*Sensors* **2011**, *11*, 4767-4779; doi:10.3390/s110504767

OPEN ACCESS

*sensors*

**ISSN 1424-8220**

www.mdpi.com/journal/sensors

*Article*

# A Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography

**Hsiu-Lien Yeh [1,*], Tien-Ho Chen [2], Pin-Chuan Liu [2], Tai-Hoo Kim [3] and Hsin-Wen Wei [4]**

[1] Institute of Information System and Applications, National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, HsinChu, 30013, Taiwan

[2] Department of Computer Science , National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, HsinChu, 30013, Taiwan; E-Mails: riverchen@ rtlab.cs.nthu.edu.tw (T.-H.C.); flash@itri.org.tw (P.-C.L.)

[3] Department of Multimedia Engineering, Hannam University, No.133 Ojeong-dong, Daeduk-gu, Daejeon 306-791, Korea; E-Mail: taihoonn@empas.com

[4] Institute of Information Science, Academia Sinica Taipei, Taiwan; E-Mail: hwwei@iis.sinica.edu.tw

* Author to whom correspondence should be addressed; E-Mail: s9865805@m98.nthu.edu.tw; Tel.: +886-3-573-3550.

*Received: 30 March 2011; in revised form: 25 April 2011 / Accepted: 26 April 2011 / Published: 2 May 2011*

**Abstract:** User authentication is a crucial service in wireless sensor networks (WSNs) that is becoming increasingly common in WSNs because wireless sensor nodes are typically deployed in an unattended environment, leaving them open to possible hostile network attack. Because wireless sensor nodes are limited in computing power, data storage and communication capabilities, any user authentication protocol must be designed to operate efficiently in a resource constrained environment. In this paper, we review several proposed WSN user authentication protocols, with a detailed review of the M.L Das protocol and a cryptanalysis of Das' protocol that shows several security weaknesses. Furthermore, this paper proposes an ECC-based user authentication protocol that resolves these weaknesses. According to our analysis of security of the ECC-based protocol, it is suitable for applications with higher security requirements. Finally, we present a comparison of security, computation, and communication costs and performances for the proposed protocols. The ECC-based protocol is shown to be suitable for higher security WSNs.

**Keywords:** authentication; security; ECC; wireless sensor network

## 1. Introduction

As wireless communication technology has matured, the deployment of Wireless Sensor Networks (WSNs) has become more common. Wireless communication is a natural fit for sensor networks for the following reasons: it reduces the cost of infrastructure, allowing sensor networks to be deployed in areas that were once cost prohibitive and it allows a greater range of applications than fixed location sensor networks [1]. WSNs are now providing economical solutions in a host of diverse industries: electric utilities use WSNs for remote voltage monitoring, museums use WSNs for humidity monitoring and control, health care providers use WSNs for patient monitoring and notification, and they are in use in the military. Other applications include environment tracking and habitat monitoring, *etc.* [2-5].

A key requirement for WSN is user authentication [6,7]. The client devices (remote wireless sensor nodes) need to be authenticated before being allowed to join the WSN and have access to the WSN's resources. To date, most user authentication methods have focused on protocol implementations in the network and link layers. Accordingly, we propose an efficient protocol implementation in the WSN application layer. It should be noted that, in order to limit power consumption by sensor nodes and to overcome limitations in computation capacity, user authentication in a WSN is typically done in dedicated gateway node (GW-node) [8].

Sastry and Wagner [9] proposed a security enhancement using access control lists (ACL's) in the GW node. In addition to verifying a client's identity and arranging the nearest sensor node, an ACL would be maintained. The ACL would be limited to 255 entries. Watro *et al.* [10] proposed a complex mathematical method for user authentication employing RSA and Diffie-Hellman algorithms to calculate an encrypted public key (TinyPK authentication), but this protocol is open to hostile attack by a user masquerading as a sensor node (spoofing). Wong *et al.* [11] proposed a less complex, light-weight, dynamic user authentication method using a hash-based protocol. Their method recommended using the security features of the IEEE 802.15.4 MAC sublayer. Das [12] and Tseng *et al.* [13] pointed out that both Watro's and Wong's user authentication methods were vulnerable to stolen-verifier, replay, and forgery attacks (made possible by allowing multiple users with a single login ID). Das [12] proposed a two factor method of user authentication. This method is designed to protect against the aforementioned stolen-verifier, replay, and forgery attacks. Tseng *et al.* [13] further pointed out that Wong's method was vulnerable to stolen passwords and that Wong's method prevented users from freely changing their password. Tseng *et al.* proposed an enhanced user authentication method that is design to prevent the various attacks and to reduce the vulnerability to stolen passwords. Khan *et al.* [14,15] and Chen *et al.* [16] reviewed the Das two factor method and found additional security issues. Chen *et al.* [16] proposed a more secure and robust two-factor user authentication in WSNs. Unfortunately, we find that the Chen *et al.* proposal fails to provide a secure method for updating user passwords and is vulnerable to the insider attack problem.

To address all of the issues raised in the above studies, we propose a novel user authentication protocol for wireless sensor networks, using Elliptic Curves Cryptography (ECC) and smart cards. Our proposal addresses the key security issues, while at the same time reducing computational load requirements. The remainder of this paper is organized as follows: in Section 2, we review the Das method and perform a detailed cryptanalysis of that method; next we present the ECC-based

authentication protocol (EAP) for WSNs in Section 3. In Section 4, we present a security and performance analysis of the related protocols. Then, in Section 5, we provide some concluding remarks.

## 2. Related Works

### *2.1. Review of Das' Scheme*

This section provides a brief review of the Das method and analyzes its protocol. Before this analysis we first summarize in Table 1 the notations used throughout this paper and their corresponding definitions.

**Table 1.** Notations.

| Symbol | Definition |
|---|---|
| U | A user |
| ID | A user's identity |
| PW | A user's password |
| DID | A user's dynamic login identity |
| GW-node | Gateway node of WSN |
| $S_n$ | Nearest sensor node of WSN |
| h(.) | A secure one-way hash function |
| $x_a$ | A permanent secret parameter generated securely by the GW-node and stored in some defined sensor nodes before deploying the WSN |
| K | A symmetric key of GW-node which shared between the GW-node, users and the sensor nodes |
| $\|\|$ | A string concatenation operation |
| $\oplus$ | A string XOR operation |
| $\Rightarrow$ | A secure channel |
| $\rightarrow$ | A public channel |

Das' protocol involves the registration phase, login phase and verification phase, and can be briefly described as follows:

(1) Registration phase:

In this phase, a user $U_i$ submits his/her $ID_i$ and $PW_i$ to the GW-node in a secured manner. Then, the GW-node issues a license to $U_i$. The steps are described as follows:

Step 1: $U_i \Rightarrow$ GW-node:$\{ID_i, PW_i\}$. A $U_i$ enters an identity $ID_i$ and a password $PW_i$ and then sends $\{ID_i, PW_i\}$ to the GW-node using a secure channel.

Step 2: GW-node $\Rightarrow$ smart card of $U_i$ :$\{ h(.), ID_i , N_i , h(PW_i ), x_a \}$. The GW-node computes $N_i = h(ID_i \| PW_i) \oplus h(K)$ after receiving the registration request. Then, the GW-node personalizes the smart card with parameters $\{h(.), ID_i, N_i, h(PW_i), x_a\}$. $U_i$ receives the smart card information using a secure channel.

(2) Login phase:

When user $U_i$ enters an $ID_i$ and a $PW_i$ in order to carry out some inquiry or to access data from the WSN, the smart card must confirm the validity of $U_i$ according to the following steps:

Step 1: Validate $U_i$. The entered $ID_i$ and $PW_i$ are validated against the $ID$ and $PW$ stored on the user's smart card. If $U_i$'s identification validation fails, the smart card will terminate this request.

Step 2: $U_i$'s smart card calculates $DID_i$ and $C_i$.

$DID_i = h(ID_i \| PW_i) \oplus h(x_a \| T)$, where $T$ is the login system timestamp.

$C_i = h(N_i \| x_a \| T)$.

Step 3: $U_i \rightarrow$GW-node: $\{DID_i, C_i, T\}$.

$\{DID_i, C_i, T\}$ is transmitted to the GW-node via public channel.

(3a) Verification phase (gateway node):

When the GW-node receives a login request $\{DID_i, C_i, T\}$ at time $T^*$, the GW-node performs the following steps to verify the identity of $U_i$:

Step 1: Validates if $T^* - T < \Delta T$.

If $(T^* - T) \leq \Delta T$ then the validity of $T$ can be certain, and the GW-node proceeds to the next step. Otherwise, the GW-node rejects the request. Here, $\Delta T$ denotes the expected time interval for transmission delay.

Step 2: Calculates $C_i^*$.

$h(IDi \| PW_i)^* = DID_i \oplus h(x_a \| T)$

$C_i^* = h(h(ID_i \| PW_i)^* \| h(K) \| x_a \| T)$.

Step 3: Confirms whether the $C_i = C_i^*$.

If the $C_i = C_i^*$, then the GW-node accepts the login request and sends a request to $Sn$.

Step 4: GW-node $\rightarrow S_n$: $\{DID_i, A_i, T'\}$.

The GW-node calculates $A_i = h(DID_i \| S_n \| x_a \| T')$ and transmits a request $\{DID_i, A_i, T'\}$ to $S_n$ over a public channel. $T'$ is the GW-node request timestamp. $A_i$ is generated using the $x_a$ parameter, thus the value of $A_i$ can be used by $S_n$ to ensure that the message originates from a valid GW-node.

(3b) Verification phase (sensor node):

When $S_n$ receives request $\{DID_i, A_i, T'\}$ at time $T$, $S_n$ performs the following steps to verify the validity of the request:

Step 1: Validates if $T - T' < \Delta T$.

If $(T - T') \leq \Delta T$ then the validity of $T'$ can be certain, and $S_n$ proceeds to the next step.

Step 2: Recalculates $A_i$.

$A_i = h(DID_i \| S_n \| x_a \| T')$

Step 3: Confirms whether the value of the locally calculated $A_i$ is the same as the value of $A_i$ in the GW-node request.

If the value of the locally calculated $A_i$ is the same as the value of $A_i$ in the GW-node request, then $S_n$ responds to $U_i$'s original request. Otherwise, $S_n$ rejects the request.

### *2.2. Cryptanalysis of Das' Protocol*

Recently, several studies have analyzed security flaws in Das' scheme [14-16]. In this section, we also discuss the requirements of security in WSNs and describe the primary flaw of Das' protocol (it omits mutual authentication) and several secondary security issues [14-16].

#### 2.2.1. Security Requirements in Wireless Sensor Networks

Sastry and Wagner [9] noted several problems with regard to the security of user authentication provided by IEEE 802.15.4 [17]. They cited ACL management problems, loss of ACL state due to power interruptions, and key management problems. They concluded that IEEE 802.15.4 provides insufficient user authentication security and provided some solutions for the noted problems. However, above and beyond the security issues noted by Sastry and Wagner, there are two additional security issues that must be addressed:

- Secure user authentication in WSNs should include, to the extent possible, methods for addressing application layer issues such as masquerade, replay, and forgery attacks.
- Secure user authentication in WSNs should be based on mutual authentication.

#### 2.2.2. No Mutual Authentication

Because Das' protocol does not provide mutual authentication [14-16], a malicious user can attack a WSN that uses the Das protocol by means of eavesdropping and masquerading. The attack could be accomplished as follows:

(i) $U_i$ sends the message $\{DID_i, C_i, T\}$ to the GW-node for accessing the WSN.
(ii) The GW-node sends the message $\{DID_i, A_i, T\}$ to $S_n$ for asking the service for $U_i$.
(iii) The attacker captures the message $\{DID_i, A_i, T\}$ via eavesdropping.
(iv) The attacker provides an $S_M$ which masquerades as $S_n$ to get the $U_i$'s request data or hold back the request.
(v) Since $S_M$ co-works with $U_i$ continuously, the $U_i$ access requests will continue to fail.

With the Das method, after accepting the login request of $U_i$, the GW-node sends a message $\{DID_i, A_i, T'\}$ to some nearest sensor node $S_n$. Here the value of $A_i$ is computed by $A_i = h(DID_i \| S_n \| x_a \| T')$, where $T'$ is the current timestamp of GW-node. The value of $A_i$ is used to assure the sensor node that the message has come from the real GW-node. The GW-node message directs the sensor node to reply to the query with the data which $U_i$ has requested. However, there is no mechanism for the GW-node to be assured that the reply message was initiated from the queried sensor node. Thus, the Das-scheme only provides unilateral authentication between the GW-node and sensor node. There is no mutual authentication between the two nodes.

#### 2.2.3. No Protection against Insider Attacks

Nowadays users use a single common password for accessing different applications or servers. The situation is common practice and this is done for their convenience. It relieves the user from having to remember multiple passwords. Nevertheless, if the system manager or a privileged user of the

GW-node obtains the common password of $U_i$, he/she may try to impersonate $U_i$ by accessing other servers where $U_i$ could be a registered user. In the Das scheme [14,15], $U_i$ performs registration with the GW-node by presenting a password in plain format. Thus, the Das protocol does not provide sufficient protection against an insider attack on a GW-node by a privileged user.

#### 2.2.4. No Provision for Changing/Updating Passwords

The fixed password is definitely suffered from threats than an updating password. It is a widely recommended security policy, for highly secure applications, that users should update or change their passwords frequently. In the scheme [14,15], there is no provision for a user to easily change his/her password.

#### 2.2.5. No Protection against Forgery Attacks

A legal user of the system can launch a forgery attack against the WSN by eavesdropping and masquerading. A forgery attack can be launched as follows [16]:

(i) A legal user of the system $U^*$ can login to the WSN at $T_A$ and $T_B$ accurately.

(ii) Suppose $U^*$ has embedded a synchronized Trojan virus into legal user $U_i$'s system.

(iii) When $U_i$ wants to login to the WSN at $T_A$ and $T_B$, $U^*$ can eavesdrop on the messages $\{DID_i, C_i, T_A\}$ and $\{DID_i, C_i, T_B\}$ between the GW-node and $U_i$ at $T_A$ and $T_B$. To judge which message is $DID_i$ or $C_i$ as follows:

Step 1. $U^*$ can obtain the following messages: $DID_i(T_A) = h(ID_i \| PW_i) \oplus h(x_a \| T_A)$ and $DID_i(T_B) = h(ID_i \| PW_i) \oplus h(x_a \| T_B)$.

Step 2. And then $U^*$ can forge the dynamic login identity $DID^*(T_A)$ and $DID^*(T_B)$.

$DID^*(T_A) = h(ID^* \| PW^*) \oplus h(x_a \| T_A)$

$DID^*(T_B) = h(ID^* \| PW^*) \oplus h(x_a \| T_B)$.

(iv) $U^*$ can use the login phase formula to compute $DID_i(T_B)$, where $DID_i(T_B)$ is calculated as $DID_i(T_B) = DID_i(T_A) \oplus DID^*(T_A) \oplus DID^*(T_B)$ $DID_i(T_B) = h(ID_i \| PW_i) \oplus \boxed{h(x_a \| T_A)} \oplus \boxed{h(ID^* \| PW^*)} \oplus \boxed{h(x_a \| T_A)} \oplus \boxed{h(ID^* \| PW^*)} \oplus h(x_a \| T_B)$

(v) After $U^*$ obtains $U_i$'s $DID_i(T_B)$, $U^*$ sends a new session message $\{DID_i(T_B), C_i, T_S\}$ within $\Delta$T timestamp for a new login request. The timestamp $T_S$, where $T_S = T_B$, is made by $U^*$ for attack on the WSN.

(vi) Thus, the GW-node will verify message $\{DID_i(T_B), C_i, T_S\}$ from $U^*$ with following steps:

$U^* \rightarrow$GW-node: $\{DID_i(T_B), C_i, T_S\}$

Step 1. The GW-node receives $\{DID_i(T_B), C_i, T_S\}$ at $T^*$ and checks for $T^* - T_S < \Delta T$. The GW-node passes the verification and proceeds to the next step.

($T^* - T_B < \Delta T$ is known and $T_S = T_B$ was made arbitrarily by $U^*$)

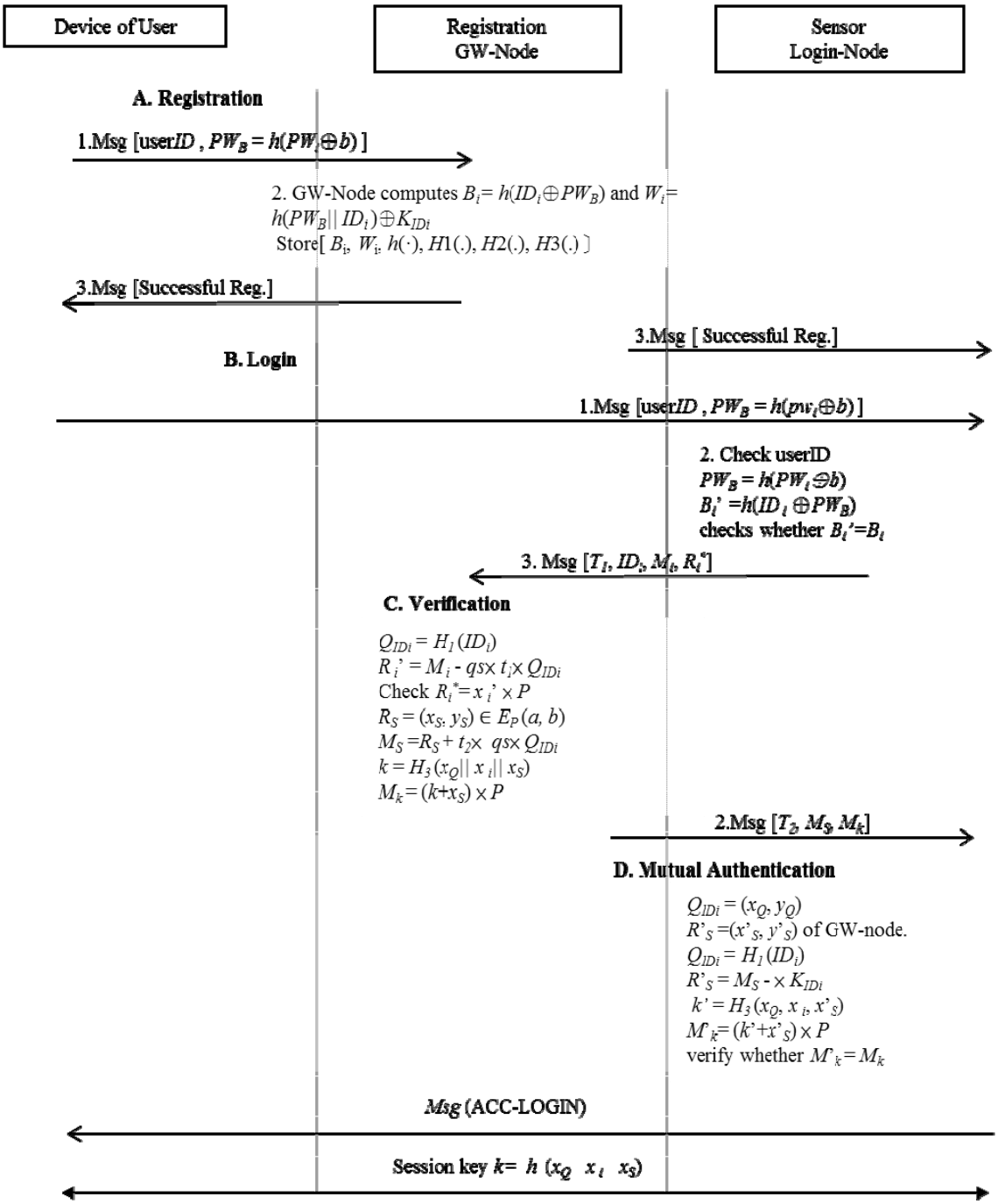
Step 2. The GW-node calculates $h(ID_i \| PW_i)^* = DIDi(T_B) \oplus h(x_a \| T)$ and obtains $C_i^* = h(h(ID_i \| PW_i)^* \| h(K) \| x_a \| T)$ ($C_i^* = C_i$) to pass the verification and proceed to the remaining steps.

Consequently, the Das protocol does not provide sufficient protection against a forgery attack by a legal user.

## 3. ECC-Based Authentication Protocol (EAP) for WSN

This section proposes a more efficient authentication mechanism using ECC. First, we review the fundamentals of Elliptic Curves and then survey the Elliptic Curves Cryptography (ECC) which is suitable for our construction of a secured authentication protocol for wireless sensor networks. The proposed five phases will be described later. The overall handshake of the proposed protocol is illustrated in Figure 1. The GW-node, $S_n$ and user use the $h(x_Q||x_i||x_S)$ as a session key with communication handshakes.

**Figure 1.** Communication handshakes of the proposed scheme.



### *3.1. ECC Based Authentication Protocol*

In 1985 Miller and Kobiltz proposed a secure and efficient elliptic curve cryptosystem (ECC) [17,18]. Because ECC provides a smaller key size than any other cryptosystem, it is suitable for application in smart card and wireless systems.

An elliptic curve is a cubic equation of the form: $\mathrm{E}: y^2 + axy + by = x^3 + cx^2 + dx + e$, where $a, b, c, d, e$ are real numbers. With regard to cryptography, we focus on the finite field of ECC and aim mainly at the prime $p$ of elliptic curve group. The mathematical equation of ECC satisfies the form: $E: y^2 = (x^3 + ax + b) \bmod p$, where $(4a^3 + 27b^2) \neq 0$. Let $F_p$ denote the finite field of points, where $p$ is a large prime number and containing $x, y, a, b$ elements. The equation points and the point at infinity $\boldsymbol{O}$ compose the elliptic curve group over real numbers. We find a large prime number $n$ such that $n \times P = \boldsymbol{O}$ using the elliptic curve addition algorithm. Here, × denotes an elliptic curve multiplication. The arithmetic of elliptic curve discrete logarithm problem (ECDLP) is given points $Q$ and $P$, where $Q, P \in F_p$ and are both publicly known, determine the random number $K$, $0 < K < n\text{-}1$, and compute $Q$ as : $Q = K \times P$ is satisfies. It is hard to determine $K$ given $Q$ and $P$, namely, ECDLP is a complex mathematical problem such that the security is achieved. The analog of Diffie-Hellman key exchange uses elliptic curve characteristics to complete key exchange. The key exchange between $U_A$ and $U_B$ can be done as follows [18-20]:

(i) The user $U_A$ chooses a random integer $r_A$ as a private key, where $r_A < n$ and computes the public key $Q_A$ as: $Q_A = r_A \times P$. Then, $U_A$ sends $Q_A$ to the user $U_B$.

(ii) The user $U_B$ selects a random integer $r_B$ as a private key, where $r_B < n$ and computes the public key $Q_B$ as: $Q_B = r_B \times P$. $U_B$ sends $Q_B$ to $U_A$.

(iii) $U_A$ can compute shared key $K_A = r_A \times Q_B = r_A \times r_B \times P$ and $U_B$ can compute shared key $K_B = r_B \times Q_A = r_B \times r_A \times P$. In this manner we find $K_A = K_B$.

### *3.2. Registration Phase*

This phase is invoked whenever user $U_i$ performs registration with the WSN. Then, $U_i$ submits $\{ID_i, PW_B\}$ to the GW-node by the secured channel. Then, the GW-node performs the license to $U_i$. The following steps are performed to complete this phase:

Step 1: $U_i \Rightarrow$ GW-node: $\{ID_i, PW_B\}$.

$U_i$ chooses his/her $ID_i$ and $PW_i$ password and randomly chooses a large number $b$ for computing $PW_B = h(PW_i \oplus b)$.

Step 2: After receiving the registration request, the GW-node computes $K_{IDi} = qs \times H_1(ID_i) \in G_p$, where $K_{IDi}$ is $U_i$'s authentication key and $G_p$ denotes a cyclic addition group of $P$.

Step 3: GW-node selects a base point $P$ with the order $n$ over $E_p(a, b)$, where $n$ is a large number for the security considerations. Then, the GW node derives its private/public key pair $(q_s, Q_S)$ by computing $Q_S = q_s \times P$. (Here × denotes an elliptic curve multiplication).

Step 4: GW node computes $B_i = h(ID_i \oplus PW_B)$ and $W_i = h(PW_B \,||\, ID_i) \oplus K_{IDi}$.

Step 5: GW-node $\Rightarrow$ smart card of $U_i$ : $\{B_i, W_i, h(\cdot), b, H_1(.), H_2(.), H_3(.)\}$.

GW-node stores $\{B_i, W_i, h(\cdot), H_1(.), H_2(.), H_3(.)\}$ on a smart card and sends the smart card to $U_i$ over a secure channel. Here $H_1(.)$, $H_2(.)$ and $H_3(.)$ are one-way hash functions, $H_1(.)$: $\{0, 1\} \rightarrow G_p$, $H_2(.):\{0,1\} \rightarrow Z_P^*$ and $H_3(.):\{0,1\} \rightarrow Z_P^*$.

Step 6: Upon $U_i$ receiving the smart card, $U_i$ stores the random number $b$ in the smart card. Such that the smart card contains $\{B_i, W_i, h(\cdot), b, H_1(.), H_2(.), H_3(.)\}$.

### *3.3. Login Phase*

Assume that $U_i$ enters in order to ask a service from the network, the smart card must perform the following steps to validate the legality of $U_i$:

Step 1: $U_i$ enters his/her $ID_i$ *and* $PW_i$ to login to obtain the message for GW-node request.

Step 2: $U_i$ computes $PW_B = h(PW_i \oplus b)$ and $B_i' = h(ID_i \oplus PW_B)$ and checks whether $B_i' = B_i$. If it holds, $U_i$ computes $Q = h(PW_B||ID_i)$ and $K_{IDi} = W_i \oplus Q$.

When the login request has been accepted, the user proceeds with the remaining steps:

Step 1: After $U_i$ obtaining his/her authentication key $K_{IDi}$, $U_i$ chooses a random point $R_i = (x_i, y_i) \in E_P(a, b)$, where $x_i$ and $y_i$ are $x$ and $y$ coordinating point of $R_i$.

Step 2: $U_i$ computes $t_1 = H_2(T_1)$ , $M_i = R_i + t_1 \times K_{IDi}$ and $R_i^* = x_i \times P$ at the timestamp $T_1$.

Step 3: $U_i \rightarrow$GW-node: $\{T_1, ID_i, M_i, R_i^*\}$.

$U_i$ sends message $Msg(T_1, ID_i, M_i, R_i^*)$ to GW-node.

### *3.4. Verification Phase*

After receiving the login request message $Msg(T_1, ID_i, M_i, R_i^*)$ at $T_1$ through the nearest sensor node ($S_n$), the GW-node executes the following steps to verify the user $U_i$:

Step 1: Compute $Q_{IDi}$ and $R_i'$

GW-node performs the following computations to obtain $Q_{IDi} = (x_Q, y_Q)$ and $R_i' = (x_i', y_i')$ of $U_i$.

$QIDi = H1\ (IDi)$

$t_1 = H_2(T_1)$

$R_i' = M_i - qs \times t_1 \times Q_{IDi}$

Step 2: The GW node verifies whether $R_i^* = x_i' \times P$. If it holds, $U_i$ is authenticated by GW-node.

Step 3: GW-node$\rightarrow U_i$: $\{T_2, M_S, M_k\}$ through $S_n$.

The GW node chooses a random point $R_S = (x_S, y_S) \in E_P(a, b)$ and computes $t_2 = H_2(T_2)$, $M_S = R_S + t_2 \times qs \times Q_{IDi}$, session key $k = H_3(x_Q \,||\, x_i \,||\, x_S)$ and $M_k = (k + x_S) \times P$.

GW-node sends a message $Msg(T_2, M_S, M_k)$ through the public channel in order to respond to the request of $S_n$ at the timestamp $T_2$.

### *3.5. Mutual Authentication Phase*

The GW-node sends $Msg(T_2, M_S, M_k)$ to the $S_n$ and then $S_n$ sends $Msg$(ACC-LOGIN) to the GW-node. The steps are described as follows:

Step 1: Compute $Q_{IDi}$ and $R'_S$

After receiving *Msg(T2, MS, Mk),* the $S_n$ execution obtains the following computation $QIDi = (x_Q, y_Q)$ and $R'_S = (x'_S, y'_S)$ of the GW-node.

$Q_{IDi} = H_1(ID_i)$

$t_2 = H_2(T_2)$

$R'_S = M_S - t_2 \times K_{IDi}$

Step 2: $S_n$ computes $k' = H_3(x_Q || x_i || x'_S)$ and $M'_k = (k' + x'_S) \times P$ to verify whether $M'_k = M_k$. If it holds, GW-node is successfully authenticated by $S_n$.

### *3.6. The Password-Changing Phase*

When a user $U_i$ enters an $ID_i$ and a $PW_i$ in order to request a password change, the smart card must compute a new value of $PW_B^* = h(PW_i^* \oplus b)$ to the GW-node. After receiving the password change request, the GW-node computes $B_i^*$ and $W_i^*$.

Step 1: GW-node computes $B_i^* = h(ID_i \oplus PW_B^*)$ and $W_i^* = h(PW_B^* || ID_i) \oplus K_{IDi}$.

Step 2: GW-node stores the new vale on smart card.

The smart card replaces the original values of $B_i$, $W_i$ with the new value $B_i^*$, $W_i^*$ and $PW_B^* = h(PW_i^* \oplus b)$.

## 4. Security and Performance Analysis

### *4.1. Security Analysis*

The studies we have referenced in this paper have discussed the security issues of remote user authentication. Below is a summary of those security issues, along with the reasons we believe our proposed ECC protocol can address those issues.

**Resistance to insider attack:** It is common practice for users to apply the same common password to access different applications. If a privileged insider has knowledge of another user $U_i$'s password, it hey may try to impersonate user $U_i$ to access network applications. Our proposed protocol registers user $U_i$ using cipher code $PW_B = h(PWi \oplus b)$ over a secure channel. This provides protection against stolen passwords. Thus, our protocol resists insider attacks.

**Resistance to masquerade attack:** To successfully complete a masquerade attack, an attacker must know $U_i$'s password in order to pass verification in the login phase and to be able to interpret the verification message correctly for mutual authentication. An attacker, even a legitimate user $U^*$, cannot masquerade as a different legitimate user $U_i$ without $U_i$'s password for forging the messages sent to the GW-node.

**Mutual authentication:** Mutual authentication is an important feature for a verification service that is resistant to server spoofing attacks. Our protocol provides a mutual authentication between the user $U_i$ and the GW-node by using ECC-based public and private keys exchange.

**Securely change/update password:** There is provision for users to update or change their password in our proposed scheme. Namely, a user can send a new password to the GW-node and then the GW-node computes new value of $B_i^*$, $W_i^*$ and stores them on the smart card.

We recall that the protocol [12-16] of Wong *et al*. does not provide for mutual authentication, and can be vulnerable to forgery and replay attacks. Besides, the proposal of Watro *et al.* has security weaknesses against masquerade attacks, and Das' protocol does not provide mutual authentication with an authenticated procedure using the hash function. Further, the weaknesses of Das' scheme are that it may suffer from an insider attack and a forgery attack. Chen *et al*.'s scheme is similar in Das' scheme, and also has the insider attack problem. Besides, the referenced proposals all fail to provide a secure method for updating user passwords. Table 2 compares our proposed protocol with the other referenced protocols in terms of protection against attacks. When compared against each other, our protocol provides a solution for user authentication that is more secure than the other referenced protocols.

**Table 2.** Security comparison among the referenced protocols.

| Item | Proposed | Chen *et al.*'s | Das' | Watro *et al.*'s | Wong *et al.*'s |
|---|---|---|---|---|---|
| Avoiding insider attack | Yes | No | No | Yes | Yes |
| Securely change/update password | Yes | No | No | No | No |
| Avoiding forgery attack | Yes | Yes | No | Yes | No |
| Mutual authentication | Yes | Yes | No | Yes | No |
| Avoiding masquerade attack | Yes | Yes | Yes | No | Yes |
| Avoiding replay attack | Yes | Yes | Yes | Yes | No |
| Avoiding guessing attack | Yes | Yes | Yes | Yes | Yes |

### *4.2. Performance Analysis*

For comparing performance between our protocol and related protocols, we estimate the computation costs. In the definition of computation costs, we define the notation $t_h$ as the hash computation time, $t_{PA}$ as the elliptic curve point addition computation time, $t_{PM}$ as the elliptic curve point multiplication computation time, $t_E$ as the elliptic curve polynomial computation time, $t_{PR}$ as the private key computation time, and $t_{PU}$ as the public key computation time. Note that the computation costs of $t_{PU}$ and $t_{PR}$ are considerably higher than $t_h$ ($t_{PU} >> t_h$ and $t_{PR} >> t_h$) because $t_{PU}$ and $t_{PR}$ usually need polynomial computation cost to obtain the public and private keys. Obviously, $t_E$, $t_{PA}$, $t_{PM}$ calculates a cubic equation at most and $t_h$ calculates a linear equation or quadratic equation at most. The comparison of related protocols is illustrated in Table 3.

When considering the computation cost in the authentication phase (which includes the verification and mutual authentication phases), our protocol requires only $11\ t_h + 4\ t_{PA} + 6\ t_{PM} + 2\ t_E$. That is, our protocol needs one point addition operation, four point multiplication operations and one polynomial operation in ECC. However, Watro *et al.*'s protocol needs two hash functions and four polynomial computations for private key and public key computation. It uses complex RSA and Diffie-Hellman algorithms for user authentication. The polynomial computation time calculates a prime exponential function which is considerably higher than cubic equation [12,17]. In addition, Watro *et al*.'s protocol needs four polynomial computations, for $t_{PR}$ and $t_{PU}$, are more than the other referenced protocols [12-16]. Besides, our proposed protocol is computed through combination of point addition and point

multiplication, point multiplication is defined by repeated addition. Considering the computation costs, ECC can generate smaller key sizes but maintain equivalent levels of security with RSA [18-20]. This is the reason the ECC-based protocol is more practical than Watro *et al*.'s protocol.

Lastly, when considering the communication cost, the proposed protocol has higher computation cost than other protocols, except for Watro *et al*.'s protocol. However, the protocol of Das does not provide mutual authentication. The method we propose solves most of the Das method problems. Furthermore, although Das's scheme needs five hash computation operations, Wong's needs four hash computation operations and Chen *et al*.'s protocol performs wireless sensor networking using seven $t_h$, their protocols suffer from security issues. Our proposed protocol addresses these issues and provides better security than the other related protocols.

**Table 3.** Performance comparison among related protocols.

| | **Proposed** | **Chen *et al.*** | **Das** | **Watro *et al.*** | **Wong *et al.*** |
|---|---|---|---|---|---|
| Authentication (Verification and Mutual Authentication) | $11\ t_h + 4\ t_{PA} + 6\ t_{PM} + 2\ t_E$ | $7\ t_h$ | $5t_h$ | $2t_h+2t_{PR}+2t_{PU}$ | $4t_h$ |

## 5. Conclusions

In this paper, we have analyzed Das' scheme for user authentication in WSNs. The Das protocol, which does not provide mutual authentication, is susceptible to insider and forgery attacks. We have also reviewed the protocols of Wong *et al*., which is vulnerable to forgery and replay attacks, of Watro *et al*., which is vulnerable to masquerade attacks, and Chen *et al*.'s protocol, which is susceptible to insider attacks. Additionally, a user cannot change his/her password with the former schemes. Since WSNs needs more efficient methods to perform mutual authentication in an insecure network environment, we use an ECC-based mechanism to accomplish this. The proposed protocol can prevent all the problems of the former schemes and provide mutual authentication to protect inside security and outside security. Furthermore, it not only inherits the merits of ECC-based mechanism but also enhances the WSN authentication with higher security than other protocols. Therefore, the proposed protocol is more suited to WSNs environments.

## Acknowledgements

This paper is supported by the National Science Council, Taiwan, R.O.C., Grant No. NSC 99-2219-E-007-007.

## References

1. Callaway, E.H. *Wireless Sensor Networks, Architectures and Protocols*; Auerbach Publications; Taylor & Francis Group: Boca Raton, FL, USA, 2003.
2. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless Sensor Network Survey. *Comput. Network.* **2008**, *52*, 2292-2330.

3. Sundararaman, B.; Buy, U.; Kshemkalyani, A.D. Clock Synchronization for Wireless Sensor Network: A Survey. *Ad-Hoc Networks* **2005**, *3*, 281-323.
4. Martinez, K.; Hart, J.K.; Ong, R. Environmental Sensor Networks. *Computer* **2004**, *37*, 50-56.
5. Szewezyk, R.; Osterweil, E.; Polastre, J.; Hamilon, M.; Mainwaring, A.; Estrin, D. Habitat Monitoring with Sensor Networks. *Commun. ACM* **2004**, *47*, 34–40.
6. Das, M.L.; Saxena, A.; Gulati, V.P. A Dynamic Id-Based Remote User Authentication Scheme. *IEEE Trans. Consum. Electron.* **2004**, *50*, 629-631.
7. Leung, K.C.; Cheng, L.M.; Fong, A.S.; Chan, C.K. Cryptanalysis of a Modified Remote User Authentication Scheme Using Smart Cards. *IEEE Trans. Consum. Electron.* **2003**, *49*, 1243-1245.
8. Akyildiz, I.F.; Weilian, S.; Sankarasubramaniam, Y.; Cayirci, E. A Survey on Sensor Networks. *IEEE Commun. Mag.* **2002**, *40*, 102-114.
9. Sastry, N.; Wagner, D. Security Considerations for IEEE 802.15.4 Networks. In *Proceedings of ACM Workshop on Wireless Security*, Philadelphia, PA, USA, 1 October 2004; pp. 32-42.
10. Watro, R.; Kong, D.; Cuti, S.-F.; Gardiner, C.; Lynn C.; Kruus, P. TinyPK: Securing Sensor Networks with Public Key Technology. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, Washington, DC, USA, 25 October 2004; pp. 59-64.
11. Wong, K.H.M.; Zheng, Y.; Cao, J.; Wang, S. A Dynamic User Authentication Scheme for Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Taichung, Taiwan, 5–7 June 2006; pp. 244-251.
12. Das, M.L. Two-Factor User Authentication in Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086-1090.
13. Tseng, H.R.; Jan, R.H.; Yang, W. An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks. In *Proceedings of IEEE Globecom*, Washington, DC, USA, 26–30 November 2007; pp. 986-990.
14. Khan, M.K.; Alghathbar, K. Security Analysis of Two-Factor Authentication in Wireless Sensor Networks. In *Proceedings of Advances in Computer Science and Information Technology: AST/UCMA/ISA/ACN 2010 Conferences*, Miyazaki, Japan, 23–25 June 2010; pp. 55-60.
15. Khan, M.K.; Alghathbar, K. Cryptanalysis and Security Improvements of 'Two-Factor User Authentication in Wireless Sensor Networks'. *Sensors J.* **2010**, *10*, 2450-2459.
16. Chen, T.H.; Shih, W.K. A Robust Mutual Authentication Protocol for Wireless Sensor Networks. *ETRI J.* **2010**, *32*, 704-712.
17. IEEE Standards for 802.15.4, Part 15, Amendment 4. *Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks*; IEEE: Washington, DC, USA, 2003.
18. Koblitz, N. Elliptic Curve Cryptosystems. *Math. Comput.* **1987**, *48*, 203-209.
19. Miller, V.S. Use of Elliptic Curves in Cryptography. In *Advances in Cryptology—CRYPTO '85: Proceedings*; LNCS Cryptography Volume; Springer: Berlin/Heidelberg, Germany, 1986; p. 417.
20. Menezes, A.J.; Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press Inc.: Boca Raton, FL, USA, 1997.