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 to 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.

Manuscript received May 19, 2018; revised August 17, 2018. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

Donghwan Ku is with the Dept. of Cyber Security at Kyungil University, Kyungbuk, Korea (e-mail: imcman5@naver.com).

Hyunsung Kim is with the Dept. of Cyber Security at Kyungil University, Kyungbuk, Korea and the Mathematical Sciences Department at University of Malawi, Zomba, Malawi (e-mail: kim@kiu.ac.kr).

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_0: A generator over E
- U_i: 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_T: 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_u: The time limit on the legal access to S by the user U_i
- ΔT: The transmission delay
- m: The sensed data collected from S
- $\|$: 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 $q|p-1$ 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 $P_{pub}=S_0 \times P_0$. Finally, *TA* keeps S_0 securely and publishes $<E, q, P_0, P_{pub}, h(\cdot)>$ 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:

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

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

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 $<W_i, a, E, q, P_0, P_{pub}, h(\cdot)>$ 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.

Step 4: U_i computes $V_i=W_i \oplus h(ID_i||PW_i||r)$, $X_i=r \oplus h(ID_i||PW_i)$ and $Y_i=h(V_i||r||h(ID_i||PW_i))$ and stores $<X_i, Y_i>$ into the SC. Finally, U_i's SC contains the parameters $<Y_i, X_i, W_i, a, E, q, P_0, P_{pub}, h(\cdot)>$.

C. Login Phase

The user inserts his/her SC into the device and inputs ID_i and PW_i. Then, SC executes the following steps:

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

Step 2: SC generates a random number α and computes $M_i=\alpha \times P_0$, $N_i=\alpha \times P_{pub}$, $O_i=h(ID_i||V_i'||T_L)$ and $Q_i=h(N_i) \oplus (ID_i||O_i)$ and sends $<M_i, Q_i, T_L>$ to *TA* through a public channel, where T_L represents U_i's login time to *TA*.

D. Verification Phase

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

Step 1: *TA* checks if $T_{now}-T_L<\Delta T$. 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*.

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*.

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 $O_i'=h(ID_i||V_i||T_L)$. If no, the session is aborted by *TA*. Otherwise, *TA* computes $E=h(b \oplus TID_i)$ and $RM=h(N_i') \oplus (ID_i||TID_i||T_u||E)$ 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.

Step 4: *TA* sends $<T_u, b, TID_i>$ 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 $(ID_i||TID_i||T_u||E)=h(N_i) \oplus RM$ 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_u 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_u, and U_i and S perform the following steps:

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

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.

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

Step 4: S computes $C'=h(a||TID_i||T') \oplus h(h(b \oplus TID_i))$ 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.

Step 5: S computes the session key $SK=h(E \oplus a \oplus T_u)$ 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.

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

Note that SK should be frequently updated when U_i's T_u is expired. If so, U_i returns to the login and verification phases for requesting a new T_u 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 ΔT 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 the 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 $S_0 \in Z_q^*$ and computes its public key $P_{pub}=\hat{e}(S_0, P_0)$. Finally, *TA* keeps s securely and publishes $<E, q, P_0, P_{pub}, h(\cdot), \hat{e}(\cdot)>$ 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 $V_i=h(ID_T||ID_i||S_0)$ 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 $W_i=ID_i||PW_i$, $V_1=V_i \oplus W_i$ and $V_2=h(W_i)$ 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.

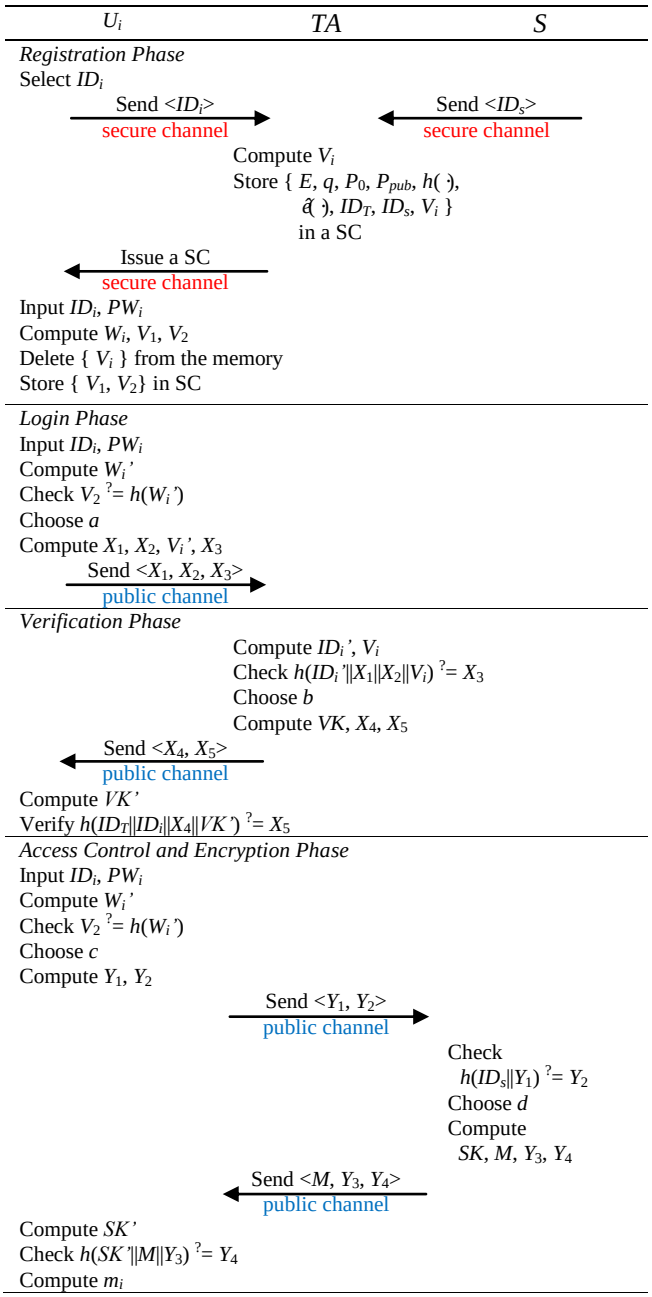

Fig. 1. The proposed user authentication scheme with privacy.

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 $W_i'=ID_i||PW_i$ and checks whether V_2 equals to $h(W_i')$. If not, SC stops the phase.

Step 2: Otherwise, SC chooses a random number a and computes $X_1=\hat{e}(a, P_0)$, $X_2=\hat{e}(a, P_{pub}) \oplus ID_i$, $V_i'=V_1 \oplus W_i'$ and $X_3=h(ID_i||X_1||X_2||V_i')$. And then, sends the message $<X_1, X_2, X_3>$ 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 $<X_1, X_2, X_3>$, it computes $ID_i' = X_2 \oplus \hat{e}(S_0, X_1)$ and $V_i=h(ID_T||ID_i'||S_0)$ and checks whether $h(ID_i'||X_1||X_2||V_i)$ equals to X_3. If not, *TA* stops

the request. Otherwise, *TA* chooses a random number *b* and computes $VK=h(ID_T||ID_i'||\hat{a}(b, X_1))$, $X_4=\hat{a}(b, P_0)$ and $X_5=h(ID_T||ID_i'||X_4||VK)$. Then, *TA* sends the reply message $<X_4, X_5>$ to U_i.

Step 2: Upon receiving $<X_4, X_5>$ from *TA*, U_i computes $VK'=h(ID_T||ID_i||\hat{a}(a, X_4))$ and verifies whether $h(ID_T||ID_i||X_4||VK')$ equals to X_5. If not, this session is aborted.

E. Access Control and Encryption Phase

After the successful login and authentication between U_i and *TA*, U_i can legally contact with *S* and get sensed data m_i from *S* securely.

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

Step 2: SC chooses a new random number *c* and computes $Y_1=\hat{a}(c, P_0)$ and $Y_2=h(ID_s||Y_1)$. And then, sends the message $<Y_1, Y_2>$ to *S*.

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

Step 4: *S* chooses a random number *d* and computes $SK=h(ID_s||\hat{a}(d, Y_1))$, $M=m_i \oplus SK$, $Y_3=\hat{a}(d, P_0)$ and $Y_4=h(SK||M||Y_3)$, where m_i is sensed data from *S*. 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 $SK'=h(ID_s||\hat{a}(c, Y_3))$ 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 $m_i=M \oplus SK$.

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 onewayness 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_s, 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_1 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. In the proposed scheme, we assume that an attacker U_a could eavesdrop all of the transmission messages $<X_1, X_2, X_3>$, $<X_4, X_5>$, $<Y_1, Y_2>$ and $<M, Y_3, Y_4>$ between U_i, *TA* and *S*. However, neither SC nor the transmission messages include U_i's password PW_i. Therefore, the proposed scheme could withstand the off-line password guessing attack.

Replay Attack: The timestamps and random numbers are common countermeasures to prevent replay attack in the authentication process. Since the messages $<X_1, X_2, X_3>$, $<X_4, X_5>$, $<Y_1, Y_2, Y_3>$ and $<M, Y_4, Y_5>$ contain freshly generated random numbers *a*, *b*, *c* and *d*. Furthermore, these random numbers are also embedded in the protected messages $X_1=\hat{a}(a, P_0)$, $X_2=\hat{a}(a, P_{pub}) \oplus ID_i$, $X_3=h(ID_i||X_1||X_2||V_i')$, $X_4=\hat{a}(b, P_0)$, $X_5=h(ID_T||ID_i'||X_4||VK)$, $Y_1=\hat{a}(c, P_0)$, $Y_2=h(ID_s||Y_1||Y_2)$, $Y_3=\hat{a}(d, P_0)$ and $Y_4=h(SK||M||Y_3)$. Thus, each participant first checks the freshness of the random numbers received and verifies whether the same random 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 $h(ID_s||Y_1)$ 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 U_a intercepts the response message $<M, Y_3, Y_4>$ and tries to generate a legitimate message *M'* with fake sensed data *m'*. However, since U_a 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 U_i is lost or stolen. The attacker U_a could get the stored parameters $\{E, q, P_0, P_{pub}, h(\cdot), \hat{a}(\cdot), ID_T, ID_s, V_1, V_2\}$ and try to impersonate U_i to successfully login to *TA*. U_a cannot guess a candidate identity ID_i and password PW_i at the same time and compute $V_1=V_i \oplus W_i$ and $V_2=h(W_i)$. The way for U_a to learn PW_i is to find out the correct pair (ID_i, PW_i) such that $V_2=h(W_i)$. In the proposed scheme, we assume the probability of guessing ID_i composed of exact *l* characters and PW_i composed of exact *m* characters is approximately $1/(2^{6l+6m})$. This probability is negligible, and U_a has no feasible way to derive ID_i 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 $X_2=\hat{a}(a, P_{pub}) \oplus ID_i$. Thus, U_a cannot reveal ID_i without using S_0 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 $SK=h(ID_s||\hat{a}(c, P_0)^d)$, $Y_4=h(SK||M||Y_3)$ 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_h and T_e, where T_h is for the time to execute a hash function and T_e 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_h is nearly 0.0002 seconds on average and T_e is nearly 0.6 seconds on average.

TABLE I: PERFORMANCE COMPARISONS

Overhead / Scheme	User side	Server side	Total
Li *et al*. in [15]	$2T_e+5T_h$	$1T_e+4T_h$	$3T_e+9T_h$
The proposed	$2T_e+2T_h$	$3T_e+3T_h$	$5T_e+5T_h$

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

[1] H. Kim, "Freshness-preserving non-interactive hierarchical key agreement protocol over WHMS," *Sensors*, vol. 14, pp. 23742-23757, August 2014.

[2] S. Shin, S. W. Lee, and H. Kim, "Authentication protocol for healthcare services over wireless body area networks," *International Journal of Computer and Communication Engineering*, vol. 5, no. 1, pp. 50-60, Jan. 2016.

[3] H. Kim, E. K. Ryu, and S. W. Lee, "Security considerations on cognitive radio based on body area networks for u-healthcare," *Journal of Security Engineering*, vol. 10, no. 1, pp. 9-20, Feb. 2013.

[4] K. Mtonga, E. J. Yoon, and H. Kim, "A pairing based authentication and key establishment scheme for remote patient monitoring systems," *Lecture Notes of the Institute for Computer Sciences*, vol. 135, pp. 79-89, August 2014.

[5] H. Ayatollahi and G. Shagerdi, "Information security risk assessment in hospitals," *Open Medical Informatics Journal*, vol. 11, pp. 37-43, Sep. 2017.

[6] H. Kim and S. W. Lee, "Enhanced novel access control protocol over wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 492-498, May 2009.

[7] J. Choi, Y. In, C. Park, S. Seok, H. Seo, and H. Kim, "Secure IoT framework and 2D architecture for end-to-end security," *Journal of Supercomputing*, pp. 1-15, March 2016.

[8] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A Security framework for the Internet of Things in the future internet architecture," *Future Internet*, vol. 9, no. 27, pp. 1-28, June 2017.

[9] S. Sridhar and S. Smys, "Intelligent security framework for IoT devices cryptography based end-to-end security architecture," in *Proc. the 2017 International Conference on Inventive Systems and Control*, 2017, pp. 1-5.

[10] K. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proc. the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2006, pp. 244-251.

[11] H. R. Tseng, R. H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in *Proc. the IEEE Globecom*, 2007, pp. 986-990.

[12] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086-1090, March 2009.

[13] C. T. Li, C. C. Lee, L. J. Wang, and C. J. Liu, "A secure billing service with two-factor user authentication in wireless sensor networks," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 8, pp. 4821-4831, August 2011.

[14] C. H. Liu and Y. F. Chung, "Secure user authentication scheme for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 59, pp. 250-261, April 2017.

[15] C. T. Li, T. Y. Wu, C. L. Chen, C. C. Lee, and C. M. Chen, "An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system," *Sensors*, vol. 17, no. 1482, pp. 1-18, June 2017.

[16] W. Dai, (Feb. 2018). *Crypto++ Library 5.6.1*. [Online]. Available: http://www.cryptopp.com

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 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.