ENC_{C_i^{[new]}}\left[D_i^{'}\right]$, and $G_i^{[new]} = h\left(C_i^{[new]} \,||\, D_i^{'}\right)$.

4. Finally, the terminal replaces $F_i$, $K_i$ and $G_i$ with $F_i^{[new]}$, $K_i^{[new]}$ and $G_i^{[new]}$ respectively into the memory of the MSD.

## VI. Security Analysis of The Proposed Scheme

The formal security analysis of the proposed scheme under the random oracle model is presented in this section. This security analysis uses the formal security analysis under the generic group model of cryptography. In the following, this work defines random oracles for the formal security analysis of the proposed scheme:

- $\mathcal{O}racle\mathcal{H}$ is a random oracle which unconditionally outputs the input *m* for the corresponding given hash value $y = h(m)$.

- $\mathcal{O}racle\mathcal{FE}$ is a random oracle which contains two parts:

  1. $\mathcal{O}racle\mathcal{FE}_{\mathrm{GEN}}$ unconditionally outputs the pair $(\psi,\theta)$ from the corresponding given biometric parameter *b*;

  2. $\mathcal{O}racle\mathcal{FE}_{\mathrm{REP}}$ unconditionally outputs $\psi$ from the corresponding biometric parameter $b'$ and uniform distribution binary string $\theta'$.

*Theorem 1:* Under the assumption that a cryptographic one-way hash function $h(\cdot)$ acts as a random oracle, the proposed scheme derived from this work (BPSUAS) is then provably secure against adversary $\mathcal{A}$ for deriving the secret key *x* of server *S* after obtaining the stored information into the memory of the MSD, and capturing the login message and the reply message of the authentication phase during communication between $U_i$ and *S*.

*Proof 1:* Consider $\mathcal{A}$ has the ability to derive the secret key *x* of *S*. Assume that the MSD of $U_i$ is lost or stolen. Thus, $\mathcal{A}$ can extract the stored parameters $\langle ID_i, F_i, K_i, G_i \rangle$ from the memory of the MSD of $U_i$ by power monitoring [23], [24]. $\mathcal{A}$ also traps the login message $LM_i = \langle ID_i, M_{i_1}, M_{i_2}, T_i \rangle$ and the reply message $RM_i = \langle M_{i_s}, R_i, T_{s1} \rangle$ of the authentication phase at timestamp $T_i$ and $T_{s1}$ respectively. $\mathcal{A}$ runs the algorithm derived from this work (BPSUAS), $ALGO1^{oracle}_{A,\ BPSUAS}$ to derive the secret key *x* of *S* as given in Algorithm 1. Define the success probability of $ALGO1^{oracle}_{A,\ BPSUAS}$ as:

$$Succ1^{oracle}_{A,BPSUAS} = \left| \Pr\left[ ALGO1^{oracle}_{A,\ BPSUAS} = 1 \right] - 1 \right| \tag{3}$$

then the advantage is given by:

$$Adv1^{oracle}_{A,\ BPSUAS}(t,\ qH) = \max_A \left( Succ1^{oracle}_{A,\ BPSUAS} \right) \tag{4}$$

where the maximum is taken over all $\mathcal{A}$ with the execution time *t*, the number of queries *qH* made to the $\mathcal{O}racle\mathcal{H}$ oracle. The proposed scheme is said to be provably secure against $\mathcal{A}$ deriving the secret key *x* of *S* if $Adv1^{oracle}_{A,\ BPSUAS}(t,\ qH) \le \xi$, for any small $\xi > 0$. According to $ALGO1^{oracle}_{A,\ BPSUAS}$, if $\mathcal{A}$ is successful in computing the inversion of $h(\cdot)$, then $\mathcal{A}$ can successfully derive the secret key *x* of *S* by using of the $\mathcal{O}racle\mathcal{H}$ random oracle. But, according to Definition 1 (see Section II), $Adv^{OracleH}_{A}(t) \le \xi_1$, for any small $\xi_1 > 0$. Since, the advantage $Adv1^{oracle}_{A,\ BPSUAS}(t,\ qH) \le \xi$, for any small $\xi > 0$ because the proposed scheme depends on $Adv_{A}^{OracleH}(t)$. Thus, this proposed scheme is secure against $\mathcal{A}$ for deriving the secret key *x* of *S*.

**Algorithm 1** $ALGO1_{A,BPSUAS}^{oracle}$

```
Input : $ID_i, K_i, G_i, M_{i_1}, M_{i_2}, T_i, R_i, M_{i_s}, T_{s1}$
Output: 0 or 1
Calls $\mathcal{O}racle\mathcal{H}$ on the input $G_i$ to retrieve the information $D_i = h(ID_i || x || b)$ and $C_i = h(pw_i || \psi_i)$ as $(C_i^* || D_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(G_i)$
Decrypts $K_i$ as $D_i^{**} = DEC_{C_i^*}[K_i]$
Calls $\mathcal{O}racle\mathcal{H}$ on the input $M_{i_2}$ to retrieve the information $ID_i$, $\alpha$, $D_i$, and $T_i$ as $(ID_i^* || \alpha^* || D_i^{***} || T_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(M_{i_2})$
Computes $[h(D_i || T_i)]^* = M_{i_1} \oplus \alpha^*$
Calls $\mathcal{O}racle\mathcal{H}$ on the input $[h(D_i || T_i)]^*$ to retrieve the information $D_i$ and $T_i$ as $(D_i^{*****} || T_i^{**}) \leftarrow \mathcal{O}racle\mathcal{H}([h(D_i || T_i)]^*)$
Calls $\mathcal{O}racle\mathcal{H}$ on the input $M_{i_s}$ to retrieve the information $ID_i$, $\beta$, $\alpha$, $D_i$ and $T_{s1}$ as $(ID_i^{**} || \beta^* || \alpha^{**} || D_i^{******} || T_{s1}^{*****}) \leftarrow \mathcal{O}racle\mathcal{H}(M_{i_s})$
Computes $D_i^{*******} = R_i \oplus \beta^*$
if ($D_i^* == D_i^{**} == D_i^{***} == D_i^{*****} == D_i^{******} == D_i^{*******}$) then
    Calls $\mathcal{O}racle\mathcal{H}$ on the input $D_i^*$ retrieve the information $ID_i$ and $x$ as $(ID_i^{***} || x^*) \leftarrow \mathcal{O}racle\mathcal{H}(D_i^*)$
    if ($ID_i == ID_i^{***}$) then
        Accepts $x^*$ as correct secret key of server $S$
        Return 1 (success)
    else
        Return 0 (failure)
    end if
else
    Return 0 (failure)
end if
```

*Theorem 2:* Under the assumption that $h(\cdot)$ and *FE* act as a random oracle, this proposed scheme is provably secure against adversary $\mathcal{A}$ for deriving the password $pw_i$ of $U_i$ after obtaining the stored information in the MSD, and capturing the login message and reply message of the authentication phase during communication between $U_i$ and *S*.

*Proof 2:* Construct $\mathcal{A}$ that has the ability to derive $pw_i$ of $U_i$. Consider the same assumptions as stated in the proof of Theorem 1. $\mathcal{A}$ runs the algorithm, $ALGO2_{A,\ BPSUAS}^{oracle}$ to derive the password $pw_i$ of $U_i$ as given in Algorithm 2. Due to the need to query two oracles, define the success probability of $ALGO2_{A,\ BPSUAS}^{oracle}$ as:

$$Succ2_{A,BPSUAS}^{oracle} = \left| 2\Pr\left[ ALGO2_{A,\ BPSUAS}^{oracle} = 1 \right] - 1 \right| \tag{5}$$

then the advantage is given by:

$$Adv2_{A,\ BPSUAS}^{oracle}(t, qH, qFE) = \max_{A}\left( Succ2_{A,\ BPSUAS}^{oracle} \right) \tag{6}$$

where the maximum is taken over all $\mathcal{A}$ with the execution time *t*, the number of queries *qH* made to the $\mathcal{O}racle\mathcal{H}$ oracle and the number of queries *qFE* made to the $\mathcal{O}racle\mathcal{FE}$. The proposed scheme is said to be provably secure against $\mathcal{A}$ deriving $pw_i$ of $U_i$ if $Adv2^{oracle}_{A,\ BPSUAS}(t,\ qH, qFE) \leq \xi$, for any small $\xi > 0$. According to algorithm $ALGO2^{oracle}_{A,\ BPSUAS}$ (see Algorithm 2), if $\mathcal{A}$ has success computing the inversion of $h(\cdot)$ and gets the same pair $(B_i, B_i^{'})$ such that $\psi = \psi_i^{'}$ then $\mathcal{A}$ can successfully derive $pw_i$ of $U_i$ by using of the $\mathcal{O}racle\mathcal{H}$ random oracle and by querying to the $\mathcal{O}racle\mathcal{FE}$ respectively. But, according to Definition 1 and Definition 2, $Adv_A^{OracleH}(t) \leq \xi_1$, for any small $\xi_1 > 0$ and $Adv_A^{OracleFE}(t) \leq \xi_2$, for any small $\xi_2 > 0$. Since, advantage $Adv2^{oracle}_{A,\ BPSUAS}(t,\ qH, qFE) \leq \xi$, for any small $\xi > 0$ because this proposed scheme depends on both $Adv_A^{OracleH}(t)$ and $Adv_A^{OracleFE}(t)$. Thus, the proposed scheme is secure against $\mathcal{A}$ for deriving $pw_i$ of $U_i$.

**Algorithm 2** $ALGO2^{oracle}_{A,BPSUAS}$

```
Input : $ID_i, K_i, F_i, G_i$
Output: 0 or 1
Calls $\mathcal{O}racle\mathcal{H}$ on the input $G_i$ to retrieve the information $D_i = h(ID_i \,||\, x \,||\, b)$ and $C_i = h(pw_i \,||\, \psi_i)$ as $(C_i^* \,||\, D_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(G_i)$
Decrypts $K_i$ as $D_i^{**} = DEC_{C_i^*}[K_i]$
if ($D_i^* == D_i^{**}$) then
    Calls $\mathcal{O}racle\mathcal{H}$ on the input $C_i^*$ to retrieve the information $pw_i$ and $\Psi_i$ as $(PW_i^* \,||\, \psi_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(C_i^*)$
else
    Return 0 (failure)
end if
Chooses $B_i^*$, and calls $\mathcal{O}racle\mathcal{FE}_{\mathrm{GEN}}$ on the input $B_i^*$ to retrieve the information $\psi_i$ and $\theta_i$ as $(\psi_i^{**}, \theta_i^*) \leftarrow \mathcal{O}racle\mathcal{FE}_{\mathrm{GEN}}(B_i^*)$
if ($\psi_i^{**} == \psi_i^*$) then
    Computes $[h(pw_i \,||\, ID_i)]^* = F_i \oplus \theta_i^*$
    Calls $\mathcal{O}racle\mathcal{H}$ on the input $[h(pw_i \,||\, ID_i)]^*$ retrieve the information $PW_i$ and $ID_i$ as $(PW_i^{**} \,||\, ID_i^*) \leftarrow \mathcal{O}racle\mathcal{H}([h(pw_i \,||\, ID_i)]^*)$
    if ($PW_i^{**} == PW_i^*$) && ($ID_i == ID_i^{**}$) then
        Return 1 (success)
    else
        Return 0 (failure)
    end if
else
    Return 0 (failure)
end if
```

*Theorem 3:* Under the assumption that $h(\cdot)$ acts as a random oracle, then this proposed scheme is provably secure against $\mathcal{A}$ deriving the shared secret session key *SK* between $U_i$ and *S* after obtaining the stored information into the memory of the MSD device, and trapping the login message and reply message of authentication phase during communication between $U_i$ and *S*.

*Proof 3:* Construct adversary $\mathcal{A}$ that has the ability to derive the shared secret session key *SK* between $U_i$ and *S*. Consider the same assumptions as stated in the proof of Theorem 1. $\mathcal{A}$ runs algorithm $ALGO3^{oracle}_{A,\ BPSUAS}$ to derive the secret shared session key *SK* between $U_i$ and *S* as given in Algorithm 3. Define the success probability of $ALGO3^{oracle}_{A,\ BPSUAS}$ as:

$$Succ3^{oracle}_{A,BPSUAS} = \left| \Pr\left[ ALGO3^{oracle}_{A,\ BPSUAS} = 1 \right] - 1 \right| \tag{7}$$

then the advantage is given by

$$Adv3^{oracle}_{A,\ BPSUAS}(t,\ qH) = \max_A \left( Succ3^{oracle}_{A,\ BPSUAS} \right) \tag{8}$$

where the maximum is taken over all $\mathcal{A}$ with the execution time *t*, the number of queries *qH* made to the $\mathcal{O}racle\mathcal{H}$ oracle. The proposed scheme is said to be provably secure against $\mathcal{A}$ deriving the shared secret key *SK* between $U_i$ and *S* if $Adv3^{oracle}_{A,\ BPSUAS}(t,\ qH) \leq \xi$ , for any small $\xi > 0$. According to $ALGO3^{oracle}_{A,\ BPSUAS}$ (see Algorithm 3), if $\mathcal{A}$ is successful in computing the inversion of $h(\cdot)$ then they can successfully derive the shared secret key *SK* between $U_i$ and *S* by using the $\mathcal{O}racle\mathcal{H}$ random oracle. But, according to the Definition 1 $Adv^{OracleH}_{A}(t)) \leq \xi_1$ , for any small $\xi_1 > 0$ . Since $Adv3^{oracle}_{A,\ BPSUAS}(t,\ qH) \leq \xi$ , for any small $\xi > 0$ and because the proposed scheme depends on $Adv^{OracleH}_{A}(t)$, then the proposed scheme is secure against $\mathcal{A}$ deriving the shared secret key *SK* between $U_i$ and *S*.

**Algorithm 3** $ALGO3^{oracle}_{A,BPSUAS}$

```
Input : $ID_i, K_i, G_i, M_{i_1}, M_{i_2}, T_i, R_i, M_{i_s}, T_{s1}, n, g$
Output: 0 or 1
Calls $\mathcal{O}racle\mathcal{H}$ on the input $G_i$ to retrieve the information $D_i = h(ID_i \| x \| b)$ and $C_i = h(pw_i \| \psi_i)$ as $(C_i^* \| D_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(G_i)$
Decrypts $K_i$ as $D_i^{**} = DEC_{C_i^*}[K_i]$
Calls $\mathcal{O}racle\mathcal{H}$ on the input $M_{i_2}$ to retrieve the information $ID_i$, $\alpha$, $D_i$, and $T_i$ as $(ID_i^* \| \alpha^* \| D_i^{***} \| T_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(M_{i_2})$
Computes $[h(D_i \| T_i)]^* = M_{i_1} \oplus \alpha^*$
Calls $\mathcal{O}racle\mathcal{H}$ on the input $[h(D_i \| T_i)]^*$ to retrieve the information $D_i$ and $T_i$ as $(D_i^{****} \| T_i^{**}) \leftarrow \mathcal{O}racle\mathcal{H}([h(D_i \| T_i)]^*)$
Calls $\mathcal{O}racle\mathcal{H}$ on the input $M_{i_s}$ to retrieve the information $ID_i$, $\beta$, $\alpha$, $D_i$ and $T_{s1}$ as $(ID_i^{**} \| \beta^* \| \alpha^{**} \| D_i^{*****} \| T_{s1}^{****}) \leftarrow \mathcal{O}racle\mathcal{H}(M_{i_s})$
Computes $D_i^{******} = R_i \oplus \beta^*$
if $(D_i^* == D_i^{**} == D_i^{***} == D_i^{****} == D_i^{*****} == D_i^{******})$ && $(ID_i == ID_i^{**} == ID_i^*)$ && $(T_i^{**} == T_i^* == T_i)$ && $(T_{s1}^* == T_{s1})$ then
        Computes $SK^* = g^{\alpha^* \times \beta^*} \bmod n$
        Computes $SK^{**} = g^{\alpha^{**} \times \beta^*} \bmod n$
        if $(SK^* == SK^{**})$ then
            Return 1 (success)
    else
            Return 0 (failure)
    end if
else
    Return 0 (failure)
end if
```

## A. Discussion of the Presented Theorems

Theorem 2 demonstrated that the proposed scheme is secure against the *off-line password guessing attack*. Theorem 3 demonstrates that the proposed scheme is secure against the *session key recovery attack* because, without knowing random numbers $\alpha$ and $\beta$ then $\mathcal{A}$ cannot compute the session key $SK_u$. In the proposed scheme, all communicating messages depend on random numbers $\alpha$ or $\beta$ and the time-stamp. So, all the communication messages are guaranteed to be different for every session. Thus, $\mathcal{A}$ cannot mount a *replay attack* on this proposed scheme. In this proposed scheme, $\mathcal{A}$ cannot mount a *forgery attack* without knowing secret password $pw_i$ of $U_i$, the secret key $x$ of the server $S$ and random numbers $\alpha$ and $\beta$ generated by $U_i$ and $S$ respectively. Theorems 1 and 2 show that the secret information of the server and the user are secure from $\mathcal{A}$. Thus, it is infeasible to mount a *forgery attack* on this proposed scheme. In this proposed scheme, $U_i$ does not need to send their password $pw_i$ to $S$ in the registration phase and also for authentication purposes. Thus, the server has no knowledge of $U_i$'s password $pw_i$. Without knowing $U_i$'s $pw_i$, then $S$ cannot apply $U_i$'s $pw_i$ to another server $S^{[a]}$ to get service from $S^{[a]}$ by accessing the MSD of user $U_i$. Therefore, this proposed scheme is secure against the *insider attack*.

## VII. Performance Evaluation of the Proposed Scheme

This section compares the performance of the proposed scheme with related schemes in the literature [1], [4], [11], [14]-[16], [18], [19]. The login and authentication phases of the proposed scheme have been compared with the related existing schemes in the literature [11], [14]-[16], [18], [19] because these phases are commonly used.

TABLE II presents the communication (overhead) and storage costs of this work compared to the literature. As can be seen, while the communication cost of this work is higher than the literature due to the session management overhead (required to solve the vulnerabilities in the literature), the storage cost is comparable to the literature.

**TABLE II**
**Comparison of Communication and Storage costs of Schemes in the Current Literature Compared to This Proposed Scheme**

| **Comparison Metric** | **SPEKE [1]** | **SAE [4]** | **Yang *et al.* [11]** | **Li and Hwang [14]** | **Das [15]** | **An [16]** | **He *et al.* [18]** | **Jiping *et al.* [19]** | **Proposed scheme (BPSUAS)** |
|---|---|---|---|---|---|---|---|---|---|
| Communication Cost(bits) | 2304 | 4352 | 4672 | 576 | 832 | 704 | 1216 | 1408 | 832 |
| Storage Cost(bits) | - | - | 2176 | 448 | 576 | 576 | 384 | 896 | 448 |

TABLE III verifies the types of attacks that are considered, the key management and the authentication that the literature uses compared to this work.

**TABLE II**
**Comparison of Communication and Storage costs of Schemes in the Current Literature Compared to This Proposed Scheme**

| **Comparison Metric** | **SPEKE [1]** | **SAE [4]** | **Yang *et al.* [11]** | **Li and Hwang [14]** | **Das [15]** | **An [16]** | **He *et al.* [18]** | **Jiping *et al.* [19]** | **Proposed scheme (BPSUAS)** |
|---|---|---|---|---|---|---|---|---|---|
| Communication Cost(bits) | 2304 | 4352 | 4672 | 576 | 832 | 704 | 1216 | 1408 | 832 |
| Storage Cost(bits) | - | - | 2176 | 448 | 576 | 576 | 384 | 896 | 448 |

TABLE IV presents the computational cost of this work compared to the literature. $T_h$ is the time required for the hashing operation, $T_e$ for exponentiation operation and $T_{enc}/T_{dec}$ for the symmetric key encryption/decryption operation. Typically, the time complexity associated with these operations can be expressed as $T_e > T_{dec}/T_{enc} \approx T_h$ [25]. Although the proposed scheme has high time complexity than the literature, the proposed scheme can resist the attacks as detailed in TABLE III.

**TABLE III**
**Comparison of Attack Vulnerability and Features of Schemes in the Current Literature Compared to This Proposed Scheme**

| **Attack Vulnerability / Feature** | **SPEKE [1]** | **SAE [4]** | **Yang *et al.* [11]** | **Li and Hwang [14]** | **Das [15]** | **An [16]** | **He *et al.* [18]** | **Jiping *et al.* [19]** | **Proposed Scheme (BPSUAS)** |
|---|---|---|---|---|---|---|---|---|---|
| Forgery attack | yes | no | No | yes | yes | yes | no | yes | no |
| Insider attack | - | - | Yes | yes | yes | no | no | yes | no |
| Off-line password guessing attack | no | no | No | yes | yes | no | no | yes | no |
| Inefficient login phase | - | - | Yes | yes | no | yes | no | no | no |
| Replay attack | no | no | Yes | yes | yes | yes | yes | yes | no |
| Session key agreement | yes | yes | Yes | no | no | no | yes | no | yes |
| Mutual authentication | no | yes | Yes | no | no | no | yes | no | yes |

Taking the practical and reasonable assumption that the length of the identity $ID_i$ and password $pw_i$ parameters are 64 bits each; cryptographic one-way hash function $h(\cdot)$, symmetric key encryption/decryption, time-stamp and random numbers returns 128 bits each, $g^{\alpha}$ and $g^{\beta}$ returns 1024 bits each, then in this research only $64+(6\times128)=832$ bits are needed for communications overhead. Therefore this work has a lower communications overhead than the Yang *et al.* [11], Jiping *et al.* [19] and He *et al.* [18] schemes. The low communications cost compared to the wide range of attacks that can be resisted means that the proposed scheme offers an improved trade-off with the computational cost, communication cost and security than in the literature. This scheme is therefore a practical solution for USB MSD security.

**TABLE IV**
**COMPARISON OF COMPUTATIONAL COST OF RELATED SCHEMES COMPARED TO THIS PROPOSED SCHEME**

| **Phase Entity** | **Login (USB MSD)** | **Authentication (USB MSD)** + | **Session Key (Server)** | **Total** |
|---|---|---|---|---|
| SPEKE [1] | - | $2T_e + 3T_h$ | $2T_e + 3T_h$ | $4T_e + 6T_h$ |
| SAE [4] | - | $3T_e + 1T_h$ | $3T_e + 1T_h$ | $6T_e + 2T_h$ |
| Yang *et al.* [11] | $1T_h+2T_e$ | $2T_e+1T_{dec} + 2T_h$ | $6T_e+3T_h + 1T_{enc}$ | $10T_e+2T_{dec/enc} + 7T_h$ |
| Li and Hwang [14] | $2T_h$ | $2T_h$ | $3T_h$ | $7T_h$ |
| Das [15] | $3T_h$ | $3T_h$ | $5T_h$ | $11T_h$ |
| An [16] | $3T_h$ | $2T_h$ | $4T_h$ | $9T_h$ |
| He *et al.* [18] | $4T_h$ | $2T_h+1T_{dec}$ | $5T_h+1T_{enc}$ | $11T_h+2T_{enc/dec}$ |
| Jiping *et al.* [19] | $3T_h$ | $3T_h$ | $5T_h$ | $11T_h$ |
| Proposed scheme | $5T_h+1T_{dec}$ | $1T_h+1T_e$ | $4T_h+1T_e$ | $10T_h+2T_e+1T_{dec}$ |

## VIII. CONCLUSION

This paper has enhanced the current state of the art in biometric security algorithms by defending against the forgery attack, the off-line password guessing attack and the replay attack. An efficient mutual authentication protocol has been presented in order to negotiate session keys which were used to encrypt data for a USB MSD device enabling secure “USB memory sticks”. Moreover, the paper has formally proved that the proposed protocol can withstand all the relevant security weaknesses. A performance comparison has also been made with the literature to confirm that the proposed scheme achieves a comparatively better trade-off among computation cost, communication cost and security than other related schemes. The overall efficiency demonstrates that USB based Mass Storage Devices (MSDs) with biometric security sensors can be implemented in order to provide significant security for the consumer and beyond.

## REFERENCES

[1] D.P. Jablon, “Strong password-only authentication key exchange,” *SIGCOMM Comput. Commun. Rev.* vol. 26, no. 5, pp. 5-26, Oct. 1996.

[2] W. Diffie, and M.E. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theor.* vol. 22, no. 6, pp. 644-654, Nov 1976.

[3] F. Hao, and S.F. Shahandashti, “The SPEKE protocol revisited,” *Security Standardisation Research, Lecture Notes in Computer Science.* vol. 8893, pp. 26-38.

[4] D. Harkins, “Simultaneous Authentication of Equals: A secure, password-based key exchange for mesh networks,” in Proc. 2nd IARIA Int. Conf. Sensor Technologies and Applications, Cap Esterel, pp. 839-844, Aug 2008.

[5] M.-S. Hwang, and L.-H. Li, “A new remote user authentication scheme using smart cards,” *IEEE*

*Trans. Consum. Electron.* vol. 46, no. 1, pp. 28-30, Feb. 2000.

[6] H.–M. Sun, “An efficient remote use authentication scheme using smart cards,” *IEEE Trans. Consum. Electron.* vol. 46, no. 4, pp. 958-961, Nov. 2000.

[7] C.-K. Chan, and L.M. Cheng, “Cryptanalysis of a remote user authentication scheme using smart cards,” *IEEE Trans. Consum. Electron.* vol. 46, no. 4, pp. 992-993, Nov. 2000.

[8] J.-J. Shen, C.-W. Lin, and M.-S. Hwang, “A modified remote user authentication scheme using smart cards,” *IEEE Trans. Consum. Electron.* vol. 49, no. 2, pp. 414-416, May 2003.

[9] W.-C. Ku, and S.-M. Chen, “Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards,” *IEEE Trans. Consum. Electron.* vol. 50, no. 1, pp. 204-207, Feb 2004.

[10] E.-J. Yoon, E.-K. Ryu, and K.-Y. Yoo, “Efficient remote user authentication scheme based on generalized Elgamal signature scheme,” *IEEE Trans. Consum. Electron.* vol. 50, no. 2, pp. 568-570, May 2004.

[11] F.-Y. Yang, T.-D. Wu, and S.-H. Chiu, “A secure control protocol for USB mass storage devices,” *IEEE Trans. Consum. Electron.* vol. 56, no. 4, pp. 2339-2343, Nov. 2010.

[12] B. Ruppert, and R. Wanner, “Protecting against insider attacks,” *SANS Institute InfoSec Reading Room,* April. 2009.

[13] D.-J. Kim, and K.-S. Hong, “Multimodal biometric authentication using teeth image and voice in mobile environment,” *IEEE Trans. Consum. Electron.* vol. 54, no. 4, pp. 1790-1797, Nov. 2008.

[14] C.-T. Li, and M.-S. Hwang, “An efficient biometrics-based remote user authentication scheme using smart cards,” *J. Network and Computer Applications.* vol. 33, no. 1, pp. 1-5, Jan. 2010.

[15] A.K. Das, “Analysis and improvement on an efficient biometric based remote user authentication scheme using smart cards,” *IET Information Security.* vol. 5, no. 3, pp. 145-151, Sept. 2011.

[16] Y. An, “Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards,” *J. Biomedicine and Biotechnology.* Article ID 519723, pp. 1-6, Sept. 2012.

[17] X. Li, J. Niu, M. K. Khan, J. Liao, and X. Zhao, “Robust three-factor remote user authentication scheme with key agreement for multimedia systems,” *J. Security and Communication Networks*, ISSN 1939-0122. Mar. 2014.

[18] D. He, N. Kumar, J.-H. Lee, and R.S. Sherratt, “Enhanced three-factor security protocol for consumer USB mass storage devices,” *IEEE Trans. Consum. Electron.* vol. 60, no. 1, pp. 30-37, Feb. 2014.

[19] L. Jiping, D. Yaoming, X. Zenggang, and L. Shouyin, “An improved biometric-based user authentication scheme for c/s system,” *Int. J. Distributed Sensor Networks.* Article ID 275341, pp. 1-9, April 2014.

[20] P. Sarkar, “A simple and generic construction of authenticated encryption with associated data,” *ACM Trans. Inf. Syst. Secur.,* vol. 13, no. 4, Article 33, Dec. 2010.

[21] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in Proc. Int. Conf. Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, Lecture Notes in Computer Science, vol. 3027, pp 523-540, 2004.

[22] X. Boyen, “Reusable cryptographic fuzzy extractors,” in Proc. 11th ACM Conf. Computer and Communications Security, Washington, DC, pp 82-91, Oct. 2004.

[23] P.C. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in Proc. 19th Annual Int. Cryptology Conf. Advances in Cryptology, Santa Barbara, CA, pp 388-397, Aug. 1999.

[24] T.S. Messerges, E.A. Dabbish, and R.H. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE Trans. Comput.* vol. 51, no. 5, pp. 541-552, May 2002.

[25] N.R. Potlapally, S. Ravi, A. Raghunathan, and N.K. Jha, “A study of the energy consumption characteristics of cryptographic algorithms and security protocols,” *IEEE Trans. Mobile Comput.* vol. 5, no. 2, pp. 128-143, Feb. 2006.

## BIOGRAPHIES

**Debasis Giri** received the Ph.D degree from the Indian Institute of Technology, Kharagpur, India in 2009. He did his masters (M.Tech and M.Sc) both from Indian Institute of Technology, Kharagpur in 2001 and 1998 respectively. Presently he is a Professor in the Department of Computer Science and Engineering, Haldia Institute of Technology, India. He has tenth All India Rank with percentile score 98.42 in the Graduate Aptitude Test in Engineering (GATE) Examination in 1999. He taught several courses such as Discrete Mathematics, Cryptography, Information Security, Coding Theory and Advanced Algorithms, etc. His current research interests include cryptography, Network security, Security in Wireless Sensor Networks and Security in VANETs.

Dr. Giri is an Editorial Board Member and a Reviewer of many reputed International Journals. Presently he is an Associate Editor of the Journal of Security and Communication Networks (Wiley), and the Journal of Electrical and Computer Engineering Innovations. He is also a Program Committee member for many International Conferences.

**R. Simon Sherratt** (M’97-SM’02-F’12) received the B.Eng. degree in Electronic Systems and Control Engineering from Sheffield City Polytechnic, UK in 1992, M.Sc. in Data Telecommunications in 1994 and Ph.D. in video signal processing in 1996 from the University of Salford, UK.

In 1996, he was appointed as a Lecturer in Electronic Engineering at the University of Reading where he is now a Professor of Consumer Electronics and Head of Wireless and Computing research. His research topic is signal processing in consumer electronic devices.

Eur Ing Professor Sherratt was an IEEE Consumer Electronics Society Vice President (08-09) and a serving AdCom member (03-08, 10-15). He received the IEEE Chester Sall Memorial 1$^{st}$ Place Award in 2006 and is now the Editor-in-Chief of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS.

**Tanmoy Maitra** received his B.E. degree in Computer Science and Engineering from Burdwan University, India in 2009 and his M.Tech degree in Computer Science and Engineering from WBUT, India in 2013. Currently, he is pursuing a Ph.D from Jadavpur University, India. His research interests include wireless sensor networks and applied cryptology.

**Ruhul Amin** received both the B.Tech and M.Tech degrees in Computer Science and Engineering from West Bengal University of Technology in 2009 and 2013 respectively. Currently, he is pursing a Ph.D in the Department of CSE, Indian school of mines, Dhanbad, India. His current research interests include cryptographic authentication protocols and security in wireless sensor networks.

.