WSN Integrated Authentication Schemes Based on Internet of Things

Tsung-Hung Lin[1], Cheng-Chi Lee[2,3], Chia-Hao Chang[1]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan
[2] Department of Library and Information Science, Fu Jen Catholic University, Taiwan
[3] Department of Photonics and Communication Engineering, Asia University, Taiwan
duke@ncut.edu.tw, cclee@mail.fju.edu.tw, sky206130@gmail.com

Abstract

The leader of the new generation of technology, namely the Internet of Things (IoT), is now already an indispensable part of many people's lives. In this research, we have analyzed the main communication models of IoT in terms of functional conformity and classified those models into two kinds. Then, for each of the two kinds, we have developed a new generalized model that is applicable in practically every possible case. Our first new model is to be applied in most common network environments where users directly connect to the authentication server, whereas the second new model is most applicable when users connect to some specified devices instead of a server. Besides the couple of new communication models we have just developed, in this paper we shall also present some integrated security authentication schemes that we have designed. Through some thorough security analyses, we have proven that our new schemes can resist a collection of attacks including the replay attack, privileged insider attack, stolen verifier attack, stolen smart card and smart card breach attack, impersonation, as well as the offline password guessing attack. In addition to the high security level, our new schemes also provide user anonymity with the communication efficiency unaffected by the security protection. On top of everything, our major concern and contribution is for every member of the worldwide IoT user community to enjoy the conveniences the industry 4.0 technologies bring without loss of privacy and personal interests.

Keywords: Internet of things, Smart card, Key agreement, Authentication, WSN

1 Introduction

With its unrivaled popularity and magical connecting power, the Internet has literally made our world a global village in real time. In recent years, the dramatically increasing prevalence of smartphones has further boosted the development of wireless networks and taken the accessibility of the Internet up to an even higher ground. At the same time, such popularity has also lead to the amazing advancement of mobile technologies, among which stands out the Internet of Things (IoT). The IoT concept involves all physical objects connected together within a network structure that are able to collect data and share them with one another. These objects have sensors embedded in them, so they can each collect data on their own [23]. Meanwhile, with RFID devices attached to them, these objects can be automatically identified and tracked. To go beyond that, through the existing Internet infrastructure or any wireless or wired system, these things can interoperate and in a sense communicate with one another. For example, the air-conditioning unit in an office can be remotely controlled by using a smartphone. This means we can monitor the office's temperature at anytime from anywhere, way beyond where a regular remote controller can reach. As the example shows, IoT can indeed bring convenience to our everyday lives. However, some serious information security issues can also arise when everything is linked to the Internet with all data open to the public. In a company with IoT, the devices connected to the Internet allow interactivity and open communication among themselves. In such a case, individuals with malicious intent could easily connect to the company's IoT and obtain some sensitive information from a particular device. Therefore, in order to benefit from the swift convenience IoT brings without getting troubled by the possible security problems, we need to have some efficient authentication schemes and security protection mechanisms that fit the IoT environment. As the smart office has nowadays come to be one of the mainstream IoT applications that attract most attention, in this paper we shall offer some novel IoT-based authentication schemes that we have designed especially for the smart office scenario.

Wireless Sensor Network (WSN) plays an important role in IoT systems because of its wide range of application. Instances include indoor temperature

[*] Corresponding Author: Cheng-Chi Lee; E-mail: cclee@mail.fju.edu.tw

DOI: 10.3966/160792642018081904008

measurement, humidity measurement, combustible gas monitoring [11], medical applications such as blood pressure measurement as well as heartbeat rate monitoring, and many more [15-16, 20]. In recent years, quite a number of WSN-related security authentication schemes have been developed. In 2013, Xue et al. offered five basic WSN authentication models [21]. Upon the basis of Xue et al.'s models, we have further divide IoT authentication models into eight types depending on the authentication steps. Then, the eight types merge into two groups: in one group the user first sends a message to the server, whereas in the other group the user's first message is sent to the device. With the similar steps in each authentication model omitted, we have classified the authentication models into four categories. Then, to strengthen the weaknesses of each model while retaining the advantages, we have combined the four kinds into two authentication models, which are entitled USD (the User-Server-Device model) and UDS (the User-Device-Server model) respectively. In the USD model, authentication steps begin when the user sends a message to the server, and the server then responds by passing the message on to the device. On the other hand, in the UDS model, authentication steps begin when the user sends a message to the device, and then the device passes the message on to the server. In this paper, we shall propose two authentication schemes. One is for applications where the UDS model is employed, and the other is for environments where the authentication steps fall into the USD family. The participants in our schemes include the user, IoT devices, and the IoT server.

The rest of this paper is organized as follows. In Section 2, some related schemes will be reviewed to set up the background for our new schemes. Then, in Section 3, we shall present the details of our proposed schemes. In Section 4, we shall offer the results of our security analyses. In Section 5, we shall show how our new schemes compare with some other similar schemes in terms of performance. Finally, Section 6 will serve to conclude this paper.

2 Related Works

WSN plays an important role in the IoT environment, and many researchers have developed various kinds of WSN authentication schemes [1-2, 4-5, 7-10, 12-13, 17-19, 21-22]. In 2006, Wong et al. [19] proposed a dynamic user authentication scheme for WSN. As a lightweight authentication scheme, their work uses only the hash function. In 2007, Tseng et al. [12] pointed out that Wong et al.'s work was vulnerable to the replay attack and the forgery attack and that the passwords could be stolen by any sensor nodes, and that the user could not change a password had it been set. To fix the above problems, Tseng et al. proposed an improvement on Wong et al.'s scheme. However, there are still security weaknesses in Tseng et al.'s scheme: it cannot resist the replay attack, forgery, and the man-in-the-middle attack, to name some. In 2009, Vaidya et al. [17] tried to mend the problems of Tseng et al.'s scheme and fixed the security flaws.

In 2009, Das [2] proposed a two-factor user authentication scheme in WSNs. Das indicated that Wong et al.'s scheme has a security flaw when many users own the same login-ID and that the scheme can be cracked by the stolen-verifier attack. Das used temporal credentials for authentication and claimed that his scheme could detect multiple logins of the same identity and was secure against a collection of attacks including stolen-verifier, password guessing, replay, and impersonation. However, Das's scheme is in fact still weak against offline password guessing, sensor node compromising, denial-of-service, and some other types of attacks. Later in the same year, Nyang et al. [10] offered an improvement on Das's scheme to mend the security weaknesses. Then, in 2010, Khan and Alghathbar [7] also proposed to improve Das's scheme. However, Vaidya et al. [18] found that Khan and Alghathbar's scheme also has security weaknesses. In fact, in order to enhance the security level of Das's scheme, many methods [1, 4-6, 17, 22] have been proposed since.

In 2012, Das et al. [3] proposed a dynamic password-based user authentication scheme for hierarchical WSNs. The scheme cannot be implemented in a realistic environment. In 2013, Turkanovic and Hölbl [13] proposed an improved dynamic password-based user authentication scheme. At the same time, Xue et al. [21] offered five basic authentication models and a temporal-credential-based mutual authentication and key agreement scheme for WSNs. In 2014, Turkanovic et al. [14], based on Xue et al.'s fifth authentication model, proposed a novel user authentication and key agreement scheme for heterogeneous ad hoc WSNs on the basis of the IoT notion. Turkanovic et al.'s scheme has a higher security level than Xue et al.'s. Their scheme uses only highly lightweight hash functions and XOR operations for encryption and decryption. However, we found that Turkanovic et al.'s scheme still has security weaknesses, as certain attacks such as off-line identity-password guessing, smart card theft, user impersonation, and sensor node impersonation can do the work of cracking the system. Figure 1 below shows the five authentication models presented by Xue et al. Please notice that these models do not cover all the possible scenarios in the entire IoT environment. To look at it from a more panoramic point of view, we offer eight IoT authentication models, each addressing a different pattern of authentication steps. Then, to strengthen the weaknesses of the models while retaining the advantages, we have had the eight models merged into two and then designed a secure IoT authentication scheme for each of the two models.

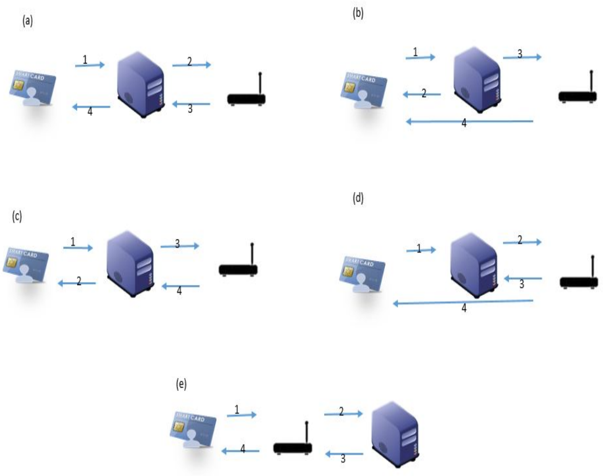

Figure 1. Xue et al.’s five authentication models

3 Our Proposed Schemes

In this section, we shall first lay out authentication models suitable for the IoT environment, and then we will present our new authentication schemes in detail.

3.1 IoT Model Definitions

Suppose a company has set up an IoT environment. The employees then use smart cards for authentication. After successfully passing the authentication procedure, they are logged on to the company’s IoT system and can use the company’s internal devices. As the authentication procedure may be set up differently, the IoT authentication procedure can be one of the eight basic authentication models shown in Figure 2. Please note that the eight authentication models are further classified into two types, namely Type A and Type B, as follows.

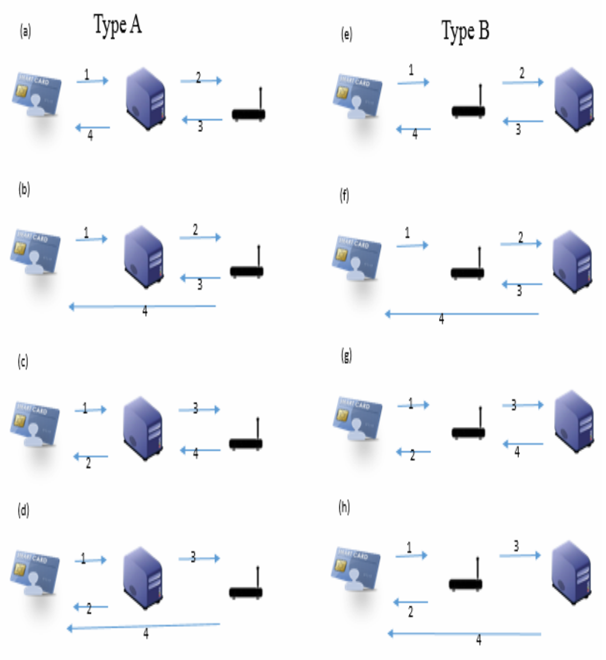

Figure 2. Eight basic authentication models

Type A. The user connects directly to the server and then connects to the device through the server.

Type B. The user connects directly to the device and then connects to the server through the device.

In Figure 2, model (a) of Type A is a traditional WSN-certified environment. The 3rd authentication step of type A-(a) shows that the authentication message gets transmitted from the device to the server. After that, the server generates a session key for the user and transmits it to the user for later message exchange. In order to reduce the number of transmissions needed, we make the device send a message to both the server and the user simultaneously, and then we change the step 3 and step 4 of model (a) to those of model (b)’s. Now we see that models (a) and (b) have the same authentication process. In other words, models (a) and (b) can now be merged into a new model named USD1, as shown in Figure 3.

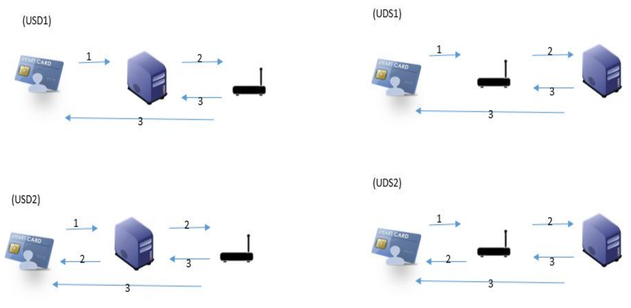

Figure 3. Four authentication models

In model (c) of Figure 2, the user’s authentication process does not involve any direct communication with the device. This way, the user and the device cannot confirm whether the session key is the same; that is to say, this model lacks an authentication check step between the user and the device. It needs to add a fifth step between the user and the device. Models (c) and (d) are different in step 4, shown as Type A-(c) and A-(d) of Figure 2. The device communicates directly with the user. Thus, the user-device pair cannot confirm each other’s session key in model (d) of Figure 2. This model lacks an authentication check step between the server and the devices. Now, what we do with models (c) and (d) is that we make both models strengthen each other by merging them into a new model named USD2, as shown in Figure 3.

In Type B, the session key for model (e) of Figure 2 is produced by the server. The 3rd step of type B-(e) shows that the server generates a session key and transmits it to the device. Then, the device transmits the session key to the user for data exchange. In order to reduce the number of transmissions needed, we adjust model (e) towards model (f) the same way we did earlier with models (a) and (b). Then models (e) and (f) have the same authentication process. Now, we merge models (e) and (f) into a new model named UDS1, as shown in Figure 3. This way, we can make sure that the server transmits the same message to both the user and the device at the same time without affecting each other.

In model (g) of Figure 2, the authentication process is unsafe because the user does not communicate with the server. When the user and the device finish communicating with each other, the device will start to exchange information with the server. This is pretty much the same thing that happens in the communication process for a credit card payment. However, model (g) lacks a server-to-user authentication step. On the other hand, in model (h), the server confirms whether the user that receives the session key is the right one. However, whether the device's session key is correct the server has no way to confirm. That is to say, model (h) of Figure 2 lacks a server-to-device authentication step. Now, just as we did earlier with models (c) and (d), we merge models g and h into a new model named UDS2, as shown in Figure 3.

Observing both USD1 and USD2 in Figure 3, we can see that USD2 includes all the moves that are taken in USD1. The only difference is that USD2 supports mutual authentication but USD1 does not. Therefore, we merge models USD1 and USD2 into a new model named USD. The USD model, as shown in Figure 4, allows mutual authentication between the user and IoT device, between the Iot device and IoT server, as well as between the Iot server and the user.

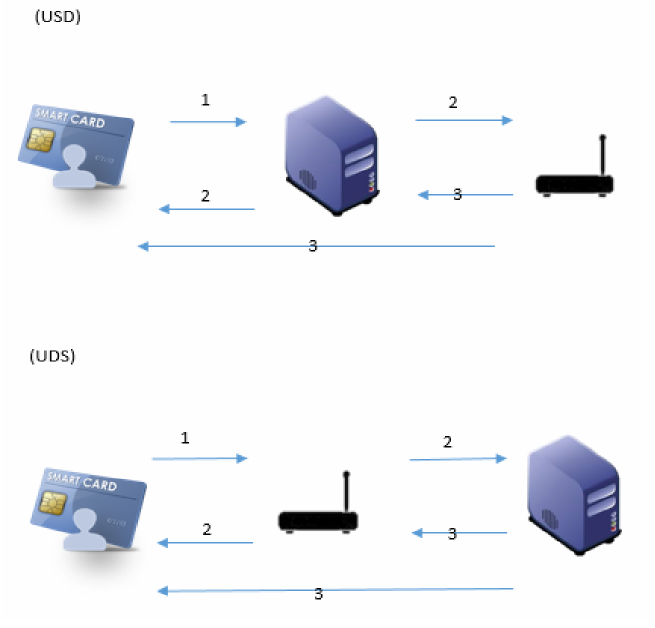

Figure 4. Authentication models USD and UDS

Similarly, observing both UDS1 and UDS2 in Figure 3, we can see that UDS2 includes all the moves that are taken in UDS1. The only difference is that UDS2 supports mutual authentication while UDS1 does not. Therefore, we merge models UDS1 and UDS2 into a new model named UDS, as shown in Figure 4.

The UDS model has three steps as follows.

Step 1: User sends login request to server.

Step 2: Server sends authentication message and session key to user and device.

Step 3: Device sends authentication message to server and at the same time sends session key authentication message to user to ensure session key is genuine.

The USD model has three steps as follows.

Step 1: User sends login request to device.

Step 2: Device sends both authentication information and user login request to server and at the same time sends authentication information to user for authentication.

Step 3: Server sends identity confirmation message and session key to device and at the same time sends authentication message and session key to user.

Now the eight authentication models we laid out earlier have been generalized into two types, which are referred to by the names USD and UDS. Upon the basis of both models, we shall propose our new IoT authentication schemes. Each of our proposed schemes contains five phases: pre-deployment, registration, login, authentication, and password-change. The notations used in our schemes are listed in Table 1. The five phases are detailed in the following subsections.

Table 1. Notations used in our schemes

Notation	Description
U_i	user
D_j	IoT device
S	IoT server
DID_j	identity of device
ID_i	identity of user
PW_i	password of user
X_{Dj}	secret key for device
X_{GWN}	secret key for server
x	random number of user that is stored in server's database
g	a primitive root of group Z_p^*
p	a large prime number
v_s, v_i	public key
$h(.)$	one-way hash function
$\|\|$	string concatenation operation
$\oplus$	bitwise xor operation
- - - - - - - -▶	secure channel
————▶	unsecure channel

3.2 Pre-deployment Phase

An IoT network environment typically includes a number of devices, and all devices must connect to the IoT master server to register in the pre-deployment phase. As mentioned earlier, we have laid out all possible IoT environment setup patterns and generalized them into two authentication models, UDS and USD, as shown in Figure 4. These two models have the same pre-deployment phase, which is detailed below and illustrated in Figure 5.

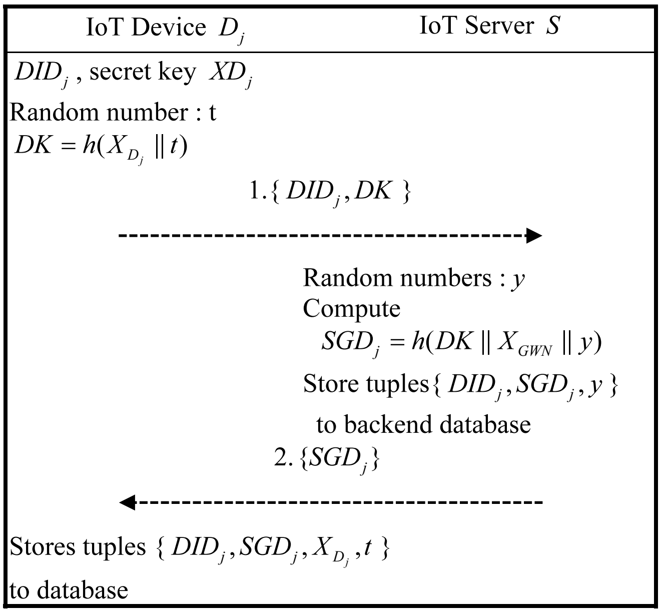

Figure 5. Pre-deployment phase

Step 1: IoT Device D_j owns identity DID_j and secret key XD_j. D_j chooses random number *t*, and computes $DK = h(X_{D_j} \| t)$. D_j sends the message tuple $\{DID_j, DK\}$ to the server through a secure channel.

Step 2: When IoT server *S* receives the message tuples, it chooses random number *y*, and computes $SGD_j = h(DK \| X_{GWN} \| y)$ using the server's secret key $X_{GWN.}$

Step 3: *S* computes SGD_j and stores tuple $\{ DID_j, SGD_j, y \}$ in the backend database. Then S sends message tuple $\{SGD_j\}$ to D_j through a secure channel.

Step 4: When D_j receives the message, it stores tuple $\{DID_j, SGD_j, S_{D_j}, t\}$ in the memory.

When the device pre-deployment phase is finished, the pre-deployment setup is complete.

3.3 Registration Phase

In the IoT working environment, all devices must be deployed first, and the user must register on demand. When the user's registration is finished, he/she receives a smart card to log on to the IoT network with. Both of our authentication models UDS and USD, as shown in Figure 4, have the same registration phase, which is detailed below and illustrated in Figure 6.

Step 1: User U_i selects his/her identity ID_i, password PW_i, and then random number X_{U_i}.

Step 2: User U_i computes $S_{U_i} = h(PW_i \| X_{U_i})$, and then sends message tuples $\{ID_i, S_{U_i}\}$ to GWN through a secure channel.

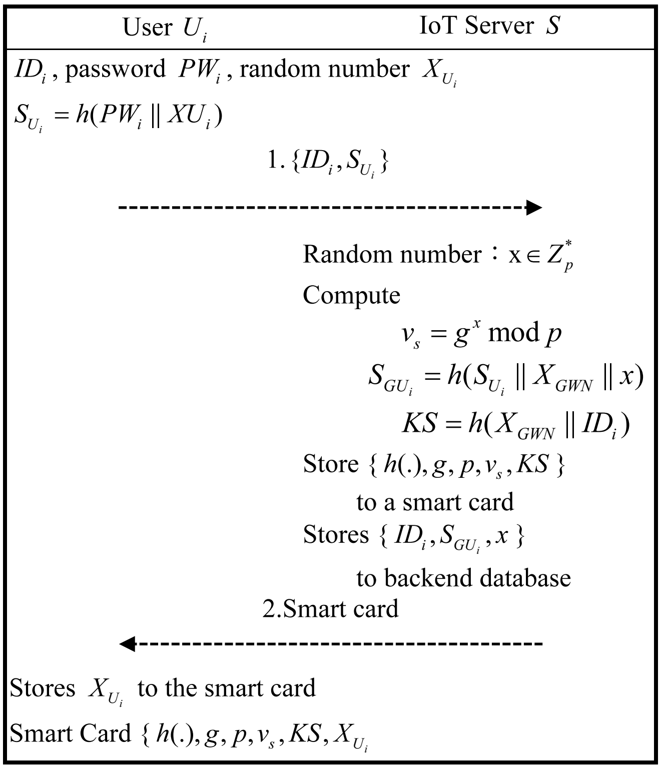
User U_i
IoT Server S
ID_i, password PW_i, random number X_{U_i}
$S_{U_i} = h(PW_i \| XU_i)$
1. $\{ID_i, S_{U_i}\}$
Random number：$x \in Z_p^*$
Compute
$v_s = g^x \bmod p$
$S_{GU_i} = h(S_{U_i} \| X_{GWN} \| x)$
$KS = h(X_{GWN} \| ID_i)$
Store { $h(.), g, p, v_s, KS$ }
to a smart card
Stores { ID_i, S_{GU_i}, x }
to backend database
2.Smart card
Stores X_{U_i} to the smart card
Smart Card { $h(.), g, p, v_s, KS, X_{U_i}$

Figure 6. Registration phase

Step 3: When *S* receives the message tuples, it selects a random number $x \in Z_p^*$ as the user's private key, and computes $v_s = g^x \bmod p$ as the public key.

Step 4: *S* computes $S_{GU_i} = h(S_{U_i} \| X_{GWN} \| x)$, $KS = h(X_{GWN} \| ID_i)$ using the secret key X_{GWN} from S.

Step 5: *S* records tuple $\{h(.), g, p, v_s, KS, X_{U_i}\}$ in the smart card and sends it via a secure channel to U_i. Finally, *S* stores message tuple $\{ID_i, S_{GU_i}, x\}$ in the backend database.

When the registration phase for U_i finishes, the registration setup is complete.

3.4 Login Phase

When a user employs an IoT device, he/she needs to send a login request to the server. The server authenticates the user's identity for logging on. Each of the two authentication schemes we propose in this paper has a different login phase, and the two login phases, one for USD and one for UDS. The USD and UDS login requests are described in the following subsections.

USD model. The login request for the USD scheme is a three-step procedure as follows:

Step 1: User U_i inserts his/her smart card into the reader and then inputs identity ID_i and password PW_i.

Step 2: The smart card generates a random number *a* $\in Z_p^*$ and computes $v_s = g^x \bmod p$, $k_i = (v_s)^a \bmod p (= g^{xa} \bmod p)$, $S_{U_i} = h(PW_i \| X_{U_i})$, $APW_i = S_{U_i} \oplus KS$,

$DH = v_i \oplus KS$, and $UID_i = ID_i \oplus KS$.

Step 3: User U_i sends message tuples $\{UID_i, APW_i, DH\}$ to the server through an unsecure channel.

UDS model. The login request for the UDS scheme is also a three-step procedure that goes:

Step 1: User U_i inserts his/her smart card into the reader and then inputs identity ID_i and password PW_i.

Step 2: The smart card generates a random number a $\in Z_p^*$ and computes $v_i = g^x \bmod p$, $k_i = (v_s)^a \bmod p (= g^{xa} \bmod p)$, $S_{U_i} = h(PW_i \| X_{U_i})$, $APW_i = S_{U_i} \oplus KS$, $DH = v_i \oplus KS$, $UID_i = ID_i \oplus KS$, and $UD_{ij} = h(DID_j \| ID_j \| k_i)$.

Step 3: User U_i sends message tuple $\{UID_i, APW_i, UD_{ij}, DH\}$ to the device through an unsecure channel.

3.5 Authentication Phase

After the user sends the login request, the authentication process enters the authentication phase. After the server verifies the identity of the user and that of the device, they produce a session key through each other's secret message. In our two authentication schemes, the authentication phases are different. The USD and UDS authentication phases are described in the following subsections.

USD model. The authentication phase for the USD scheme is as follows:

Step 1: When S receives a login request, it computes $ID_i = UID_i \oplus KS$, and then S uses the user's ID_i to retrieve S_{GU_i} and x from the database.

Step 2: computes $v_i = DH \oplus KS$, $k_i = v_i^x \bmod p$, and $S_{U_i} = APW_i \oplus KS$, and check $S_{U_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$. If $S_{GU_i} = h(S_{U_i} \| X_{GWN} \| x)$, S proceeds to the next step. Otherwise, the message is rejected.

Step 3: S uses DID_i to retrieve SGD_j and y from the database. Then, S selects random numbers b and c and then computes $SK = h(S_{U_i} \| SGD_j \| b \| c)$.

Step 4: S computes $RGD_j = b \oplus SGD_j$, $GDk_j = h(SGD_j \| b)$, $DSK_{ij} = SK \oplus b$ and $RSU_i = c \oplus S_{U_i}$, $US_i = h(S_{U_i} \| c)$, $USK_{ij} = SK \oplus c \oplus k_i$.

Step 5: Finally, S sends message tuple $\{RGD_j, GDk_j, DSK_{ij}\}$ to D_j and $\{USK_{ij}, RSU_j, US_i\}$ to U_i U_i simultaneously.

Step 6: When the smart card receives message tuple $\{USK_{ij}, RSU_j, US_i\}$, it computes $c = RSU_j \oplus SU_i$ and check $US_i \overset{?}{=} h(S_{U_i} \| c)$. If $US_i = h(S_{U_i} \| c)$, the smart card computes $SK = USK_{ij} \oplus c \oplus k_i$ and waits for D_j to send messages. Otherwise, the message is rejected. When D_j receives message tuple $\{RGD_j, GDk_j, DSK_{ij}\}$, it computes $b = RGD_j \oplus SGD_j$ and check $GDK_j \overset{?}{=} h(SGD_j \| b)$.

Step 7: If $GDK_j = h(SGD_j \| b)$, D_j selects random numbers d and e, and computes $SR = d \oplus b$, $DGK_j = h(SGD_i \| d)$, $SK = DSK_{ij} \oplus b$, $MSK = h(SK \| e)$, and $RSK = SK \oplus e$.

Step 8: D_j sends message tuple $\{DGK_j, SR\}$ to S and $\{MSK, RSK\}$ to the user.

Step 9: When S receives message tuple $\{DGK_j, SR\}$, it computes $d = SR \oplus b$, $DGK'_j = h(SGD_j \| d)$ and checks $DGK_j \overset{?}{=} DSK'_j$. If $DGK_j = DSK'_j$, S completely authenticates the device's identity. Otherwise, the message is rejected. When the smart card receives message $\{MSK, RSK\}$, it computes $e = RSK \oplus SK$ and checks $MSK \overset{?}{=} h(SK \| e)$. If $MSK = h(SK \| e)$, the smart card confirms that the session key for both the smart card and D_j are the same. Otherwise, the message is rejected.

After finishing the authentication phase, the user and the IoT device use $SK = h(S_{U_i} \| SGD_j \| b \| c)$ as their common session key in their subsequent communications.

UDS model. The authentication phase for the UDS scheme is as follows:

Step 1: When D_j receives the user's login request, it selects a random number b and computes $RGD_j = b \oplus SGD_j, UDS_{ij} = h(SGD_j \| b)$. D_j sends message tuple $\{UDS_{ij}\}$ to U_i and $\{UID_i$, APW_i, UD_{ij}, DH, DID_j, RGD_j, $UDS_{ij}\}$ to S.

Step 2: When the smart card receives message $\{UDS_{ij}\}$, it waits for D_j to send messages. When S receives message tuple $\{UID_i$, APW_i, UD_{ij}, D, DID_j, RGD_j, $UDS_{ij}\}$, it computes $ID_i = UID_i \oplus KS$ and uses ID_i to retrieve S_{GU_i} x from the database.

Step 3: S computes $v_i = DH_i \oplus KS$, $k_i = v_i^x \bmod p$, $S_{U_i} = APW_i \oplus KS$ and checks $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$. If $S_{GU_i =} h(S_{U_i} \| X_{GWN} \| x)$, S proceeds to the next step. Otherwise, the message is rejected.

Step 4: S checks $UD_{ij} \overset{?}{=} h(DID_j \| ID_i \| k_i)$. If $UD_{ij} = h(DID_j \| ID_i \| k_i)$, S proceeds to the next step. Otherwise, the message is rejected.

Step 5: S uses DID_j to retrieve SGD_j and y from the database. Then, S computes $b = RGD_j \oplus SGD_j$ and checks $UDS_{ij} \overset{?}{=} h(SGD_j \| b)$. If $UDS_{ij} = h(SGD_j \| b)$, S proceeds to the next step. Otherwise, the message is rejected.

Step 6: S selects random numbers c and d and computes $SK = h(S_{U_i} \| SGD_j \| b \| c)$, $SDR = d \oplus b$, $DSK = SK \oplus SGD_j \oplus b$, $SKD = h(SK \| d)$, $USK = SK \oplus k_i \oplus$, $RSU = c \oplus S_{U_i}$, and $SKU = h(SK \| c \| UDS_{ij})$.

Step 7: S sends message tuple $\{DSK, SKD, SDR\}$ to D_j and sends message tuple $\{USK, RKU, SKU\}$ to U_i.

Step 8: When D_j receives message $\{DSK, SKD, SDR\}$, it computes $d = SDR \oplus b$, $SK = DSK \oplus SGD_j \oplus b$ and checks $SKD \stackrel{?}{=} h(SK \| d)$. If $SKD = (SK \| d)$, D_j confirms that its session key is correct. When the smart card receives message $\{USK, RSU, SKU\}$, it computes $c = RSU \oplus S_{U_i}$, $SK = USK \oplus k_i \oplus c$ and checks $SKU \stackrel{?}{=} h(SK \| c \| UDS_{ij})$. If $SKU = h(SK \| c \| UDS_{ij})$, the smart card confirms that the user's session key is correct.

With the authentication phase completed, the user and the device can then use $SK = h(S_{U_i} \| SGD_j \| b \| c)$ as their common session key for later communication.

3.6 Password-change Phase

When the user wishes to change his/her password, he/she needs to apply to the server to initiate the password change phase. Our two authentication schemes share the same password change phase. The password change phase has the following steps:

Step 1: U_i inserts his/her smart card into the reader and then inputs identity ID_i and password PW_i.

Step 2: The smart card generates a random number a $\in Z_p^*$ and computes $v_I = g^a \bmod p$, $k_i = (v_s)^a \bmod p (= g^{xa} \bmod p)$, $S_{U_i} = h(PW_i \| X_{U_i})$, $APW_i = S_{U_i} \oplus KS$, $DH = v_i \oplus KS$, $UID_i = ID_i \oplus KS$, $S_{U_{new}} = h(PW_{new} \| X_{U_i})$, and $UNP = S_{U_{new}} \oplus KS$.

Step 3: The smart card sends message $\{UID_i, APW_i, UH, UNP\}$ to the server through an unsecure channel.

Step 4: When S receives a login request, it computes $ID_i = UID_i \oplus KS$ and uses ID_i to retrieve S_{GU_i} and x from the database.

Step 5: S computes $v_i = DH \oplus KS$, $k_i = v_i^x \bmod p$, and $S_{U_i} = APW_i \oplus KS$, and then checks $S_{GU_i} \stackrel{?}{=} h(S_{U_i} \| X_{GWN} \| X)$. If S_{GU_i} equals $h(S_{U_i} \| X_{GWN} \| X)$, then the next step is on. Otherwise, the request is rejected.

Step 6: S selects a random number x_{new} and computes $v_{new} = g^x \bmod p$, $DN = v_{new} \oplus KS$, $S_{U_{new}} = UNP \oplus KS$, $S_{GU_{new}} = h(S_{U_{new}} \| X_{GWN} \| x)$ and $PWC = h(v_{new} \| S_{U_{new}} \| S_{U_i})$.

Step 7: S sends message $\{DN, PWC\}$ to the user. Finally, the server updates S_{GU_i} to $S_{GU_{new}}$ and x to x_{new}.

Step 8: When the smart card receives message $\{DN, PWC\}$, it checks $PWC \stackrel{?}{=} h(v_{new} \| S_{new} \| S_{U_i})$. If $PWC = h(v_{new} \| S_{U_{new}} \| S_{U_i})$, the smart card updates v_s to v_{new}.

4 Security Analysis

In this section, we provide the security details of the proposed schemes. We shall show that our proposed schemes satisfy the user anonymity requirement and can resist the replay attack, privileged-insider attack, stolen-verifier attack, stolen smart card and smart card breach attack, impersonation attack, and off-line password guessing attack.

4.1 User Anonymity

User anonymity is a very important property of wireless communication authentication. During the login process we must confirm that the user's ID is secure. The user anonymity security of USD and that of UDS are shown as follows.

USD scheme. In USD's login phase, the user retrieves the server exchange key KS from the database to compute $UID_i = ID_i \oplus KS$ to protect the user's ID. The user sends UID_i to the IoT server through an unsecure channel. Even though illegal users may succeed in their attempts to steal communication message $\{UID_i, APW_i, DH\}$, they cannot work out ID_i from UID_i because illegal users do not have the knowledge of $KS = h(X_{GWN} \| ID_i)$. Even if an illegal user steals the smart card, he/she is still unable to obtain the user's identity because the user's identity is not stored in the smart card $\{h(.), g, p, v_s, KS, X_{U_i}\}$. Therefore, we claim that our USD authentication scheme satisfies the user anonymity requirement.

UDS scheme. In UDS's login phase, the user retrieves the server exchange key KS from the database to compute $UID_i = ID_i \oplus KS$ to protect the user's ID. The user sends UID_i to the IoT device through an unsecure channel. Even though illegal users may be able to steal message $\{UID_i, APW_i, UD_{ij}, DH\}$, they cannot derive ID_i from UID_i since they do not have the knowledge of $KS = h(X_{GWN} \| ID_i)$.. Moreover, illegal users cannot obtain the user's identity even if they can steal the smart card because the user's identity is not stored in the smart card. Therefore, we claim that our UDS authentication scheme satisfies the user anonymity requirement.

4.2 Replay Attack

A replay attack happens when a malicious party intercepts data during the authentication process and later impersonates a legitimate user by retransmitting the intercepted data in an attempt to log on to the remote IoT server. Here we shall discuss how our new schemes can resist the replay attack.

USD scheme. In order to resist the replay attack, the user generates a random one-time number a and computes $v_i = g^a \bmod p$, and $k_i = (v_s)^a \bmod p$ in USD's login phase. Then the user employs v_i and k_i to compute $S_{U_i} = h(PW_i \| X_{U_i})$, $APW_i = S_{U_i} \oplus KS$, $DH = v_i \oplus KS$, and $UID_i = ID_i \oplus KS$. Then the user sends message tuple $\{UID_i, APW_i, DH\}$ to the IoT server.

The server uses DH to compute v_i, and employs v_i and x to compute the communication common key $k_i = v_i^x \bmod p$. Then the server uses k_i and APW_i to compute S_{U_i}. The server then checks $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$. Since the user generates a different one-time random number every time, even if an illegal user somehow comes by message $\{UID_i, APW_i, DH\}$ during the authentication process, he/she still cannot use the data obtained to log on to the server. The reason is that for a user to successfully log on to the server, he/she has to not only verify $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$ but also compute the session key $SK = h(S_{U_i} \| SGD_j \| b \| c)$. Therefore, the replay attack will not take effect on our USD scheme.

UDS scheme. In order to resist the replay attack, the user generates a random one-time number a and computes $v_i = g^a \bmod p$ and $k_i = (v_s)^a \bmod p$ in UDS's login phase. Then the user employs v_i and k_i to compute $S_{U_i} = h(PW_i \| X_{U_i})$, $APW_i = S_{U_i} \oplus KS$, $DH = v_i \oplus KS$, $UID_i = ID_i \oplus KS$, and $UD_{ij} = h(DID_j \| ID_i \| k_i)$. Then the user sends message tuple $\{UID_i, APW_i, UD_{ij}, DH\}$ to the IoT device.

The device sends message tuple $\{UID_i, APW_i, UD_{ij}, DH, DID_j, RGD_j, UDS_{ij}\}$ to the IoT server, and then the server uses DH to compute v_i and employs v_i and x to compute the communication common key $k_i = v_i^x \bmod p$. Then the server uses k_i and APW_i to compute S_{U_i}. Then the server checks $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$. Since users generate a different one-time random number every time, even if an illegal user can steal message tuple $\{UID_i, APW_i, UD_{ij}, DH\}$ during the authentication process, he/she still cannot use it to log on to the IoT server. The reason is that besides verifying $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$, a legal user also has to compute the session key $SK = h(S_{U_i} \| SGD_j \| b \| c)$. Therefore, the replay attack will not take effect on our UDS scheme.

4.3 Privileged-insider Attack

A privileged-insider attack is when a remote server's administrator or any party with privilege obtains a legal user's password and then pretends to be the user and logs on to the server and utilizes the resources. Here we shall discuss how our new schemes can prevent an insider from abusing privileges.

USD scheme. In our USD scheme's authentication process, the user utilizes X_{U_i} to compute $S_{U_i} = h(PW_i \| X_{U_i})$, and then computes $APW_i = S_{U_i} \oplus KS$ and sends it to the IoT server. The server receives APW_i and checks $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$. The user's password does not appear in this authentication process. This way, even a privileged insider has no access to the user's password. Therefore, our USD scheme can resist the privileged-insider attack.

UDS scheme. In our UDS scheme's authentication process, the user utilizes X_{U_i} to compute $S_{U_i} = h(PW_i \| X_{U_i})$, and then the user computes $APW_i = S_{U_i} \oplus KS$ and sends it to the IoT device. The device sends message tuple $\{UID_i, APW_i, UD_{ij}, DH, DID_i, RGD_i, UDS_{ij}\}$ to IoT server, and then the server checks $S_{GU_i} \overset{?}{=} h(S_{U_i} \| X_{GWN} \| x)$. The user's password does not appear in this authentication process. This way, even a privileged insider has no access to the user's password. Therefore, our UDS scheme can resist the privileged-insider attack.

4.4 Stolen-verifier Attack

A stolen-verifier attack is when an illegal user steals any legal user's password from the remote server's authentication table and then pretends to be the legitimate user and logs on to the server. In our proposed schemes, the IoT server stores the user's message $\{ID_i, S_{GU_i}, x\}$ in the verifier table. Even if an illegal user has a way to steal the verifier table, the user's password is still under proper protection because the password is guarded by $S_{GU_i} = h(h(PW_i \| X_{U_i}) \| X_{GWN} \| x)$. Therefore, both the USD scheme and the UDS scheme can resist the stolen-verifier attack.

4.5 Stolen Smart Card and Smart Card Breach Attack

A stolen smart card and smart card breach attack happens when an unauthorized user steals a legal user's smart card and cracks it to obtain the information stored inside. Then the illegal user utilizes the smart card information to impersonate the legal user in order to log on to the IoT server and access resources. As mentioned earlier, in our proposed schemes USD and UDS, the message stored in the smart card includes $\{h(.), g, p, v_s, KS, X_{U_i}\}$. In other words, the user's password and identity are not stored in the smart card. Therefore, the illegal user cannot impersonate the legitimate user and log on to the server because of the lack of ID_i and $PW._i$ This is how our proposed schemes prevent the stolen smart card and smart card breach attack from doing harm.

4.6 Impersonation Attack

An impersonation attack takes effect when some illegal user intercepts communication contents during data transmission in the authentication process and later utilizes the data intercepted to get verified by the server. As mentioned earlier in subsection 4-2, in the login phase of both our proposed schemes, no illegal user can pretend to be a legitimate user since there is no session key $SK = h(S_{U_i} \| SGD_j \| b \| c)$. Moreover, even if the illegal user can steal a legitimate user's smart card and break the core to obtain the information inside, the illegal user still cannot user the smart card's information to impersonate the legitimate user without the legitimate user's password. Therefore, our proposed schemes USD and UDS can both resist the impersonation attack.

4.7 Off-line Password Guessing Attack

An off-Line password guessing attack is most likely to work when the attacker has some data intercepted and uses the data to lead the way to the correct password. In our new schemes USD and UDS, the password of user U_i is protected by X_{U_i}. User U_i utilizes X_{U_i} and k_i to compute $S_{U_i} = h(PW_i \| X_{U_i})$ and $APW_i = S_{U_i} \oplus KS$, respectively. An illegal user might be able to obtain APW_i when it is being transmitted; however, the illegal users does not have X_{U_i}, which is stored in the smart card, and k_i, which can only be derived by computing $(v_s)^a \bmod p$. Meanwhile, as described in subsection 4-5, if an illegal user steals the smart card and cracks it to obtain the information stored inside, he/she still cannot use it to obtain the password.

5 Performance Analysis

In this section, we shall discuss how our new schemes compare with some other schemes in terms of security and efficiency performance. Among the schemes compared, Turkanovic et al.'s work, which also focuses on IoT applications, shares the most similarities with our schemes such as focusing on IoT applications and using smart cards for authentication. Hence, we should pay more attention to the difference between our schemes and Turkanovic et al.'s. Table 2 is a list of the security properties of all the schemes included in the comparison. Obviously, our new schemes have a higher security level than the other schemes, Turkanovic et al.'s work included.

Table 2. Security comparison among similar schemes

Security property	Ours	[14]	[21]	[3]	[13]	[22]	[1]	[7]	[4]	[18]	[6]	[5]
User Anonymity	✓	×	×	×	×	×	✓	✓	✓	✓	✓	✓
Replay Attack	✓	✓	✓	✓	✓	×	✓	✓	×	×	✓	×
Privileged-Insider Attack	✓	✓	✓	×	✓	✓	×	✓	✓	×	×	✓
Stolen-Verifier Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Stolen Smart Card and Smart Card Breach Attack	✓	×	✓	×	✓	×	×	×	×	×	×	×
Impersonation Attack	✓	×	✓	×	✓	✓	×	✓	×	×	×	×
Off-Line Password Guessing Attack	✓	×	×	×	✓	✓	×	✓	✓	×	✓	✓

On the other hand, for the performance comparison, the time consumption of the computations needed in each scheme's login phase as well as authentication phase has been estimated, and the results of the efficiency performance comparison we have made are listed in Table 3. We can see that the computation cost of Turkanovic et al.'s [14] scheme is much higher than those of our proposed schemes.

6 Conclusion

In this paper, we proposed a couple of novel authentication schemes to be applied in IoT environments. We have analyzed how IoT systems are put together and laid out eight basic models of IoT authentication system setup, and then we have merged them into two generalized models USD and UDS. Then, based on these two generalized models, we have developed our new schemes that utilize only a one-way hash function and some exponential operations for authentication. Our new schemes satisfy the requirement of user anonymity and can resist the replay attack, privileged insider attack, stolen verifier attack, stolen smart card and smart card breach attack, impersonation attack, as well as off-line password

Table 3. Performance comparison among similar schemes

Authentication scheme	User	Device\Sensor	Server\GWN
Our schemes	$3T_h + 2T_E$ $3T_h + 2T_E$	$3T_h$ $2T_h$	$5T_h + 1T_E$ $6T_h + 1T_E$
Turkanovic et al. [14]	$7T_h$	$5T_h$	$7T_h$
Xue et al. [21]	$7T_h$	$6T_h$	$13T_h$
Das et al. [3]	$5T_h + 1T_s$	N/A	$2T_h + 1T_s$
Turkanovic and Hölbl [13]	$4T_h + 1T_s$	N/A	$1T_h + 1T_s$
Yeh et al. [22]	$1T_h + 2T_s$	$3T_h + 2T_s$	$4T_h + 4T_s$
Chen and Shih [1]	$4T_h$	$1T_h$	$5T_h$
Khan and Alghathbar [7]	$4T_h$	$2T_h$	$6T_h$
Fan et al. [4]	$7T_h$	$2T_h$	$8T_h$
Vaidya et al. [18]	$6T_h$	$2T_h$	$5T_h$
He et al. [6]	$5T_h$	$1T_h$	$5T_h$
Huang et al. [5]	$4T_h$	$1T_h$	$6T_h$

T_h - time for a one-way hash function; T_E - time for an exponential operation; T_S -time for a symmetric encryption/decryption operations;

guessing attack. The results of our security comparison and efficiency performance comparison among similar schemes have shown that our new schemes have the highest security level and the lowest computation cost of them all.

References

[1] T. H. Chen, W. K. Shih, A Robust Mutual Authentication Protocol for Wireless Sensor Networks, *ETRI Journal*, Vol. 32, No. 5, pp. 704-712, October, 2010.

[2] 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.

[3] A. K. Das, P. Sharma, S. Chatterjee, J. K. Sing, *A Dynamic Password-based User Authentication Scheme for Hierarchical Wireless Sensor Networks, Journal of Network and Computer Applications*, Vol. 35, No. 5, pp. 1646-1656, September, 2012.

[4] R. Fan, L. D. Ping, J. Q. Fu, X. Z. Pan, A Secure and Efficient User Authentication Protocol for Two-tiered Wireless Sensor Networks, *Second Pacific-Asia Conference on Circuits, Communications and System*, Vol. 1, Beijing, China, 2010, pp. 425-428.

[5] H. F. Huang, Y. F. Chang, C. H. Liu, Enhancement of Two-factor User Authentication in Wireless Sensor Networks, *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Darmstadt, Germany, 2010, pp. 27-30.

[6] D. He, Y. Gao, S. Chan, C. Chen, J. Bu, An Enhanced Two-factor User Authentication Scheme in Wireless Sensor Networks, *Ad Hoc & Sensor Wireless Networks*, Vol. 10, No. 4, pp. 361-371, 2010.

[7] M. K. Khan, K. Alghathbar, Cryptanalysis and Security Improvements of "Two-factor User Authentication in Wireless Sensor Networks", *Sensors*, Vol. 10, No. 3, pp. 2450-2459, March, 2010.

[8] S. Kumari, M. K. Khan, M. Atiquzzaman, User Authentication Schemes for Wireless Sensor Networks: A Review, *Ad Hoc Networks*, Vol. 27, pp. 159-194, April, 2015.

[9] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, M. K. Khan, A User Friendly Mutual Authentication and Key Agreement Scheme for Wireless Sensor Networks using Chaotic Maps, *Future Generation Computer Systems*, Vol. 63, pp. 56-75, October, 2016.

[10] D. Nyang, M. K. Lee, *Improvement of Das's Two-factor Authentication Protocol in Wireless Sensor Networks*, http://eprint.iacr.org/20091631.pdf.

[11] A. Somov, A. Baranov, D. Spirjakin, A. Spirjakin, V. Sleptsov, R. Passerone, Deployment and Evaluation of A Wireless Sensor Network for Methane Leak Detection, *Sensors and Actuators A: Physical*, Vol. 202, pp. 217-225, November, 2013.

[12] H. R. Tseng, R. H. Jan, W. Yang, An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks, *IEEE Global Telecommunications Conference, GLOBECOM '07*, Washington, DC, 2007, pp. 986-990.

[13] M. Turkanovic, M. Hölbl, An Improved Dynamic Password-based User Authentication Scheme for Hierarchical Wireless Sensor Networks, *Elektronika Ir Elektrotechnika*, Vol. 19, No. 6, pp. 109-116, June, 2013.

[14] M. Turkanovic, B. Brumen, M. Hölbl, A Novel User Authentication and Key Agreement Scheme for Heterogeneous Ad Hoc Wireless Sensor Networks, Based on the Internet of Things Notion, *Ad Hoc Networks*, Vol. 20, pp. 96-112, September, 2014.

[15] M. V. Ramesh, Design, Development, and Deployment of A Wireless Sensor Network for Detection of Landslides, *Ad Hoc Networks*, Vol. 13, pp. 2-18, February, 2014.

[16] Z. Sun, P. Wang, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan, I. F. Akyildiz, Bordersense: Border Patrol through Advanced Wireless Sensor Networks, *Ad Hoc Networks*, Vol.

9, No. 3, pp. 468-477, May, 2011.

[17] B. Vaidya, J. S. Silva, J. J. P. C. Rodrigues, Robust Dynamic User Authentication Scheme for Wireless Sensor Networks, *Proceedings of the 5th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2Swinet)*, Canary Islands, Spain, 2009, pp. 88-91.

[18] B. Vaidya, D. Makrakis, H. T. Mouftah, Improved Two-factor User Authentication in Wireless Sensor Networks, *IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, Niagara Falls, Canada, 2010, pp. 600-606.

[19] K. H. M. Wong, Y. Zheng, J. Cao, S. Wang, A Dynamic User Authentication Scheme for Wireless Sensor Networks, *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Vol. 01, Taichung, Taiwan, 2006, pp. 1-8.

[20] M. Winkler, K. D Tuchs, K. Hughes, G. Barclay, Theoretical and Practical Aspects of Military Wireless Sensor Networks, *Journal of Telecommunications and Information Technology*, Vol. 2, pp. 37-45, June, 2008.

[21] K. Xue, C. Ma, P. Hong, R. Ding, A Temporal-credential-based Mutual Authentication and Key Agreement Scheme for Wireless Sensor Networks, *Journal of Network and Computer Applications*, Vol. 36, No. 1, pp. 316-323, January, 2013.

[22] H. Yeh, T. Chen, P. Liu, T. Kim, H. Wei, *A Secured Authentication Protocol for Wireless Sensor Networks using Elliptic Curves Cryptography*, Sensors, Vol. 11, No. 5, pp. 4767-4779, May, 2011.

[23] H. C. Hsieh, K. D. Chang, L. F. Wang, J. L. Chen, H. C. Chao, ScriptIoT: A Script Framework for and Internet-of-Things Applications, *IEEE Internet of Things Journal*, Vol. 3, No. 4, pp 628-636, August, 2016.

Biographies

Tsung-Hung Lin received the M.S. and Ph.D. degrees in Computer Science and Information Engineering from the National Chung Cheng University, Taiwan, R.O.C, in June 1993 and July 2005. He joined the faculty of Department of Computer Science and Information Engineering at National Chin-Yi University of Technology, Taiwan, R.O.C., as an assistant professor in August 2005, and has been as an associate professor in August 2011. His research interests include wireless personal area network, wireless sensor network, Internet of Things, Big data applications, machine learning, Steganography, and multimedia security.

Cheng-Chi Lee received the Ph.D. degree in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently a Professor with the Department of Library and Information Science at Fu Jen Catholic University. Dr. Lee is currently as an editorial board member of International Journal of Network Security, Journal of Computer Science, Cryptography, and International Journal of Internet Technology and Secured Transactions. He also served as a reviewer in many SCI-index journals, other journals, other conferences. His current research interests include data security, cryptography, network security, mobile communications and computing, wireless communications. He is a member of IEEE, the Chinese Cryptology and Information Security Association (CCISA), the Library Association of The Republic of China, and the ROC Phi Tau Phi Scholastic Honor Society.

Chia Hao Chang was born in Taiwan 1991. He received the M.S. degree in Computer Science and Information Engineering from National Chin-Yi University of Technology, Taiwan, in 2016. His research interests include Internet of Things, Steganography, and multimedia security.