Enhanced three-factor security protocol for consumer USB mass storage devices


It is advisable to refer to the publisher’s version if you intend to cite from the work.

To link to this article DOI: http://dx.doi.org/10.1109/TCE.2014.6780922

Publisher: IEEE

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the End User Agreement.

www.reading.ac.uk/centaur

CentAUR
Abstract

The Universal Serial Bus (USB) is an extremely popular interface standard for computer peripheral connections and is widely used in consumer Mass Storage Devices (MSDs). While current consumer USB MSDs provide relatively high transmission speed and are convenient to carry, the use of USB MSDs has been prohibited in many commercial and everyday environments primarily due to security concerns. Security protocols have been previously proposed and a recent approach for the USB MSDs is to utilize multi-factor authentication. This paper proposes significant enhancements to the three-factor control protocol that now makes it secure under many types of attacks including the password guessing attack, the denial-of-service attack, and the replay attack. The proposed solution is presented with a rigorous security analysis and practical computational cost analysis to demonstrate the usefulness of this new security protocol for consumer USB MSDs.

Index Terms

Authentication, Consumer Storage, Mass Storage Device, USB.
I. Introduction

The Universal Serial Bus (USB) is a ubiquitous interface standard being widely used for connecting storage to consumer devices [1]. Because of its convenience and ease of connectivity, the USB port has become an essential component of consumer electronics devices such as flash disks, keyboards, cell phones, chargers, speakers, and printers. However, the USB interface has the following three weaknesses when it is used for consumer storage devices [2]: (1) anyone (e.g., an unauthorized user) could read or steal confidential information easily since the information is stored in plaintext format; and (2) an adversary could intercept or attack the transmitted information since the transmit channel between the device and the computer is not secure. Therefore, despite their practicality, USB Mass Storage Devices (MSDs) have been prohibited in an enormous number of environments. To solve these problems, and extend the applications of USB consumer storage devices, an authentication protocol can be implemented to ensure secure communications between the device and the computer.

Ever since Lamport proposed the first authentication protocol [2], many authentication protocols have been proposed for different applications. Hwang and Li [3] proposed an authentication protocol using a smart card. However, their protocol could not withstand the masquerade attack. To improve security, Ku and Chen [4] proposed an improved authentication protocol using a smart card. Later, Yoon, Ryu and Yoo [5] found that Ku and Chen’s improved authentication protocol was however vulnerable to the parallel session attack, and subsequently proposed a new authentication protocol using a smart card, but Hsiang and Shih [6] later demonstrated that it was vulnerable to three kinds of attacks. Hsiang and Shih proposed their new authentication protocol using a smart card; however, Shim [7] found that Hsiang and Shih’s protocol was vulnerable to the off-line password guessing attack.

Kim and Hong [8] proposed a multimodal biometric authentication protocol that employed teeth, image and voice in mobile environments. To improve performance, Kim, Chung and Hong [9], and Lee, Kim and Cho [10] proposed two new protocols that all used person specific authentication using personal biometric characteristics such as face, teeth, and voice. However, all these protocols are not ideally suitable for USB MSDs because their stored information can easily be read out or require significant local complex computations.

To protect the privacy of a file transferred to a storage device, Yang, Wu and Chiu [11] proposed the first secure control protocol using the Schnorr signature scheme [12]. However, Chen, Qin and Yu [13] indicated that Yang et al.’s protocol [11] was vulnerable to the forge and replay attacks. Besides, Lee, Chen and Wu [14] found that the performance of Yang et al. protocol [11] was computationally heavy due to significant modular exponentiation operations. To solve those problems, Lee et al. [14] proposed a three-factor authentication protocol based on Elliptic Curve Cryptosystem (ECC) that requires the password, smart card, and biometric characteristic for authentication.

Compared with the use of only password, biometric keys have the following advantages [15]:

1) Biometric keys cannot be lost or forgotten;
2) Biometric keys are very difficult to copy or share;
3) Biometric keys are extremely hard to forge or distribute;
4) Biometric keys cannot be guessed easily.

Compared with the traditional public key cryptosystem, the ECC can provide better performance because it can achieve the same security level using a smaller key size. For example, the 160-bit ECC and 1024-bit from the popular Rivest–Shamir–Adleman (RSA) cryptosystem have the same level of security [16]. Therefore, Lee et al. protocol [14] was previously considered to be more suitable for USB consumer
storage devices. However, this paper will demonstrate that the protocol is vulnerable to the password guessing attack, the Denial-of-Service (DoS) attack, and the replay attack.

In this paper, an enhanced three-factor security protocol is introduced that removes the shortcomings of past three-factor security protocols. Detailed operations of the new protocol are provided with comprehensive security analysis that proves the robustness of the protocol against various attacks.

The organization of this paper is as follows. Section II gives a review of the three-factor authentication protocol followed by its security issues discussed in Section III. Section IV introduces the proposed security protocol as part of this work. The protocol’s immunity from various attacks and other related features is analyzed in Section V. Section VI analyzes the proposed protocol’s computational cost. Section VII concludes the paper.

II. Review of the Three-Factor Elliptic Curve Cryptosystem Protocol

There are three phases in Lee et al. protocol [14]: (1) the registration phase; (2) the verification and data encryption phase; and (3) the key agreement phase. The details of these phases are described in this section. Notations used in this paper are first defined as follows:

\begin{itemize}
    \item \(p, q\): two large prime numbers;
    \item \(F_p\): finite field;
    \item \(E(F_p)\): elliptic curve over \(F_p\) defined by the equation \(y^2 = x^3 + ax + b\), where \(a, b \in F_p\) and \(4a^3 + 27b^2 \neq 0\);
    \item \(G\): cyclic additive group consisting of points on \(E(F_p)\) that has a specific point called the infinite point;
    \item \(P\): generator point of \(G\) with the order \(q\);
    \item \(AS\): authentication server;
    \item \(U\): user;
    \item \(A\): adversary;
    \item \(ID_U\): user \(U\)’s identity;
    \item \(B_U\): user \(U\)’s biometric characteristic (e.g., fingerprint);
    \item \(pw_U\): user \(U\)’s password;
    \item \(x\): authentication server \(AS\)’s secret key;
    \item \(
\): concatenate operation;
    \item \(h()\): one-way hash function;
    \item \(F_{\text{enc}}\): encrypted filename;
    \item \(E_k()\): symmetric encryption algorithm using a key \(K\);
    \item \(D_k()\): symmetric decryption algorithm using a key \(K\).
\end{itemize}
A. System Environment

To manage security for a USB MSD, AS restricts the data transfers over the USB interface. U is allowed to transfer data via the USB interface only when U could pass AS’s authentication. When U wishes to transfer a file to a storage device via the USB interface, U is required to input their username, password and biometric characteristic to verify legitimacy.

When U is successfully authenticated, a shared session key is generated between U and AS. Then, the session key will be used to encrypt the files transferred via the USB interface. When U decrypts the files on the storage devices, U must do the same authentication and generate the same session key for the original file. Every filename and user’s identity will have a session key and different files or users’ identity have different session keys. To ensure system security, the temporarily stored session key will be deleted after encrypting or decrypting the file. Lee et al.’s protocol [14] has the following three characteristics: (1) only authorized users can access the USB consumer storage devices; (2) files taken from the storage devices cannot be decrypted without the session key; and (3) other legal users cannot decrypt a legal classified file even if it is copied to their storage device. Therefore the original file is secure.

Fig. 1. Lee et al.’s three-factor authentication protocol.

Lee et al.’s three-factor authentication protocol [14] is illustrated in Fig. 1. U inserts their storage device into a client terminal and inputs their password, identity and biometric signature (phase 1). Mutual authentication is then executed between U and AS (phase 2). U obtains a session key from AS if they are successfully authenticated (phase 3). With this key, U can store an encrypted file on the storage device.

B. Registration Phase

When U wants to be a legal user of AS, then U has to be registered through the following steps:

1) U inputs their biometric characteristic $B_u$ through a specified biometric device and provides a password, $pw_u$ and identity $ID_u$. The system sends $\{ID_u, h(pw_u \parallel B_u)\}$ to AS.

2) Upon receiving $\{ID_u, h(pw_u \parallel B_u)\}$, AS computes $e_u = h(h(ID_u \parallel x) \parallel h(pw_u \parallel B_u))$ and $s_u = h(ID_u \parallel x) \oplus h(pw_u \parallel B_u)$, where $x$ is AS’s secret key. Then AS stores $\{e_u, s_u\}$ in U’s USB MSD and delivers it to U securely.

3) Upon receiving data from the storage device, U computes $BPW_u = B_u \oplus h(pw_u)$ and stores it in their storage device. Accordingly, the information $\{e_u, s_u, BPW_u\}$ is stored in the storage device.
C. Verification and Data Encryption Phase

When $U$ accesses the storage device, the following steps are executed between $U$ and AS for mutual authentication:

1) $U$ inserts their USB MSD into the client terminal, and inputs a password $pw_U$, identity $ID_u$ and biometric characteristic $B_u$. The device then computes $B'_u = BPW_u \oplus h(pw_u)$ and checks $B_u$ and $B'_u$ are equal. If they are not equal, the device rejects $U$’s request; otherwise, the device computes $w_u = s_u \oplus h(pw_u \parallel B_u)$ and checks if $h(w_u \parallel h(pw_u \parallel B_u))$ and $e_u$ are equal. If they are not equal, the device again rejects $U$’s request; otherwise, the device generates a random number $a \in \mathbb{Z}_q^*$, computes $aP$ and $\alpha = h(ID_u \parallel aP \parallel w_u)$, where $\mathbb{Z}_q^*$ denotes the set $\{1, 2, ..., q-1\}$. The message $m_1 = \{ID_u, aP, F_u, \alpha\}$ is sent to AS.

2) Upon receiving the message $m_1$, AS first checks the user’s identity. If it is not valid then AS rejects the request; otherwise, AS computes $w_u = h(ID_u \parallel x)$ and checks if $h(ID_u \parallel aP \parallel w_u)$ and $\alpha$ are equal. If they are not equal, AS rejects the request; otherwise, AS generates a random number $b \in \mathbb{Z}_q^*$ and computes $bP$, $sk = b(aP) = abP$, $n = h(x \parallel F_u)$ and $\beta = h(ID_u \parallel sk \parallel bP \parallel n \parallel w_u)$. Then, AS sends the message $m_2 = \{bP, E_a(n), \beta\}$ to $U$.

3) Upon receiving the message $m_2$, $U$ computes $sk = a(bP) = abP$ and uses it to decrypt $E_a(n)$. Then, $U$ obtains $n = h(x \parallel F_u)$. $U$ then checks if $h(ID_u \parallel sk \parallel bP \parallel n \parallel w_u)$ and $\beta$ are equal. If they are equal then $U$ has been successfully authenticated.

D. Key Agreement Phase

After completing mutual authentication, $U$ computes an encrypted key, $K = h(ID_u \parallel n)$. When $U$ wishes to access a file on the USB MSD, $U$ uses the key to encrypt a file as $E_k(\text{file})$ to ensure the security of the file on the storage device. If $U$ needs to decrypt the file, $U$ must follow the same steps to decrypt the file as $D_k(E_k(\text{file}))$ on the device.

III. Security Analysis of the Three-Factor authentication Protocol Based on ECC

In this section, the security of the three-factor authentication protocol is analyzed.

A. Password Guessing Attack

Assume that $A$ has obtained $U$’s USB storage device. Then, $A$ could read the stored information $\{e_u, s_u, BPW_u\}$ from the device, where $e_u = h(h(ID_u \parallel x) \parallel h(pw_u \parallel B_u))$, $s_u = h(ID_u \parallel x) \oplus h(pw_u \parallel B_u)$ and $BPW_u = B_u \oplus h(pw_u)$. However, $A$ could obtain the password through the following steps:

1) $A$ guesses a password $pw'_u$ from a directory, $D$.

2) $A$ computes $B'_u = BPW_u \oplus h(pw'_u)$ and $w'_u = s_u \oplus h(pw'_u \parallel B'_u)$.

3) $A$ checks if $h(w'_u \parallel h(pw'_u \parallel B'_u))$ and $e_u$ are equal. If they are equal, $pw'_u$ is the correct password; otherwise, $A$ repeats steps 1)-3) until the correct password is found.

With the found password and stored information $BPW_u = B_u \oplus h(pw_u)$, $A$ could generate a legal login message like $U$ normally does. Then, $A$ could impersonate $U$ to login to AS and obtain the secure data. Therefore, the protocol [14] is vulnerable to the password guessing attack as specifically could be the case where $A$ has gained possession of $U$’s USB MSD this allowing $A$ to do the attack.

5
B. Denial of Service (DoS) Attack

In step 1) of the verification and data encryption phase, \( U \) inserts their USB storage device into the client terminal, and inputs their password \( pw_v \), identity \( ID_v \) and biometric characteristic \( B_v \). The device then computes \( B'_v = BPW_v \otimes h(pw_v) \) and checks if \( B_v \) and \( B'_v \) are identical. If they are not equal, the device rejects \( U \)’s request. However, it is known that the inputted biometric characteristic of the same person can be somewhat different every time [17]. Then \( B_v \) and \( B'_v \) are not equal and the device may reject \( U \)’s valid request. Therefore, the protocol [14] is somewhat vulnerable to the DoS attack due to the unrepeatability of biometric characteristic.

C. Replay Attack

Suppose \( A \) could control the communication channel between \( U \) and \( AS \) since messages are transmitted via an insecure channel in the login and key agreement phase. Therefore, \( A \) could intercept, insert, delete, or interpolate any messages at will. \( A \) could intercept a message \( m_i \) sent by \( U \). Then, \( A \) could replay it to \( AS \). Although \( A \) cannot compute the session key, \( A \) is successful as long as \( AS \) accepts the login request. Therefore, the protocol [14] is vulnerable to the replay attack.

IV. The Proposed Protocol

This section proposes significant enhancements to the three-factor authentication protocol. Before the proposed protocol operations are described, a fuzzy extractor [18] used in the proposed protocol is defined as illustrated below:

**Definition 1:** - Metric Space [18]. A metric space is a set \( \mathcal{Y} \) with a distance function \( \text{dis} : \mathcal{Y} \times \mathcal{Y} \to R' = [0, \infty) \) which obeys various natural properties. One example of metric spaces is the **Hamming metric** : \( \mathcal{Y} = \Gamma^n \) is over some alphabet \( \Gamma \) (e.g., \( \Gamma = \{0, 1\} \) ) and \( \text{dis}(\omega, \omega') \) is the number of positions in which they differ.

**Definition 2:** - Statistic Distance [18]. Statistic Distance is the distance between two probability distributions \( \alpha \) and \( \beta \) and is denoted by \( SD(\alpha, \beta) = \frac{1}{2} \sum \left| \text{Pr}[\alpha - v] - \text{Pr}[\beta - v] \right| \).

**Definition 3:** - Entropy [18]. The min-entropy \( H_\omega(\alpha) \) of a random variable \( \alpha \) is \( -\log(\max_\omega \text{Pr}[\alpha = a]) \).

A fuzzy extractor extracts a nearly random string \( \sigma \) from its biometric characteristic input \( \omega \) in an error-tolerant way. If the input changes but remains close to \( \omega \), then the extracted \( \sigma \) remains the same. To assist in recovering \( \sigma \) from a biometric characteristic input \( \omega' \), a fuzzy extractor outputs an auxiliary string \( \theta \). However, \( \sigma \) remains uniformly random for a given \( \theta \). The fuzzy extractor is formally defined as below:

**Definition 4:** - Fuzzy Extractor [18]. A (\( \mathcal{Y}, m, l, t, \epsilon \)) fuzzy extractor is given by two procedures, \( \text{Gen} \) and \( \text{Rep} \):

1) **Gen** is a probabilistic generation procedure, which on (biometric characteristic) input \( \omega \in \mathcal{Y} \) outputs an “extracted” string \( \sigma \in \{0, 1\}^l \) and an auxiliary string \( \theta \). For any distribution \( W \) on \( \mathcal{Y} \) of min-entropy \( m \), if \( \langle \sigma, \theta \rangle \leftarrow \text{Gen}(W) \), then \( SD(\langle \sigma, \theta \rangle, \langle U_l, \theta \rangle) \leq \epsilon \). Here, \( U_l \) denotes the uniform distribution on \( l \)-bit binary strings.

2) **Rep** is a deterministic reproduction procedure allowing to recover \( \sigma \) from the corresponding
auxiliary string \( \vartheta \) and any vector \( \omega' \) close to \( \omega \): for all \( \omega, \omega' \in Y \) satisfying \( \text{dis}(\omega, \omega') \leq t \), if \( <\sigma, \vartheta> \leftarrow \text{Gen}(W) \), then \( \text{Rep}(\omega', \vartheta) = \sigma \).

Like Lee et al. protocol [14], the proposed protocol in this paper also consists of the three phases, i.e., the registration phase, the verification and data encryption phase, and the key agreement phase. The system environment of the proposed protocol is the same as Lee et al. protocol.

A. Registration phase

When \( U \) wants to be a legal user of \( AS \), the following steps are executed:

1) \( U \) inputs their biometric characteristic \( B_U \) through a suitable biometric device and provides their password \( pw_U \) and identity \( ID_U \). \( U \) then computes \( (\sigma_U, \vartheta_U) = \text{Gen}(B_U) \) and submits \( \{ID_U, h(pw_U \parallel \sigma_U)\} \) to \( AS \).

2) Upon receiving \( \{ID_U, h(pw_U \parallel \sigma_U)\} \), \( AS \) computes \( e_U = h(h(ID_U \parallel x) \parallel h(pw_U \parallel \sigma_U)) \) and \( s_U = h(ID_U \parallel x) \oplus h(pw_U \parallel \sigma_U) \), where \( x \) is \( AS \)'s secret key. Then, \( AS \) stores \( \{e_U, s_U\} \) in \( U \)'s storage device and delivers it to \( U \) securely.

3) Upon receiving the USB consumer storage device information, \( U \) computes \( BPW_U = \vartheta_U \oplus h(pw_U) \) and stores the result in the storage device. The storage device thus contains the information \( \{e_U, s_U, BPW_U\} \).

B. Verification and Data Encryption Phase

When \( U \) wants to access the USB MSD, the following steps are executed between \( U \) and \( AS \) for mutual authentication:

1) \( U \) inserts their USB storage device into the client USB port and inputs their password \( pw_U \), identity \( ID_U \) and biometric characteristic \( B_U' \). The device computes \( \vartheta_U = BPW_U \oplus h(pw_U) \), \( \sigma_U = \text{Rep}(B_U', \vartheta_U) \), and \( w_U = s_U \oplus h(pw_U \parallel \sigma_U) \). Then, it checks if \( h(w_U \parallel h(pw_U \parallel \sigma_U)) \) and \( e_U \) are equal. If they are not equal, the device rejects \( U \)'s request; otherwise, the device generates a random number \( a \in Z_q^* \), and then computes \( aP \) and \( \alpha = h(ID_U \parallel aP \parallel F_u \parallel w_U) \). The message \( m_1 = \{ID_U, aP, F_u, \alpha\} \) is then sent to \( AS \).

2) Upon receiving the message \( m_1 \), \( AS \) checks the user’s identity first. If it not true, \( AS \) rejects the request; otherwise, \( AS \) computes \( w_1 = h(ID_U \parallel x) \) and checks if \( h(ID_U \parallel aP \parallel F_u \parallel w_U) \) and \( \alpha \) are equal. If they are not equal, \( AS \) rejects the request; otherwise, \( AS \) generates a random number \( b \in Z_q^* \) and computes \( bP, sk = b(aP) = abP \), \( n = h(x \parallel F_u) \) and \( \beta = h(0^* \parallel ID_U \parallel aP \parallel F_u \parallel bP \parallel n \parallel sk \parallel w_U) \).

The \( AS \) then sends the message \( m_2 = \{bP, E_{sk}(n), \beta\} \) to \( U \).

3) Upon receiving the message \( m_2 \), \( U \) computes \( sk = a(bP) = abP \) and uses it to decrypt \( E_{sk}(n) \). Then, \( U \) obtains \( n = h(x \parallel F_u) \) and \( U \) checks if \( h(0^* \parallel ID_U \parallel aP \parallel F_u \parallel bP \parallel n \parallel sk \parallel w_U) \) and \( \beta \) are equal. If they are not equal, \( U \) stops the session; otherwise, it is authenticated. Next, \( U \) computes \( \gamma = h(1^* \parallel ID_U \parallel aP \parallel F_u \parallel bP \parallel n \parallel sk \parallel w_U) \) and sends the message \( m_3 = \{\gamma\} \) to \( AS \).

4) Upon receiving the message \( m_3 \), \( AS \) checks if \( \gamma \) and \( h(1^* \parallel ID_U \parallel aP \parallel F_u \parallel bP \parallel n \parallel sk \parallel w_U) \) are equal.

If they are not equal, \( AS \) stops the session; otherwise, \( U \) is authenticated.

C. Key Agreement Phase

After completing mutual authentication, \( U \) computes an encrypted key \( K = h(ID_U \parallel n) \). When \( U \) wants to
access the USB MSD, U uses the key to encrypt a file as \( E_k(\text{file}) \) to ensure the security of the file on the storage device. If U needs to decrypt the file, then they must follow the same steps to decrypt the file as \( D_k(E_k(\text{file})) \) on the storage device.

V. Security Analysis

In this section, the security of the proposed protocol is analyzed for USB consumer storage devices. Burrows–Abadi–Needham (BAN) logic [19], [20] has been used to demonstrate that the proposed protocol provides secure authentication. Then, the security assessment is performed to test whether the proposed protocol can overcome weaknesses in past security algorithms.

A. Authentication Proof Based on BAN-logic

The notations of BAN logic are as follows:

- \( P \models X \) : The principal \( P \) believes a statement \( X \), or \( P \) is entitled to believe \( X \).
- \( #(X) \) : The formula \( X \) is fresh.
- \( P \Rightarrow X \) : The principal \( P \) has jurisdiction over the statement \( X \).
- \( P \neg X \) : The principal \( P \) sees the statement \( X \).
- \( P \neg X \) : The principal \( P \) once said the statement \( X \).
- \( (X,Y) \) : The formula \( X \) or \( Y \) is one part of the formula \( (X,Y) \).
- \( <X>_p \) : The formula \( X \) combined with the formula \( Y \).
- \( \{X\}_p \) : The formula \( X \) is encrypted under the key \( K \).
- \( (X)_y \) : The formula \( X \) is hash with the key \( K \).
- \( P \leftrightarrow Q \) : The principals \( P \) and \( Q \) use the shared key \( K \) to communicate. The key \( K \) will never be discovered by any principal except \( P \) and \( Q \).
- \( sk \) : The session key used in the current session.

Main logical postulates of the BAN logic are as follows:

The message-meaning rule:

\[
\begin{align*}
P &\models P \leftrightarrow Q, \ P \models \{X\}_K \\
\models P \models Q \models X
\end{align*}
\]

The freshness-conjunctenation rule:

\[
\begin{align*}
P &\models #(X) \\
\models P \models #(X,Y)
\end{align*}
\]

The nonce-verification rule:

\[
\begin{align*}
P &\models #(X), \ P \models Q \models X \\
\models P \models Q \models X
\end{align*}
\]

The jurisdiction rule:

\[
\begin{align*}
P &\models Q \Rightarrow X, \ P \models Q \models X \\
\models P \models X
\end{align*}
\]
According to the analytic procedures of the BAN logic, the proposed protocol must then satisfy the following test goals in order to prove the system is secure:

**Goal 1:** \( U \models (U \xrightarrow{K} AS) \);
**Goal 2:** \( U \models AS \models (U \xrightarrow{K} AS) \);
**Goal 3:** \( AS \models (U \xleftarrow{K} AS) \);
**Goal 4:** \( AS \models U \models (U \xrightarrow{K} AS) \);
**Goal 5:** \( U \models (U \xleftarrow{K} AS) \);
**Goal 6:** \( U \models AS \models (U \xleftarrow{K} AS) \);
**Goal 7:** \( AS \models (U \xleftarrow{K} AS) \);
**Goal 8:** \( AS \models U \models (U \xleftarrow{K} AS) \);

First, the proposed protocol is transformed to the idealized form as:

**Msg 1.** \( U \rightarrow AS \):
\[
(ID_U, aP, F_n)_{U \xrightarrow{APIB} AS}
\]

**Msg 2.** \( AS \rightarrow U \):
\[
("0", aP, bP, U \xleftarrow{K} AS, U \leftarrow \alpha AS)_{U \xleftarrow{APIB} AS}
\]

**Msg 3.** \( U \rightarrow S \):
\[
("1", aP, bP, U \xleftarrow{K} AS, U \leftarrow \alpha AS)_{U \xleftarrow{APIB} AS}
\]

Second, the following assumptions about the initial state of the protocol are made to analyze the proposed protocol:

\( A_1 : U \models \#(aP) \);
\( A_2 : AS \models \#(bP) \);
\( A_3 : U \models (U \xrightarrow{h(ID_U)} AS) \);
\( A_4 : AS \models (U \xleftarrow{h(ID_U)} AS) \);
\( A_5 : U \models AS \models (U \leftarrow K AS) \);
\( A_6 : AS \models U \models (U \xrightarrow{K} AS) \);
\( A_7 : U \models AS \models (U \leftarrow \alpha AS) \);
\( A_8 : AS \models U \models (U \xleftarrow{\alpha AS}) \);

Third, the idealized form of the proposed protocol is analyzed based on the BAN logic rules and the assumptions. The main proofs are stated as follows:

According to **Msg 1**, the following is obtained:
\[
S_1 : AS \triangleleft (ID_U, aP, F_n)_{U \xrightarrow{APIB} AS}
\]

According to \( A_4 \), the message-meaning rule is applied:
\[
S_2 : AS \models U \models (ID_U, aP, F_n)
\]

According to **Msg 2**, the following is obtained:
\[
S_3 : U \triangleleft ("0", aP, bP, U \xrightarrow{K} AS, U \leftarrow \alpha AS)_{U \xleftarrow{APIB} AS}
\]
According to $A_1$, the message-meaning rule is applied:

\[ S_4 : U \models AS \models ("0", aP, bP, U \leftarrow k \rightarrow AS, U \leftarrow d \rightarrow AS) \]

According to $A_1$, the freshness-conjunction rule is applied:

\[ S_5 : U \models AS \models ("0", aP, bP, U \leftarrow k \rightarrow AS, U \leftarrow d \rightarrow AS) \]

According to $S_5$, the BAN logic rule is applied to break conjunctions to produce:

\[ S_6 : U \models AS \models (U \leftarrow k \rightarrow AS) \quad \text{Goal 2} \]
\[ S_7 : U \models AS \models (U \leftarrow d \rightarrow AS) \quad \text{Goal 6} \]

According to $A_1$, the jurisdiction rule is applied to get:

\[ S_8 : U \models (U \leftarrow k \rightarrow AS) \quad \text{Goal 1} \]

According to $A_1$, the jurisdiction rule is applied to get:

\[ S_9 : U \models (U \leftarrow d \rightarrow AS) \quad \text{Goal 5} \]

According to $Msg\ 3$, the following is obtained:

\[ S_{10} : AS \models ("1", aP, bP, U \leftarrow k \rightarrow AS, U \leftarrow d \rightarrow AS) \]

According to $A_1$, the message-meaning rule is applied:

\[ S_{11} : U \models AS \models ("0", aP, bP, U \leftarrow k \rightarrow AS, U \leftarrow d \rightarrow AS) \]

According to $A_1$, the freshness-conjunction rule is applied to get:

\[ S_{12} : AS \models U \models ("1", aP, bP, U \leftarrow k \rightarrow AS, U \leftarrow d \rightarrow AS) \]

According to $S_{12}$, the BAN logic rule is applied to break conjunctions to produce:

\[ S_{13} : AS \models U \models (U \leftarrow k \rightarrow AS) \quad \text{Goal 3} \]
\[ S_{14} : AS \models U \models (U \leftarrow d \rightarrow AS) \quad \text{Goal 7} \]

According to $A_1$, the jurisdiction rule is applied to get

\[ S_{15} : AS \models (U \leftarrow k \rightarrow AS) \quad \text{Goal 4} \]

According to $A_1$, the jurisdiction rule is applied to get

\[ S_{16} : AS \models (U \leftarrow d \rightarrow AS) \quad \text{Goal 8} \]

According to Goal 1 – Goal 8, both $U$ and $AS$ know that a session key $sk = abP$ and an encrypted key $K = h(ID_U \parallel n)$ have successfully been shared between $U$ and $AS$.

### B. Security Assessment

1) Password guessing attack

Assume an adversary $A$ has stolen the user $U$’s USB MSD. Then, they could read the stored information \{e$_U$, s$_U$, BPW$_U$\} from the device. $A$ could guess a password $pw'_U$ and compute $\sigma'_U = BPW_U \oplus h(pw'_U)$ . However, $A$ cannot compute the corresponding $\sigma_U$ without $U$’s biometric characteristic. Therefore, they cannot verify the correctness of $pw'_U$. Therefore, the proposed protocol should withstand the password
guessing attack.

2) **DoS attack**

In the proposed protocol, the inputted biometric characteristic of the same person are also different every time. However, the device can get the correct $\sigma_{0}$ through the fuzzy extractor algorithm. Therefore, $U$ can pass the device’s verification and the proposed algorithm thus withstands the DoS attack.

3) **Replay attack**

Suppose adversary $A$ intercepts the message $m_1$ sent by $U$ and replays it back to $AS$. Without knowing value $w_{U}$, $A$ cannot compute $\gamma$ for a newly generated $bP$. Then, $AS$ could determine the attack by checking the correctness of $\gamma$. $A$ may intercept the message $m_2$ sent by $AS$ and replay it to $U$. However, $U$ can identify the attack by checking the correctness of $\beta$ since $a$ is generated for every session. Therefore, the proposed algorithm should withstand the replay attack.

4) **Stolen-verifier attack**

In the proposed protocol, $AS$ maintains no password table at all. Therefore, the proposed algorithm should withstand the stolen-verifier attack.

5) **Impersonation attack**

Suppose an adversary $A$ wants to impersonate $U$ to $AS$. $A$ could generate a random number $a \in Z_{p}$ and compute $aP$. However, $A$ cannot compute $\alpha$ since $A$ does not know $w_{U}$. Furthermore, $AS$ can find the attack by checking the correctness of $\alpha$. Suppose $A$ intercepts the message $m_1$ and wants to impersonate $U$ to $AS$, however in this case $A$ cannot compute $\beta$ without the value $w_{U}$. $U$ can also identify the attack by checking the correctness of $\beta$. Therefore, the proposed protocol should withstand the impersonation attack.

6) **Mutual authentication:**

The proposed protocol allows that only $U$ knows $U$’s secret key, $w_{U} = h(ID_{U} \parallel x)$, otherwise $AS$’s secret key $x$ could generate the legal message $\beta$ and $\gamma$. Then, $U$ and $AS$ can confirm $m_2$ and $m_3$ are sent by $AS$ and $U$ by checking the correctness of $\beta$ and $\gamma$ separately. The proposed protocol thus should provide mutual authentication between $U$ and $AS$.

7) **Man-in-the-middle attack:**

From the above description, it has been shown that the proposed protocol should provide mutual authentication between $U$ and $AS$, therefore by definition, the proposed algorithm should also withstand the man-in-the-middle attack.

VI. **Computational Cost Analysis**

In this section, the proposed protocol is compared with Yang *et al.* protocol [11] and Lee *et al.* protocol [14] in terms of relative computational cost. This work analyzed the target protocols [11], [14] and explicitly divided the protocols’ operations in terms of crypto-operations. Then, the relative
computational times and the absolute times were subsequently calculated as before [14], [21]. Notations are as follows:

\[ T_{\text{hash}}: \text{Time for executing a hash function;} \]
\[ T_{\text{sym}}: \text{Time for executing a symmetric key cryptography;} \]
\[ T_{\text{pm}}: \text{Time for executing an elliptic curve point multiplication;} \]
\[ T_{\text{exp}}: \text{Time for executing a modular exponentiation;} \]
\[ T_{\text{fe}}: \text{Time for executing a fuzzy extractor.} \]

TABLE I shows the relative cost comparisons for Yang et al. protocol, Lee et al. protocol, and the proposed protocol in this paper. The total computational cost of the verification and data encryption phase of Yang et al. protocol, Lee et al. protocol, and the proposed protocol are \(10T_{\text{sym}} + 5 T_{\text{hash}} + 2 T_{\text{exp}},\)
\(4T_{\text{pm}} + 9T_{\text{hash}} + 2T_{\text{sym}}\) and \(4T_{\text{pm}} + 9T_{\text{hash}} + 2T_{\text{sym}} + T_{\text{fe}}\) respectively. To be precise, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, modular exponentiation operation and an elliptic curve point relative multiplication operation is 0.00032 s, 0.0056 s, 0.0192 s and 0.0171 s respectively [14]. The total relative computational time of Yang et al. protocol, Lee et al. protocol, and the proposed protocol are 0.20488 s, 0.08248 s and 0.09958 s, respectively. The proposed protocol requires the fuzzy extractor that can be constructed from universal hash functions or error-correcting codes requiring only lightweight operations [18]. It is here assumed that the time for executing a fuzzy extractor is the same as that for executing an elliptic curve point multiplication at the most. Note that the elliptic curve point multiplication is considered as a complicated and time-consuming operation among the cryptographic operations.

The proposed protocol and Lee et al. protocol both show better computation performance than Yang et al. protocol as expected; while at the same time the proposed protocol addresses the vulnerabilities in Lee et al. protocol with a small extra computational cost. Hence, the proposed protocol is suitable for practical applications in terms of security reliability and computational efficiency.

**TABLE I**

<table>
<thead>
<tr>
<th>User</th>
<th>Authentication Server</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang et al. protocol [11]</td>
<td>(4T_{\text{sym}} + 3T_{\text{hash}} + T_{\text{exp}})</td>
<td>(6T_{\text{sym}} + 2T_{\text{hash}} + T_{\text{exp}})</td>
</tr>
<tr>
<td>Lee et al. protocol [14]</td>
<td>(2T_{\text{pm}} + 5T_{\text{hash}} + T_{\text{exp}})</td>
<td>(2T_{\text{pm}} + 4T_{\text{hash}} + T_{\text{exp}})</td>
</tr>
<tr>
<td>Proposed protocol</td>
<td>(2T_{\text{pm}} + 5T_{\text{hash}} + T_{\text{exp}})</td>
<td>(2T_{\text{pm}} + 4T_{\text{hash}} + T_{\text{exp}})</td>
</tr>
<tr>
<td></td>
<td>(T_{\text{sym}} + T_{\text{fe}} + T_{\text{exp}})</td>
<td>(T_{\text{sym}} + T_{\text{fe}} + T_{\text{exp}})</td>
</tr>
<tr>
<td></td>
<td>(0.0836)</td>
<td>(0.12144)</td>
</tr>
<tr>
<td></td>
<td>(0.0414)</td>
<td>(0.04108)</td>
</tr>
<tr>
<td></td>
<td>(0.0585)</td>
<td>(0.04108)</td>
</tr>
</tbody>
</table>

**VII. Conclusion**

The three-factor authentication protocol based on Elliptic Curve Cryptosystem for USB consumer storage devices has been shown to have significant advantages, but as presented in this paper, there were still existing security vulnerability issues needed to be solved, specifically the password guessing attack, the DoS attack and the replay attack. This paper has presented a significantly enhanced security protocol to address previous weaknesses. The proposed protocol has been presented and rigorously analyzed in
terms of security and computational cost. As shown, the proposed protocol is robust against conceivable attacks while at the same time having the same computational cost compared to the literature. The work is ideal to be embedded in the firmware of consumer based USB Mass Storage Devices thus relieving the user of extra security burdens and enabling the devices to be confidently used in the knowledge that the data stored is secure.

REFERENCES


BIographies

Debiao He received the Ph.D. degree in applied mathematics from School of Mathematics and Statistics, Wuhan University in 2009. He is currently a lecturer at Wuhan University, China. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Neeraj Kumar received the Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (India) and PDF from Coventry University, Coventry, UK. He is now an Assistant Professor in the Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India. He is a senior member of ACEEE and IACSIT. His research is focused on mobile computing, parallel/distributed computing, multi-agent systems, service oriented computing, routing and security issues in mobile ad hoc, sensor and mesh networks.

Jong-Hyouk Lee (M’07-SM’12) received the M.S. and Ph.D. degrees in Computer Engineering from Sungkyunkwan University, Korea. He was a researcher at INRIA, France and was an Assistant Professor at TELECOM Bretagne, France. He is now an Assistant Professor at the Department of Computer Software Engineering, Sangmyung University, Korea. His research interests include authentication, privacy, smart networking technologies, and Internet mobility management.

Dr. Lee is an Associate Editor of Wiley Security and Communication Networks and the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS. He won the Best Paper Award at the IEEE WiMob 2012 and was a tutorial speaker at the IEEE WCNC 2013.

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 both from the University of Salford. In 1996, he has appointed as a Lecturer in Electronic Engineering at the University of Reading where he is currently a Professor of Consumer Electronics and the Head of Wireless and Computing research. He is also a Guest Professor at Nanjing University of Information Science and Technology, China. His research topic is on signal processing in consumer electronic devices concentrating on equalization and DSP architectures, specifically for Personal Area Networks, USB and Wireless-USB.

Eur Ing Professor Sherratt has served the IEEE Consumer Electronics Society as a Vice President (Conferences) (2008/9), AdCom member (2003-2008, 2010-) and Awards chair (2006/7). He is a member of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS Editorial Board (2004-) and is currently the Editor-in-Chief (2011-), the IEEE International Conference on Consumer Electronics general chair (2009) and the IEEE International Symposium on Consumer Electronics general chair (2004). He received the IEEE Chester Sall 1st place best Transactions on Consumer Electronics paper award for 2004 and the best paper in the IEEE International Symposium on Consumer Electronics in 2006.