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

Article

Accepted Version

Full Text of article

Title: **Enhanced Three-factor Security Protocol for Consumer USB Mass Storage Devices**

Publication: **IEEE Transactions on Consumer Electronics**
Volume: **60**
Issue: **1**
pp.: **30-37**
URL: **http://dx.doi.org/10.1109/TCE.2014.6780922**
DOI: **10.1109/TCE.2014.6780922**

Authors:

Debiao He, School of Mathematics and Statistics, Wuhan University, Wuhan, China (e-mail: hedebiao@163.com).

Neeraj Kumar, Department of Computer Science and Engineering, Thapar University, Patiala, India (e-mail: neeraj.kumar@thapar.edu).

Jong-Hyouk Lee, *Senior Member*, IEEE, Department of Computer Software Engineering, Sangmyung University, Republic of Korea (e-mail: jonghyouk@smu.ac.kr).

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

This work was supported by a 2013 Research Grant from Sangmyung University, Republic of Korea.

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:

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

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.

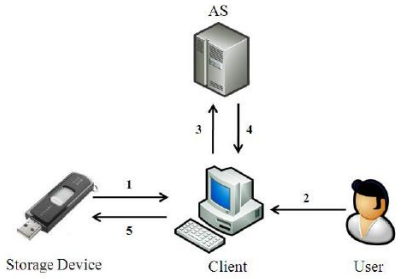


Fig. 1. Lee *et al.'s* three-factor authentication protocol.

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 $\{ID_U, h(pw_U \| B_U)\}$ to *AS*.
2) Upon receiving $\{ID_U, h(pw_U \| B_U)\}$, *AS* computes $e_U = h(h(ID_U \| x) \| h(pw_U \| B_U))$ and $s_U = h(ID_U \| x) \oplus h(pw_U \| B_U)$, 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 $w_U = s_U \oplus h(pw_U \| B_U)$ and checks if $h(w_U \| h(pw_U \| B_U))$ and e_U are equal. If they are not equal, the device again rejects *U*'s request; otherwise, the device generates a random number $a \in Z_q^*$, computes aP and $\alpha = h(ID_U \| aP \| w_U)$, where Z_q^* denotes the set $\{1, 2, \ldots, q-1\}$. The message $m_1 = \{ID_U, aP, F_n, \alpha\}$ 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 $w_U = h(ID_U \| x)$ and checks if $h(ID_U \| aP \| w_U)$ and α are equal. If they are not equal, *AS* rejects the request; otherwise, *AS* generates a random number $b \in Z_q^*$ and computes bP, $sk = b(aP) = abP$, $n = h(x \| F_n)$ and $\beta = h(ID_U \| sk \| bP \| n \| w_U)$. Then, *AS* 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 $n = h(x \| F_n)$. *U* then checks if $h(ID_U \| sk \| bP \| n \| w_U)$ and β 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, $K = h(ID_U \| n)$. When *U* wishes to access a file on the USB MSD, *U* uses the key to encrypt a file as $E_K(\textbf{file})$ 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 $D_K(E_K(\textbf{file}))$ 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 $e_U = h(h(ID_U \| x) \| h(pw_U \| B_U))$, $s_U = h(ID_U \| x) \oplus h(pw_U \| B_U)$ 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 $w'_U = s_U \oplus h(pw'_U \| B'_U)$.
3) *A* checks if $h(w'_U \| h(pw'_U \| B'_U))$ 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.

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_U, identity ID_U and biometric characteristic B_U. The device then computes $B'_U = BPW_U \oplus h(pw_U)$ and checks if B_U and B'_U 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_U and B'_U 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_1 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 Υ with a distance function $dis: \Upsilon \times \Upsilon \to R^+ = [0,\infty)$ which obeys various natural properties. One example of metric spaces is the *Hamming metric*: $\Upsilon = \Gamma^n$ is over some alphabet Γ^n (e.g., $\Gamma^n = \{0,1\}$) and $dis(\omega,\omega')$ is the number of positions in which they differ.

Definition 2: - Statistic Distance [18]. Statistic Distance is the distance between two probability distributions α and β and is denoted by $SD(\alpha,\beta) = \frac{1}{2}\sum_v |\Pr[\alpha - v] - \Pr[\beta - v]|$.

Definition 3: - Entropy [18]. The min-entropy $H_\infty(\alpha)$ of a random variable α is $-\log(\max_a \Pr[\alpha = a])$.

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

Definition 4: - Fuzzy Extractor [18]. A $(\Upsilon, m, l, t, \varepsilon)$ fuzzy extractor is given by two procedures, *Gen* and *Rep*:

1) *Gen* is a probabilistic generation procedure, which on (biometric characteristic) input $\omega \in \Upsilon$ outputs an "extracted" string $\sigma \in \{0,1\}^l$ and an auxiliary string ϑ. For any distribution *W* on Υ of min-entropy *m*, if $<\sigma,\vartheta> \leftarrow Gen(W)$, then $SD(<\sigma,\vartheta>, <U_l,\vartheta>) \le \varepsilon$. Here, U_l denotes the uniform distribution on *l*-bit binary strings.
2) *Rep* is a deterministic reproduction procedure allowing to recover σ from the corresponding

auxiliary string ϑ and any vector ω' close to ω : for all $\omega, \omega' \in \Upsilon$ satisfying $dis(\omega, \omega') \le t$, if $< \sigma, \vartheta > \leftarrow Gen(W)$, then $Rep(\omega', \vartheta) = \sigma$.

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 $(\sigma_U, \vartheta_U) = Gen(B_U)$ and submits $\{ID_U, h(pw_U \| \sigma_U)\}$ to *AS*.
2) Upon receiving $\{ID_U, h(pw_U \| \sigma_U)\}$, *AS* computes $e_U = h(h(ID_U \| x) \| h(pw_U \| \sigma_U))$ and $s_U = h(ID_U \| x) \oplus h(pw_U \| \sigma_U)$, 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 $\vartheta_U = BPW_U \oplus h(pw_U)$, $\sigma_U = Rep(B'_U, \vartheta_U)$, and $w_U = s_U \oplus h(pw_U \| \sigma_U)$. Then, it checks if $h(w_U \| h(pw_U \| \sigma_U))$ 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 $\alpha = h(ID_U \| aP \| F_n \| w_U)$. The message $m_1 = \{ID_U, aP, F_n, \alpha\}$ 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 $w_U = h(ID_U \| x)$ and checks if $h(ID_U \| aP \| F_n \| w_U)$ and α are equal. If they are not equal, *AS* rejects the request; otherwise, *AS* generates a random number $b \in Z_q^*$ and computes bP , $sk = b(aP) = abP$, $n = h(x \| F_n)$ and $\beta = h("0" \| ID_U \| aP \| F_n \| bP \| n \| sk \| w_U)$. 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 $n = h(x \| F_n)$ and *U* checks if $h("0" \| ID_U \| aP \| F_n \| bP \| n \| sk \| w_U)$ and β are equal. If they are not equal, *U* stops the session; otherwise, it is authenticated. Next, *U* computes $\gamma = h("1" \| ID_U \| aP \| F_n \| bP \| n \| sk \| w_U)$ and sends the message $m_3 = \{\gamma\}$ to *AS*.
4) Upon receiving the message m_3, *AS* checks if γ and $h("1" \| ID_U \| aP \| F_n \| bP \| n \| sk \| w_U)$ 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 $K = h(ID_U \| n)$. When *U* wants to

access the USB MSD, *U* uses the key to encrypt a file as $E_K(\textbf{file})$ 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 $D_K(E_K(\textbf{file}))$ 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*.

$P \lhd X$: The principal *P* sees the statement *X*.

$P \mid\sim X$: The principal *P* once said the statement *X*.

(X,Y) : The formula *X* or *Y* is one part of the formula (*X, Y*).

$< X >_Y$: The formula *X* combined with the formula *Y*.

$\{X\}_Y$: The formula *X* is encrypted under the key *K*.

$(X)_Y$: The formula *X* is hash with the key *K*.

$P \xleftrightarrow{K} Q$: 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: $\dfrac{P \models P \overset{K}{\leftrightarrow} Q, P \lhd \{X\}_K}{P \models Q \mid\sim X}$

The freshness-conjuncatenation rule: $\dfrac{P \models \#(X)}{P \models \#(X,Y)}$

The nonce-verification rule: $\dfrac{P \models \#(X), P \models Q \mid\sim X}{P \models Q \models X}$

The jurisdiction rule: $\dfrac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X}$

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: $U \mid\equiv (U \overset{K}{\longleftrightarrow} AS)$;
Goal 2: $U \mid\equiv AS \mid\equiv (U \overset{K}{\longleftrightarrow} AS)$
Goal 3: $AS \mid\equiv (U \overset{K}{\longleftrightarrow} AS)$
Goal 4: $AS \mid\equiv U \mid\equiv (U \overset{K}{\longleftrightarrow} AS)$
Goal 5: $U \mid\equiv (U \overset{sk}{\longleftrightarrow} AS)$
Goal 6: $U \mid\equiv AS \mid\equiv (U \overset{sk}{\longleftrightarrow} AS)$
Goal 7: $AS \mid\equiv (U \overset{sk}{\longleftrightarrow} AS)$
Goal 8: $AS \mid\equiv U \mid\equiv (U \overset{sk}{\longleftrightarrow} AS)$

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

Msg **1.** $U \rightarrow AS$:
$(ID_U, aP, F_n)_{U \overset{h(ID_U \| x)}{\longleftrightarrow} AS}$
Msg **2.** $AS \rightarrow U$:
$("0", aP, bP, U \overset{K}{\longleftrightarrow} AS, U \overset{sk}{\longleftrightarrow} AS)_{U \overset{h(ID_U \| x)}{\longleftrightarrow} AS}$
Msg **3.** $U \rightarrow S$:
$("1", aP, bP, U \overset{K}{\longleftrightarrow} AS, U \overset{sk}{\longleftrightarrow} AS)_{U \overset{h(ID_U \| x)}{\longleftrightarrow} AS}$

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

A_1: $U \mid\equiv \#(aP)$;
A_2: $AS \mid\equiv \#(bP)$;
A_3: $U \mid\equiv (U \overset{h(ID_U \| x)}{\longleftrightarrow} AS)$;
A_4: $AS \mid\equiv (U \overset{h(ID_U \| x)}{\longleftrightarrow} AS)$;
A_5: $U \mid\equiv AS \mid\Rightarrow (U \overset{K}{\longleftrightarrow} AS)$;
A_6: $AS \mid\equiv U \mid\Rightarrow (U \overset{K}{\longleftrightarrow} AS)$;
A_7: $U \mid\equiv AS \mid\Rightarrow (U \overset{sk}{\longleftrightarrow} AS)$;
A_8: $AS \mid\equiv U \mid\Rightarrow (U \overset{sk}{\longleftrightarrow} AS)$;

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: $AS \triangleleft (ID_U, aP, F_n)_{U \overset{h(ID_U \| x)}{\longleftrightarrow} AS}$

According to A_4, the message-meaning rule is applied:
S_2: $AS \mid\equiv U \mid\sim (ID_U, aP, F_n)$

According to *Msg* **2**, the following is obtained:
S_3: $U \triangleleft ("0", aP, bP, U \overset{K}{\longleftrightarrow} AS, U \overset{sk}{\longleftrightarrow} AS)_{U \overset{h(ID_U \| x)}{\longleftrightarrow} AS}$

According to A_3, the message-meaning rule is applied:

$S_4: U \mid\equiv AS \mid\sim ("0", aP, bP, U \xleftrightarrow{K} AS, U \xleftrightarrow{sk} AS)$

According to A_1, the freshness-conjuncatenation rule is applied:

$S_5: U \mid\equiv AS \mid\equiv ("0", aP, bP, U \xleftrightarrow{K} AS, U \xleftrightarrow{sk} AS)$

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

$S_6: U \mid\equiv AS \mid\equiv (U \xleftrightarrow{K} AS)$ **Goal 2**

$S_7: U \mid\equiv AS \mid\equiv (U \xleftrightarrow{sk} AS)$ **Goal 6**

According to A_5, the jurisdiction rule is applied to get:

$S_8: U \mid\equiv (U \xleftrightarrow{K} AS)$ **Goal 1**

According to A_7, the jurisdiction rule is applied to get:

$S_9: U \mid\equiv (U \xleftrightarrow{sk} AS)$ **Goal 5**

According to *Msg* **3**, the following is obtained:

$S_{10}: AS \lhd ("1", aP, bP, U \xleftrightarrow{K} AS, U \xleftrightarrow{sk} AS)_{U \xleftrightarrow{h(ID_U \| x)} AS}$

According to A_4, the message-meaning rule is applied:

$S_{11}: AS \mid\equiv U \mid\sim ("0", aP, bP, U \xleftrightarrow{K} AS, U \xleftrightarrow{sk} AS)$

According to A_2, the freshness-conjuncatenation rule is applied to get:

$S_{12}: AS \mid\equiv U \mid\equiv ("1", aP, bP, U \xleftrightarrow{K} AS, U \xleftrightarrow{sk} AS)$

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

$S_{13}: AS \mid\equiv U \equiv (U \xleftrightarrow{K} AS)$ **Goal 3**

$S_{14}: AS \mid\equiv U \mid\equiv (U \xleftrightarrow{sk} AS)$ **Goal 7**

According to A_6, the jurisdiction rule is applied to get

$S_{15}: AS \mid\equiv (U \xleftrightarrow{K} AS)$ **Goal 4**

According to A_8, the jurisdiction rule is applied to get

$S_{16}: AS \mid\equiv (U \xleftrightarrow{sk} AS)$ **Goal 8**

According to **Goal 1** – **Goal 8,** both *U* and *AS* know that a session key $sk = abP$ and an encrypted key $K = h(ID_U \| n)$ 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 $\vartheta'_U = BPW_U \oplus h(pw'_U)$. However, *A* cannot compute the corresponding σ_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 σ_U 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 γ for a newly generated bP. Then, *AS* could determine the attack by checking the correctness of γ. *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 β 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 $a \in Z_q^*$ and compute aP. However, *A* cannot compute α since *A* does not know w_U. Furthermore, *AS* can find the attack by checking the correctness of α. Suppose *A* intercepts the message m_1 and wants to impersonate *U* to *AS*, however in this case *A* cannot compute β without the value w_U. *U* can also identify the attack by checking the correctness of β. Therefore, the proposed protocol should withstand the impersonation attack.

6) Mutual authentication:

The proposed protocol allows that only *U* knows *U* 's secret key, $w_U = h(ID_U \| x)$, otherwise *AS*'s secret key x could generate the legal message β and γ. Then, *U* and *AS* can confirm m_2 and m_3 are sent by *AS* and *U* by checking the correctness of β and γ 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_{hash} : Time for executing a hash function;
T_{sym} : Time for executing a symmetric key cryptography;
T_{pm} : Time for executing an elliptic curve point multiplication;
T_{me} : Time for executing a modular exponentiation;
T_{fe} : 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 $10T_{me}+5T_{hash}+2T_{sym}$, $4T_{pm}+9T_{hash}+2T_{sym}$ and $4T_{pm}+9T_{hash}+2T_{sym}+T_{fe}$ 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
COMPUTATIONAL COST COMPARISONS

	User	***Authentication Server***	***Total***
Yang *et al.* protocol [11]	$4T_{me}+3T_{hash}+1T_{sym}\approx 0.08336$	$6T_{me}+2T_{hash}+1T_{sym}\approx 0.12144$	$10T_{me}+5T_{hash}+2T_{sym}\approx 0.20488$
Lee *et al.* protocol [14]	$2T_{pm}+5T_{hash}+1T_{sym}\approx 0.0414$	$2T_{pm}+4T_{hash}+1T_{sym}\approx 0.04108$	$4T_{pm}+9T_{hash}+2T_{sym}\approx 0.08248$
Proposed protocol	$2T_{pm}+5T_{hash}+1T_{sym}+T_{fe}\approx 0.0585$	$2T_{pm}+4T_{hash}+1T_{sym}\approx 0.04108$	$4T_{pm}+9T_{hash}+2T_{sym}+T_{fe}\approx 0.09958$

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

[1] M. Alzarouni, "The reality of risks from consented use of USB devices," in Proc. 4th in Proc. 4th Australian Information Security Management Conference, pp. 312-317, December 2006.

[2] C. Wu, W. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, no. 10, pp. 722-723, Oct. 2008.

[3] M. S. Hwang, and L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Trans. Consumer Electron.*, vol. 46, no. 1, pp. 28-30, Feb. 2000.

[4] 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.

[5] E. Yoon, E. Ryu, and K. 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.

[6] H. C. Hsiang, and W. K. Shih "Weaknesses and improvements of the Yoon-Ryu-Yoo remote user authentication scheme using smart cards," *Computer Communications*, Elsevier, vol. 32, no. 4, pp. 649-652, Mar. 2009.

[7] K. -A. Shim, "Security flaws in three password-based remote user authentication schemes with smart cards," *Cryptologia*, Taylor and Francis, vol. 36, no. 1, pp. 62-69, Jan. 2012.

[8] 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.

[9] D. -J. Kim, K. -W. Chung, and K. -S. Hong, "Person authentication using face, teeth and voice modalities for mobile device security", *IEEE Trans. Consumer Electron*, vol. 56, no. 4, pp. 2678-2685, Nov. 2010.

[10] S. -H. Lee, D. -J. Kim, and J. -H. Cho, "Illumination-robust face recognition system based on differential components," *IEEE Trans. Consumer Electron.*, vol. 58, no. 3, pp. 963-970, Aug. 2012.

[11] 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.

[12] C. Schnorr, "Efficient identification and signatures for smart cards," *Journal of Cryptology*, Springer, vol. 4, no. 3, pp. 161-174, 1991.

[13] B. Chen, C. Qin, and L. Yu, "A Secure Access Authentication Scheme for Removable Storage Media," *Journal of Information & Computational Science*, Binary Information Press, vol. 9, no. 15, pp. 4353-4363, Nov. 2012.

[14] 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.

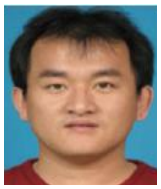
[15] C. -T. Li, and M. -S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, Elsevier, vol. 33, no. 1, pp. 1-5, Jan. 2010.

[16] D. Hankerson, S. Vanstone, and A. Menezes, "Guide to elliptic curve cryptography," *Lecture Notes in Computer Science*, 2004.

[17] A. 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.

[18] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in Proc. 2004 Int. Conf. Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, in *Lecture Notes in Computer Science*, pp. 523-540, 2004.
[19] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Computer Systems*, vol. 8, no. 1, pp. 18-36, Feb. 1990.
[20] J.-H. Lee and J.-M. Bonnin, "HOTA: Handover Optimized Ticket-based Authentication in Network-based Mobility Management," *Information Sciences*, Elsevier, vol. 230, pp. 64-77, May 2013.
[21] B. Schneier, *Applied cryptography protocols algorithms*, 2nd ed., Wiley, 1996.

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.