Research Article

A New User Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography

Wenbo Shi¹ and Peng Gong²

¹ Department of Electronic Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China
² National Key Laboratory of Mechatronic Engineering and Control, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Wenbo Shi; swb319@hotmail.com

Received 12 January 2013; Revised 16 March 2013; Accepted 16 March 2013

Academic Editor: Muhammad Khurram Khan

User authentication in wireless sensor networks (WSNs) is a critical security issue due to their unattended and hostile deployment in the field. Since sensor nodes are equipped with limited computing power, storage, and communication modules, authenticating remote users in such resource-constrained environments is a paramount security concern. To overcome the weaknesses of Yeh et al.’s protocol, we proposed a new authentication protocol for wireless sensor networks using elliptic curves cryptography. The comparisons show that our protocol is more suitable for WSNs.

1. Introduction

Wireless sensor networks (WSNs) are becoming more and more popular in everyday life as they offer economically viable, real-time monitoring solutions. These wireless sensors can be quickly and easily deployed in hostile environments, and WSNs are now widely used in a variety of real-time applications, such as vehicular tracking, habitat monitoring, environment control, military surveillance, healthcare monitoring, wildlife monitoring, and traffic monitoring. One recent survey declared that, in the near future, WSNs will become an intelligent and integral part of daily lives [1].

A WSN consists of a discrete group of independent, low cost, and low power nodes with limited memory and computation power. They communicate wirelessly over limited frequency and low bandwidth [1]. More specifically, sensor nodes collectively monitor the area and sense substantial amounts of data, which are transmitted to the base station traversing some nodes via RF signals and routing schemes.

A key requirement for WSN is user authentication [2, 3]. 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. 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) [1].

In 2004, Sastry and Wagner [4] proposed a security enhancement using access control lists (ACLs) in the GW node. In Sastry and Wagner’s protocol, an ACL would be maintained besides the client’s identity and the arranging of the nearest sensor node. Watro et al. [5] proposed a user authentication protocol employing RSA and Diffie-Hellman algorithms, but this protocol is open to hostile attack by a user masquerading as a sensor node. Wong et al. [6] proposed a dynamic user authentication protocol using hash function. Das [7] and Tseng et al. [8] demonstrated that both Watro’s and Wong’s user authentication methods were vulnerable to stolen-verifier, replay, and forgery attacks. To improve the security, Das [7] proposed a two-factor user authentication protocol. In 2007, Tseng et al. [8] show that Wong’s protocol was vulnerable to stolen passwords. Tseng et al. also proposed an enhanced user authentication protocol to improve overcome the weakness. However, Khan and Alghathbar [9, 10] show that Das’ protocol did not provide mutual authentication between gateway node and
sensor node and was vulnerable to gateway node bypassing attack and privileged-insider attack. Chen and Shih [11] also demonstrated that Das’ protocol did not provide mutual authentication between gateway node and sensor node. Chen and Shih [11] also proposed a more secure and robust two-factor user authentication in WSNs. Unfortunately, Yeh et al. [12] found that Chen and Shih’s protocol failed to provide a secure method for updating user passwords and was vulnerable to the insider attack problem. To improve the performance and the security, Yeh et al. [12] proposed the first user authentication protocol for WSNs using the elliptic curve cryptography (ECC). ECC was first proposed by Miller [13] and Koblitz [14], and its security was based upon the difficulty of elliptic curve discrete logarithm problem. Compared with the other cryptography, ECC offers a better performance because it can achieve the same security with a smaller key size. For example, 160-bit ECC and 1024-bit RSA have the same security level in practice [15]. Thus, ECC-based authentication schemes are very suitable for WSNs.

Unfortunately, Han [16] found that the Yeh et al. protocol had the following weaknesses: (1) no mutual authentication between the user and the sensor node, (2) no perfect forward secrecy, and (3) no key agreement between the user and the sensor node. To overcome the weaknesses of Yeh et al.’s protocol, we propose a new ECC-based user authentication protocol for WSNs.

The remainder of this paper is organized as follows. In Section 2, we propose our ECC-based authentication protocol for WSNs. The security analysis of the proposed protocol is presented in Section 3. In Section 4, performance analysis is presented. Conclusions are given in Section 5.

2. The Proposed Protocol

To solve the weakness of Yeh et al.’s scheme, we propose a new ECC-based user authentication protocol for WSNs. Thus, before issuing a query to a sensor node, each user must register with the gateway in a secure manner so that they can access the real-time sensors’ data. Upon the successful user registration request, the gateway node personalizes a smart card for every registered user. Then, a user can submit his query in an authentic way and access the sensor network data at any time within an administratively configurable period [6].

In order to execute the proposed framework, we considered that the gateway is a trusted node and it holds two master keys (x and y), which are sufficiently large for the sensor network. Before starting the system, it is assumed that the gateway and the sensor nodes share a long-term common secret key, that is, SKGS = h(Sn∥y) using any key agreement protocol. For example, Watro et al. [17] demonstrated that, with the careful design, D-H key agreement protocol [18] can be easily deployed on most constrained devices. Here, h(·) is a collision-free one-way hash function (i.e., SHA-1), which has an output length of 160 bits [19] and is used throughout this paper.

It is assumed that some identical secure symmetric cryptosystems are publicly available and stored in the gateway and the sensor node. As a result only the users registered with the gateway have access privileges to the sensors, which share a long-term secret with the gateway. The framework is divided into four phases, namely, user registration phase, login phase, authentication phase, and password update phase. For convenience, the notations used throughout this paper are summarized as follows:

\( p, n \): two large prime numbers;
\( F_p \): a finite field;
\( E \): an elliptic curve defined on finite field \( F_p \) with large order;
\( G \): the group of elliptic curve points on \( E \);
\( P \): a point on elliptic curve \( E \) with order \( n \);
\( U \): a user;
\( ID_U \): the user identity;
\( pw_U \): the user password;
\( GW \): the gateway node of WSN;
\( S_n \): a sensor node of WSN;
\( x, y \): the master keys of GW node;
\( h(\cdot) \): a secure one-way hash function;
\( || \) : a string concatenation operation;
\( \oplus \) : a string XOR operation;
ECDLP: the discrete logarithm problem, that is, given \( Q \in G \) compute \( x \in Z_n^* \) such that \( Q = xP \);
ECCDHP: the computational Diffie-Hellman problem, that is, given \( aP, bP \in G \) compute \( abP \).

2.1. Registration Phase. In this phase, user \( U \) has to submit an identity, \( ID_U \), and a password, \( pw_U \), to the GW node in a secured way. Then, the GW node issues a license to \( U \). The detailed steps are depicted as follows.

1. \( U \) chooses his identity \( ID_U \) and password \( pw_U \), generates a random number \( h_U \), and computes \( \overline{pw_U} = h(pw_U \oplus h_U) \). Then, \( U \) sends \( ID_U \) and \( \overline{pw_U} \) to the GW node.
2. Upon receiving the registration request, GW node computes \( K_G = h(ID_U||x) \times P \), \( B_U = h(ID_U \oplus \overline{pw_U}) \), and \( W_U = h(ID_U||\overline{pw_U}) \oplus K_G \). Then GW node stores \( \{B_U, W_U, h(\cdot)\} \) into a smart card and sends it to the user \( U \).
3. After receiving the smart card, the user \( U \) inputs \( h_U \) into it and finishes the registration.

2.2. Login Phase. When \( U \) enters an \( ID_U \) and a \( pw_U \), in order to deliver some query to or access data from the WSN, the smart card must perform the following steps to validate the legitimacy of \( U \). Figure 1 shows both the login phase and the authentication phase.
(1) User \( U \) inserts his smart card into the terminal and enters his identity \( \text{ID}_U \) and password \( \text{pw}_U \).

(2) The smart card computes \( \overline{\text{pw}}_U = h(\text{pw}_U \oplus b_U) \) and \( B'_U = h(\text{ID}_U \oplus \overline{\text{pw}}_U) \) and checks whether \( B'_U = B_U \). If it does not hold, the smart card stops the request. Otherwise, the smart card computes \( K_U = h(\text{ID}_U | \overline{\text{pw}}_U) \oplus W_U \). Then the smart card generates a random number \( r_U \in Z_n^* \) and computes \( X = r_U \times P \), \( X' = r_U \times K_U \), and \( \alpha = h(\text{ID}_U | X | X' | T_U) \), where \( T_U \) is the current timestamp of \( U \)’s system. At last, the smart card sends the login message \( M_1 = \{ \text{ID}_U, X, T_U, \alpha \} \) to \( S_n \).

2.3. Authentication Phase. After receiving the login request message \( M_1 \) at time \( T' \), the sensor node \( S_n \) executes the
following steps to authenticate $U$’s requests by the following steps.

(1) $S_n$ checks whether $T' - T_U \leq \Delta T$ holds, where $\Delta T$ is the legal time interval for transmission delay. If the answer is yes, the validity of $T_U$ can be assured, and $S_n$ proceeds to the next step. If no, the $S_n$ rejects the request.

(2) $S_n$ generates a random number $r_S \in Z^*_n$ and computes $Y = r_S \times P$ and $\beta = h(SK_{CS}|ID_U||X||T'_U||Y||T_S)$, where $T_S$ is the current timestamp of $S_n$’s system. At last, $S_n$ sends $M_2 = \{ID_U, X, T'_U, \alpha, ID_S, Y, T_S, \beta\}$ to GW node.

After receiving the message $M_2$ at time $T''$, GW node performs the following actions.

(1) GW node checks whether $T'' - T_U \leq \Delta T$ and $T'' - T_S \leq \Delta T$ hold, where $\Delta T$ is the legal time interval for transmission delay. If the answer is yes, the validity of $T_U$ and $T_S$ can be assured, and GW node proceeds to the next step. If no, GW node rejects the request.

(2) GW node uses long-term key $SK_{CS}$ to check whether the equation $\beta = h(SK_{CS}|ID_U||X||T'_U||Y||T_S)$ holds. If the equation does not hold, GW node stops the session. Otherwise, GW node computes $X' = h(ID_U||x||X)$ using his master key $x$. Then, GW node checks whether the equation $\alpha = h(ID_U||X||X'||T'_U)$ holds. If the equation does not hold, GW node stops the session. Otherwise, GW node computes $\gamma = h(SK_{CS}|ID_U||X||T'_U||Y||T_S||T'_U||Y||T_S)$ and $\delta = h(ID_U||X||X'||T'_U||Y||T_S)$. At last, GW node sends the message $M_3 = \{T_G, Y, \delta\}$ to $S_n$.

After receiving the message $M_3$ at time $T'''$, $S_n$ performs the following actions to authenticate $U$ and GW node.

(1) $S_n$ checks whether $T''' - T_C \leq \Delta T$ holds, where $\Delta T$ is the legal time interval for transmission delay. If the answer is yes, the validity of $T_C$ can be assured, and $S_n$ proceeds to the next step. If no, $S_n$ rejects the request.

(2) $S_n$ uses long-term key $SK_{CS}$ to check whether the equation $\gamma = h(SK_{CS}|ID_U||X||T'_U||\alpha||ID_S||Y||Y||T_S||T'_U||Y||T_S)$ holds. If the equation does not hold, $S_n$ stops the session. Otherwise, $S_n$ computes $K_{SU} = r_S \times X$, $\tau = h(Y||T'_S||\delta||K_{SU})$, and the session key $sk = h(X||Y||K_{SU})$ and sends $M_4 = \{Y, T_S, \delta, \tau\}$ to the smart card.

After receiving the message $M_4$ at time $T'''$, the smart card performs the following actions to authenticate $S_n$.

(1) The smart card checks whether $T''' - T_S \leq \Delta T$ holds, where $\Delta T$ is the legal time interval for transmission delay. If the answer is yes, the validity of $T_S$ can be assured, and the smart card proceeds to the next step. If no, the smart card rejects the request.

(2) The smart card computes $K_{US} = r_U \times Y$ and checks whether the equations $\delta = h(ID_U||X'||T'_U||Y||T_S)$ and $\tau = h(Y||T'_S||\delta||K_{US})$ hold. If either of the two equations does not hold, the smart card stops the session. Otherwise, $S_n$ is authenticated and the smart card computes the session key $sk = h(X||Y||K_{US})$.

2.4. Password Update Phase. The password update phase is invoked whenever user $U$ wants to update his old password $pw_U$. The password update phase is described below.

(1) User $U$ inserts his smart card into the terminal and enters his identity $ID_U$, the old password $pw_U$, and the new password $\bar{pw}_U$.

(2) The smart card computes $\bar{pw}_U = h(pw_U \oplus b_1)$ and $B'_U = h(ID_U \oplus \bar{pw}_U)$ and checks whether $B'_U = B_U$. If it does not hold, the smart card stops the request. Otherwise, the smart card computes $K_U = h(ID_U||\bar{pw}_U) \oplus W_U$, $\bar{pw}_U = h(pw_U \oplus b_2)$, and $W'_U = K_U \oplus h(ID_U||\bar{pw}_U)$. At last, the smart card replaces $W_U$ with $W'_U$.

3. Security Analysis

In this section, we will discuss the security of our protocol as follows.

Mutual Authentication. Our scheme provides mutual authentication, where all entities (i.e., user, gateway, and sensor nodes) are mutually authenticating each other. More specifically, when the GW node receives the message $M_2 = \{ID_U, X, T'_U, \alpha, ID_S, Y, T_S, \beta\}$, it can make sure that the user message $M_1 = \{ID_U, X, T'_U, \alpha\}$ is included in the sensor node message $M_2$. When the sensor node receives message $M_3 = \{T_G, Y, \delta\}$, it ensures that this message is generated by the GW node. Furthermore, when the user receives message $M_4 = \{Y, T_S, \delta\}$, he can also confirm that this message is generated by the sensor node. Hence, mutual authentication is achieved.

Replay Attacks. Our scheme is resistant to replay attacks, because the authenticity of messages $M_1, M_2, M_3, \text{and } M_4$ is validated by checking the freshness of four timestamps. Let us assume an intruder intercepts a login request message $M_1 = \{ID_U, X, T'_U, \alpha\}$ and attempts to access the sensor node by replaying the same message $M_1$. The verification of this login attempt fails since the time difference expires (i.e., $T' - T_U > \Delta T$). Similarly, if an intruder intercepts a valid message $M_2 = \{ID_U, X, T'_U, \alpha, ID_S, Y, T_S, \beta\}$ and attempts to replay it to the GW node, the verification request will fail at the GW node because the time difference expires again (i.e., $T'' - T_S > \Delta T$). Thus, our protocol is secure against replaying of messages.

User Impersonation Attacks. An attacker cannot impersonate the user. Suppose an attacker forges a login message $M_1 = \{ID_U, X, T'_U, \alpha\}$. Now, he will again try to login into the system with the modified message $\{ID_U, X', T'_U, \alpha'\}$. However, the
attacker cannot forge $\alpha^*$ without knowing $K_{U} = h(ID_U||x)\times P$ or the master key $x$ since he will be faced with ECDLP. Therefore, it is not possible to impersonate the user.

**Sensor Impersonation Attacks.** As long as an attacker does not know the secret key $SK_{GS}$, he cannot generate a legal message $M_2 = \{ID_D, X, T_U, \alpha, ID_S, Y, T_S, \beta\}$. Then he cannot cheat the gateway. At the same time, he cannot generate a legal message $M_4 = \{Y, T_S, \beta\}$ without knowing the master key $x$. Therefore, it is not possible to impersonate the sensor.

**Gateway Impersonation Attacks.** As long as an attacker does not possess the secret key $SK_{GS}$, he cannot impersonate the gateway and cannot cheat the sensor node. Hence, it frustrates attackers to generate the valid message $M_4$ to the sensor node. Therefore, it is not possible to impersonate the gateway.

**Man-in-the-Middle Attack.** Man-in-the-middle attack means that an active attacker intercepts the communication line between a legal user and the server and uses some means to successfully masquerade as both the server to the user and the user to the server. Then, the user will believe that he is talking to the intended server and vice versa. From the above discussion we know that our protocol can provide mutual authentication, and then the "man-in-the-middle" attack can be resisted.

**Stolen-Verifier Attacks.** An attacker who steals the password verifer (e.g., hashed passwords) from the gateway can use the stolen verifer to impersonate a legal user to login to the system. The proposed scheme is free from the stolen verifier attack. There is no such information stored at the server, by which an adversary can make a fabricated login request to impersonate a legal user to login the server or can impersonate the gateway to cheat the legal user and the sensor node.

**Insider Attacks.** It is possible in a real-time environment, when the gateway manager or system administrator can use the user password $pw_{U}$ (e.g., weak password), to impersonate the user $U$ through any other network gateways. In this case, our scheme does not give any room for privileged insiders, since, in the registration phase, the user $U$ is passing $pw_{U} = h(\alpha_2, x_2, b_2)$ instead of the plain password. Thus, the insider of the GW node cannot get $pw_{U}$ easily. Here, $b_2$ is a sufficiently high entropy number, which is not revealed to the GW node. Furthermore, the proposed scheme does not store any verifier table and can resist the insider attacks.

**Perfect Forward Secrecy.** A protocol is said to be perfect forward secrecy if compromise of the three private keys of the participating entities does not affect the security of the previous session keys. Two aspects are related to this notion, that is, perfect forward secrecy (p-FS) and master key perfect forward secrecy. p-FS means that the compromise of both user’s and sensor node’s long-term private keys would not affect the secrecy of the previously established session keys. Master key p-FS is satisfied if the session key secrecy still holds even when the server’s master key is compromised. Our protocol satisfies both p-FS and master key p-FS by using $sk = h(X||Y||r_5 \times X)$ or $sk = h(X||Y||[r_5 Y] X)$ as the shared secret. If user’s private keys or gateway’s master key is compromised, the adversary cannot compute $r_5$, or $r_{5}$ from $X$ and $Y$ since he has to solve the ECCDHP, thus satisfying both p-FS and master key p-FS.

### 4. Performance Comparison

For the convenience of evaluating the computational cost, we define some notations as follows.

$TG_{mul}$: the time of executing a scalar multiplication operation of point.

$TG_{add}$: the time of executing an addition operation of points.

$TG_{grp}$: the time of generating a random number point.

$TG_{mph}$: the time of executing a map-to-point hash function.

$TG_{h}$: the time of executing a one-way hash function.

|                             | Yeh et al.’s protocol | Our protocol            |
|------------------------------|------------------------|-------------------------|
| Computational cost (user)    | $2TG_{mul} + 1TG_{grp} + 1TG_{add} + 4TG_{h} \approx 2.5TG_{mul}$ | $3TG_{mul} + 5TG_{h} \approx 3TG_{mul}$ |
| Computational cost (sensor node) | $2TG_{mul} + 1TG_{grp} + 1TG_{add} + TG_{mph} + 1TG_{h} \approx 3TG_{mul}$ | $2TG_{mul} + 3TG_{h} \approx 2TG_{mul}$ |
| Computational cost (gateway) | $3TG_{mul} + 1TG_{grp} + TG_{mph} + 1TG_{h} \approx 3TG_{mul}$ | $1TG_{mul} + 4TG_{h} \approx 1TG_{mul}$ |
| Mutual authentication        | No                     | Yes                     |
| Perfect forward secrecy      | No                     | Yes                     |
| Key agreement between user and node | No              | Yes                     |
| User impersonation attack    | No                     | No                      |
| Sensor node impersonation attack | Yes               | No                      |
| Gateway impersonation attack | No                     | No                      |
| Inside attack                | No                     | No                      |
| Replay attacks               | No                     | No                      |
| Stolen-verifier attacks      | No                     | No                      |
| Man-in-the-middle attack     | No                     | No                      |
In Table 1, we summarize the performance results of the proposed protocol. In Table 1, we know that the user, the sensor node, and the gateway require $3T_{\text{mul}} + 5T_{\text{h}}$, $2T_{\text{mul}} + 3T_{\text{h}}$, and $1T_{\text{mul}} + 4T_{\text{h}}$, separately. From the theoretical analysis [20] and the experimental result [21, 22], we know that the relative computation cost of generating a random number and executing a map-to-point hash function is about 1/2 times that of the scalar multiplication.

Besides, the computation costs of $T_{\text{mul}}$ are considerably higher than $T_{\text{add}}$ and $T_{\text{h}}$. Then, the computational costs of the user, the sensor node, and the gateway in Yeh et al.'s protocol are about $3T_{\text{mul}}$, $3T_{\text{mul}}$, and $3T_{\text{mul}}$, separately. The computational costs of the user, the sensor node, and the gateway in our protocol are about $3T_{\text{mul}}$, $2T_{\text{mul}}$, and $1T_{\text{mul}}$, separately. Then our protocol has better performance at the sensor node side and the gateway side. Moreover, Yeh et al.'s protocol cannot provide (1) mutual authentication between the user and the sensor node, (2) perfect forward secrecy, and (3) key agreement between the user and the sensor node, and then our protocol enhances the security at the cost of increasing user's computation cost slightly.

5. Conclusions

This paper provides a new ECC-based user authentication protocol for WSNs. The proposed protocol performs more efficiently in terms of computation cost, communication cost, and security. Compared with the protocol of Yeh et al., the proposed protocol in this paper can prevent general security issues and provide mutual authentication to protect inside security and outside security. Therefore, the proposed protocol is more suited to WSNs environments.

Acknowledgments

The authors thank the editors and the anonymous reviewers for their valuable comments. This research was supported by National Natural Science Foundation of China (nos. 61202447 and 61201810), Natural Science Foundation of Hebei Province of China (no. F2013501066), Northeastern University at Qinhuangdao Science and Technology Support Program (no. xnk201307), Beijing Natural Science Foundation (no.4132055), and Excellent Young Scholars Research Fund of Beijing Institute of Technology.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. A. Cayirci, "Survey on saensor network," IEEE Communications Magazine, vol. 40, pp. 102–114, 2002.
[2] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," IEEE Transactions on Consumer Electronics, vol. 50, no. 2, pp. 629–631, 2004.
[3] K. C. Leung, L. M. Cheng, A. S. Fong, and C. K. Chan, "Cryptanalysis of a modified remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 49, no. 4, pp. 1243–1245, 2003.
[4] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in Proceedings of the ACM Workshop on Wireless Security (WiSe '04), pp. 32–42, Philadelphia, Pa, USA, October 2004.
[5] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04), pp. 59–64, Washington, DC, USA, October 2004.
[6] K. H. M. Wong, Z. Yuan, C. Iannong, and W. Shengwei, "A dynamic user authentication scheme for wireless sensor networks," in Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 244–251, Taichung, Taiwan, June 2006.
[7] M. L. Das, "Two-factor user authentication in wireless sensor networks," IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086–1090, 2009.
[8] H. R. Tseng, R. H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07), pp. 986–990, Washington, DC, USA, November 2007.
[9] M. K. Khan and K. Alghathbar, "Security analysis of two-factor authentication in wireless sensor networks," in Proceedings of the Advances in Computer Science and Information Technology (AST/UCMA/ISA/ACN '10), pp. 55–60, Miyazaki, Japan, June 2010.
[10] M. K. Khan and K. Alghathbar, "Cryptography and security improvements of two-factor user authentication in wireless sensor networks," Sensors, vol. 10, no. 3, pp. 2450–2459, 2010.
[11] T. H. Chen and W. K. Shih, "A robust mutual authentication protocol for wireless sensor networks," ETRI Journal, vol. 32, no. 5, pp. 704–712, 2010.
[12] H. L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim, and H. W. Wei, "A secured authentication protocol for wireless sensor networks using Elliptic Curves Cryptography," Sensors, vol. 11, no. 5, pp. 4767–4779, 2011.
[13] V. S. Miller, "Use of elliptic curves in cryptography," in Proceedings of the Advances in Cryptology (CRYPTO ’85), vol. 218 of Lecture Notes in Computer Science, pp. 417–426, Springer, 1986.
[14] N. Kobitz, "Elliptic curve cryptosystem," Mathematics of Computation, vol. 48, pp. 203–209, 1987.
[15] D. Hankerson, A. Menezes, and S. Vanstone, Guide to Elliptic Curve Cryptography, Lecture Notes in Computer Science, Springer, New York, NY, USA, 2004.
[16] W. Han, "Weakness of a Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography," http://eprint.iacr.org/2011/293 .
[17] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN ’04), pp. 59–64, October 2004.
[18] W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644–654, 1976.
[19] "National Institute of Standards and Technology, FIPS PUB 180-1, Secure Hash Standard," http://www.techheap.com/cryptography/hash/fips180-1.pdf.
[20] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," International Journal of Information Security, vol. 6, no. 4, pp. 213–241, 2007.
[21] X. Cao, X. Zeng, W. Kou, and L. Hu, “Identity-based anonymous remote authentication for value-added services in mobile networks,” IEEE Transactions on Vehicular Technology, vol. 58, no. 7, pp. 3508–3517, 2009.

[22] H. Debiao, C. Jianhua, and H. Jin, “An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security,” Information Fusion, vol. 13, no. 3, pp. 223–230, 2012.
