Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 2484268, 9 pages
https://doi.org/10.1155/2018/2484268

Research Article

A Provably Secure Anonymous Authenticated Key Exchange Protocol Based on ECC for Wireless Sensor Networks

Ke Zhang[1,2] **Kai Xu**[3] **and Fushan Wei**[2]

[1]*Network Information Center, Shaanxi Normal University, Xi'an 710062, China*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China*
[3]*School of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China*

Correspondence should be addressed to Ke Zhang; kezhang2017@163.com

Received 22 January 2018; Revised 7 April 2018; Accepted 19 April 2018; Published 16 July 2018

Academic Editor: Ding Wang

In wireless sensor networks, users sometimes need to retrieve real-time data directly from the sensor nodes. Many authentication protocols are proposed to address the security and privacy aspects of this scenario. However, these protocols still have security loopholes and fail to provide strong user anonymity. In order to overcome these shortcomings, we propose an anonymous authenticated key exchange protocol based on Elliptic Curves Cryptography (ECC). The novel protocol provides strong user anonymity such that even the gateway node and the sensor nodes do not know the real identity of the user. The security of the proposed protocol is conducted in a well-defined security model under the CDH assumption. Compared with other related protocols, our protocol is efficient in terms of communication and enjoys stronger security. The only disadvantage is that our protocol consumes more computation resources due to the usage of asymmetric cryptography mechanisms to realize strong anonymity. Consequently, our protocol is suitable for applications which require strong anonymity and high security in wireless sensor networks.

1. Introduction

1.1. Background. As an important part of Internet of things, wireless sensor networks (WSNs) draw more and more attentions from the industrial and the researchers. Typically, a WSN is composed of numerous tiny sensor nodes. These sensor nodes can be deployed in unattended or hostile environments to collect valuable data of interest. For example, a large amount of visual data such as images and videos can be collected by visual sensor nodes [1]. These nodes have the characteristics of easy deployment, low cost, and high mobility [2]. Due to these merits, WSNs are very useful in many application scenarios, such as natural disaster prevention, machine health monitoring, air temperature monitoring, health care monitoring, and battlefield surveillance.

Usually, the data collected by the sensor nodes will be transmitted to and aggregated by a gateway node periodically. Whenever a user wants to get access to the aggregated data from the gateway node, he will authenticate himself to the gateway node. However, in some application scenarios such as health care monitoring and battlefield surveillance, users have great needs to access the data directly from the sensor nodes. Under such circumstances, the user first sends a request to the gateway node for accessing the real-time data. The gateway will authenticate whether the user is valid or not. If the user is valid, a common session key will be established between the user and the sensor node with the help of the gateway node. The session key can later be used to protect confidentiality and integrity of the data [3].

1.2. Related Work. In order to address the security aspects under the above-mentioned application scenario, many authentication protocols are proposed [4–7]. In 2009, Das [8] proposed a two-factor user authentication protocol, which is claimed to have strong authentication and session key establishment and achieves efficiency. Unfortunately, Khan et al. [9] soon found that Das's protocol is vulnerable to the gateway node bypassing attack and the privileged-insider attack. Besides, Das's protocol also fails to provide password update and mutual authentication. Khan et al. also presented

an improved protocol to fix the susceptibilities of the original protocol. In 2011, Yeh et al. [10] pointed out several weaknesses of Das's protocol. They also design an ECC-Based authentication protocol to meet the needs of applications with higher security requirements. In 2013, Xue et al. [11] proposed a temporal-credential-based mutual authentication scheme among the user, the gateway node, and the sensor node. A credential is issued by the gateway node to each user for authentication. Their protocol only involves lightweight operations such as XOR and hash and is suitable for resource-constrained WSNs. Nevertheless, He et al. [12] soon pointed out that Xue et al.'s protocol is vulnerable to the offline password guessing attack, the user impersonation attack, the sensor node impersonation attack, and the modification attack. They also proposed an improved temporal-credential-based protocol to remedy the weaknesses. Yuan et al. [13] proposed an authentication scheme for WSNs based on pubic key mechanism and biometric characteristics of the user to realize strong authentication. In 2014, Wang et al. [14] analyzed two authentication schemes for WSNs and demonstrated several loopholes. They also investigate the underlying rationale of the security failures and put forward three basic principles for designing secure authentication protocols in WSNs. In 2016, Shen et al. [15] proposed an efficient multilayer authentication protocol and a secure session key generation method for WSNs. They also designed a one-to-many group authentication protocol and a certificate-less authentication protocol, which is of independent interest.

Recently, researchers begin to focus on user's privacy protection in WSNs. Wu et al. [16] proposed an anonymous authentication scheme based on ECC for WSNs with formal security proof. Jiang et al. [17] designed an anonymous lightweight three-factor authentication scheme for WSNs. The security of their protocol is conducted using ProVerif. Wang et al. [20] put forward a new authentication which can resist all known attacks for WSNs. Moreover, they explore the design principle of authentication schemes. They also designed a biometric-based authentication scheme and proved its security using the Burrows-Abadi-Needham (BAN) logic [18]. Li et al. [19] proposed a three-factor anonymous authentication scheme for WSNs. They use fuzzy commitment to deal with the user's biometric template.

1.3. Motivation and Contribution. Until now, there are many authentication protocols designed for protection of security and privacy when accessing real-time data in WSNs. However, there are still some problems which are not solved. Firstly, most of these protocols only have informal heuristic security arguments. It is quite common such that a protocol which is claimed to be secure is soon found to be vulnerable to several attacks. What is worse, the improved protocol still has many vulnerabilities. Secondly, the existing protocol pays little attention to user's privacy. Only few protocols provide user anonymity and these protocols only achieve weak anonymity; i.e., the real identity is hidden to an adversary but is known to the gateway node and sometimes even the sensor node knows the real identity of the user. Last but not least, the existing protocols rely on efficient XOR, symmetric encryption, and hash operations to provide better efficiency. Although these protocols can meet the characteristics of constrained storage, computation, and communication capabilities of sensor nodes, they fail to provide strong security guarantee. For security-critical applications such as battlefield surveillance, security and privacy are more important than the computation and communication efficiency.

In this paper, we investigate the design of anonymous and strongly secure authenticated key exchange protocol in WSNs. We propose an efficient authenticated key exchange protocol for the scenario in which the user wants to access the real-time data directly from the sensor node. The novel protocol has the following advantages. First of all, our protocol enjoys formal security proof in a well-defined security model. The security is conducted in the random oracle model under the CDH assumption. Second, our protocol provides the strongest anonymity in the sense the real identity of the user is only known to himself. Neither the gateway node nor the sensor node can obtain any information of the user's identity, not to mention the adversary. Thirdly, our protocol achieves more security attributes than other related protocols. Consequently, it is more secure than other related protocol and is particularly suitable for security-critical applications in WSNs. The only disadvantage is that it needs more computation resources. However, security and privacy are more important than the computation efficiency in security-critic application. So our protocol is suitable for security-critical applications in WSNs.

The rest of the paper is organized as follows. In Section 2, we present the security model and some preliminaries. We describe the details of the proposed protocol in Section 3. The security proof is given in the random oracle model in Section 4. The performance comparison with other related protocols is summarized in Section 5. We conclude this paper in Section 6.

2. Security Model

In this section, we briefly recall the security model presented in [21, 22]. The security of our protocol will be conducted in this formal security model.

Protocol Participants. The participants of an authentication and key exchange protocol for real-time data retrieval in WSNs involves users U, a gateway node GN, and a sensor node S_i. Each user U registers with the gateway node and each sensor node S_i shares a common secret key with the gateway node.

Protocol Execution. All the participants are modeled as a PPT Turing machine. The i-th instance of a participant P is denoted by P^i. All the communication channels are managed by a probabilistic polynomial time adversary $\mathcal{A}$. The adversary $\mathcal{A}$ can intercept, delay, modify, and even forge a message at will. The capabilities of the adversary are captured through oracle queries. The adversary can make the following oracle queries:

(i) $Execute(U^x, GN^y, S_i^z)$: the execution query captures the passive eavesdrop ability of $\mathcal{A}$. In reply to this oracle query, $\mathcal{A}$ will get all the transcripts of

the authentication instance executed among a user instance U^x, a gateway node instance GN^y, and a sensor node instance S_i^z.

(ii) $Send(P^i, m)$: the send query captures the active attack ability of $\mathscr{A}$. Through the $Send(P^i, m)$ query, $\mathscr{A}$ sends a modified or forged message m to instance P^i in the name of another participant instance. $\mathscr{A}$ will get the message generated by the participant instance P^i upon receiving the message m according to the description of the protocol. The participant P can be a user, a gateway node, or a sensor node.

(iii) $Corrupt(U, PW)$: this query captures the compromise of the user's password. The adversary $\mathscr{A}$ only gets the password of the victim user; it can neither control nor compromise the credential of the user.

(iv) $Corrupt(U, cred)$: this query captures the compromise of the user's terminal. The adversary $\mathscr{A}$ can extract the credential issued by the gateway node and control the victim user's terminal. However, the password of the user is still unknown to $\mathscr{A}$.

(v) $Corrupt(S_i)$: this query captures the compromise of a sensor node S_i. The adversary $\mathscr{A}$ will get the secret key and control the sensor node through this query.

(vi) $Reveal(P^i)$: this query can only be asked to a user instance or a sensor node instance. If the instance P^i accepts the session and generates a session key, $\mathscr{A}$ will get the session key. Otherwise, $\mathscr{A}$ will get the symbol $\perp$ which means the instance P^i does not hold a session key.

(vii) $Test(P^i)$: this query does not capture any real attack ability of $\mathscr{A}$ but is used to measure the security of the session key held by instance P^i. Upon receiving this query, the simulator will flip a coin b. If the result is 1, then it returns the real session key to $\mathscr{A}$. If the result is 0, the simulator will send a random session key of the same length with the real session key to $\mathscr{A}$. $\mathscr{A}$ has to distinguish if the key is real or random. In other words, $\mathscr{A}$ has to guess the coin flip result.

The session identification (sid) is defined as the transcripts shared between a user instance and a sensor node instance. The partner identification (pid) of an instance is defined to be the participant with whom the instance wants to establish a common session key. We say a user instance U^x and a sensor node instance S_i^z are partners if the following conditions are satisfied: (1) these two instances both accept and generate the same session key; (2) these two instances share the same sid; (3) the pid of U^x is S_i and the pid of S_i^z is U; and (4) no other instances accept the same sid with U^x and S_i^z.

If the adversary $\mathscr{A}$ asks both $Corrupt(U, cred)$ and $Corrupt(U, PW)$, the user MU is said to be fully corrupted. When defining the AKE security of the session key, we do not consider the corruption of the gateway node. This is because once the gateway node is corrupted; there is nothing we can do to guarantee the security of the protocol. A user instance or a sensor node instance P^i is said to be fresh if (1) $\mathscr{A}$ does not send *Reveal* queries to the instance or its partner; and (2) the user or the sensor node is not fully corrupted by $\mathscr{A}$.

AKE Security. The security of the session keys is captured by the AKE security. The adversary $\mathscr{A}$ is restricted to ask *Test* queries to fresh instances only; otherwise the adversary $\mathscr{A}$ can trivially win the attack game. The adversary $\mathscr{A}$ is given access to all the oracle queries; the only restriction is that $\mathscr{A}$ only can ask one *Test* query to a fresh instance. The adversary $\mathscr{A}$ needs to guess the hidden bit b used by the simulator when answering the *Test* query. If $\mathscr{A}$ correctly guesses the random bit, then we say $\mathscr{A}$ wins the AKE security game. We denote this event by *Succ*. With respect to the distribution of the passwords, we use the Zipf's law put forward by Wang et al. [21] instead of assuming a uniform distribution. The adversary $\mathscr{A}$'s advantage in attacking the AKE security of a protocol $\mathscr{P}$, when passwords are chosen according to the Zipf's law of a dictionary D, is defined as follows:

$$Adv_{\mathscr{P},\mathscr{D}}^{ake}(\mathscr{A}) = 2 \cdot Pr[Succ] - 1 \quad (1)$$

An authentication and key exchange protocol $\mathscr{P}$ is said to be AKE secure if for all PPT adversary $\mathscr{A}$, the advantage $Adv_{\mathscr{P},\mathscr{D}}^{ake}(\mathscr{A})$ is only negligible larger than $C' \cdot q_{send}^{s'}$, where C' and s' are Zipf parameters and q_{send} is the number of active attack sessions. Moreover, C' and s' are constants depending on the password data set and can be calculated by linear regression.

3. Description of the Protocol

In this section, we describe the proposed anonymous authenticated key exchange protocol based on ECC for WSNs. The most important benefit of ECC is that it provides the same level of security with a smaller key size compared to other cryptography mechanisms such as RSA. So it suits the needs of the resource-constrained nature of the WSN. Our protocol has three phases: the setup phase, the registration phase, the authentication, and key exchange phase. The detailed steps of each phase are described in the following. The symbols used in this paper are summarized in Table 1.

3.1. The Setup Phase. Let p be a large prime and F_p be a finite field of prime order p. Let E be an elliptic curve cryptosystem satisfying the equation $y^2 = (x^3 + ax + b) \bmod p$ such that $a, b \in F_p$ and $4a^3 + 27b^2 \bmod p \neq 0$. The set of rational points in E over finite field F_p is denoted by $E(F_p)$. More precisely, $E(F_p) = \{(x; y) : x, y \in F_p$ such that $y^2 = (x^3 + ax + b) \bmod p\} \cup \{O\}$, where O is the point at infinity. Let G be a cyclic group generated by P, where $P \in E(F_p)$ has a large prime order n. These parameters $(F_p, E, E(F_p), G, P)$ are the system parameters and can be chosen by a trusted third party or the gateway node. The gateway node (GN) chooses a random number $s_{GN} \in Z_n^*$ as his private key and computes the corresponding public key $Q_{GN} = s_{GN}P$. The public key Q_{GN} is published in the whole network. Define six hash functions such that $H_1 : \{0,1\}^* \longrightarrow Z_n^*$,

Table 1: Notations.

notation	meaning	notation	meaning
ID_{GW}	identity of the gateway node	ID_U	identity of the user U
ID_{S_i}	identity of the sensor node S_i	p, n	large prime numbers
F_p	a finite field	E	an elliptic curve defined on F_p
$E(F_p)$	the set of rational points in E	s_{GN}	secret key of the gateway node
PW_U	the password of the user U	$\oplus$	exclusive OR
$\|$	concatenation	$h(m)$	cryptographic hash of message m
$sign_{s_{GN}(m)}$	signature of m singed by s_{GN}	T_{GN/S_i}	timestamp of GN/S_i

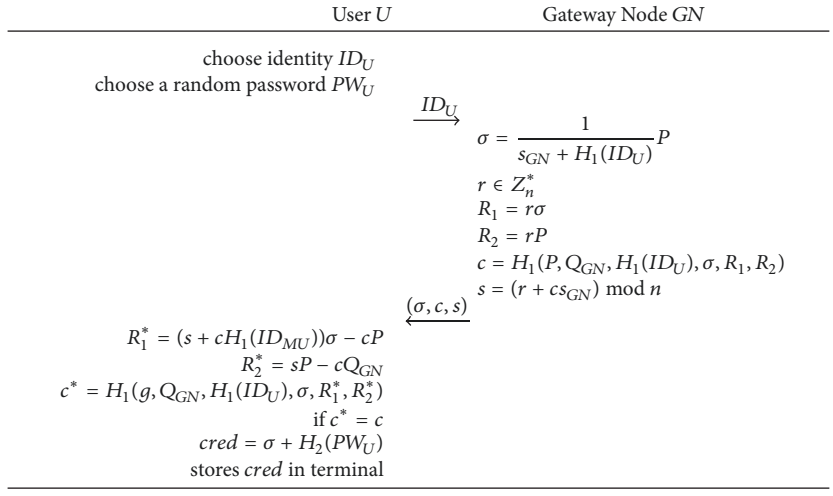

Figure 1: Registration phase of mobile user.

$H_2 : \{0,1\}^* \longrightarrow G^*$, $H_0, H_3, H_4, H_5 : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$, where κ is the security parameter. All these parameters $(F_p, E, E(F_p), G, P, Q_{GN}, H_i(i = 0, 1, \ldots, 5))$ are available to all the entities in the WSN.

3.2. The Registration Phase. If a user U wants to access the data collected by the sensor nodes in the WSN, U has to register himself to the gateway node. For a pictorial illustration of the user registration, please refer to Figure 1. The detailed steps are described in the following.

Step 1. The user U randomly chooses his identity ID_U and his password PW_U from the password dictionary. U sends his identity ID_U to the gateway node GN through a secure channel.

Step 2. When the gateway node GN receives the registration request from the user, GN verifies the validity of the $U's$ identity ID_U. If it is valid and there is no other user in its database registers using the same identity, GN first computes the credential $\sigma = (1/(s_{GN} + H_1(ID_U)))P$. Then GN chooses a random number $r \in Z_n^*$ and computes $c = H_1(P, Q_{GN}, H_1(ID_U), \sigma, R_1, R_2)$ and $s = (r + cs_{GN}) \bmod n$, where $R_1 = r\sigma$, $R_2 = rP$. At last, GN sends the registration message (σ, c, s) to the user U through a secure channel.

Step 3. When the user U receives the registration message (σ, c, s) from GN, U will verify the validity of the message. U computes $R_1^* = (s + cH_1(ID_U))\sigma - cP$, $R_2^* = sP - cQ_{GN}$, and $c^* = H_1(g, Q_{GN}, H_1(ID_U), \sigma, R_1^*, R_2^*)$. U verifies whether c^* is equal to c or not. If the verification is successful, U will accept σ as a valid credential. Finally, U computes $cred = \sigma + H_2(PW_U)$ and then stores his password-protected credential $cred$ in his terminal.

The registration of the sensor node is rather simple compared with the user registration. The sensor node S_i sends the registration request to the gateway node GN through a secure channel. Upon receiving the request, the gateway node GN will compute a symmetric key $K_{(GN,S_i)} = H_3(GN, S_i, s_{GN})$ and send the symmetric key $K_{(GN,S_i)}$ to S_i through a secure channel.

3.3. The Authentication and Key Exchange Phase. Suppose a user U wants to get the real-time data from the sensor node S_i, U has to execute the authentication and key exchange phase with the gateway node GN and the sensor node S_i. During this phase, the user U, the gateway node GN, and the sensor node S_i will authenticate each other. At the end of this phase, a session key will be established between U and S_i to protect the upcoming data transmission. The detailed steps

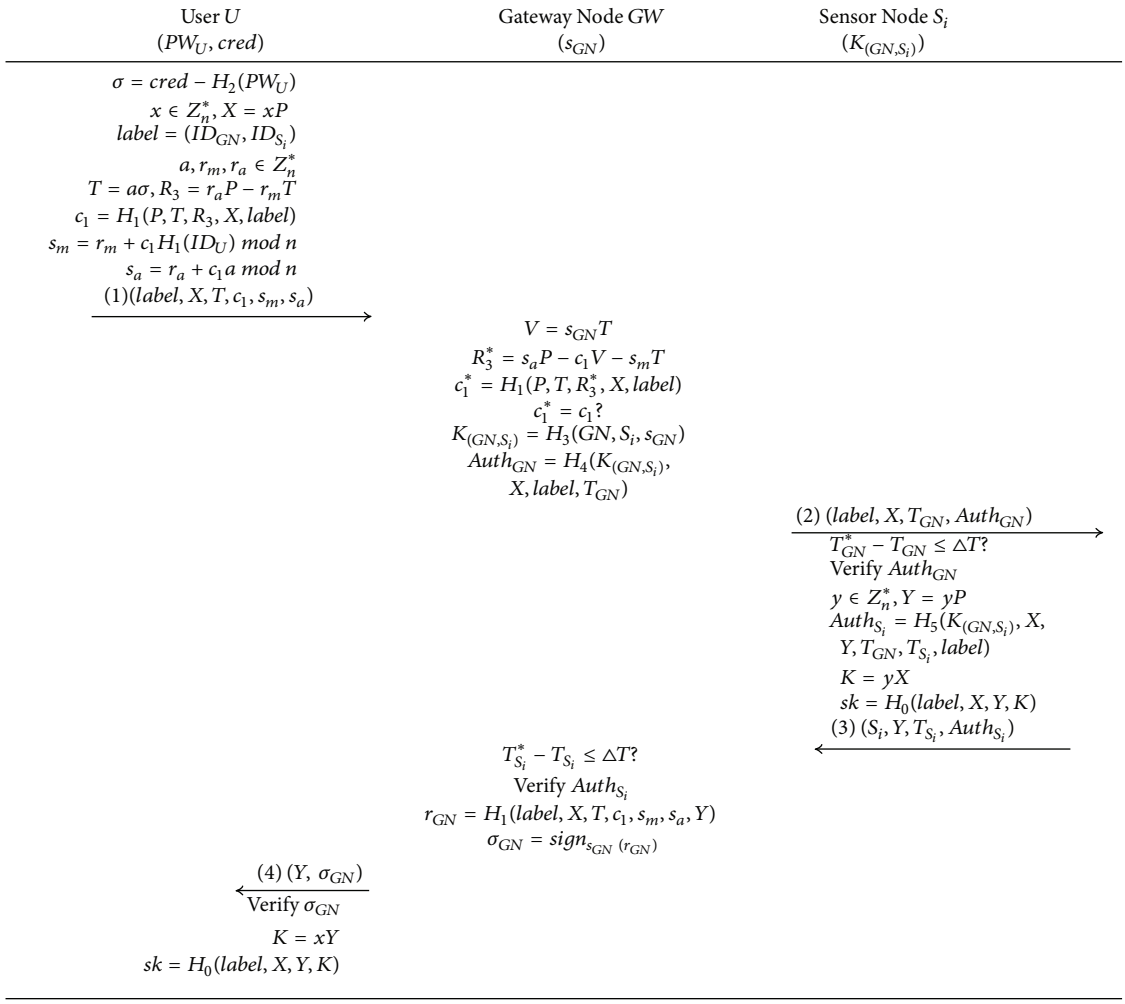

Figure 2: Authentication and key exchange phase.

of the authentication and key exchange phase are described as follows. For a pictorial illustration, please refer to Figure 2.

Step 1. The user U types his password PW_U to his terminal. The terminal will compute $H_2(PW_U)$ and recovers the credential σ from the stored $cred$. U then chooses a random number $x \in Z_n^*$ and computes $X = xP$. U defines the label of this session as $label = (ID_{GN}, ID_{S_i})$. U chooses three random numbers $a, r_m, r_a \in Z_n^*$ and computes $T = a\sigma$, $R_3 = r_aP - r_mT$, $c_1 = H_1(P, T, R_3, X, label)$, $s_m = r_m + c_1 H_1(ID_U) \bmod n$, and $s_a = r_a + c_1 a \bmod n$. Finally, U sends the message $(label, X, T, c_1, s_m, s_a)$ to the gateway node GN.

Step 2. Upon receiving the message $(label, X, T, c_1, s_m, s_a)$ from the user, GN needs to authenticate the user U. GN computes $V = s_{GN}T$, $R_3^* = s_aP - c_1V - s_mT$, and $c_1^* = H_1(P, T, R_3^*, X, label)$. GN checks whether c_1^* is equal to c_1 or not. If the verification is successful, GN authenticates the user U and believes the user U is a valid user. GN then computes the shared key with the sensor node $K_{(GN,S_i)} = H_3(GN, S_i, s_{GN})$ and the authenticator $Auth_{GN} = H_4(K_{(GN,S_i)}, X, lable, T_{GN})$, where T_{GN} is the current timestamp of GN. Finally, GN send the message $(label, X, T_{GN}, Auth_{GN})$ to the sensor node S_i.

Step 3. Upon receiving the message $(label, X, T_{GN}, Auth_{GN})$ from GN at time T_{GN}^*, the sensor node S_i first checks whether $|T_{GN}^* - T_{GN}| \le \triangle T$, where $\triangle T$ is the expected time interval for the transmission delay. If this is true, S_i then verifies the validity of the authenticator $Auth_{GN}$ using its private key $K_{(GN,S_i)}$. If the authenticator is valid, S_i chooses a random number $y \in Z_n^*$ and computes $Y = yP$. S_i then computes the authenticator $Auth_{S_i} = H_5(K_{(GN,S_i)}, X, Y, T_{GN}, T_{S_i}, label)$, where T_{S_i} is the current timestamp of S_i. S_i computes the Diffie-Hellman key $K = yX$ and the session key $sk = H_0(label, X, Y, K)$. Finally, S_i sends the message $(S_i, Y, T_{S_i}, Auth_{S_i})$ to the gateway node GN.

Step 4. Upon receiving the message $(S_i, Y, T_{S_i}, Auth_{S_i})$ from S_i at time $T_{S_i}^*$, GN first checks whether $|T_{S_i}^* - T_{S_i}| \le \triangle T$, where $\triangle T$ is the expected time interval for the transmission delay. If this is true, GN then computes the shared key with the sensor node $K_{(GN,S_i)} = H_3(GN, S_i, s_{GN})$ and verifies the validity of the authenticator $Auth_{S_i}$. If the verification is successful, GN

computes $r_{GN} = H_1(label, X, T, c_1, s_m, s_a, Y)$ and signs the random number r_{GN} using his private key s_{GN}; the signature is denoted by σ_{GN}. Finally, GN sends the message (Y, σ_{GN}) to the user U.

Step 5. Upon receiving the message (Y, σ_{GN}) from GN, U first verifies the validity of the signature σ_{GN}. U computes the random number $r^*_{GN} = H_1(label, X, T, c_1, s_m, s_a, Y)$ and checks if σ_{GN} is a valid signature for r^*_{GN} singed by GN. If the verification is successful, U computes the Diffie-Hellman key $K = xY$ and the session key $sk = H_0(label, X, Y, K)$. U will accept the session and wait for the upcoming communication.

4. Security Proof

In this section, we present the security proof of our protocol. The security proof is conducted in the security model presented in Section 2.

Theorem 1. *Suppose $\mathcal{P}$ is the anonymous authentication and key exchange protocol for WSN described in the previous section and $\mathcal{A}$ is a PPT time adversary against the AKE security of $\mathcal{P}$ who runs in time t and makes at most Q_{send} queries of Send oracle to different instances. If the signature scheme used in our protocol is existential unforgeable against adaptive chosen message attacks, the hash functions $H_i(\cdot)(i = 0, 2, \ldots, 5)$ are all modeled as random oracles; then under the CDH assumption, the advantage of the adversary $\mathcal{A}$ in violating the AKE security of the protocol $\mathcal{P}$ is at most*

$$Adv^{ake}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) \leq C' \cdot Q^{s'}_{send} + negl(\kappa). \quad (2)$$

Proof. We use the hybrid experiments technique to prove Theorem 1. These hybrid experiments start with the real attack scenario. We gradually change the simulation rules in each experiment. In the last experiment, the advantage of the adversary in distinguishing the session key is negligible. We also estimate the advantage difference of the adversary between two hybrid experiments and the advantage of the adversary in breaking the AKE security can be calculated. We denote the adversary's advantage in hybrid Exp_i using $Adv_i(\mathcal{A})$. □

Experiment Exp_0. This is the real attack scenario defined in the security model. In this experiment, the adversary has access to all the oracles. According to the definition of $\mathcal{A}$'s advantage, we have the following result:

$$Adv^{ake}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) = Adv_0(\mathcal{A}). \quad (3)$$

Experiment Exp_1. In this experiment, we simulate all the hash function $H_i(\cdot)(i = 0, 1, \ldots, 5)$ by maintaining hash lists $\Lambda_{H_i}(i = 0, 1, \ldots, 5)$ using the following rules:

(i) On a query $H_i(m)$, if a record (i, m, r) exists in Λ_{H_i}, then return r. Otherwise, the output r is chosen according to the following rule: $RuleH_i$.

if $i = 1$, choose a random element r from z^*_n. Then add the record $(1, m, r)$ to Λ_{H_i}.

if $i = 2$, choose a random element r from G. Then add the record $(2, m, r)$ to Λ_{H_2}.

if $i = 0, 3, 4, 5$, choose a random element r from $\{0, 1\}^\kappa$. Then add record (i, m, r) to Λ_{H_i}.

In addition to these lists, we also simulate six private hash oracles $H'_i(i = 0, 1, \ldots, 5)$ by maintaining hash lists $\wedge'_{H_i}(i = 0, 1, \ldots, 5)$. We will use these private hash functions in the following hybrid experiments. It is well known that a hash function can be simulated perfectly in PPT time using the above rules; thus, we have

$$\left|Adv_1(\mathcal{A}) - Adv_0(\mathcal{A})\right| \leq negl(\kappa). \quad (4)$$

Experiment Exp_2. In this experiment, we cancel the sessions if some unlikely collisions occur in these sessions. To be more specific, if some collisions occur in the simulation of the hash functions or on the transcripts of $((X, Y, T, c_1, s_m, s_a, \sigma_{GN}))$, we will terminate the session and let the adversary win. Based on the birthday paradox, we have the following result:

$$\left|Adv_2(\mathcal{A}) - Adv_1(\mathcal{A})\right| \leq negl(\kappa). \quad (5)$$

Experiment Exp_3. In this experiment, we modify the simulation rules of sessions by *Execute* queries. Whenever we need to compute the session key in a passive session, we use the private hash oracle H'_0 instead of H_0. Moreover, the Diffie-Hellman key K is not used as an input. In other words, the session key of a passive session is computed as $sk = H'_0(label, X, Y)$. The adversary can distinguish the experiment Exp_3 and the previous experiment Exp_2 if and only if the adversary sends a hash query $(label, X, Y, K)$ to the hash oracle H_0 in which X, Y is generated in a passive session and $K = CDH(X, Y)$. However, if the adversary can issue such a query, we can use the ability of the adversary to solve the CDH problem.

Given a CDH instance (U, V), we can embed the instance to all the passive session using the self-reducibility of the CDH problem. In order to do so, we choose four random numbers $a_0, b_0, a_1, b_1 \in Z^*_n$ for each passive session. In simulation the transcripts, we simply set $X = a_0U + b_0P$ and $Y = a_1V + b_1P$. All other transcripts are simulated as usual until the computation of the session key. The session key is computed as $sk = H'_0(label, X, Y)$. If an adversary can distinguish between this experiment and the previous one, then a query $(label, X, Y, K)$ must be issued to the hash oracle H_0. We can compute the Diffie-Hellman value of (U, V) by selecting a random record $(0, (label, X, Y, K), r)$ in $\wedge_{H_0}$ and computing $(K - a_0b_1U - a_1b_0V - b_0b_1P)/a_0a_1$.

Under the intractability assumption of the CDH problem, we have

$$\left|Adv_3(\mathcal{A}) - Adv_2(\mathcal{A})\right| \leq negl(\kappa). \quad (6)$$

Experiment Exp_4. In this experiment, we begin to deal with the active sessions. For a $Send(U, (Y, \sigma_{GN}))$ query, if the signature σ_{GN} is a valid signature for this active session, we simply terminate the simulation and let the adversary

win. Since the user U is honest in this session, the message (X, T, c_1, s_m, s_a) is generated by the user U. Besides, we cancel the experiment in which the collision occurs in the output of the hash functions and the transcripts in Exp_2, so the signature σ_{GN} is valid if it is a signature for the random number r_{GN}. The adversary wins the game in this experiment if and only if a new signature is forged. The signature scheme used in our protocol is existential unforgeable against the chosen message attacks, so the advantage of the adversary $\mathcal{A}$ in forging a signature for a new random number is negligible. It is obvious that

$$\left|Adv_4(\mathcal{A}) - Adv_3(\mathcal{A})\right| \leq negl(\kappa). \quad (7)$$

Experiment Exp_5. In this experiment, we continue to deal with the active sessions. For a $Send(GN, (S_i, Y, T_{S_i}, Auth_{S_i}))$ query, if the sensor node S_i is uncorrupted, the timestamp T_{S_i} is within the transmission delay and $Auth_{S_i}$ is a valid authenticator; then we simply terminate the simulation and let the adversary win the attack game. Since the sensor node S_i is uncorrupted, the symmetric key $K_{(GN,S_i)}$ is unknown to the adversary. Moreover, the timestamp T_{S_i} makes the replay attack impossible. The adversary can only produce a valid authenticator $Auth_{S_i}$ by issuing a query $(K_{(GN,S_i)}, X, Y, T_{GN}, T_{S_i}, label)$ to the hash oracle H_5 or the adversary correctly guesses the output of the hash function H_5 without asking the corresponding message. $K_{(GN,S_i)}$ and $Auth_{S_i}$ are two random values chosen from $\{0,1\}^\kappa$; the success probability of the adversary is negligible. Consequently we have the following equation:

$$\left|Adv_5(\mathcal{A}) - Adv_4(\mathcal{A})\right| \leq negl(\kappa). \quad (8)$$

Experiment Exp_6. In this experiment, we deal with the active sessions once again. For a $Send(S_i, (label, X, T_{GN}, Auth_{GN}))$ query, if the timestamp T_{GN} is within the transmission delay and $Auth_{GN}$ is a valid authenticator, then we simply terminate the simulation and let the adversary win the attack game. Since the gateway node is not allowed to be corrupted, the symmetric key $K_{(GN,S_i)}$ is unknown to the adversary and the timestamp T_{GN} ensures the adversary cannot replay an old authenticator. The adversary can only produce a valid authenticator $Auth_{GN}$ by issuing a query $(K_{(GN,S_i)}, X, lable, T_{GN})$ to the hash oracle H_4 or the adversary correctly guesses the output of the hash function H_4 without asking the corresponding message. $K_{(GN,S_i)}$ and $Auth_{S_i}$ are two random values chosen from $\{0,1\}^\kappa$; the success probability of the adversary is negligible. Similarly with the previous experiment, we have

$$\left|Adv_6(\mathcal{A}) - Adv_5(\mathcal{A})\right| \leq negl(\kappa). \quad (9)$$

Experiment Exp_7. In this experiment, we change the simulation rule of *Send* queries for the last time. For a $Send(GN, (label, X, T, c_1, s_m, s_a))$ query, the gateway node will first check the validity of the credential proof. If the credential proof is valid and the message is forged by the adversary, we then terminate the simulation and the adversary is claimed successful. However, the success probability of the adversary in producing a fake proof is bounded by the presentation of an algebraic MAC. With a similar analysis with [23], we get the following result:

$$\left|Adv_7(\mathcal{A}) - Adv_6(\mathcal{A})\right| \leq negl(\kappa). \quad (10)$$

In the last experiment, we can see that all the session keys of passive sessions are chosen randomly from the domain and all the active sessions are terminated without accepting. The only way for the adversary to succeed is to steal the terminal of the user and recover the credential by guessing the password. The adversary has to verify the correctness of the recovered credential by executing the protocol. Consequently, we have

$$\left|Adv_7(\mathcal{A})\right| \leq C' \cdot Q_{send}^{s'}. \quad (11)$$

5. Performance Analysis

In this section, we evaluate the computation and communication costs and the security attributes of our protocol with other related protocols with user anonymity [16–19]. In terms of computation, let "T_M" denote the time of one modular exponentiation computation, "T_{PM}" denote the time cost of one point multiplication computation on elliptic curve, 'T_H" denote the time of one hash function computation, and "T_S" denote the time of one symmetric encryption/decryption operation. According to [24], $T_M \approx 1.169ms$, $T_{PM} \approx 0.508ms$, $T_H \approx 0.069ms$, and $T_S \approx 0.069ms$. Moreover, we only evaluate the computation cost of the authentication and key exchange phase because the registration phase is a one-time job. In terms of communication cost, we assume the length of the identity is 32 bits, the secure parameter κ is 160 bits, the length of the timestamp is 64 bits, an element of cyclic group of ECC can be represented with 320 bits, and an element of cyclic group of RSA can be presented with 1024 bits. We also instantiate the signature scheme using the famous ECDSA signature scheme [25]. The performance of communication and computation is summarized in Table 2. We can see from Table 2 that our protocol is inefficient in terms of computation. However, the communication performance of the compared protocols is more or less the same. The computation cost of our protocols mainly arises from the strong user anonymity; i.e., no one except the user knows his real identity in our protocol, while the gateway node knows the user's real identity in other protocols.

Table 3 summarizes security properties of the proposed protocol with related protocols. It can be seen from Table 3 that our protocol provides all the security features. Moreover, our protocol is the only one which provides strong user anonymity and formal security proof. Considering the computation cost, communication cost, and security attributes as a whole, our protocol outperforms to other protocols. Consequently, the proposed protocol is more suitable for security and privacy critic applications scenarios in WSNs.

Table 2: Comparisons of computation and communication costs.

Protocols	Wu et al.'s [16]	Jiang et al.'s [17]	Wang et al.'s [18]	Li et al.'s [19]	Our protocol
Computation time of user (ms)	$2T_{PM} + T_S + 11T_H \approx 1.04$	$T_{PM} + 8T_H \approx 1.18$	$2T_{PM} + 8T_H \approx 1.04$	$2T_{PM} + 8T_H \approx 1.05$	$4T_{PM} + 4T_H \approx 2.03$
Computation time of gateway (ms)	$2T_S + 11T_H \approx 1.04$	$T_{PM} + 12T_H \approx 1.19$	$2T_{PM} + T_S + 11T_H \approx 1.05$	$T_{PM} + 9T_H \approx 0.52$	$4T_{PM} + 5T_H \approx 2.03$
Computation time of sensor (ms)	$2T_{PM} + T_S + 4T_H \approx 1.05$	$5T_H \approx 0.04$	$2T_{PM} + T_S + 11T_H \approx 1.06$	$4T_H \approx 0.03$	$2T_{PM} + 3T_H \approx 1.02$
Rounds	4	4	4	4	4
Bandwidth	3168bits	2689bits	3968bits	2912bits	2976bits

Table 3: Comparisons of security features.

Protocols	Wu et al.'s [16]	Jiang et al.'s [17]	Wang et al.'s [18]	Li et al.'s [19]	Our protocol
The replay attack	secure	secure	secure	secure	secure
The privileged insider attack	secure	secure	secure	secure	secure
The GW-node impersonation attack	secure	secure	secure	secure	secure
The stolen verifier attack	secure	secure	secure	secure	secure
The off-line dictionary attack	secure	secure	secure	secure	secure
The compromised sensor node attack	secure	secure	secure	secure	secure
Mutual authentication	yes	yes	yes	yes	yes
Session key establishment	yes	yes	yes	yes	yes
Key privacy	yes	no	yes	no	yes
User anonymity	weak	weak	weak	weak	strong
Formal security proof	yes	yes	yes	yes	yes

6. Conclusions

In this paper, we propose an anonymous authentication and key exchange protocol for WSNs. The most attractive property of our protocol is its strong user anonymity such that no one except the user knows the real identity of himself. Besides this, our protocol also enjoys formal security proof in the random oracle model and efficient communication complexity. The only disadvantage is that it consumes more computation resources. In wireless communication networks, establishing a channel usually consumes more energy than computation does. As a result, the heavy computation cost is not a serious problem. Due to its high security and strong anonymity, our protocol is very suitable for security and privacy critic application scenarios in WSNs.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Funding of Science and Technology on Information Assurance Laboratory (no. KJ-17-001) and Key Scientific and Technological Project of Henan Province (no. 122102210126).

References

[1] Y. Liu, W. Guo, C. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Transactions on Industrial Informatics*, pp. 1-1, 2018.

[2] D. He, N. Kumar, H. Wang, L. Wang, K. R. Choo, and A. Vinel, "A Provably-Secure Cross-Domain Handshake Scheme with Symptoms-Matching for Mobile Healthcare Social Network," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-1, 2016.

[3] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, 2017.

[4] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[5] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[6] Q. Jiang, Z. Chen, B. Li et al., "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, 2017.

[7] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers and Electrical Engineering*, 2017.

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

[9] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[10] 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.

[11] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316–323, 2013.

[12] D. He, N. Kumar, H. Shen, and J.-H. Lee, "One-to-many authentication for access control in mobile pay-TV systems," *Science China Information Sciences*, vol. 59, no. 5, pp. 1–14, 2016.

[13] J.-J. Yuan, "An enhanced two-factor user authentication in wireless sensor networks," *Telecommunication Systems*, vol. 55, no. 1, pp. 105–113, 2014.

[14] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 1–15, 2014.

[15] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Generation Computer Systems*, vol. 78, no. 3, pp. 956–963, 2018.

[16] F. Wu, L. Xu, S. Kumari, and X. Li, "A new and secure authentication scheme for wireless sensor networks with formal proof," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 16–30, 2017.

[17] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[18] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks," *Sensors*, vol. 17, no. 12, article no. 2946, 2017.

[19] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.

[20] C. Wang, D. Wang, G. Xu, and Y. Guo, "A lightweight password-based authentication protocol using smart card," *International Journal of Communication Systems*, vol. 30, no. 16, pp. 1–11, 2017.

[21] D. Wang, H. Cheng, P. Wang et al., "Zipfs law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[22] F. Wei, P. Vijayakumar, J. Shen, R. Zhang, and L. Li, "A provably secure password-based anonymous authentication scheme for wireless body area networks," *Computers and Electrical Engineering*, 2017.

[23] Z. Zhang, K. Yang, X. Hu, and Y. Wang, "Practical anonymous password authentication and TLS with anonymous client authentication," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security, CCS 2016*, pp. 1179–1191, October 2016.

[24] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[25] C. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.