Full-Text version

Title: **A Novel and Efficient Session Spanning Biometric and Password based Three-factor Authentication Protocol 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*
Department of Biomedical 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)

Publication: **IEEE Transactions on Consumer Electronics**
Volume: 62
Issue: 3
pp.: 283 - 291
Date: August 2016
DOI: 10.1109/TCE.2016.7613195

Abstract

This paper proposes a key agreement scheme after secure authentication to prevent the unauthorized access of the data stored in a Universal Serial Bus (USB) Mass Storage Device (MSD). Due to the system architecture of this proposed scheme, authorized users can store their data in a secure encrypted form after performing authentication. The novelty of this work is that users can retrieve the encrypted data in not only the current session but also across different sessions, thus reducing the required communications overhead. This paper then analyses the security of the proposed protocol through a formal analysis to demonstrate that the information has been stored securely and is also protected offering strong resilience to relevant security attacks. The computational and communication costs of the proposed scheme is analyzed and compared to related works to show that the proposed scheme has an improved tradeoff for computational cost, communication cost and security.

Index Terms

Biometrics, Consumer Storage, Mass Storage Device, Password, USB, FLASH.

I. INTRODUCTION

Universal Serial Bus (USB) is a well-accepted ubiquitous serial interface. It is commonly used to connect peripheral devices such as keyboards, mice, cell phones, printers, scanner, Mass Storage Devices (MSD), etc. to a host Personal Computer (PC). The high availability, high data transfer rate and ease of connectivity are primary advantages of USB. However, at an application level, MSD's suffer from significant security weaknesses such as (1) an unauthorized user could easily read or steal confidential information as the information is stored in plaintext form, and (2) an attacker could intercept all the information sent over the bus as the channel between device and the host computer USB port can be open to the attacker (e.g. physical, virus or malware). User authentication and session key agreement are designed in such a way that they can resolve the aforementioned difficulties.

Many authors have proposed authentication schemes. In 2004, Ku and Chen [1] proposed an authentication scheme which was solely based on passwords, but Yoon, Ryu and Yoo [2] found that Ku and Chen's scheme [1] was vulnerable to the parallel session attack and so they proposed an enhanced authentication scheme. In 2010, Yang, Wu and Chiu [3] also proposed a password based protocol for USB MSDs. Research then focused on complementing passwords with user biometrics [4] to offer a further factor to the authentication protocol in order to provide improved security. In 2013, Lee, Chen and Wu [5] proposed a biometric based three-factor authentication protocol for USB MSDs. However, in 2014, He *et al.* [6] showed that Lee, Chen and Wu's scheme [5] could not resist the password guessing attack, the Denial of Service (DoS) attack and the replay attack, so He *et al.* [6] proposed a biometrics based three-factor security protocol for USB MSDs. In 2015, Giri *et al*. [7] also proposed an authentication scheme for USB MSDs using biometrics and password protection, where after performing mutual authentication, a session key was established to encrypt user's data.

This research has identified that there are two drawbacks in the previously presented schemes, (1) each time a user sends a request to the authentication server to decrypt a stored file the server sends the same part of the key. If an authorized user stores that part of the key then it is no longer needed to send a subsequent request to the server to obtain the same part of key at a later date. However, in security protocols for consumer USB MSDs it is required to authenticate each and every time a user wants to store and retrieve the file; (2) if an authorized user wishes to retrieve a stored encrypted file within the same login session as when the file was originally stored into the memory of USB device, then the user has to perform all the authentication steps (mainly login phase) which unfortunately increases the computational and communication costs, and reduces device usability.

This research presents a system architecture for creating a Three-factor Security Protocol (TSP), where authorized users can store their files in encrypted form and not only can retrieve the original file in the current session but additionally in subsequent sessions after performing the authentication procedure respectively.

The rest of the paper is organized as follows: Section II presents the required mathematical concepts. Section III introduce the proposed scheme by describing the system architecture. Section IV presents the authentication scheme proposed in this work and the security of the proposed scheme is analyzed in Section V. Section VI presents the performance evaluation and Section VII concludes the paper. Table I shows the nomenclature that is used throughout the paper.

TABLE I
NOMENCLATURE

Term	Usage
U_i	*i-th* user
AS	Remote authentication server
x	Secret key of authentication server *AS*
pw_i	Password of user U_i
B_i	Biometric parameter of user U_i
ID_i	Identity of user U_i
$ENC_k[\ ]$	Symmetric key encryption by a key k
$DEC_k[\ ]$	Symmetric key decryption by a key k
SK_i	Shared secret session key between user U_i and server *AS*
α_i *and* δ_i	Random numbers generated by *USB* device
β_i *and* γ_i	Random numbers generated by server *AS*
F_i	Unique identity or index of a file
F_{name}	File name
$des(\cdot)$	Difference measurement function
d	Threshold value
r	Integer, first r number of bits of a file
n	Integer, total number of bits of a file
T	Current timestamp
ΔT	Estimated time delay
$h(\cdot)$	Cryptographic one-way hash function
$\oplus$	Bitwise xor operation
$\|\|$	Concatenation operation

II. BACKGROUND

This section defines the collision resistant cryptographic one-way hash function [7], the collision resistant fuzzy extractor [7], [8] and collision resistant secure encryption/decryption technique [9], [10] in order to analyze the security of this proposed scheme.

Definition 1: A collision resistant cryptographic one-way hash function maps a binary string of an arbitrary length to a binary string of fixed length called the hashed value. This paper considers the cryptographic one-way hash function as previously defined [7].

Definition 2: A collision resistant fuzzy extractor can be modeled as two procedures: 1) *GEN*, which takes a binary string as input, and generates two strings, namely, arbitrary string of length *l* bit and auxiliary string of length *r*-bit; 2) *REP*, which takes a binary string and an auxiliary string, and produces an arbitrary string. Furthermore, this paper also defines the same definition of collision resistant fuzzy extractor as previously defined [7].

Definition 3: A secret key is used to encrypt plaintext into ciphertext and the same key is used to decrypt the ciphertext using the symmetric key block encryption/decryption technique. This process can be symbolized as: $C \leftarrow ENC_k[M]$ and $M \leftarrow DEC_k[C]$, where $M = \{0,1\}^*$, $C = \{0,1\}^*$ and $k = \{0,1\}^n$, given plaintext M and ciphertext C being binary strings of arbitrary length, secret key k being a binary string of fixed length *n* and ENC/DEC are encryption and decryption algorithms respectfully. If $Adv_A^{ENC/DEC}(t_3^1)$ is the advantage to an adversary $\mathcal{A}$ to choose a key $k \in_R \{0,1\}^n$ randomly such that $M = DEC_k[ENC_k[M]]$ or $C = ENC_k[DEC_k[C]]$

for the time duration t_3^1, it can be considered that $Adv_A^{ENC/DEC}\left(t_3^1\right)$ is the advantage computed over random choices made by adversary $\mathcal{A}$ for time duration t_3^1. Then the symmetric key encryption/decryption ENC/DEC can be called secure if $Adv_A^{ENC/DEC}(t_3^1) \le \xi_3^1$, for any small $\xi_3^1 > 0$. $Adv_A^{ENC/DEC}(t_3^1)$ is represented as:

$$Adv_A^{ENC/DEC}\left(t_3^1\right) = \Pr\left[\begin{array}{c} k \in_R \{0,1\}^n \mid C = ENC_k\left[DEC_k\left[C\right]\right] \vee \\ M = DEC_k\left[ENC_k\left[M\right]\right] \end{array}\right] \tag{1}$$

The symmetric key encryption/decryption algorithm also follows the property of collision resistance as described in Definition 1. If $Adv_A^{ENC/DEC}\left(t_3^2\right)$ is the advantage to $\mathcal{A}$ to choose a pair $\left(M, M^{'}\right) \in_R \{0,1\}^* \times \{0,1\}^*$ randomly such that $ENC_k\left[M\right] = ENC_k\left[M^{'}\right]$ where $M \neq M^{'}$ or $\left(C, C^{'}\right) \in_R \{0,1\}^n \times \{0,1\}^n$ such that $DEC_k\left[C\right] = DEC_k\left[C^{'}\right]$ where $C \neq C^{'}$ for the time duration t_3^2, it can be considered that $Adv_A^{ENC/DEC}\left(t_3^2\right)$ is the advantage computed over random choices made by $\mathcal{A}$ for time duration t_3^2. Then the symmetric key encryption/decryption *ENC/DEC* is called collision resistant if $Adv_A^{ENC/DEC}\left(t_3^2\right) \le \xi_3^2$, for any small $\xi_3^2 > 0$. $Adv_A^{ENC/DEC}\left(t_3^2\right)$ is represented as:

$$Adv_A^{ENC/DEC}\left(t_3^2\right) = \Pr\left[\begin{array}{c} \left(\begin{array}{c} \left(M, M^{'}\right) \in_R \{0,1\}^* \times \{0,1\}^* \mid \\ \left(M \neq M^{'}\right) \wedge \\ ENC_k\left[M\right] = ENC_k\left[M^{'}\right] \end{array}\right) \\ \vee \\ \left(\begin{array}{c} \left(C, C^{'}\right) \in_R \{0,1\}^n \times \{0,1\}^n \mid \\ \left(C \neq C^{'}\right) \wedge \\ DEC_k\left[C\right] = DEC_k\left[C^{'}\right] \end{array}\right) \end{array}\right] \tag{2}$$

$Adv_A^{ENC/DEC}\left(t_3\right)$ is the advantage to $\mathcal{A}$ for the time duration t_3, given by:

$$Adv_A^{ENC/DEC}\left(t_3\right) = \max_A\left\{Adv_A^{ENC/DEC}\left(t_3^1\right), Adv_A^{ENC/DEC}\left(t_3^2\right)\right\} \tag{3}$$

Then the symmetric key encryption/decryption *ENC/DEC* is called collision resistant and secure, if $Adv_A^{ENC/DEC}\left(t_3\right) \le \xi_3$, for any small $\xi_3 > 0$.

III. ARCHITECTURE

This section presents the details of the system architecture of the proposed scheme.

At first, user U_i registers with an authentication server, *AS*, in order to request authorized access to the USB MSD as shown in Fig. 1(a). *AS* has the responsibility to verify the legitimacy of U_i when U_i wishes to read or write a file to a storage device via the USB interface. U_i enters their identity, password and biometric parameters to verify legitimacy. Fig. 1(b) shows the communication procedure when U_i wants to store a new file into the USB MSD in an encrypted form or wants to decrypt an already stored encrypted file in the USB device.

This research has considered two conditions (1) U_i can store a new file in an encrypted form after performing mutual authentication between U_i and *AS* in session S_i and can decrypt the newly stored encrypted file within the session S_i after obtaining permission from *AS*, (2) U_i can decrypt the stored encrypted file within a different session, S_j, after performing mutual authentication between U_i and *AS*. Moreover in this proposed system architecture, the authentication plays a vital role whenever the users try to store/extract a file to/from their USB MSD, each and every time the users need to be authenticated to *AS*.

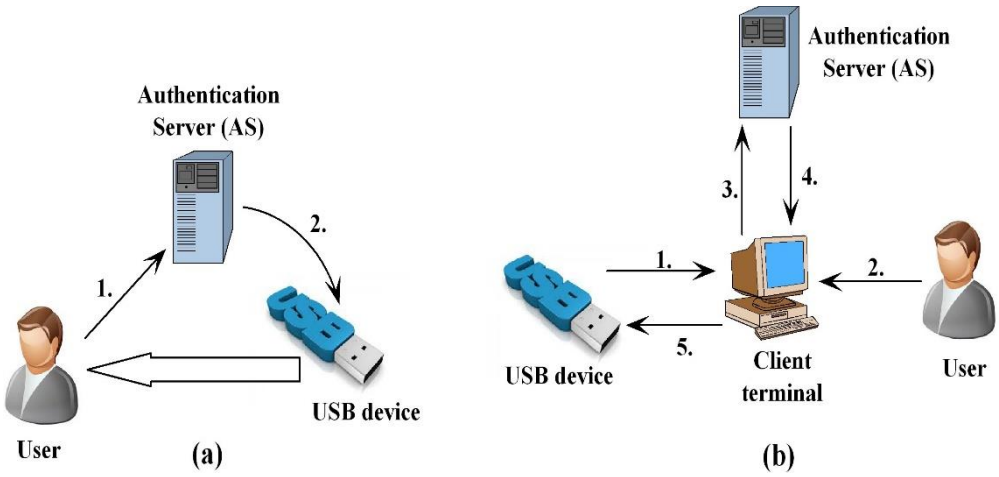

Fig. 1. System architecture of the proposed scheme
(a) Registration procedure (b) File encryption/decryption and authentication procedure.

IV. PROPOSED SCHEME FROM THIS RESEARCH

The proposed scheme consists of (1) registration phase, (2) authentication with file encryption and decryption phase in current session, (3) authentication with file decryption phase in different session and (4) password update phase.

1) *AS* selects a cryptographic one-way hash function $h(\cdot)$. This process can be symbolized as $h:\{0,1\}^* \rightarrow \{0,1\}^l$, where l is a fixed length (128 bits) integer.
2) *AS* also selects a symmetric key encryption /decryption algorithm *ENC/DEC* and a secret key x. *AS* then publishes $\langle ENC/DEC, h(\cdot)\rangle$ as the public parameters while keeping x secret.

A. Registration Phase

In the registration phase, U_i and *AS* perform the following steps to obtain authorized information from the USB MSD:

1) U_i inputs their biometric parameter B_i through a suitable biometric device (e.g. fingerprint reader) to calculate $(\psi_i, \theta_i) \leftarrow GEN(B_i)$. U_i also provides their password pw_i and identity ID_i. U_i then computes $pwr_i = h(pw_i \| \psi_i)$ and submits $\langle ID_i, pwr_i \rangle$ to *AS*.
2) After receiving $\langle ID_i, pwr_i \rangle$, *AS* computes $A_i = h(ID_i \| x) \oplus pwr_i$ and $C_i = h(h(ID_i \| x) \| pwr_i)$. *AS* then stores $\langle A_i, C_i, des(\cdot) \rangle$ in U_i's USB MSD and delivers it to U_i securely.
3) After getting the authorized information for the USB MSD, U_i computes $K_i = \theta_i \oplus h(ID_i \| pw_i)$, $G_i = B_i \oplus h(ID_i \| pw_i)$ and stores $\langle K_i, G_i \rangle$ into the USB MSD. Finally, the USB MSD contains the parameters $\langle A_i, C_i, des(\cdot), K_i, G_i \rangle$.

B. Authentication with File Encryption and Decryption Phase in the Current Session

When U_i wants to store a new file (larger than *r* bits) in the USB MSD in an encrypted format, the following steps are executed between U_i and *AS*:

1) U_i inserts their USB MSD into the client USB port and inputs their password pw_i, identity ID_i and biometric parameter B_i. The MSD then computes $B_i = G_i \oplus h(ID_i \| pw_i)$ and then checks condition $des(B_i, B_i^{'}) \le d$. If false, the device rejects U_i; otherwise the device computes $\theta_i = K_i \oplus h(ID_i \| pw_i)$, $\psi_i = REP(B_i, \theta_i)$ and $L_i \left(= h(ID_i \| x)\right) = A_i \oplus h(pw_i \| \psi_i)$. Then the device checks $C_i = h(L_i \| h(pw_i \| \psi_i))$. If the equality fails then the device rejects U_i; otherwise the device executes the next step.
2) The USB MSD generates random number α_i, and computes $Q_i = ENC_{h(ID_i \| \psi_i)}[File_{r\,bits}] \oplus \alpha_i$, $Z_i = L_i \oplus \alpha_i$ and $W_i = h(L_i \| ID_i \| \alpha_i \| F_i \| T_i)$, where *r*, T_i and F_i are the first *r* number of bits of the file, the current timestamp of U_i and index or identity of the file respectively. Finally, message $MSG1 = \langle ID_i, F_i, Q_i, Z_i, W_i, T_i \rangle$ is sent to *AS*.
3) After receiving $MSG1$ at timestamp T_s, *AS* checks the format of ID_i and condition $(T_s - T_i) \le \Delta T$. If any are invalid *AS* rejects the message; otherwise *AS* computes $L_i^{'} = h(ID_i \| x)$ and $\alpha_i^{'} = L_i^{'} \oplus Z_i$. Then *AS* checks $W_i = h(L_i^{'} \| ID_i \| \alpha_i^{'} \| F_i \| T_i)$. If the equality does not hold, *AS* rejects the message; otherwise *AS* generates random number β_i and computes $X_i \left(= ENC_{h(ID_i \| \psi_i)}[file_{r\,bits}]\right) = Q_i \oplus \alpha_i^{'}$, $Y_i = \beta_i \oplus L_i^{'}$, $SK_i = h(\alpha_i^{'} \| \beta_i \| L_i^{'} \| T_s)$, $H_i = h(x \| F_i) \oplus SK_i$, $E_i = ENC_x[X_i \| F_i] \oplus SK_i$ and $D_i = h(ID_i \| \beta_i \| \alpha_i^{'} \| T_s \| SK_i \| h(x \| F_i) \| ENC_x[X_i \| F_i])$. *AS* then sends the message $MSG2 = \langle E_i, D_i, Y_i, T_s, H_i \rangle$ to U_i.
4) After receiving $MSG2$ at timestamp T_j, U_i checks condition $(T_j - T_s) \le \Delta T$. If invalid U_i terminates the session; otherwise U_i computes $\beta_i^{'} = L_i \oplus Y_i$, $SK_i^{'} = h(\alpha_i \| \beta_i^{'} \| L_i \| T_s)$, $[h(x \| F_i)]^{'} = H_i \oplus SK_i^{'}$ and $[Enc_x[X_i \| F_i]]^{'} = E_i \oplus SK_i^{'}$. Then U_i checks $D_i = h(ID_i \| \beta_i^{'} \| \alpha_i \| T_s \| SK_i^{'} \| [h(x \| F_i)]^{'} \| [Enc_x[X_i \| F_i]]^{'})$. If the equality fails then U_i terminates the session; otherwise U_i selects a suitable file name F_{name} for the file and stores the encrypted file as $\langle Enc_x[X_i \| F_i], F_i, F_{name}, Enc_{h(x \| F_i)}[File_{(n-r)\,bits}] \rangle$ into the memory of the USB MSD, where (*n*-*r*) bits are the remaining bits of the file. Note that if U_i maintains a database for

the file plus key combinations then U_i has to also communicate with *AS* to extract the first *r* bits of the file. If *r* or (*n-r*) bits are larger than key size *k* then U_i breaks *r* or (*n-r*) bits into blocks of size equal to the key size *k* (here 128 bits) and encrypt each block with the counter (CTR) mode [10] of operation to produce the corresponding ciphertext. Plaintext can be retrieved by the same procedure reversely.

When U_i wants to decrypt that encrypted file from the USB MSD in the current session, the following steps are executed between U_i and *AS:*

5) U_i sends message $\left\langle Enc_{SK_i^{'}}\left[Enc_x\left[X_i \| F_i\right]\right], F_i, ID_i, T_i^d\right\rangle$ to *AS*, where T_i^d is the current timestamp.
6) After receiving the message at timestamp T_s^d, *AS* checks the format of ID_i and $\left(T_s^d - T_i^d\right) \le \Delta T$. If either is invalid *AS* rejects the message; otherwise *AS* computes $\left(X_i^{'} \| F_i^{'}\right) = Dec_x\left[Dec_{SK_i}\left[Enc_{SK_i^{'}}\left[Enc_x\left[X_i \| F_i\right]\right]\right]\right]$ and checks $F_i = F_i^{'}$. If equal *AS* sends $\left\langle Enc_{SK_i}\left[X_i^{'}\right], H_i\left(=h\left(x \| F_i\right) \oplus SK_i\right), T_s^d\right\rangle$ to U_i.
7) After receiving $\left\langle Enc_{SK_i}\left[X_i^{'}\right], H_i\left(=h\left(x \| F_i\right) \oplus SK_i\right), T_s^d\right\rangle$ at timestamp T_j^d, U_i checks $\left(T_s^d - T_j^d\right) \le \Delta T$. If false U_i rejects the message, otherwise U_i obtains the plaintext of the file by decoding file by computing

$$Dec\_File = \begin{bmatrix} Dec_{h(ID_i \| \psi_i)}\left[Dec_{SK_i^{'}}\left[Enc_{SK_i}\left[X_i\right]\right]\right] \\ \| \, Dec_{h(x \| F_i)}\left[Enc_{h(x \| F_i)}\left[File_{(n-r)bits}\right]\right] \end{bmatrix}$$

C. Authentication with File Decryption Phase in a Different Session

When U_i wants to decrypt the stored encrypted file from the USB MSD in a different session, the following steps are executed between U_i and *AS*:

1) After checking the provided password pw_i, identity ID_i and biometric parameter $B_i^{'}$ of U_i as described in IV.B, the USB MSD generates random number δ_i and then computes $\hat{W}_i = h(L_i \| ID_i \| \delta_i \| F_i \| \hat{T}_i)$, $\hat{Z}_i = L_i \oplus \delta_i$ and $\hat{Q}_i = ENC_x\left[X_i \| F_i\right] \oplus \delta_i$, where F_i and $\hat{T}_i$ are the indexes of the file and the current timestamp of U_i respectively. Message $MSG3 = \left\langle ID_i, F_i, \hat{Q}_i, \hat{Z}_i, \hat{W}_i, \hat{T}_i\right\rangle$ is sent to *AS*.
2) After receiving $MSG3$ at timestamp $\hat{T}_s$, *AS* checks the format of ID_i and $\left(\hat{T}_s - \hat{T}_i\right) \le \Delta T$. If either are invalid *AS* rejects $MSG3$; otherwise *AS* computes $L_i^{'} = h\left(ID_i \| x\right)$ and $\delta_i^{'} = L_i^{'} \oplus \hat{Z}_i$. Then *AS* checks $\hat{W}_i = h\left(L_i^{'} \| ID_i \| \delta_i^{'} \| F_i \| \hat{T}_i\right)$. If the equality fails *AS* rejects $MSG3$; otherwise *AS* generates a random number γ_i and computes $\left(X_i^{'} \| F_i^{'}\right) = DEC_x\left[\hat{Q}_i \oplus L_i^{'}\right]$. Then *AS* checks $F_i = F_i^{'}$. If true, *AS* further computes $\hat{Y}_i = \gamma_i \oplus L_i^{'}$, $S\hat{K}_i = h\left(\delta_i^{'} \| \gamma_i \| L_i^{'} \| \hat{T}_s\right)$, $H_i = h\left(x \| F_i\right) \oplus S\hat{K}_i$, $\hat{E}_i = X_i^{'} \oplus S\hat{K}_i$ and $\hat{D}_i = h\left(ID_i \| \gamma_i \| \delta_i^{'} \| \hat{T}_s \| S\hat{K}_i \| h\left(x \| F_i\right) \| X_i^{'}\right)$. *AS* then sends message $MSG4 = \left\langle \hat{E}_i, \hat{D}_i, \hat{Y}_i, H_i, \hat{T}_s\right\rangle$ to U_i.
3) After receiving $MSG4$ at timestamp $\hat{T}_j$, U_i checks $\left(\hat{T}_j - \hat{T}_s\right) \le \Delta T$. If false, U_i rejects message $MSG4$; otherwise U_i computes $\gamma_i^{'} = \hat{Y}_i \oplus L_i$, $S\hat{K}_i^{'} = h\left(\delta_i \| \gamma_i^{'} \| L_i \| \hat{T}_s\right)$, $\left[h\left(x \| F_i\right)\right]^{'} = H_i \oplus S\hat{K}_i^{'}$, $\hat{X}_i^{'} = \hat{E}_i \oplus S\hat{K}_i^{'}$ and checks equality $\hat{D}_i = h\left(ID_i \| \gamma_i^{'} \| \delta_i \| \hat{T}_s \| S\hat{K}_i^{'} \| \left[h(x \| F_i)\right]^{'} \| \hat{X}_i^{'}\right)$. If it fails U_i rejects message $MSG4$;

otherwise U_i decrypts $\hat{X}_i^{'}$ by using the key $h(ID_i \| \psi_i)$ to recover the plaintext of the file as $File_{r\,bits} = DEC_{h(ID_i \| \psi_i)}\left[\hat{X}_i^{'}\right]$ and also decrypts $ENC_{h(x\|F_i)}\left[File_{(n-r)bits}\right]$ by using the key $\left[h(x \| F_i)\right]^{'}$ as $File_{(n-r)bits} = Dec_{h(x\| F_i)}\left[Enc_{h(x\| F_i)}\left[File_{(n-r)bits}\right]\right]$, where $File = \left(File_{r\,bits} \| File_{(n-r)bits}\right)$.

D. Password Update Phase

When a user U_i wants to change their password, the password update phase is invoked.

1) U_i inserts their USB MSD into the client USB port and inputs their current password pw_i, identity ID_i, biometric parameter $B_i^{'}$ and their new password $pw_i^{[new]}$. The USB MSD computes $B_i = G_i \oplus h(ID_i \| pw_i)$ and checks $des\left(B_i, B_i^{'}\right) \le d$. If it fails the MSD rejects U_i; otherwise the device computes $\theta_i = K_i \oplus h(ID_i \| pw_i)$, $\psi_i = Rep(B_i, \theta_i)$ and $L_i \left(= h(ID_i \| x)\right) = A_i \oplus h(pw_i \| \psi_i)$. Then it checks $C_i = h\left(L_i \| h(pw_i \| \psi_i)\right)$. If the equality fails then the device rejects U_i; otherwise the USB MSD executes the next step.
2) Then the USB MSD computes $A_i^{[new]} = L_i \oplus h\left(pw_i^{[new]} \| \psi_i\right)$, $C_i^{[new]} = h\left(L_i \| h\left(pw_i^{[new]} \| \psi_i\right)\right)$, $G_i^{[new]} = B_i \oplus h\left(ID_i \| pw_i^{[new]}\right)$ and $K_i^{[new]} = \theta_i \oplus h\left(ID_i \| pw_i^{[new]}\right)$. The MSD then replaces $\langle A_i, C_i, G_i, K_i \rangle$ with $\left\langle A_i^{[new]}, C_i^{[new]}, G_i^{[new]}, K_i^{[new]} \right\rangle$ respectively.

V. SECURITY ANALYSIS OF 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 maintains a tuple $\langle y, m \rangle$ such that $y = h(m)$. It returns m from y upon receiving a query (qH, y) if $\langle y, m \rangle$ is present in the tuple; otherwise returns a random number r_1. Then it stores a new entry $\langle y, r_1 \rangle$ into its tuple.
- $\mathcal{O}racle\mathcal{FE}$ is a random oracle which contains two parts:
  1. $\mathcal{O}racleFE_{\text{GEN}}$ unconditionally outputs the pair (ψ,θ) from the corresponding tuple $\langle b, \theta, \psi \rangle$ upon receiving a query ($qGEN$,b) such that $(\psi, \theta) \leftarrow GEN(b)$ if $\langle b, \theta, \psi \rangle$ is present in its tuple; otherwise returns two random numbers r_2 and r_3. Then it stores new entry $\langle b, r_2, r_3 \rangle$ into its tuple.
  2. $\mathcal{O}racle\mathcal{FE}_{\text{REP}}$ unconditionally outputs ψ from the corresponding tuple $\left\langle b^{'}, \theta, \psi \right\rangle$ upon receiving a query ($qREP$,$b^{'}$,θ) such that $\psi \leftarrow REP\left(b^{'}, \theta\right)$ if $\left\langle b^{'}, \theta, \psi \right\rangle$ is present in its tuple; otherwise returns random number r_4. Then it stores new entry $\left\langle b^{'}, \theta, r_4 \right\rangle$ into its tuple.

- $\mathcal{O}racle\mathcal{SK}$ is a random oracle which contains two parts:
  1. $\mathcal{O}racleSK_{\text{ENC}}$ unconditionally outputs the ciphertext C from its tuple $\langle M, C\rangle$ upon receiving a query (*qENC,M)* such that $C = ENC_k[M]$ if $\langle M, C\rangle$ is present in its tuple; otherwise returns random number r_5 from the ciphertext space. It then stores a new entry $\langle M, r_5\rangle$ in its tuple.
  2. $\mathcal{O}racleSK_{DEC}$ unconditionally outputs a key k and plaintext M from its tuple $\langle C, k, M\rangle$ upon receiving a query (*qDEC,C*) such that $M = ENC_k[C]$ if $\langle C, k, M\rangle$ is present in its tuple; otherwise returns two random numbers r_6 and r_7 from key space. Then it stores new entry $\langle C, r_6, r_7\rangle$ in its tuple.

Theorem 1: Under the assumption that $h(\cdot)$ and *ENC/DEC* act as random oracles, the proposed scheme derived from this Three-factor Security Protocol (TSP) work is then provably secure against an adversary $\mathcal{A}$ for deriving the secret key x of an authentication server *AS* after obtaining the stored information into the memory of the MSD, and capturing the login message and the reply messages of the authentication plus file encryption plus file retrieval phase during communication between U_i and *AS* in the current session as well as in a different session.

Proof 1: Assume that $\mathcal{A}$ has the ability to derive the secret key x of *AS*, and the MSD of U_i is lost or stolen. Thus, $\mathcal{A}$ can extract the stored parameters $\langle A_i, C_i, K_i, G_i\rangle$ from the memory of the MSD of U_i by power monitoring [11], [12]. $\mathcal{A}$ also traps the login message $MSG1$, the reply messages $MSG2$, $\left\langle Enc_{SK_i'}[Enc_x[X_i \| F_i]], F_i, ID_i, T_i^d\right\rangle$ and $\left\langle Enc_{SK_i}[X_i'], T_s^d\right\rangle$ of the authentication plus data retrieval phase at timestamp T_i, T_s, T_i^d and $T_s{}^d$ respectively. $\mathcal{A}$ runs the algorithm derived from this work (TSP), $EXP1_{A,TSP}^{Oracle}$ to derive the secret key x of *AS* as given in Algorithm 1. Define the success probability of $EXP1_{A,TSP}^{Oracle}$ as:

$$Succ1_{A,TSP}^{Oracle} = \Pr\left[EXP1_{A,TSP}^{Oracle} = 1\right] \tag{4}$$

then the advantage is given by:

$$Adv1_{A,TSP}^{Oracle}(t, qH, qDEC) = \max_A\left\{Succ1_{A,TSP}^{Oracle}\right\} \tag{5}$$

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 $qDEC$ made by $\mathcal{O}racleSK_{\text{DEC}}$.

The proposed scheme is said to be provably secure against $\mathcal{A}$ deriving the secret key x of *AS* if $Adv1_{A,TSP}^{Oracle}(t, qH, qDEC) < \xi$, for any small $\xi > 0$. According to $EXP1_{A,TSP}^{Oracle}$, if $\mathcal{A}$ is successful in computing the inversion of $h(\cdot)$ as well as extracting the correct secret key, then $\mathcal{A}$ can successfully derive the secret key x of *AS* by using the $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racleSK_{\text{DEC}}$ random oracles respectively. But, according to Definition 1 and 3, $Adv1_{A,TSP}^{Oracle}(t, qH, qDEC) \le \xi_1$, for any small $\xi_1 > 0$. Since the advantage $Adv1_{A,TSP}^{Oracle}(t, qH, qDEC) < \xi$, for any small $\xi > 0$ because the proposed scheme depends on $Adv1_A^H(t)$ and $Adv1_A^{DEC}(t)$. Thus, this proposed scheme is secure against $\mathcal{A}$ for deriving the secret key x of *S*.

Algorithm 1 $EXP1_{A,TSP}^{Oracle}$

Input : $A_i, C_i, ID_i, F_i, W_i, Z_i, T_i, E_i, Y_i, H_i, D_i, T_s, Enc_{SK_i}\left[Enc_x\left[X_i \| F_i\right]\right], Enc_{SK_i}\left[X_i^{'}\right]$

Output : 0 *or* 1

1: Calls $\mathcal{O}racle\mathcal{H}$ on the input C_i to retrieve the information $h(ID_i \| x)$ and pwr_i $(= h(pw_i \| \psi_i))$ as $([h(ID_i \| x)]^* \| pwr_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(C_i)$

2: Computes $[h(ID_i \| x)]^{**} = A_i \oplus pwr_i^*$

3: Calls $\mathcal{O}racle\mathcal{H}$ on the input W_i to retrieve the information $L_i\ (= h(ID_i \| x))$, ID_i, α_i, F_i and T_i as $(L_i^* \| ID_i^* \| \alpha_i^* \| F_i^* \| T_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(W_i)$

4: Calls $\mathcal{O}racle\mathcal{H}$ on the input D_i to retrieve the information ID_i, β_i, α_i, T_s, SK_i , $h(x \| F_i)$ and $Enc_x[X_i \| F_i]$ as $(ID_i^{**} \| \beta_i^* \| \alpha_i^{**} \| T_s^* \| SK_i^* \| [h(x \| F_i)]^* \| [Enc_x[X_i \| F_i]]^*) \leftarrow \mathcal{O}racle\mathcal{H}(D_i)$

5: Calls $\mathcal{O}racle\mathcal{SK}_{\text{DEC}}$ on the input $Enc_{SK_i}[Enc_x[X_i \| F_i]]$ to retrieve the information SK_i and $Enc_x[X_i \| F_i]$ as $(SK_i^{**}, [Enc_x[X_i \| F_i]]^{**}) \leftarrow \mathcal{O}racle\mathcal{SK}_{\text{DEC}}(Enc_{SK_i}[Enc_x[X_i \| F_i]])$

6: Calls $\mathcal{O}racle\mathcal{SK}_{\text{DEC}}$ on the input $[Enc_x[X_i \| F_i]]^{**}$ to retrieve the information x and $(X_i \| F_i)$ as $(x^*, (X_i^* \| F_i^{**})) \leftarrow \mathcal{O}racle\mathcal{SK}_{\text{DEC}}([Enc_x[X_i \| F_i]]^{**})$

7: Calls $\mathcal{O}racle\mathcal{SK}_{\text{DEC}}$ on the input $Enc_{SK_i}[X_i]$ to retrieve the information SK_i and X_i as $(SK_i^{***}, X_i^{**}) \leftarrow \mathcal{O}racle\mathcal{SK}_{\text{DEC}}(Enc_{SK_i}[X_i])$

8: **if** $(T_i^* == T_i)$ && $(T_s^* == T_s)$ && $(\alpha_i^* == \alpha_i^{**})$ && $(ID_i^{**} == ID_i^* == ID_i)$ && $(F_i^* == F_i == F_i^{**})$ && $(SK_i^* == SK_i^{***} == SK_i^{****})$ && $(X_i^* == X_i^{**})$ **then**

9: Computes $L_i^{**} = Z_i \oplus \alpha_i^*$, $[Enc_x[X_i \| F_i]]^{***} = E_i \oplus SK_i^*$, $L_i^{***} = Y_i \oplus \beta^*$ and $[h(x \| F_i)]^{**} = H_i \oplus SK_i^*$

7. **else**

8: **Return** 0 (failure)

9: **if** $([h(x \| F_i)]^* == [h(x \| F_i)]^{**})$ && $([Enc_x[X_i \| F_i]]^{***} == [Enc_x[X_i \| F_i]]^* == [Enc_x[X_i \| F_i]]^{**})$ **then**

10: Calls $\mathcal{O}racle\mathcal{H}$ on the input $[h(x \| F_i)]^*$ to retrieve the information F_i and x as $(x^{**} \| F_i^{***}) \leftarrow \mathcal{O}racle\mathcal{H}([h(x \| F_i)]^*)$

11: Calls $\mathcal{O}racle\mathcal{SK}_{\text{DEC}}$ on the input $[Enc_x[X_i \| F_i]]^*$ to retrieve the information X_i, F_i and x as $(x^{***}, (X_i^{***} \| F_i^{****})) \leftarrow \mathcal{O}racle\mathcal{SK}_{\text{DEC}}[Enc_x[X_i \| F_i]]^*)$

12: **else**

13: **Return** 0 (failure)

14: **if** $(L_i^* == L_i^{**} == L_i^{***} == [h(ID_i \| x)]^* == [h(ID_i \| x)]^{**})$ **then**

15: Calls $\mathcal{O}racle\mathcal{H}$ on the input L_i^* to retrieve the information ID_i and x as $(ID_i^{***} \| x^{****}) \leftarrow \mathcal{O}racle\mathcal{H}(L_i^*)$

16: **else**

17: **Return** 0 (failure)

18: **if** $(ID_i^{***} == ID_i)$ && $(F_i^{***} == F_i == F_i^{****})$ && $(X_i^* == X_i^{****})$ && $(x^* == x^{**} == x^{***} == x^{****})$ **then**

19: **Return** 1 (success)

20: **else**

21: **Return** 0 (failure)

Theorem 2: Under the assumption that $h(\cdot)$ and *FE* act as random oracles, the proposed Three-factor Security Protocol (TSP) scheme derived from this work 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 plus file encryption plus file retrieval phase during communication between U_i and *AS* in the current session as well as in any other session.

Proof 2: Apply the same assumptions as described in Theorem 1. There is no chance to derive password pw_i of U_i from communication messages because they are independent of the password. Then $\mathcal{A}$ runs the algorithm derived from this work (TSP), $EXP2_{A,TSP}^{Oracle}$ to derive the password pw_i of U_i as given in Algorithm 2. Define the success probability of $EXP2_{A,TSP}^{Oracle}$ as:

$$Succ2_{A,TSP}^{Oracle} = \Pr\left[EXP2_{A,TSP}^{Oracle} = 1 \right] \tag{6}$$

then the advantage is given by:

$$Adv2_{A,TSP}^{Oracle}(t, qH, qFE) = \max_A \left\{ Succ2_{A,TSP}^{Oracle} \right\} \tag{7}$$

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 by $\mathcal{O}racle\mathcal{FE}_{\text{REP}}$. Again, the proposed scheme is said to be provably secure against $\mathcal{A}$ deriving the password pw_i of U_i if $Adv2_{A,TSP}^{Oracle}(t, qH, qFE) < \xi$, for any small $\xi > 0$. According to $EXP2_{A,TSP}^{Oracle}$, if $\mathcal{A}$ is successful in computing the inversion of $h(\cdot)$ as well as extract the correct derived biometric parameter ψ_i, they can successfully derive the password pw_i of U_i by using of the $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racle\mathcal{FE}_{\text{REP}}$ random oracles respectively. But, according to Definition 1 and 2, $Adv2_{A,TSP}^{Oracle}(t, qH, qFE) \le \xi_1$, for any small $\xi_1 > 0$. Since, the advantage $Adv2_{A,TSP}^{Oracle}(t, qH, qFE) \le \xi$, for any small $\xi > 0$ because the proposed scheme depends on $Adv2_A^H(t)$ and $Adv1_A^{FE}(t)$. Thus, this proposed scheme is secure against $\mathcal{A}$ for deriving the password pw_i of U_i.

Algorithm 2 $EXP2_{A,TSP}^{Oracle}$

Input : K_i, G_i, A_i, C_i, ID_i

Output : 0 *or* 1

1: Calls $\mathcal{O}racle\mathcal{H}$ on the input C_i to retrieve the information $h(ID_i \| x)$ and pwr_i (= $h(pw_i \| \psi_i)$) as $([h(ID_i \| x)]^* \| pwr_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(C_i)$

2: Computes $pwr_i^{**} = A_i \oplus [h(ID_i \| x)]^*$

3: **Repeat**

4: Chooses a password pw_i^*

5: Computes $B_i^* = G_i \oplus h(ID_i \| pw_i^*)$ and $\theta_i^* = K_i \oplus h(ID_i \| pw_i^*)$

6: Calls $\mathcal{O}racle\mathcal{FE}_{\text{REP}}$ on the input (B_i^*, θ_i^*) to retrieve the information ψ_i as $(\psi_i^*) \leftarrow \mathcal{O}racle\mathcal{FE}_{\text{REP}}(B_i^*, \theta_i^*)$

7: Computes $pwr_i^{***} = h(pw_i^* \| \psi_i^*)$

8: **Until** $(pwr_i^{***} == pwr_i^{**})$

9: **if** $(pwr_i^{***} == pwr_i^{**})$ **then**

10: Accepts pw_i^* as correctly guessed password

11: **Return** 1 (success)

12: **else**

13: **Return** 0 (failure)

Theorem 3: Under the assumption that $h(\cdot)$ and *ENC/DEC* act as random oracles, then the Three-factor Security Protocol (TSP) scheme derived from this work is provably secure against $\mathcal{A}$ deriving the shared secret session key *SK* between U_i and *AS* after getting the stored information into the memory of the MSD device, and trapping the login message and reply message of authentication plus file encryption plus file retrieval phase during communication between U_i and *AS* in the current session as well as any other session.

Proof 3: Apply the same assumptions as described in Theorem 1. Then $\mathcal{A}$ runs the algorithm derived from this work (TSP), $EXP3_{A,TSP}^{Oracle}$ to derive the session key SK_i between U_i and *AS* as given in Algorithm 3. Define the success probability of $EXP3_{A,TSP}^{Oracle}$ as:

$$Succ3_{A,TSP}^{Oracle} = \Pr\left[EXP3_{A,TSP}^{Oracle} = 1\right] \tag{8}$$

then the advantage is given by:

$$Adv3_{A,TSP}^{Oracle}(t, qH, qDEC) = \max_A \left\{ Succ3_{A,TSP}^{Oracle} \right\} \tag{9}$$

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 *qDEC* made by $\mathcal{O}racleSK_{\text{DEC}}$. Again, the proposed scheme is said to be provably secure against $\mathcal{A}$ deriving the session key SK_i between U_i and *AS* if $Adv3_{A,TSP}^{Oracle}(t, qH, qDEC) \le \xi$, for any small $\xi > 0$. According to $EXP3_{A,TSP}^{Oracle}$, if $\mathcal{A}$ is successful in computing the inversion of $h(\cdot)$ and extracting the correct key, they can successfully derive the shared secret session key SK_i between U_i and *AS* by using of the $\mathcal{O}racle\mathcal{H}$ and $\mathcal{O}racleSK_{\text{DEC}}$ random oracles respectively. But, according to Definition 1 and 3, $Adv3_{A,TSP}^{Oracle}(t, qH, qDEC) \le \xi_1$, for any small $\xi_1 > 0$. Since, the advantage $Adv3_{A,TSP}^{Oracle}(t, qH, qDEC) \le \xi$, for any small $\xi > 0$ because the proposed scheme depends on $Adv3_A^H(t)$ and $Adv3_A^{DEC}(t)$. Thus, the proposed scheme is secure against $\mathcal{A}$ deriving the shared secret session key SK_i between U_i and *AS*.

Algorithm 3 $EXP3_{A,TSP}^{Oracle}$

Input : $A_i, C_i, ID_i, F_i, W_i, Z_i, T_i, E_i, Y_i, H_i, D_i, T_s, Enc_{SK_i}\left[Enc_x\left[X_i \| F_i\right]\right], Enc_{SK_i}\left[X_i^{'}\right]$

Output : 0 *or* 1

1: Calls $\mathcal{O}racle\mathcal{H}$ on the input C_i to retrieve the information $h(ID_i \| x)$ and pwr_i $(= h(pw_i \| \psi_i))$ as $([h(ID_i \| x)]^* \| pwr_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(C_i)$

2: Computes $[h(ID_i \| x)]^{**} = A_i \oplus pwr_i^*$

3: Calls $\mathcal{O}racle\mathcal{H}$ on the input W_i to retrieve the information $L_i (= h(ID_i \| x))$, ID_i, α_i, F_i and T_i as $(L_i^* \| ID_i^* \| \alpha_i^* \| F_i^* \| T_i^*) \leftarrow \mathcal{O}racle\mathcal{H}(W_i)$

4: Calls $\mathcal{O}racle\mathcal{H}$ on the input D_i to retrieve the information ID_i, β_i, α_i, T_s, SK_i , $h(x \| F_i)$ and $Enc_x[X_i \| F_i]$ as $(ID_i^{**} \| \beta_i^* \| \alpha_i^{**} \| T_s^* \| SK_i^* \| [h(x \| F_i)]^* \| [Enc_x[X_i \| F_i]]^*) \leftarrow \mathcal{O}racle\mathcal{H}(D_i)$

5: Calls $\mathcal{O}racle\mathcal{SK}_{\text{DEC}}$ on the input $Enc_{SK_i}[Enc_x[X_i \| F_i]]$ to retrieve the information SK_i and $Enc_x[X_i \| F_i]$ as $(SK_i^{**}, [Enc_x[X_i \| F_i]]^{**}) \leftarrow \mathcal{O}racle\mathcal{SK}_{\text{DEC}}(Enc_{SK_i}[Enc_x[X_i \| F_i]])$

6: Calls $\mathcal{O}racle\mathcal{SK}_{\text{DEC}}$ on the input $Enc_{SK_i}[X_i]$ to retrieve the information SK_i and X_i as $(SK_i^{***}, X_i^{**}) \leftarrow \mathcal{O}racle\mathcal{SK}_{\text{DEC}}(Enc_{SK_i}[X_i])$

7: **if** $(T_i^* == T_i)$ && $(T_s^* == T_s)$ && $(\alpha_i^* == \alpha_i^{**})$ && $(ID_i^{**} == ID_i^* == ID_i)$ && $(F_i^* == F_i)$ && $([Enc_x[X_i \| F_i]]^* == [Enc_x[X_i \| F_i]]^{**})$ **then**

8: Computes $L_i^{**} = Z_i \oplus \alpha_i^*$, $SK_i^{****} = E_i \oplus [Enc_x[X_i \| F_i]]^*$, $L_i^{***} = Y_i \oplus \beta^*$ and $SK_i^{*****} = H_i \oplus [h(x \| F_i)]^*$

9. **else**

10: **Return** 0 (failure)

11: **if** $(L_i^* == L_i^{**} == L_i^{***} == [h(ID_i \| x)]^* == [h(ID_i \| x)]^{**})$ **then**

12: Computes $SK_i^{******} = h(\alpha_i^* \| \beta_i^* \| L_i^* \| T_s)$

13: **else**

14: **Return** 0 (failure)

15: **if** $(SK_i^* == SK_i^{**} == SK_i^{***} == SK_i^{****} == SK_i^{*****} == SK_i^{******})$ **then**

16: **Return** 1 (success)

17: **else**

18: **Return** 0 (failure)

A. Discussion of 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 α, β (for current session), and δ, γ (for a different session) then $\mathcal{A}$ cannot compute the session key *SK*. In the proposed scheme, all communicating messages depend on random numbers and the timestamp. 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 *AS* and random numbers generated by U_i and *AS* respectively.

Theorems 1 and 2 show that the secret information of the authentication server and the user are secure from $\mathcal{A}$. Thus, it is infeasible to mount a *forgery attack* on this proposed scheme.

VI. PERFORMANCE EVALUATION OF PROPOSED SCHEME

This section compares the performance of the proposed scheme with related schemes in the literature [1]-[3], [5], [6]. The login and authentication phases for both file encryption and file decryption of the proposed scheme have been compared with the related existing schemes in the literature [1]-[3], [5], [6] because these phases are commonly used.

Table II presents the communication (overhead) and storage costs of this work compared to the literature. It can be seen that the communication cost of this work is the same for data encryption in the literature, but has a significant advantage in the data decryption within the current session as was the objective of this work. The storage cost of this work is also 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	Communication Cost (bits)				Storage Cost (bits)
	File encryption		Authentication + File retrieval		
	Login	Authentication	(Different session) Login + Authentication	(Current session) Login + Authentication	
Ku and Chen [1]	320	256	-	-	384
Yoon, Ryu and Yoo [2]	320	256	-	-	512
Yang, Wu and Chiu [3]	3328	1280	4608 = (3328 + 1280)	4608 = (3328 + 1280)	2176
Lee *et al.* [5]	512	512	1024 = (512 + 512)	1024 = (512 + 512)	384
He *et al.*[6]	512	640	1152 = (512 + 640)	1152 = (512 + 640)	384
Proposed scheme	640	768	1408 = (640 + 768)	768 = (384 + 384)	640

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

TABLE III
COMPARISON OF ATTACK VULNERABILITY OF SCHEMES IN THE CURRENT LITERATURE COMPARED TO THIS PROPOSED SCHEME

Attack Vulnerability / Feature	Ku and Chen [1]	Yoon, Ryu and Yoo [2]	Yang, Wu and Chiu [3]	Lee *et al.* [5]	He *et al.* [6]	Proposed scheme
User impersonation attack	yes [2] [14]	yes [14]	no	no	no	no
Server masquerading attack	yes [14]	yes [14]	yes [15]	no	no	no
Insider attack	no	no	no	no	no	no
Off-line password guessing attack	yes [14]	yes [14]	no	yes [6]	no	no
Inefficient login phase	yes [14]	yes [14]	yes	no	no	no
Denial of service (DoS) attack	yes [14]	yes [14]	no	yes [6]	no	no
Password change phase attack	yes [2] [14]	yes [14]	no	-	-	no
Replay attack	no	no	yes	yes [6]	yes	no
Session key agreement	no	no	yes	yes	yes	yes
Mutual authentication	no	no	yes	no	yes	yes

Table IV presents the computational cost of this work compared to the literature. T_h is the time required for the hashing operation, T_{Pm} for point multiplication operation, T_m for scalar multiplication operation, T_{div} for division operation and T_{En}/T_{De} for the symmetric key encryption/decryption operation. Typically, the time complexity associated with these operations can be expressed as $T_{En}/T_{De} = T_h > T_{Pm} > T_m \approx T_{div}$ [13]. It can be seen that this work significantly reduces the computation cost for both data encryption and data retrieval as well as authentication.

It can be assumed that the identity ID_i, the file index F_i and the password pw_i are length of 64 bits each, cryptographic one-way hash function $h(\cdot)$ and symmetric key encryption/decryption, random number and timestamp returns 128 bits for each block. Since the communication overhead for the login, file encryption and the authentication phase as well as login, file decryption and authentication in different sessions is $(2\times64)+(8\times128)+256=1408$ bits, and login, file decryption and authentication in current session phase is $(2\times64)+(5\times128)=768$ bits. Therefore, this work achieves a much lower communication overhead than the work in the literature [3], [5], [6]. Low communication overhead, low storage cost, low computational cost and resistance of all possible attacks indicate that the proposed scheme provides an efficient security protocol offering a practical solution for enhanced security of mass market USB MSD.

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

	File encryption	**Authentication + File retrieval**					
	Login	**Authentication**		**Different session**		**Current session**	
	(USB MSD)	**(USB MSD)**	**(Server)**	**(USB MSD)**	**(Server)**	**(USB MSD)**	**(Server)**
Ku and Chen [1]	$2T_h$	$1T_h$	$3T_h$	-	-	-	-
Yoon, Ryu and Yoo [2]	$2T_h$	$1T_h$	$3T_h$	-	-	-	-
Yang, Wu and Chiu [3]	$1T_m+1T_h+2T_e$	$2T_h+2T_e+1T_{De}+1T_{En}$	$1T_{en}+1T_{div}+2T_m+3T_e+3T_h$	$1T_m+3T_h+4T_e+2T_{De}$	$1T_{en}+1T_{div}+2T_m+3T_e+3T_h$	$1T_m+3T_h+4T_e+2T_{De}$	$1T_{en}+1T_{div}+2T_m+3T_e+3T_h$
Lee *et al.* [5]	$4T_h+1T_{Pm}$	$2T_h+1T_{Pm}+1T_{En}+1T_{De}$	$4T_h+2T_{Pm}+1T_{En}$	$6T_h+2T_{Pm}+2T_{De}$	$4T_h+2T_{Pm}+1T_{En}$	$6T_h+2T_{Pm}+2T_{De}$	$4T_h+2T_{Pm}+1T_{En}$
He *et al.* [6]	$4T_h+1T_{Pm}$	$3T_h+1T_{Pm}+1T_{En}+1T_{De}$	$5T_h+2T_{Pm}+1T_{En}$	$7T_h+2T_{Pm}+2T_{De}$	$5T_h+2T_{Pm}+1T_{En}$	$7T_h+2T_{Pm}+2T_{De}$	$5T_h+2T_{Pm}+1T_{En}$
Proposed scheme	$5T_h+1T_{En}$	$2T_h$	$4T_h+1T_{En}$	$7T_h+1T_{De}$	$4T_h+1T_{De}$	$1T_{En}+2T_{De}$	$2T_{De}+1T_{En}$

VII. CONCLUSION

An efficient mutual authentication protocol has been presented to encrypt files in a Universal Serial Bus (USB) Mass Storage Device (MSD) enabling secure and usable "USB memory sticks." This paper has contributed a novel concept to the current state of the art in biometric security algorithms by defending against security attacks and improving device usability across different sessions. Moreover, the paper has formally proved that the proposed protocol can withstand relevant security weaknesses. A performance comparison has also been made with the literature to confirm that the proposed scheme achieves a significantly lower computation cost and communication cost than other related schemes. The overall efficiency demonstrates that USB based MSDs with biometric security sensors can be implemented in order to provide significant security and usability for the consumer and beyond.

REFERENCES

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

[2] E.-J. Yoon, E.-K. Ryu, and K.-Y. Yoo, "Further improvement of an efficient password based remote user authentication scheme using smart cards," *IEEE Trans. Consumer Electron.,* vol. 50, no. 2, pp. 612-614, May 2004.

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

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

[5] C. Lee, C. Chen, and P. Wu, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices," *IET Computers & Digital Techniques,* vol. 7, no. 1, pp. 48-55, Jan. 2013.

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

[7] D. Giri, R. S. Sherratt, T. Maitra, and R. Amin, "Efficient Biometric and Password Based Mutual Authentication for Consumer USB Mass Storage Devices," *IEEE Trans. Consumer Electron.,* vol. 61, no. 4, pp. 491-499, Nov. 2015.

[8] 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, LNCS, Vol. 3027, pp 523-540, 2004.

[9] B. A. Forouzan and D. Mukhopadhyay, *Cryptography and Network Security 2/e*. Tata-McGraw Hill, TMH, 2nd edition, 2010.

[10] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering Design Principles and Practical Applications,* John Wiley & Sons, Mar. 2010.

[11] 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.

[12] 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.

[13] 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.

[14] X.-M. Wang, W.-F. Zhang, J.-S. Zhang, and M. K. Khan, "Cryptanalysis and improvement on two efficient remote user authentication scheme using smart card," *Computer Standards & Interfaces,* vol. 29, no. 5, pp. 507-512, Jul. 2007.

[15] M. H. Eldefrawy, M. K. Khan, and H. Elkamchouchi, "The use of two authentication factors to enhance the security of mass storage device," in Proc. 11th Int. Conf. Information Technology: New Generations, Las Vegas, NV, pp. 196-200, Apr. 2014.

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. 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 has appointed as a Lecturer in Electronic Engineering at the University of Reading where he is now Professor of Biosensors. His research topic is signal processing and communications in consumer devices focusing on wearable devices and healthcare. He received the 1st place IEEE Chester Sall Memorial Award in 2006 and 2nd place in 2016. He is a reviewer for the IEEE SENSORS JOURNAL 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.