Sensors **2015**, *15*, 17057-17075; doi:10.3390/s150717057

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

A User Authentication Scheme Based on Elliptic Curves Cryptography for Wireless Ad Hoc Networks

Huifang Chen [1,2,*], Linlin Ge [1] and Lei Xie [1,2]

[1] Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China; E-Mails: lynkeh13@zju.edu.cn (L.G.); xiel@zju.edu.cn (L.X.)

[2] Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China

* Author to whom correspondence should be addressed; E-Mail: chenhf@zju.edu.cn; Tel.: +86-571-8795-1820 (ext. 217); Fax: +86-571-8795-2017.

Academic Editor: Leonhard Reindl

Received: 7 June 2015 / Accepted: 10 July 2015 / Published: 14 July 2015

Abstract: The feature of non-infrastructure support in a wireless ad hoc network (WANET) makes it suffer from various attacks. Moreover, user authentication is the first safety barrier in a network. A mutual trust is achieved by a protocol which enables communicating parties to authenticate each other at the same time and to exchange session keys. For the resource-constrained WANET, an efficient and lightweight user authentication scheme is necessary. In this paper, we propose a user authentication scheme based on the self-certified public key system and elliptic curves cryptography for a WANET. Using the proposed scheme, an efficient two-way user authentication and secure session key agreement can be achieved. Security analysis shows that our proposed scheme is resilient to common known attacks. In addition, the performance analysis shows that our proposed scheme performs similar or better compared with some existing user authentication schemes.

Keywords: wireless ad hoc network (WANET); self-certified public key (SCPK); elliptic curves cryptography (ECC); user authentication; session key agreement

1. Introduction

Wireless ad hoc network (WANET) is a decentralized type of wireless network. It has widely practical applications, such as tactical communication, emergency communication, temporary

communication, and so on. However, the WANET is vulnerable to various attacks due to the absence of infrastructure support [1]. Security of the WANET is critical for its deployment and management. Moreover, the user authentication is the first safety barrier in a network. That is, each node needs to ensure that the peer node with which it is communicating is he/she claims. On the other hand, wireless devices have limited computation capability, memory and energy. For the resource-constrained WANET, an efficient and lightweight user authentication scheme is necessary.

Many user authentication schemes have been proposed for the WANET in recent years. In [2], Bechler, M. *et al*. proposed a cluster-based user authentication scheme, where a cluster head controls the cluster. Since the cluster structure is useful for enhancing the scalability, the cluster-based authentication scheme is more suitable for large-scale networks. However, this scheme is exposed to the single point of failure since all cluster members depend on the cluster head. A distributed key management and user authentication approach is proposed in [3], where the concepts of identity-based key cryptography and threshold secret sharing are used. This approach works in a self-organizing way to provide the key generation and management service, and effectively solves the single point of failure problem. However, the security is breached when a threshold number of shareholders are compromised. Other user authentication schemes were proposed in [4] and [5], where a certificate server (CS) is used to issue user's certificate and public key. In addition, users perform the identity authentication with the assistance of CS. However, the CS is hard to be set up because of the dynamics of nodes in WANETs. Moreover, if the identity authentication needs the help of CS, the storage and management requirements of certificates increase the burden for CS.

Most user authentication schemes mentioned above use the public key infrastructure (PKI) [6] or the identity-based public key cryptosystem (ID-PKC) [7]. However, the high complexity for certificates in PKI increases the system burden greatly. In addition, the key escrow problem of ID-PKC is also a serious problem.

Unlike the prior work, the self-certified public key (SCPK) cryptosystem [8] is another kind of scheme. In this scheme, certificate authority (CA) embeds its signature in user's public key, and computes user's private key cooperatively with users. The advantage of the SCPK scheme is that the authenticity of a user's public key can be verified publicly without using any certificate issued by the CA and the private key known to the user only. Hence, this scheme does not need the digital certificates as in the PKI scheme, as well as avoids the key escrow problem of the ID-PKC scheme.

Compared with RSA, one of most widely accepted and traditional public key cryptographies, elliptic curves cryptography (ECC), has attracted considerable attention due to its smaller key size and lower resource consumption for achieving the same security level. This is because the addition operation in ECC is the counterpart of modular multiplication in RSA, and multiple addition is the counterpart of modular exponentiation. Furthermore, ECC is based on the intractability of the elliptic curve discrete logarithm problem (ECDLP). That is, finding an effective and rapid solution to the ECDLP is still a hard problem [9].

Hence, the user authentication scheme based on SCPK and ECC is a feasible alternative for resource-constrained wireless networks, such as WANET, mobile ad hoc networks and wireless sensor networks. Several user authentication schemes using SCPK and ECC have been proposed [10–12]. In [10], a distributed user authentication scheme based on SCPK was presented. In this scheme, each user gets his/her public/private key from CA through a secure communication channel. However,

providing a secure communication channel in a wireless network is not a trivial thing. A user authentication and key agreement scheme was proposed in [11], where the timestamp mechanism is used to resist the replay attack. However, it is a difficult task to maintain time synchronization in a WANET. In addition, the session key cannot resist key compromise impersonation attack in this scheme. In [12], a novel self-certified secure access authentication protocol was proposed. In this scheme, a challenge-response mechanism is adopted to resist the replay attack. However, the user's private key can be compromised easily.

In this paper, we propose a user authentication scheme based on SCPK and ECC for a WANET. In order to reduce the computational complexity, the SCPK proposed in [13] is modified using ECC. The proposed user authentication scheme consists of three phases, namely the setup phase, the user registration phase, and the user authentication phase. CA selects and generates the global system parameters, and publishes them to the whole network in the setup phase. Users register with CA to obtain the private/public key pairs for authentication in the user registration phase. In the user authentication phase, users complete their identities authentication using their private/public keys and the CA's public key. Finally, we analyze the performance of the proposed user authentication scheme, in terms of the security, the storage overhead, the communication overhead and the computation overhead. Analysis results show that our proposed scheme achieves efficient two-way user authentication and secure session key agreement. Hence, the proposed scheme is efficient, and suitable for the resource-constrained WANET.

Our proposed user authentication scheme differs from other existing user authentication schemes in [10–12] are: (1) A secure communication channel for distributing user's public/private key does not need; (2) A modified challenge-response mechanism is adopted to resist the replay attack; (3) The authentication mechanism between user and CA in the user registration phase is used to resist the user masquerade attack.

The remainder paper is organized as follows. In Section 2, the system model for the proposed user authentication scheme is introduced. In Section 3, the proposed user authentication scheme based on SCPK and ECC is presented. The security and performance of the proposed scheme are analyzed in Sections 4 and 5, respectively. Finally, we conclude the paper in Section 6.

2. System Model

Figure 1 shows the system architecture for our proposed user authentication scheme.

In this system, a CA is deployed to generate user's private/public key pairs cooperatively with users. Each user knows the public key of the CA. With the public key of CA, each user can verify the peer user's identity with whom he/she is communicating.

To clarify the proposed user authentication scheme, notations and their denotations are summarized in Table 1.

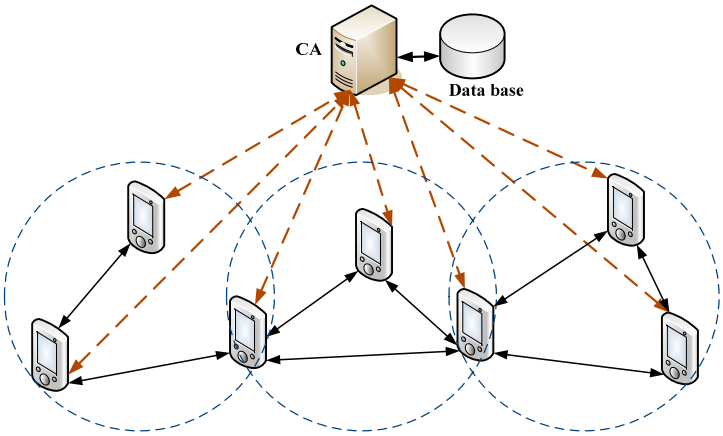

Figure 1. The system architecture of the proposed user authentication scheme in a wireless ad hoc network (WANET).

Table 1. Notations and their denotations.

Notations	Denotations
p	A large prime number
GF(p)	The finite field
a, b	The elliptic curve parameters, real numbers
$E_p(a, b)$	The elliptic curve over GF(p) consisting of the elliptic group of points defined by $y^2 = x^3 + ax + b \pmod p$, where $(4a^3 + 27b^2) \bmod p \neq 0$
G	A base point (x, y) selected on $E_p(a, b)$ with a large order
n	The order of point G, where n is the smallest positive integer such that $nG = O$ (infinity point), and n is a large prime number
$SHA(\cdot)$	A one-way hash function
s_{CA}	The private key of CA
P_{CA}	The public key of CA
N_{CA}	A nonce randomly generated by CA from [2, n−2]
N_i	A nonce randomly generated by U_i from [2, n−2]
s_i	The private key of U_i
P_i	The public key of U_i
ID_i	The identity of U_i
$signature_i$	The signature of U_i
MIC_i	The massage integrity code of the message generated by U_i
$\oplus$	The simple exclusive-OR operation
$\|\|$	The message concatenation operation

3. The Proposed User Authentication Scheme

In this section, a user authentication scheme based on SCPK and ECC for a WANET is presented.

The proposed scheme is divided into three phases, namely the setup phase, the user registration phase, and the user authentication phase. In the setup phase, CA generates the system parameters and publishes them to users. In the user registration phase, users obtain their private/public key pairs by registering with CA. In the user authentication phase, users complete their identities authentication with the help of their private/public keys and the public key of CA.

The detail of the proposed user authentication scheme is described as follows.

3.1. The Setup Phase

We adopt an elliptic curve defined over GF(p) is recommended by SEC 2 [14]. First, the elliptic curve $E_p(a, b)$ over GF(p) is defined by $y^2 = x^3 + ax + b \pmod p$, where a and b are real numbers, and $(4a^3 + 27b^2) \bmod p \neq 0$. Next, a base point $G = (x_G, y_G)$ with a very large value order is selected on $E_p(a, b)$. The order of G, n, is the smallest positive integer such that $n \cdot G = O$, where O is infinity point. The global parameters of the system, (p, a, b, G, n), are known by all users in networks.

CA randomly chooses an integer s_{CA}, from [2, n−2] as its private key. In addition, CA's paired public key is generated with:

$$R_i' = r_i' \cdot G \tag{1}$$

And then, CA publishes P_{CA} to the whole network, but keeps s_{CA} as a secret.

3.2. The User Registration Phase

When a user, U_i with identity ID_i, wants to join the system, he/she performs the following operations to register with CA.

First, U_i generates a nonce, N_i, using a pseudo-random number generator (PRNG), and randomly chooses an integer, r_i', from [2, n−2]. Then, U_i computes:

$$R_i' = r_i' \cdot G \tag{2}$$

And:

$$ID_i' = ID_i \oplus SHA(r_i' \cdot P_{\mathrm{CA}}) \tag{3}$$

After that, U_i transmits *Message 1* (N_i, R_i', ID_i') to CA. That is, $\mathrm{U}_i \rightarrow \mathrm{CA} : N_i \,||\, R_i' \,||\, ID_i'$.

Receiving (N_i, R_i', ID_i') from U_i, CA checks whether the message is fresh according to N_i. If the message has been received, CA discards it and cancels the user registration. Otherwise, CA computes:

$$SHA(s_{\mathrm{CA}} \cdot R_i') = SHA(s_{\mathrm{CA}} \cdot r_i' \cdot G) = SHA(r_i' \cdot P_{\mathrm{CA}}) \tag{4}$$

The user's identity is extracted by:

$$ID_i = ID_i' \oplus SHA(s_{\mathrm{CA}} \cdot R_i') \tag{5}$$

CA checks ID_i. If ID_i has existed, CA cancels the user registration. Otherwise, CA randomly chooses an integer $\tilde{r}_{\mathrm{CA}}$ from [2, n−2], and computes:

$$R_i = R_i' + \tilde{r}_{\mathrm{CA}} \cdot G \tag{6}$$

And:

$$\tilde{s}_i = (s_{\mathrm{CA}} \cdot SHA(ID_i \,||\, R_i.x) + \tilde{r}_{\mathrm{CA}}) \bmod n \tag{7}$$

where $R_i.x$ is the x-coordinate of the point R_i.

CA generates a nonce, N_{CA}, using a PRNG, and returns *Message 2* $(N_i, N_{\mathrm{CA}}, R_i, \tilde{s}_i)$ to U_i. That is, $\mathrm{CA} \rightarrow \mathrm{U}_i : N_i \,||\, N_{\mathrm{CA}} \,||\, R_i \,||\, \tilde{s}_i$.

After receiving $(N_i, N_{CA}, R_i, \tilde{s}_i)$ from CA, U$_i$ derives the private key as:

$$s_i = \tilde{s}_i + r_i' = (s_{CA} \cdot SHA(ID_i \| R_i.x) + \tilde{r}_{CA}) \bmod n + r_i' \tag{8}$$

And U$_i$ verifies the authenticity of P_i by:

$$P_i = s_i \cdot G = P_{CA} \cdot [(SHA(ID_i \| R_i.x)) \bmod n] + R_i. \tag{9}$$

If this verification succeeds, U$_i$ accepts P_i as his/her public key.

In the following, we demonstrate why the verification procedure described in (9) works correctly. According to Equations (6)–(8), we obtain:

$$\begin{aligned} s_i \cdot G &= [(s_{CA} \cdot SHA(ID_i \| R_i.x) + \tilde{r}_{CA}) \bmod n + r_i'] \cdot G \\ &= P_{CA} \cdot [SHA(ID_i \| R_i.x) \bmod n] + \tilde{r}_{CA} \cdot G + r_i' \cdot G \\ &= P_{CA} \cdot [SHA(ID_i \| R_i.x) \bmod n] + R_i. \end{aligned}$$

Hence, U$_i$ computes $SHA(N_{CA} \| r_i' \cdot P_{CA})$ and returns *Message 3* ($SHA(N_{CA} \| r_i' \cdot P_{CA})$) to CA. That is, $U_i \rightarrow CA : SHA(N_{CA} \| r_i' \cdot P_{CA})$.

Receiving ($SHA(N_{CA} \| r_i' \cdot P_{CA})$), CA computes $SHA(N_{CA} \| s_{CA} \cdot R_i')$ and compares it with $SHA(N_{CA} \| r_i' \cdot P_{CA})$ received from U$_i$. If $SHA(N_{CA} \| s_{CA} \cdot R_i') = SHA(N_{CA} \| r_i' \cdot P_{CA})$, CA is be convinced that U$_i$ has verified the authenticity of his/her public key. Then, CA stores the registration information in the registration file. If $SHA(N_{CA} \| s_{CA} \cdot R_i') \neq SHA(N_{CA} \| r_i' \cdot P_{CA})$, CA cancels the user registration.

The interaction diagram of the user registration phase mentioned above is shown in Figure 2.

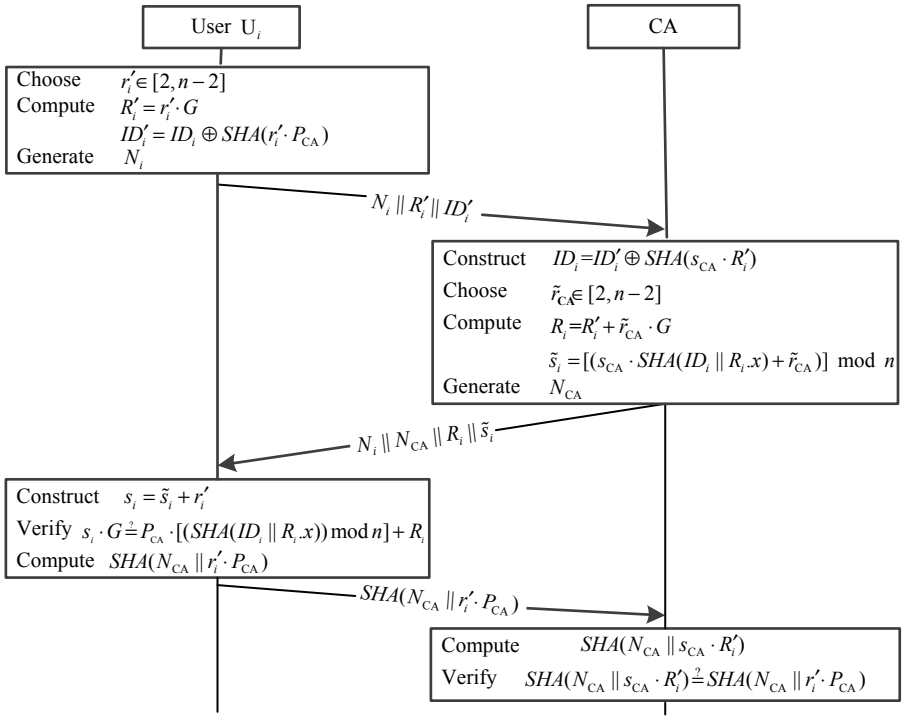

Figure 2. The user registration phase.

After U$_i$ finishes the registration successfully, he/she stores (R_i, ID_i, s_i, P_i). Other users can use G, n, P_{CA}, R_i and ID_i to construct the public key of U$_i$, P_i.

3.3. The User Authentication Phase

The user authentication and session key agreement between Alice and Bob operates as follows, where Alice is an initiator and Bob is a responder.

Alice wants to set up a session key with Bob securely.

Step 1: $\text{Alice} \rightarrow \text{Bob}: N_{\text{A}} || C_{\text{A}} || ID_{\text{A}} || ID_{\text{B}} || R_{\text{A}} || signature_{\text{A}}$

First, Alice generates a nonce, N_{A}, using a PRNG, and randomly chooses an integer, r_{A}, from $[2, n-2]$. Next, Alice computes $C_{\text{A}} = r_{\text{A}} \cdot G$ Then,Alice generates a signature using her private key as:

$$signature_{\text{A}} = (r_{\text{A}} + s_{\text{A}} \cdot SHA(N_{\text{A}} || C_{\text{A}} || ID_{\text{A}} || ID_{\text{B}} || R_{\text{A}})) \bmod n \tag{10}$$

Thereafter, Alice sends $(N_{\text{A}}, C_{\text{A}}, ID_{\text{A}}, ID_{\text{B}}, R_{\text{A}}, signature_{\text{A}})$ to Bob.

Step 2: $\text{Bob} \rightarrow \text{Alice}: N_{\text{A}} || N_{\text{B}} || C_{\text{B}} || ID_{\text{B}} || ID_{\text{A}} || R_{\text{B}} || MIC_{\text{B}}$

Receiving the message from Alice, Bob performs the following operations.

(1) According to N_{A}, Bob checks whether the message is fresh or not. If the message is fresh, Bob goes on the user authentication process. Otherwise, Bob rejects Alice's authentication request.

(2) Bob computes Alice's public key as:

$$P_{\text{A}} = P_{\text{CA}} \cdot [(SHA(ID_{\text{A}} || R_{\text{A}}.x)) \bmod n] + R_{\text{A}} \tag{11}$$

Bob verifies the Alice's signature as:

$$\begin{aligned} signature_{\text{A}} \cdot G &= [(r_{\text{A}} + s_{\text{A}} \cdot SHA(N_{\text{A}} || C_{\text{A}} || ID_{\text{A}} || ID_{\text{B}} || R_{\text{A}})) \bmod n] \cdot G \\ &= r_{\text{A}} \cdot G + [(s_{\text{A}} \cdot SHA(N_{\text{A}} || C_{\text{A}} || ID_{\text{A}} || ID_{\text{B}} || R_{\text{A}})) \bmod n] \cdot G \\ &= C_{\text{A}} + P_{\text{A}} \cdot [(SHA(N_{\text{A}} || C_{\text{A}} || ID_{\text{A}} || ID_{\text{B}} || R_{\text{A}})) \bmod n]. \end{aligned}$$

If the signature is valid, Alice is a valid user and Bob continues the user authentication process. Otherwise, Bob cancels the user authentication process.

(3) Bob generates a nonce N_{B}, using a PRNG, and randomly chooses an integer r_{B}, from $[2, n-2]$. Next, Bob computes $C_{\text{B}} = r_{\text{B}} \cdot G$. Then, Bob computes the session key,

$$K_{\text{BA}} = SHA((r_{\text{B}} + s_{\text{B}}) \cdot (C_{\text{A}} + P_{\text{A}})) \tag{12}$$

and the message integrity code,

$$MIC_{\text{B}} = SHA(K_{\text{BA}} || N_{\text{A}} || N_{\text{B}} || C_{\text{B}} || ID_{\text{B}} || ID_{\text{A}} || R_{\text{B}}) \tag{13}$$

Finally, Bob sends $(N_{\text{A}}, N_{\text{B}}, C_{\text{B}}, ID_{\text{B}}, ID_{\text{A}}, R_{\text{B}}, MIC_{\text{B}})$ to Alice.

Step 3: $\text{Alice} \rightarrow \text{Bob}: N_{\text{B}} || ID_{\text{A}} || ID_{\text{B}} || MIC_{\text{A}}$

Receiving the response from Bob, Alice executes the following operations.

(1) According to N_{A}, Alice checks whether the message is fresh or not. If the message is fresh, Alice continues the user authentication process. Otherwise, Alice cancels the user authentication process.

(2) Alice construct Bob's public key as:

$$P_{\text{B}} = P_{\text{CA}} \cdot [(SHA(ID_{\text{B}} || R_{\text{B}}.x)) \bmod n] + R_{\text{B}} \tag{14}$$

(3) Alice computes the session key as:

$$K_{AB} = SHA((r_A + s_A) \cdot (C_B + P_B)) \tag{15}$$

and the message integrity code as:

$$MIC'_B = SHA(K_{AB} \| N_A \| N_B \| C_B \| ID_B \| ID_A \| R_B) \tag{16}$$

Alice compares MIC'_B with MIC_B. If $MIC'_B = MIC_B$, Alice passes the identity verification and regards Bob as a valid user.

Bob's identity verification works as follows.

$$\begin{aligned} K_{AB} &= SHA((r_A + s_A) \cdot (C_B + P_B)) \\ &= SHA(r_A \cdot C_B + r_A \cdot P_B + s_A \cdot C_B + s_A \cdot P_B) \\ &= SHA(r_A \cdot r_B \cdot G + r_A \cdot P_B + r_B \cdot P_A + s_A \cdot s_B \cdot G) \end{aligned}$$

$$\begin{aligned} K_{BA} &= SHA((r_B + s_B) \cdot (C_A + P_A)) \\ &= SHA(r_B \cdot C_A + r_B \cdot P_A + s_B \cdot C_A + s_B \cdot P_A) \\ &= SHA(r_A \cdot r_B \cdot G + r_A \cdot P_B + r_B \cdot P_A + s_A \cdot s_B \cdot G) \end{aligned}$$

Hence, we have $K_{AB} = K_{BA}$, and $SHA(K_{AB} \| N_A \| N_B \| C_B \| ID_B \| ID_A \| R_B) = SHA(K_{BA} \| N_A \| N_B \| C_B \| ID_B \| ID_A \| R_B)$ which implies the identity verification is valid.

(4) Alice computes $MIC_A = SHA(K_{AB} \| N_B \| ID_A \| ID_B)$, and returns($N_B$, ID_A, ID_B, MIC_A) to Bob.

Receiving the message from Alice, Bob executes the following operations.

(1) According to N_B, Bob checks whether the message is fresh or not. If the message is fresh, Bob continues the user authentication process. Otherwise, Bob cancels the user authentication process.

(2) Bob computes $MIC'_A = SHA(K_{BA} \| N_B \| ID_A \| ID_B)$, and compares it with $MIC_A = SHA(K_{AB} \| N_B \| ID_A \| ID_B)$ received from Alice. If $MIC'_A = MIC_A$, Bob regards that Alice has verified his identity. At the same time, the session key agreement is successful, and the session key can be used for future communication.

Since $K_{AB} = K_{BA}$, it is obvious that $MIC'_A = MIC_A$.

The interaction diagram of the user authentication phase mentioned above is illustrated in Figure 3.

The overall process of the proposed user authentication scheme is illustrated in Figure 4.

4. Security Analysis

The security of the proposed user authentication scheme is based on the intractability of reversing ECDLP and one-way hash function problem (OWHFP).

Let $E_p(a, b)$ be an elliptic curve over $\mathrm{GF}(p)$. P is a point with order n on the elliptic curve $E_p(a, b)$. Q is another point on the same curve.

The ECDLP is to determine m satisfying $Q = m \cdot P$ with given P and Q, which is difficult.

Let h be a one-way hash function. Given $h(x)$, it is computationally infeasible to find x. Furthermore, for a given value x and $h(x)$, it is computationally infeasible to find a y such that $h(y) = h(x)$.

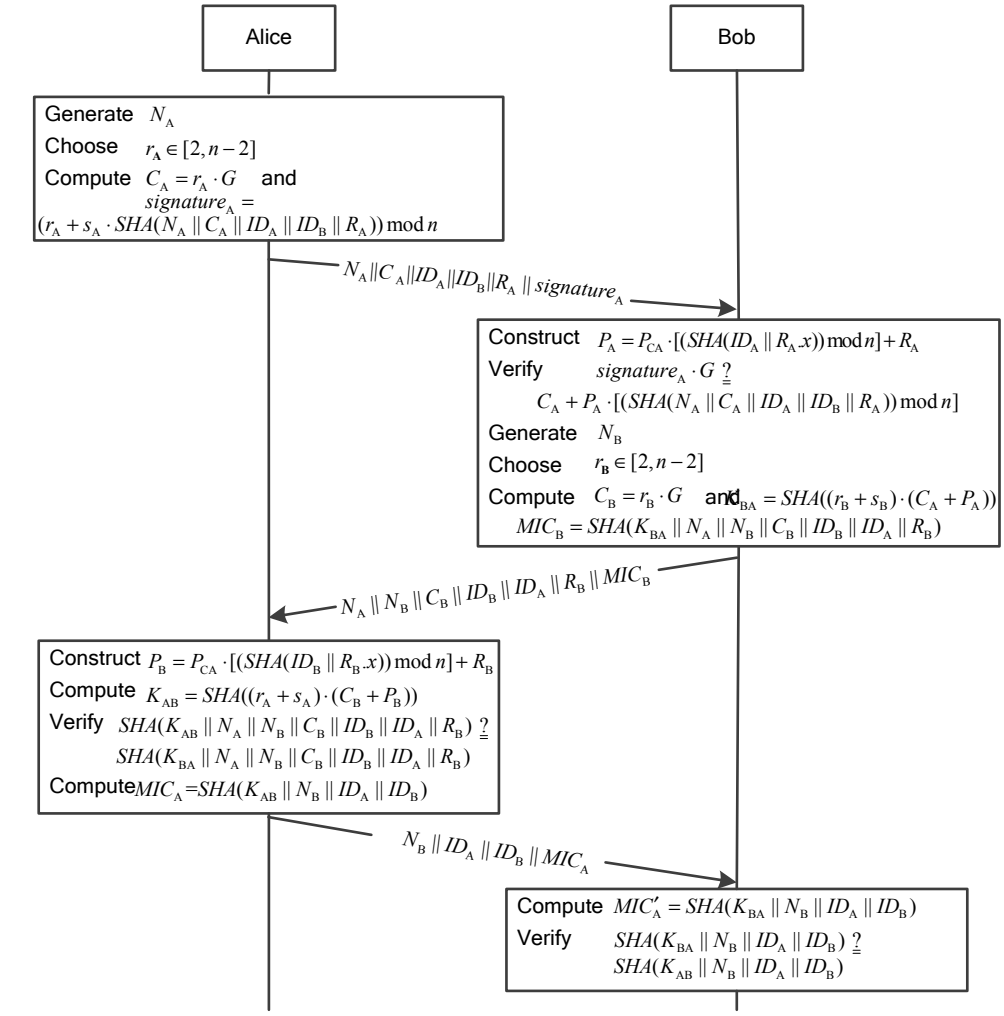

Figure 3. The user authentication phase.

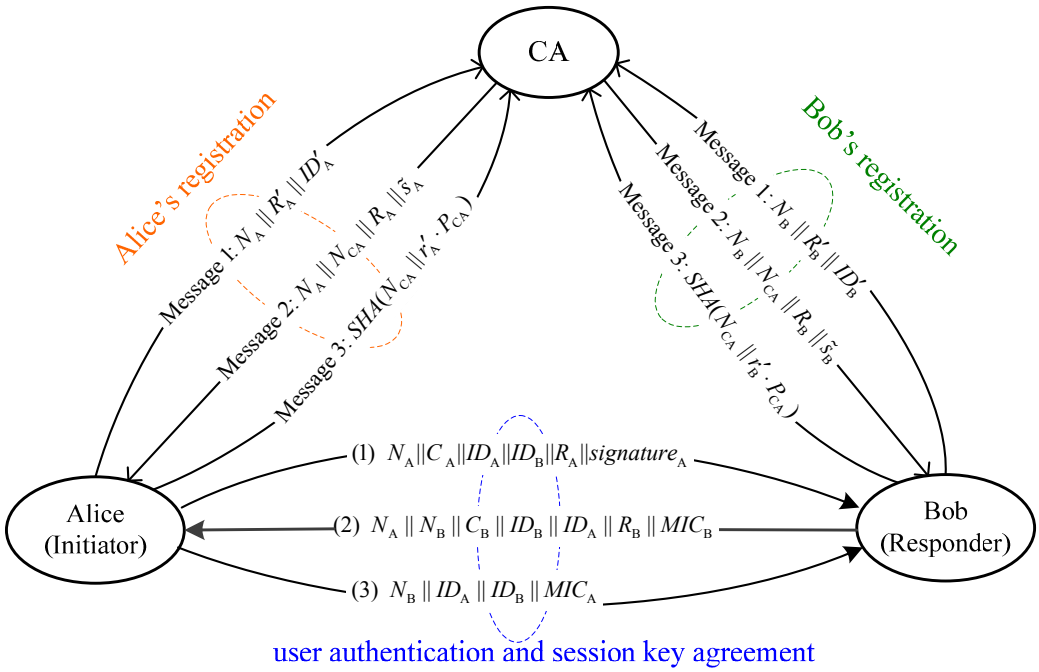

Figure 4. The proposed user authentication scheme.

4.1. Security Analysis in User Registration Phase

Theorem 1. *The proposed user authentication scheme is secure against user masquerade attack, message-forgery attack, impersonate attack from CA in user registration phase.*

Proof.

(1) User masquerade attack resistance

We assume that an adversary (Eve) intercepts the legal user's registration information and attempts to masquerade the legal user (U_i) to join in the network. However, Eve will be faced with some difficulties in following scenarios.

Although Eve intercepts ID_i', he cannot masquerade the valid user. Because U_i's identity is hidden in $ID_i' = ID_i \oplus SHA(r_i' || P_{CA})$. If Eve wants to obtain ID_i from ID_i', he must first obtain $SHA(r_i' || P_{CA})$ which is protected under the OWHFP and ECDLP.

Although Eve intercepts message $(N_i, N_{CA}, R_i, \tilde{s}_i)$ and wants to masquerade the valid user, he should derive r_i' from $R_i' = r_i' \cdot G$. It is not possible because solving the ECDLP is computationally infeasible. Meantime, he cannot return *Message 3* ($SHA(N_{CA} || r_i' \cdot P_{CA})$) to CA without the knowledge of r_i'.

Although Eve gets ID_i, he attempts to re-register with CA on the purpose of masquerading a valid user. Even if this attack is successful, the attack can be easily detected. This is because CA is convinced that the user has verified the authenticity of his public key since receiving Message 3. And CA stores the user's registration information in the registration file. As a registration request is accepted, CA will check the submitted user's identity information of the user in the registration file to prevent the re-registration attempt.

Therefore, our proposed scheme can resist the user masquerade attack.

(2) Message-forgery attack resistance

We assume that Eve intercepts $(N_i, N_{CA}, R_i, \tilde{s}_i)$ when CA returns it to U_i and attempts to forge $(R_i, \tilde{s}_i)$.

U_i verifies the condition $s_i \cdot G = P_{CA} \cdot [(SHA(ID_i, R_i.x)) \bmod n] + R_i$. The verification does not hold because Eve needs to have the private key of CA, P_{CA}. Hence, Eve should compute s_{CA} from $P_{CA} = s_{CA} \cdot G$. It is not possible because solving the ECDLP is computationally infeasible. Therefore, our proposed scheme can resist the message-forgery attack.

(3) Resistance of the impersonate attack from CA

We assume that CA generates another pair of valid private/public key, (s_i', P_i'), satisfying (9), CA can impersonate U_i. However, this fraud can be detected by U_i because two different valid keys exist. It can prove that CA is cheating. Therefore, our proposed scheme can resist the impersonate attack from CA.

4.2. Security Analysis in User Authentication Phase

Theorem 2. *The proposed user authentication scheme achieves mutual trust, and is secure against man-in-the-middle attack, replay attack, masquerading and tampering attacks in user authentication phase.*

Proof.

(1) Mutual trust

The signature of the message sent by Alice is generated in Step 1, which is verified by Bob in Step 2. In this way, Bob authenticates Alice's identity.

Moreover, a message integrity code of the message sent by Bob, $MIC_B = SHA(K_{BA} \| N_A \| N_B \| C_B \| ID_B \| ID_A \| R_B)$, is applied in Step 2. This provides the evidence of authentication and integrity for the message received by Alice. In the proposed scheme, MIC_B contains $K_{BA} = SHA((r_2 + s_B) \cdot (C_A + P_A))$ generated by Bob's private key. Hence, MIC_B can be used to authenticate Bob's identity.

Therefore, the proposed scheme provides the two-way authentication between Alice and Bob.

(2) Man-in-the-middle attack resistance

In the user registration phase, it prevents from the re-registration attempt so that adversaries can hardly masquerade other valid users to perform the man-in-the-middle attack.

In the user authentication phase, the proposed scheme exchanges $C_A = r_A \cdot G$ and $C_B = r_B \cdot G$ along with $signature_A$ and MIC_B, and generates the session keys, $K_{AB} = SHA((r_A + s_A) \cdot (C_B + P_B))$ and $K_{BA} = SHA((r_B + s_B) \cdot (C_A + P_A))$, using the private keys, s_A and s_B, and two random values, r_A and r_B. Man-in-the-middle attack is only possible if an adversary (Eve) can forge $signature_A$ and MIC_B. Hence, Eve must compute s_A and s_B from the pair $(P_A, P_B) = (s_A \cdot G, s_B \cdot G)$. It is not possible because solving the ECDLP is computationally infeasible.

Therefore, the proposed scheme can resist man-in-the-middle attack.

(3) Replay attack resistance

Two types of replay attacks are considered. Type-I replay attack is defined as an adversary intercepts an authentication message and attempts to masquerade as a sender by replaying it without modifying any content of the authentication message. Type-II replay attack is defined as an adversary intercepts an authentication message and replays a forged authentication message modified from the original one.

Since the proposed scheme uses the nonce to ensure the fresh of message, the type-I replay attack will be excluded by checking the nonce. If Eve intercepts the message $(N_A, C_A, ID_A, ID_B, R_A, signature_A)$ and replays it to impersonate Alice, Bob checks whether the message is fresh or not according to N_A. If the nonce has been received, Bob discards the message.

In order to pass the authentication of Alice, Eve must change the nonce. It is assumed that Eve only changes the nonce from N_A to N'_A in $(N_A, C_A, ID_A, ID_B, R_A, signature_A)$ to forge the authentication message. Bob verifies $signature_A \cdot G \stackrel{?}{=} C_A + P_A \cdot [(SHA(N'_A \| C_A \| ID_A \| ID_B \| R_A)) \bmod n]$. The message verification does not hold since Eve needs to have the private key of Alice, P_A, to generate a new signature. It is not possible because solving the ECDLP is intractable. In the same way, an adversary

impersonating Bob cannot pass the authentication. Hence, the nonce cannot be forged in the proposed scheme, which means that the proposed scheme is also resistant to the type-II replay attack.

Therefore, the proposed scheme can resist the replay attack.

(4) Masquerading and tampering attacks resistance

It is assumed that an adversary (Eve) intercepts an authentication message and replays it to masquerade as a valid user.

Eve intercepts an authentication message sent by Alice and attempts to masquerade as Alice by launching the type-I replay attack. After Bob receives the authentication message, he will check whether the message is fresh or not according to N_A. If the nonce has been received, Bob discards the message. On the other hand, Eve intercepts an authentication message and launches the type-II replay attack. It is difficult to succeed since Eve needs to use P_A to generate a new signature. Computing s_A from $P_A = s_A \cdot G$ is not possible because solving the ECDLP is computationally infeasible.

It is assumed that an adversary (Eve) intercepts the message $(N_A, C_A, ID_A, ID_B, R_A, signature_A)$ and attempts to tamper the message. This action will not pass the user authentication of Alice. As explained in the replay attack resistance, Eve needs to use P_A to generate a new signature. Hence, Eve encounters the intractability of solving the ECDLP. In addition, the one-way hash function is adopted in the user authentication phase to guarantee the integrity of message, which contains the session key generated by Alice and Bob's private keys. Computing (s_A, s_B) from (P_A, P_B) is not possible because solving the ECDLP is computationally infeasible.

Therefore, the proposed scheme can resist the masquerading and tampering attacks.

Theorem 3. *Based on the difficulty in solving the ECDLP, the proposed user authentication scheme provides perfect forward secrecy, backward secrecy, key compromise impersonation attack resistance, known-key security, unknown key-share resistance, and known session-specific temporary information attack resistance*.

Proof.

(1) Perfect forward secrecy and backward secrecy

It is assumed that the private keys, s_A and s_B, are compromised, and an adversary (Eve) attempts to compute the key $K_{AB} = SHA(r_A \cdot r_B \cdot G + s_B \cdot C_A + s_A \cdot C_B + s_A \cdot s_B \cdot G)$. Here, the forward secrecy is achieved by means of the term $r_A \cdot r_B \cdot G$. However, in order to compute the session key, Eve needs the knowledge of the random values, r_A and r_B. Solving C_A and C_B to get r_A and r_B is equivalent to the problem of solving ECDLP.

In addition, the session key relies on the random values, r_A and r_B, which are generated in each session independently and changed for each authentication phase.

Furthermore, another important aspect of our proposed scheme is that the session key is protected by the secure hash function. Although an adversary obtains a certain period session key, he/she cannot use the current session key to get forward and backward session keys. Hence, the session key in the proposed scheme achieves perfect forward secrecy and backward secrecy.

(2) Key compromise impersonation attack resistance

As defined in [15], the key compromise impersonation attack resistance is that an adversary (Eve) can masquerade as Alice if Alice's private key is compromised, while Eve cannot masquerade as another user to interact with Alice.

It is assumed that the long-term private key of Alice, s_A, is compromised and known to Eve. Obviously, Eve can impersonate Alice using s_A. However, to impersonate any other user (Bob) to interact with Alice, Eve would need the session key, $K_{BA} = SHA(r_B \cdot C_A + r_A \cdot P_B + r_B \cdot P_A + s_A \cdot P_B)$. Thus, Eve needs to have the private key of Bob, s_B, or the random value generated by Alice, r_A. Solving P_B and C_A to get s_B and r_A is equivalent to the problem of solving ECDLP. In addition, in most circumstances, the private key of a user is updated periodically.

Hence, the key compromise impersonation vulnerability can be limited to some considerably low extent.

(3) Known-key security

The proposed scheme achieves the known-key security if the knowledge of previous generated session keys does not allow an adversary to compromise the past or future session keys.

It is assumed that a session key generated by the proposed scheme is obtained by an adversary (Eve). Eve cannot derive all past and future session keys from the knowledge of the compromised session key. To derive a session key, Eve has to compute (r_A, r_B) and (s_A, s_B) from (C_A, C_B) and (P_A, P_B), respectively. It is not possible because solving the ECDLP is computationally infeasible.

(4) Unknown key-share resistance

A key agreement protocol achieves unknown key-share attack resistance if a user cannot be forced to share a session key with a different user rather than the one intended without their knowledge. That is, Alice cannot be forced to share a key with Eve when Alice believes that the key is shared with Bob.

In the user authentication phase of the proposed scheme, Bob sends a message to Alice, $N_A \| N_B \| C_B \| ID_B \| ID_A \| R_B \| MIC_B$. And MIC_B contains $K_{BA} = SHA((r_B + s_B) \cdot (C_A + P_A))$ generated by Bob's private key, s_B. Similarly, Alice responds to Bob with the message, $N_B \| ID_A \| ID_B \| MIC_A$. And MIC_A contains $K_{AB} = SHA((r_A + s_A) \cdot (C_B + P_B))$ generated by Alice's private key s_A. The verification of MIC_B and MIC_A at Alice and Bob confirms the generation of same session key.

Therefore, the proposed scheme resists the unknown key-share attack.

(5) Known session-specific temporary information attack resistance

The security of the generated session key should not be compromised even if two random values are compromised by an adversary (Eve).

In the proposed scheme, Eve cannot derive the session key $K_{AB} = SHA((r_A + s_A) \cdot (C_B + P_B))$ and $K_{BA} = SHA((r_B + s_B) \cdot (C_A + P_A))$ even if r_A and r_B are compromised. This is because Eve does not know Alice's private key and Bob's private key, s_A and s_B. Moreover, Eve cannot derive from (P_A, P_B)=($s_A \cdot G$, $s_A \cdot G$) because solving the ECDLP is computationally infeasible.

Therefore, the proposed scheme resists the known session-specific temporary information attack.

5. Performance Analysis

In this section, we analysis the performance of the proposed user authentication scheme, in terms of security, storage overhead, communication overhead and computation overhead.

(1) Attack resistance and functionality

The attack resistance and functionality of the proposed user authentication scheme are compared with other three schemes, namely Diffie-Hellman key agreement scheme in [4] (abbreviated as DHKA scheme), the user authentication phase of secure MAC protocol for cognitive radio networks in [5] (abbreviated as SecureMAC protocol), and authentication and key agreement scheme in [11] (abbreviated as AKA scheme).

The comparison results are listed in Table 2. From Table 2, we observe that our proposed user authentication scheme provides two-way user authentication and session key agreement. However, SecureMAC protocol in [5] does not achieve the session key agreement.

Table 2. The functionality comparison.

Functionality	DHKA Scheme in [4]	SecureMAC Protocol in [5]	AKA Scheme in [11]	Proposed Scheme
Mutual trust	Yes	Yes	Yes	**Yes**
Session key agreement	Yes	No	Yes	**Yes**
Time synchronization	Not need	Not need	Need	**Not need**
Replay attack resistance	No	Yes	Yes	**Yes**
Man-in-the middle attack resistance	Yes	Yes	Yes	**Yes**
Forward secrecy	No	No	Yes	**Yes**
Backward secrecy	No	No	Yes	**Yes**
Key compromise impersonation attack resistance	No	No	No	**Yes**

Moreover, the session key of our proposed scheme achieves perfect forward secrecy and backward secrecy, and key compromise impersonation attack resistance compared with DHKA scheme in [4] and AKA scheme in [11].

In addition, our proposed scheme also defends against the replay attack with modified challenge-response mechanism, but DHKA scheme in [4] is vulnerable to the replay attack. AKA scheme in [11] defends against the replay attack using timestamp mechanism.

(2) Storage overhead

Each user needs store parameters (p, a, b, G, n, P_{CA}, R_i, ID_i) and the private/public key pair (s_i, P_i). In our proposed scheme, we assume that the key length of ECC is 160 bits, and the length of ID value is 160 bits. The storage overhead of each user is listed in Table 3.

Table 3. Storage overhead of each user.

Parameters	Storage Overhead (bits)
The parameters of ECC, (p, a, b, G, n)	960/(160 + 160 + 160 + 320 + 160)
CA's public key, P_{CA}	320
Point R_i	320
User identity, ID_i	160
User's private key, s_i	160
User's public key, P_i	320
Total	**2240**

The total storage overhead is only 2,240 bits, which is quite suitable for resource-constrained wireless network.

For security, the private key of U_i, s_i, needs to be stored in the form of ciphertext, and the public key of U_i, P_i, and other parameters, (p, a, b, G, n, P_{CA}, R_i, ID_i) are stored in the form of plaintext. Since other users can use n, P_{CA}, R_i and ID_i to construct the public key of U_i, P_i, users does not need to store the public keys of other users with whom he/she is communicating. In addition, since the generated session key between two users is temporary, it does not need to be stored.

(3) Communication overhead

Let the length of nonce be 64 bits, and the hash value of the one way hash function is 256 bits. The communication overhead in the user authentication phase of our proposed scheme is listed in Table 4.

Table 4. Communication overhead of each user.

Message	Communication Overhead (bits)
Step 1	1184
Step 2	1344
Step 3	640
Total	**3168**

From Table 4, it is obvious that the communication overhead in the user authentication phase of our proposed scheme is relatively light.

(4) Computation overhead

The computational complexity is analyzed in detail and compared with some other user authentication schemes, namely DHKA scheme in [4], AKA scheme in [11], time stamp mechanism and key management scheme in [16] (abbreviated as TSMKM scheme), authentication scheme based on bilinear pairings) in [17] (abbreviated as BP-A scheme), ECC-based authentication key agreement scheme in [18] (abbreviated as ECC-AKA scheme), and ECC-based improved authentication key agreement scheme in [19] (abbreviated as ECC-IAKA scheme).

The notations of various operations and the denotations used in this subsection are listed in Table 5.

Table 5. Definition of various operations.

Notations	Denotations
T_{EM}	The time for computing a point multiplication on GF(p)
T_{EA}	The time for computing a point addition on GF(p)
T_{BP}	The time for computing a bilinear pairing
T_{MI}	The time for computing modular inversion
T_{MM}	The time for computing modular multiplication
T_{MA}	The time for computing modular addition
T_{ME}	The time for computing modular exponentiation
T_{H}	The time for computing the one-way hash function
$T_{RSA\text{-}Ver}$	The time for computing RSA signature verification operation
T_{X}	The time for computing symmetric encryption/decryption operation

According to [19–23], $T_{BP} \approx 3T_{EM}$, $T_{EM} \approx 29T_{MM}$, $T_{EA} \approx 0.12T_{MM}$, $T_{ME} \approx 240T_{MM}$, and $T_{MI} \approx 3T_{MM}$. Compared to the computational time for performing other operations, the time for performing the modular addition and one-way hash function can be negligible. The comparison of computation overhead is listed in Table 6.

As shown in Table 6, our proposed user authentication scheme does not involve modular exponentiation and bilinear pairing operations, while DHKA scheme in [4] and the BP-A scheme in [17] require two modular exponentiation operations and three bilinear pairing operations, respectively. Meanwhile, our proposed scheme reduces the amount of point multiplication operations compared with the AKA scheme in [11], the TSMKM scheme in [16] and the ECC-IAKA scheme in [19]. The ECC-AKA scheme in [18] utilizes both RSA and ECC to achieve mutual authentication, which increases the computation burden on user's side. Hence, the computation overhead of our proposed scheme is obviously less than that of other compared schemes.

Table 6. Computation overhead of each user.

Schemes	Computation Overhead	Equivalent Computation Overhead
DHKA scheme in [4]	$2T_{ME}$	$480T_{MM}$
AKA scheme in [11]	$15T_{EM} + 4T_{MM} + 4T_{MA} + 6T_{EA} + T_{MI} + 6T_{H}$	$442.72T_{MM}$
TSMKM scheme in [16]	$15T_{EM} + 5T_{EA} + 2T_{MI} + 4T_{MM} + T_{MA} + 8T_{H}$	$445.6T_{MM}$
BP-A scheme in [17]	$2T_{EM} + 3T_{BP} + 8T_{H}$	$319T_{MM}$
ECC-AKA scheme in [18]	$10T_{EM} + 4T_{EA} + 8T_{MA} + 8T_{MM} + 4T_{MI} + 10T_{H} + 2T_{RSA\text{-}Ver}$	$310.48T_{MM} + 2T_{RSA\text{-}Ver}$
ECC-IAKA scheme in [19]	$17T_{EM} + 5T_{EA} + 3T_{H} + T_{X}$	$493.6T_{MM} + T_{X}$
Our proposed scheme	$8T_{EM} + 5T_{EA} + 3T_{MA} + T_{MM} + 10T_{H}$	$233.6T_{MM}$

Moreover, as the performance analysis in [24], some parameters can be pre-computed to reduce the computational complexity. In our proposed scheme, C_A and C_B can be computed in advance. In this way, the computational complexity can be reduced in some extent.

In addition, if some applications require lower computational complexity, a higher clock frequency for hardware implementations or binary-field based elliptic curves [25] can be selected for our proposed scheme.

6. Conclusions

The WANET will play an important role in the next generation wireless networking. In addition, security issue is critical to deploy and manage WANETs. Furthermore, the user authentication is the first safety barrier in a network.

We proposed a user authentication scheme based on SCPK and ECC for the WANET, in which an efficient two-way user authentication and a secure session key agreement are achieved. Based on the security and performance analysis, our proposed scheme resists various common known attacks, such as man-in-the-middle attack, replay attack, masquerading and tampering attacks, as well as achieves lower storage, communication, and computation overheads. Therefore, the proposed user authentication scheme based on SCPK and ECC is efficient and suitable for the resource-constrained WANET.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China (No. 61071127, No. 61471318), and the Fundamental Research Funds for the Central Universities.

Author Contributions

Huifang Chen, Linlin Ge and Lei Xie proposed the user authentication scheme based on SCPK and ECC, and compared the performance; Huifang Chen and Linlin Ge wrote the paper.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Kumaar, S.S.; Mangai, M.; Fernado, N.; Daniel, J.V. A survey of various attacks in mobile ad hoc networks. *Int. J. Comput. Sci. Mob. Comput.* **2013**, *2*, 171–185.
2. Bechler, M.; Hof, H.J.; Kraft, D.; Pahlke, F.; Wolf, L. A cluster-based security architecture for ad hoc networks. In Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), Hongkong, China, 7–11 March 2004; pp. 2393–2403.
3. Deng, H.; Mukherjee, A.; Agrawal, D.P. Threshold and identity-based key management and authentication for wireless ad hoc networks. In Proceedings of Information Technology: Coding and Computing (ITCC 2004), Las Vegas, NV, USA, 5–7 April 2004; pp. 107–111.
4. Zhu, X.; Xu, S. A new authentication scheme for wireless ad hoc network. In Proceedings of the 2012 International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII 2012), Sanya, China, 20–21 October 2012; pp. 312–315.
5. Alhakami, W.; Mansour, A.; Safdar, G.A.; Albermany, S. A secure MAC protocol for cognitive radio networks (SMCRN). In Proceedings of the 2013 Science and Information Conference (SAI 2013), London, UK, 7–9 October 2013; pp. 796–803.

6. Kohnfelder, L. Towards a Practical Public-Key Cryptosystem. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, UK, 1978.
7. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of Advances in Cryptology (CRYPTO 84), Berlin, Germany, 19–22 August 1984; pp. 47–53.
8. Girault, M. Self-certified public keys. In Proceedings of Advances in Cryptology (EUROCRYPT 91), Brighton, UK, 8–11 April 1991; pp. 490–497.
9. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63.
10. Jing, C.; Li, B.; Xu, H. An efficient scheme for user authentication in wireless sensor networks. In Proceedings of the 21th IEEE International Conference on Advanced Information Networking and Applications Workshops (AINAW 2007), Niagara Falls, ON, Canada, 21–23 May 2007; pp. 438–442.
11. Zhao, X.; Lv, Y.; Yeap, T.H.; Hou, B. A novel authentication and key agreement scheme for wireless mesh networks. In Proceedings of the 5th IEEE International Joint Conference on INC, IMS and IDC (NCM 2009), Seoul, Korea, 25–27August 2009; pp. 471–474.
12. Zhang, C.; Wang, X. A novel self-certified security access authentication protocol in the space network. In Proceeding of the 2012 IEEE International Conference on Communication and Technology (ICCT 2012), Chengdu, China, 9–11 November 2012; pp. 635–639.
13. Petersen, H.; Horster, P. Self-certified keys concepts and applications. *Commun. Multimed. Secur.* **1997**, *3*, 102–116.
14. Daniel, R.L.B. *Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters*; Certicom Corp.: Mississauga, ON, Canada, 2012.
15. Giruka, V.; Chakrabarti, S.; Singhal, M. A distributed multi-party key agreement protocol for dynamic collaborative groups using ECC. *J. Parallel Distrib. Comput.* **2006**, *66*, 959–970.
16. Indra, G.; Taneja, R. A time stamp-based elliptic curve cryptosystem for wireless ad-hoc sensor networks. *Int. J. Space-Based Situat. Comput.***2014**, *4*, 39–54.
17. Zhang, J.; Li, X.; Ma, J.; Wang, W. Secure and efficient authentication scheme for mobile sink in WSNs based on bilinear pairings. *Int. J. Distrib. Sens. Netw.***2014**, *2014*, 1–11.
18. Ammayappan, K.; Negi, A.; Sastry, V; Das, A. An ECC-based two-party authenticated key agreement protocol for mobile ad hoc networks. *J. Comput.* **2011**, *6*, 2408–2416.
19. Li, X.; Wen, Q.; Zhang, H; Jin, Z. An improved authentication with key agreement scheme on elliptic curve cryptosystem for global mobility networks. *Int. J. Netw. Manag.* **2013**, *23*, 311–324.
20. Wu, T; Hsu, C; Lin, H. Self-certified multi-proxy signature schemes with message recovery. *J. Zhejiang Univ.* **2009**, *10*, 290–300.
21. Babamir, F.S; Norouzi, A. Achieving key privacy and invisibility for unattended wireless sensor networks in healthcare. *Comput. J.* **2014**, *57*, 624–635.
22. Holbl, M.; Welzer, T.; Brumen, B. An improved two-party identity-based authenticated key agreement protocol using pairings. *J. Comput. Syst. Sci.* **2012**, *78*, 142–150.

23. Tsaur, W.J.; Yeh, Y. A novel mobile agent authentication scheme for multi-host environments using self-certified pairing-based public key cryptosystem. *Int. J. Innov. Comput. Inf. Control* **2011**, *7*, 2389–2404.
24. Jiang, Y.; Lin, C.; Shen, X.; Shi, M. Mutual authentication and key exchange protocols for roaming services in wireless mobile networks. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 2569–2577.
25. Wenger, E. Hardware architectures for MSP430-based wireless sensor nodes performing elliptic curve cryptography. In Proceedings of the 11th International Conference on Applied Cryptography and Network Security (ACNS 2013), Banff, AB, Canada, 25–28 June 2013; pp. 290–306.