Enhanced User Authentication with Privacy for IoT-Based Medical Care System

Donghwan Ku and Hyunsung Kim

Abstract—With the rapid development of wireless communication technologies and the growing prevalence of smart devices, medical care system allows patients to receive medical treatments from the doctors in remote over wireless sensor networks via Internet of things (IoT). However, the medical data transmission through IoT concerns the privacy issue of patient. To solve this problem, Li et al. proposed an efficient user authentication and user anonymity scheme for medical care system over IoT and claimed their scheme is provably secure. This paper shows that Li et al.’s scheme has some security weaknesses and presents an enhanced scheme to solve the problems in Li et al.’s scheme. The proposed scheme has a bit of overhead in computation but provides security with privacy.

Index Terms—Internet of things, medical care system, privacy, authentication, public key cryptography.

I. INTRODUCTION

The use of information and communication technology for medical care system in hospitals and medical institutions enables medical personnel and patients to perform remote medical services via IoT [1]-[4]. IoT based wireless sensor networks (WSNs) has been getting considerable attention from a variety of applications, especially for medical care system. Ensuring good level of security is not a trivial task in medical care system as it uses wireless communications based on IoT. Threats and attacks are more diverse and often very large in scale [5]. The foundation of security is authentication of the legitimacy of system entity and confidentiality of data transmissions [6]-[9].

A diversity of user authentication schemes in WSNs have been presented [10]-[15]. Wong et al. introduced user authentication scheme for WSNs based on hash functions and XOR operations [10]. Tseng et al. provided cryptanalysis on Wong et al.’s scheme against replay, forgery and password guessing attacks [11]. Das suggested two factor authentication mechanism for WSNs in 2009 [12]. However, Li et al. pointed out the vulnerability of Das’s scheme to off-line password guessing, user impersonation, node impersonation and unknown user attacks [13]. Recently Liu and Chung proposed a bilinear pairing-based authentication scheme for IoT-based medical care system [14]. Unfortunately, Li et al. showed the security weaknesses in Liu and Chung’s scheme focused on password disclosure attack, replay attack, sensed data disclosure attack, sensed data forgery attack, stolen smart card (SC) attack and off-line password guessing attack [15].

This paper, first of all, provides Li et al. scheme’s security analysis focused on replay attack, stolen verifier attack, sensed data forgery attack and unfreshness of session key. For the remedy of Li et al.’s scheme, we also propose an enhanced user authentication scheme with privacy for IoT-based medical care system. Security analysis shows that the proposed scheme is more secure than the previous well-known schemes including Li et al.’s scheme.

II. REVIEW OF LI ET AL.’S SCHEME

This section briefly reviews Li et al.’s scheme [15]. Their scheme consists of five phases: setup phase, registration phase, login phase, verification phase and access control and encryption phase. For simple description, the terminology and notations used in Li et al.’s scheme in [15] and our scheme are summarized as follows:

- \( p, q \): Big prime numbers such that \( q | p - 1 \)
- \( F_p \): A finite field that has \( p \) elements
- \( E \): An elliptic curve defined over a finite field
- \( P \): A generator over \( E \)
- \( U \): The user
- \( TA \): The trusted authority
- \( S \): The sensor node
- \( S_0 \): The secret key of \( TA \)
- \( P_{pub} \): The public key of \( TA \)
- \( ID_i \): The identity of \( U_i \)
- \( ID_0 \): The identity of \( TA \)
- \( ID_S \): The identity of \( S \)
- \( PW_i \): The password of \( U_i \)
- \( h(\cdot) \): A one-way hash function
- \( \hat{e}(a, b) \): A bilinear pairing function using \( a \) and \( b \)
- \( a \): A private parameter generated by \( TA \)
- \( T_L \): The login time of \( U_i \)
- \( T_{now} \): The current time
- \( T_c \): The time limit on the legal access to \( S \) by the user \( U_i \)
- \( \Delta T \): The transmission delay
- \( m \): The sensed data collected from \( S \)
- \( \| \cdot \| \): The message concatenation
- \( \oplus \): The XOR operation

A. Setup Phase

The trusted authority, \( TA \) selects an elliptic curve \( E \) over \( F_p \) and a base point \( P_0 \) over the \( E \) and chooses a secure one-way
function $h(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^l$, where $p$ is a large prime such that $qp - 1$ for some great prime $q$ and $l$ means the length of the output. In addition, TA chooses the secret key $S_0 \in Z_q^*$ and computes its public key $P_{pub} = S_0 Q_0$. Finally, TA keeps $S_0$ securely and publishes $<E_0, q_0, P_{pub}, h(\cdot)>$ as system parameters.

### B. Registration Phase

The user registers with TA through a secure channel to be a legal user, and the details of registration phase are as follows:

1. Step 1: $U_i$ registers an authenticated identity $ID_i$ and password $PW_i$ with TA and chooses a random number $r$ for computing $r = h(ID_i) || PW_i)$.

2. Step 2: $U_i$ sends the registration request $<ID_i, R_i>$ to TA through a secure channel.

3. Step 3: TA checks whether $ID_i$ has been registered or not. If $ID_i$ has not been registered, TA computes $V_i = h(ID_i) || S_0(a)$ and $W_i = V_i \oplus R_i$. Then, TA stores the parameters $<W_i, a, E_i, q_0, P_{pub}, h(\cdot)>$ in $U_i$’s SC and issues the SC to $U_i$, where $a$ represents a private parameter generated by TA and all the sensor nodes of TA include a.

4. Step 4: $U_i$ computes $V_i = W_i \oplus h(ID_i) || PW_i)$ and stores $<V_i >$ in the SC. Finally, $U_i$’s SC contains the parameters $<Y_i, X_i, W_i, a, E_i, q_0, P_{pub}, h(\cdot)>$.

### C. Login Phase

The user inserts his/her SC into the device and inputs $ID_i$ and $PW_i$.

1. Step 1: SC checks $ID_i$ and $PW_i$ entered by $U_i$ against those stored in the SC. First, SC computes $r = X_i \oplus h(ID_i) || PW_i)$ and $V_i = W_i \oplus h(ID_i) || PW_i)$ and checks if $V_i = V_i$ in the SC. If it holds, SC executes Step 2. Otherwise, SC terminates this phase.

2. Step 2: SC generates a random number $\alpha$ and computes $M_i = \alpha X_i$, $N_i = \alpha P_{pub}$, $O_i = h(ID_i) || T_i)$ and $Q_i = h(N_i) \oplus (ID_i) || O_i)$ and sends $<M_i, Q_i, T_i>$ to TA through a public channel, where $T_i$ represents $U_i$’s login time to TA.

### D. Verification Phase

When $TA$ receives the login request $<M_i, Q_i, T_i>$ from $U_i$, $TA$ authenticates $U_i$ through the following steps:

1. Step 1: $TA$ checks if $T_{now} - T_i \leq \Delta T$. If yes, $TA$ executes Step 2. Otherwise, it means that the login time exceeds the transmission delay, and the login request will be rejected by $TA$.

2. Step 2: $TA$ computes $N_i = S_0 \times M_i$ and $(ID_i) || O_i) = Q_i \oplus h(N_i)$ and checks if user’s $ID_i$ is recorded by TA. If yes, $TA$ executes Step 3. Otherwise, the login request is denied by $TA$.

3. Step 3: $TA$ goes on to compute $V_i = h(ID_i) || S_0(a)$ by using the identity $ID_i$ and checks that the decrypted $O_i$ is the same as computed $O_i = h(ID_i) || V_i)$ in the SC. If no, the session is aborted by $TA$. Otherwise, $TA$ computes $E_i = h(b || TID_i)$ and $RM = h(N_i) \oplus (ID_i) || TID_i || E_i)$ sends the response message $<RM>$ to $U_i$ through a public channel, where $b$ represents a random number and $TID_i$ represents a temporary identity for $U_i$.

4. Step 4: $TA$ sends $<T_{now}, b, TID_i>$ to all of the sensor nodes $S$ via a secure channel and notifies $S$ that the temporary identity $TID_i$ is legal in the next access control and encryption phase.

### Step 5: When $U_i$ receives $<RM>$ from $TA$, $U_i$ authenticates $TA$ by computing $(ID_i) || TID_i || E_i)$ $\oplus h(N_i) \oplus RM$ and checks that the decrypted $ID_i$ is involved in $RM$ or not. If yes, $U_i$ confirms that $TA$ is legal and the parameters $TID_i$, $T_{now}$ and $E$ will be used in access control and encryption phase. Otherwise, $U_i$ ends this session. Note that $TID_i$ and $E$ must be kept secret by $U_i$ and temporarily stored into $U_i$’s SC until the end of the access control and encryption phase.

### E. Access Control and Encryption Phase

When the user $U_i$ is authenticated as legal, $U_i$ can legally access sensed data $m$ in $S$ within a permitted time $T_a$, and $U_i$ and $S$ perform the following steps:

1. Step 1: In this step, the executed operations are the same as Step 1 of the login phase.

2. Step 2: SC calculates $C = h(a || TID_i || T' \oplus h(E))$ and sends $<C, TID_i, T'>$ to $S$ through a public channel, where $T'$ represents a timestamp.

3. Step 3: Upon receiving $<C, TID_i, T'>$ from $U_i$, $S$ verifies if $T_{now} - T' \leq \Delta T$ and $T_{now} \leq T_a$. If yes, $S$ executes Step 4.

4. Step 4: $S$ computes $C' = h(a || TID_i || T' \oplus h(b || TID_i))$ by using $b$ transmitted by $TA$ and the temporary identity $TID_i$ of $U_i$ to examine whether $C = C'$. If yes, the validity of $U_i$ is authenticated by $S$, and the sensed data $m$ will be transmitted by $S$. If no, $S$ terminates this session.

5. Step 5: $S$ computes the session key $SK = h(E \oplus a \oplus T_a)$ and encrypts the sensed data by computing $M = m \oplus SK$. Then, $S$ sends $<M>$ to $U_i$ through a public channel. Note that the session key $SK$ provides a secure channel for protecting data transmission between $S$ and $U_i$.

6. Step 6: When $U_i$ receives $<M>$ from $S$, $U_i$ uses the parameters $<E_i, a, T_a>$ to calculate the session key $SK = h(E \oplus a \oplus T_a)$ and decrypts the sensed data $m$ by computing $m = M \oplus SK$.

Note that $SK$ should be frequently updated when $U_i$’s $T_a$ is expired. If so, $U_i$ returns to the login and verification phases for requesting a new $T_a$ with $TA$. Finally, a new $SK$ will be established and updated among $U_i$ and $S$ in the access control and encryption phase.

### III. SECURITY WEAKNESSES IN LI ET AL.’S SCHEME

In this section, we present security weaknesses of Li et al.’s scheme in [15]. The scheme is weak against replay attack, stolen verifier attack and sensed data forgery attack. Furthermore, it does not provide freshness of the session key.

#### A. Replay Attack

Replay attack is an attack of maliciously replaying valid data. In verification phase and access control and encryption phase, the scheme uses timestamp to prevent replay attack. However, timestamp requires continuous time synchronization among the devices. It is not appropriate in resource constraint IoT devices. Furthermore, replays in the permitted transmission delay $\Delta T$ is allowed. Therefore, there is less accuracy of detecting replay attacks. To avoid overheads of correspondence and ensure the correctness of protecting replay attacks, noble ways to prevent replay attacks should be presented.
B. Stolen Verifier Attack

The stolen verifier attack means that an attacker who stole verifier from the server impersonates a legitimate user. If someone who is an insider of trusted authority or invades the server gets the verifier, he/she can acquire the user’s information and impersonate the legal user. The attack can also pass the authentication by disguising him/herself as a legitimate user. Therefore, cryptographic scheme that does not store the verifier is required.

C. Sensed Data Forgery Attack

Sensed data forgery attack can be occurred in this scheme. It is an attack that an attacker catches and forges the data that is transmitted to receiver. In access control and encryption phase, S transmits the message $<M>$ to $U_i$ via public channel. At this time, although an attacker cannot decrypt this message, he/she can make a user confused by transmitting forged message with the same length of the normal message. That is why we have to accommodate the integrity check mechanism of the transmitted data.

D. Unfreshness of Session Key

An attack caused by unfresh session key can be occurred in this scheme. In access control and encryption phase, a session key is computed as $SK = h(E\oplus a \oplus T_u)$. However, $E$, $a$ and $T_u$ which consist the session key are not changed in a period of $T_u$. It may cause some problems related from the reuse of session key. It cannot provide future prevention. Therefore, a way that provides the freshness of the session key should be proposed.

IV. ENHANCED USER AUTHENTICATION SCHEME

In this section, we propose an enhanced user authentication scheme for IoT-based medical care system. The proposed scheme is based on Li et al.’s scheme and also consists of five phases: setup phase, registration phase, login phase, verification phase and access control and encryption phase. Fig. 1 shows the conceptual phases of the proposed scheme.

A. Setup Phase

TA preforms this system setup for the enhanced user authentication scheme. First of all, TA selects an elliptic curve $E$ over $F_p$ and the generator $P_0$ of $E$, where $p$ is a large prime such that $q|p - 1$ for any big prime number $q$. Also, TA selects a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. TA chooses a secure hash function $h(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^l$, where $l$ is the length of output, selects its own secret key $S_I \in Z_q^*$ and computes its public key $P_{pub} = \hat{e}(S_I, P_0)$. Finally, TA keeps $s$ securely and publishes $<E, q, P_0, P_{pub}, h(\cdot), \hat{e}(\cdot), ID_T, ID_S, S_I>$ as the system parameters.

B. Registration Phase

When a user $U_i$ wants to register with TA, this phase is necessary to be performed through a secure channel as follows:

Step 1: $U_i$ selects his/her identity $ID_i$ and sends it to TA. $S$ also sends its own identity $ID_S$ to TA.

Step 2: TA computes $V_i = h(ID_T || ID_S || S_I)$ and issues a SC to $U_i$, which stores $<E, q, P_0, P_{pub}, h(\cdot), \hat{e}(\cdot), ID_T, ID_S, V_i>$.

Step 3: $U_i$ computes $W_i = ID_T || PW_i$, $V_i = V_i \oplus W_i$ and $V_i = h(W_i)$ by using his/her identity $ID_i$ and password $PW_i$. After that, $U_i$ deletes $V_i$ from the memory of SC and writes $\{V_1, V_2\}$ on it.

<table>
<thead>
<tr>
<th>$U_i$</th>
<th>TA</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Registration Phase</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select $ID_i$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send $&lt;ID_i&gt;$ secure channel</td>
<td>Compute $V_i$</td>
<td>Store ${E, q, P_0, P_{pub}, h(\cdot), \hat{e}(\cdot), ID_T, ID_S, V_i}$ in a SC</td>
</tr>
<tr>
<td><strong>Issue a SC</strong> secure channel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input $ID_i$, PW_i, Compute $W_i$</td>
<td>Check $V_i = h(W_i)$</td>
<td>Choose $a$</td>
</tr>
<tr>
<td>Compute $X_i$, $V_i'$, $X_1$</td>
<td>Send $&lt;X_1, X_2, X_i&gt;$ public channel</td>
<td></td>
</tr>
</tbody>
</table>

| **Verification Phase** | | |
| Compute $ID_T$, $V_i$ | Check $h(ID_T || X_i || X_1) \equiv X_1$ | Choose $b$ |
| Compute $VK$, $X_1$, $X_5$ | | |
| **Access Control and Encryption Phase** | | |
| Input $ID_T$, PW_i, Compute $W_i'$ | Check $V_i' = h(W_i')$ | Choose $c$ |
| Compute $Y_i$, $V_2$ | Send $<Y_1, Y_2>$ public channel | |
| Check $h(ID_T || Y_i) \equiv Y_2$ | Choose $d$ | Compute $SK'$ |
| Compute $M$, $Y_i$, $Y_2$ | Send $<M, Y_i, Y_2>$ public channel |

C. Login Phase

When $U_i$ wants to access $S$, $U_i$ performs this login phase with TA. The details of this phase are as follows:

Step 1: $U_i$ inserts his/her SC into the reader and inputs $ID_i$ and $PW_i$. SC computes $W_i = ID_T || PW_i$ and checks whether $V_i$ equals to $h(W_i')$. If not, SC stops the phase.

Step 2: Otherwise, SC chooses a random number $a$ and computes $X_i = \hat{e}(a, P_0)$, $X_1 = \hat{e}(a, P_{pub}) \oplus ID_T$, $V_i' = V_i \oplus W_i'$ and $X_2 = h(ID_T || X_i || X_1 || V_i')$. And then, sends the message $<X_1, X_2, X_i>$ to TA through a public channel.

D. Verification Phase

After TA receives the message from $U_i$, TA verifies $U_i$ by following the steps:

Step 1: After TA receives the message $<X_1, X_2, X_i>$, it computes $ID_T' = X_2 \oplus \hat{e}(S_I, X_i)$ and $V_i = h(ID_T' || ID_S || S_I)$ and checks whether $h(ID_T' || X_1 || X_2 || V_i)$ equals to $X_3$. If not, $TA$ stops...
the request. Otherwise, TA chooses a random number $b$ and computes $VK=h(ID_2||ID_1||e(b, X_1))$. Then, TA sends the reply message $<X_3, X_5>$ to $U_i$.

Step 2: Upon receiving $<X_4, X_6>$ from TA, $U_i$ computes $VK'=h(ID_2||ID_1||e(a, X_i))$ and verifies whether $h(ID_1||ID_2||X_2||VK)$ equals to $X_5$. If not, this session is aborted.

Step 2: SC chooses a new random number $c$ and computes $Y_1=e(c, P_0)$ and $Y_2=h(ID_1||Y_1)$. Then, sends the message $<Y_1, Y_2>$ to $S$.

Step 3: Once $S$ receives the message $<Y_1, Y_2>$, it checks whether $h(ID_1||Y_1)$ equals to $Y_2$. If it holds, $S$ executes the next step.

Step 4: $S$ chooses a random number $d$ and computes $SK=h(ID_1||e(d, Y_1))$. Then $S$ sends the reply message $<M, Y_3, Y_4>$ to $U_i$.

Step 5: When $U_i$ receives the reply message $<M, Y_3, Y_4>$, $U_i$ computes the session key $SK'=h(ID_1||e(c, Y_3))$ and checks whether $h(SK'||M|Y_3)$ equals to $Y_5$. If not, the session is terminated. Otherwise, $U_i$ could get the sensed data by computing $m=M\oplus SK$.

V. ANALYSIS

This section provides security and performance analysis of the proposed scheme.

A. Security Analysis

The security of the proposed scheme is based on the one-wayness of the hash function and the bilinear pairing.

Password Guessing Attack: In the registration phase, the user’s password $PW_i$ is used in $W_i$ but not transmitted to TA. Although the privileged-insider of TA can obtain the registration message $ID_1$, it is unable to know the user’s sensitive password $PW_i$ because $PW_i$ related computation is performed only by the user. Note that deriving $PW_i$ from $V_i$ is equal to implementing the brute-force attack to crack the hash function. Moreover, during the login, verification and access control and encryption phases, neither SC nor the transmitted messages include user’s password $PW_i$. Hence, the proposed scheme eliminates the possibility of password guessing attack.

Replay Attack: The timestamps and random numbers are common countermeasures to prevent replay attack in the authentication process. Since the messages $<X_1, X_2, X_3>, <X_4, X_6>, <Y_1, Y_2, Y_5>$ and $<M, Y_3, Y_4>$ contain freshly generated random numbers $a, b, c$ and $d$. Furthermore, these random numbers are also embedded in the protected messages $X_1=e(a, P_0), X_2=e(a, P_{pub}), X_3=e(b, P_0), X_4=h(ID_1||ID_2||X_2||VK), Y_1=e(c, P_0), Y_2=h(ID_1||Y_2), Y_3=e(d, P_0)$. Thus, each participant first checks the freshness of the random numbers received and verifies whether these numbers are present in the transmitted messages. Hence, this design discards the possibility of replay attack in our proposed scheme.

Sensed Data Forgery Attack: In the access control and encryption phase of the proposed scheme, $S$ first authenticates $U_i$ by verifying whether $h(ID_1||Y_1)$ equals to $Y_2$. Due to the protection of using random number $c$, no one can forge a valid message $<Y_1, Y_2>$ to pass $S$’s verification. In addition, we assume that the attacker can intercept the response message $<M, Y_3, Y_4>$ and tries to generate a legitimate message $M'$ with fake sensed data $m'$. However, since $U_o$ does not know the secret parameter $SK$, it cannot generate the legitimate message $<M, Y_3, Y_4>$ due to Elliptic Curve Discrete Logarithm Problem (ECDLP). Thus, the proposed scheme could withstand the sensed data forgery attack.

Stolen Smart Card Attack: Suppose that $SC$ of $SC$ is lost or stolen. The attacker could get the stored parameters $\{E, q, P_0, P_{pub}, h(\cdot), e(\cdot), ID_7, ID_1, V_1, V_2\}$ and try to impersonate $U_i$ to successfully login to TA. $U_i$ cannot guess a candidate identity $ID_5$ and password $PW_i$ at the same time and compute $V_i=\gamma V_i W_i$ and $V_2=h(W_2)$. The way for $U_o$ to learn $PW_i$ is to find out the correct pair $(ID_5, PW_i)$ such that $V_2=h(W_i)$. In the proposed scheme, we assume the probability of guessing $ID_5$ composed of exact $l$ characters and $PW_i$ composed of exact $m$ characters is approximately $1/(2^{64l})$. This probability is negligible, and $U_o$ has no feasible way to derive $ID_5$ and $PW_i$ of $U_i$ in polynomial time.

User Anonymity: Based on the design of our proposed scheme, the excellent property of user anonymity can be guaranteed at every phase. We used masking for the real identity of $U_i$ via a public channel, and no attacker can compromise $U_i$’s real identity by launching security attacks. First, in the login phase, $U_i$’s real identity is included in $X_1=h(a, P_{pub})\oplus ID_1$. Thus, $U_o$ cannot reveal $ID_1$, $P_{pub}$ without using $S_{0i}$ to $X_1$ due to ECDLP. That is to say, all of the identities are transmitted in cipher format instead of plaintext, and these identities will be randomized at each new session. As a result, our proposed scheme can provide the property of user anonymity.

Session Key Security: Since the common session key $SK$ is only shared and established between $U_i$ and $S$, in order to establish a secure and authenticated channel for late successive transmission, $SK$ not only ensures confidentiality, but also achieves authenticity of participants and messages. Based on the design of session key $SK'=h(ID_1||e(c, P_0))$, $Y_2=h(SK'||M|Y_3)$ is used for verifying the integrity of the transmitted messages, whereas two random numbers are used for preventing possible replay and misuse service attacks. As a result, the session key security and data confidentiality can be provided in the proposed authentication scheme parameters.

B. Performance Analysis

This section provides performance analysis of the proposed scheme in terms of the computation complexities focused on
the login phase and the verification phase only. We thus present a performance evaluation to compare the proposed scheme to Li et al.’s scheme [15]. We present a comparison of the computational costs and measure the execution time. The computational analysis of an authentication scheme is generally conducted by focusing on operations performed by each party within the schemes. Therefore, for analysis of the computational costs, we concentrated on the operations that are conducted by the parties in the network: namely a user and a server. In order to facilitate the analysis of the computational costs, we define two notations, \( T_0 \) and \( T_1 \), where \( T_0 \) is for the time to execute a hash function and \( T_1 \) is the time to compute an ECC operation.

In addition, in order to achieve accurate measurement, we performed an experiment. This experiment was performed using the Crypto++ Library [16] on a system using the 64-bits Windows 7 operating system, 3.2 GHz processor, 4 GB memory, Visual C++ 2013 Software, the SHA-1 hash function, the AES symmetric encryption/decryption function, and the ECC-160 function. According to our experiment, \( T_0 \) is nearly 0.0002 seconds on average and \( T_1 \) is nearly 0.6 seconds on average.

### Table I. Performance Comparisons

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Overhead</th>
<th>User side</th>
<th>Server side</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al. in [15]</td>
<td>( T_0+5T_1 )</td>
<td>( 1T_0+4T_1 )</td>
<td>( 3T_0+9T_1 )</td>
<td></td>
</tr>
<tr>
<td>The proposed</td>
<td>( T_0+2T_1 )</td>
<td>( 3T_0+3T_1 )</td>
<td>( 5T_0+5T_1 )</td>
<td></td>
</tr>
</tbody>
</table>

Table I shows a comparative analysis of the computational cost among the related schemes. In addition, even though the proposed scheme has a bit of computational overhead than Li et al.’s scheme, the proposed scheme assures higher security, and affords resistance to the most well-known attacks, while providing functionality.

### VI. CONCLUSION

This paper gave a brief review of Li et al.’s authentication scheme in [15] and showed the scheme is vulnerable to some attacks. Furthermore, an enhanced scheme was proposed to repair the security flaws and improve the performance. The security analysis shows that the proposed scheme is secure against replay attack, stolen verifier attack, sensed data forgery attack and the other related attacks. Comparing the efficiency with the related scheme, the proposed scheme is comparable in terms of the computational overheads.

### REFERENCES


### Donghwan Ku

Donghwan Ku is a student at the Department of Cyber Security, Kyungil University, Korea from 2017. He is a member of KICOM and a researcher at the Research Center of Information Cross-over Security, Kyungbuk, Korea. He is working on devising security and privacy protocols for Internet of Things funded by the National Research Foundation of Korea. His research interests include cyber security, security protocol, ubiquitous computing, Internet of Things, network security, wireless communication security and cryptography.

### Hyunsung Kim

Hyunsung Kim is a full professor at the Department of Cyber Security, Kyungil University, Korea from 2012 and is also a visiting professor at the Department of Mathematical Sciences, Chancellor College, University of Malawi, Malawi from 2015. He received the M.S. and the Ph.D. degrees in computer engineering from Kyungpook National University, Korea, in 1998 and 2002, respectively. From 2000 to 2002, he worked as a senior researcher at Ditto Technology. He had been an associate professor from 2002 to 2012 with the Department of Computer Engineering, Kyungil University. His research interests include cryptography, VLSI, authentication technologies, network security and ubiquitous computing security.