A Pairing-Free ID-based One-Pass Authenticated Key Establishment Protocol for Wireless Sensor Networks

Rehana Yasmin, Eike Ritter
School of Computer Science, University of Birmingham
Birmingham, B15 2TT, UK
Email: {R.Yasmin,E.Ritter}@cs.bham.ac.uk

Guilin Wang
School of Computer Science, University of Wollongong
Wollongong, NSW 2522, Australia
Email: guilin@uow.edu.au

***Abstract*—Due to resource constraints and unique features of wireless sensor networks (WSNs), designing a key establishment protocol is much harder for WSNs than for traditional wired and wireless counterparts. In this paper, we propose a new efficient and secure ID-based one-pass authenticated key establishment protocol between an outside user and a sensor node. The proposed protocol does not require sensor nodes to compute any expensive pairing function. Moreover, it imposes very light computational and communication overheads and also provides scalability. We analyze security and efficiency of the proposed protocol by comparing firstly the session key establishment protocols for WSNs and secondly the existing ID-based one-pass key establishment protocols. The comparison shows that the proposed protocol is the most secure and efficient one for WSNs applications providing both security features of user authentication and session key establishment.**

***Keywords*-Wireless Sensor Networks; Security; ID-based One-Pass Key Establishment; User Authentication;**

I. INTRODUCTION

Recent advances in embedded technologies, as well as wireless communications, have broadened the prospects for many applications of wireless sensor networks (WSNs), for instance, environmental monitoring, ocean reading and many military applications [1]. However, the vulnerability of wireless communication and the ad-hoc nature of deployment open the door for a wide variety of malicious attacks, making security a key concern for these applications. On the other hand, the resource constrained nature of sensor nodes, i.e., limited *power*, *computing* and *storage* resources, poses a need for highly efficient security solutions. This restriction has significantly impacted the field of application security. For such applications, the efficiency of a security scheme is as important as its security. Any security scheme which is computationally expensive, no matter how secure it is, does not suit resource constrained sensor nodes.

To protect the communication in WSNs, one security requirement is the ability to encrypt and decrypt confidential data entailing the establishment of a session key. An authenticated **session key establishment** protocol provides the communicating parties with a secret and authentic shared session key which is used for the encryption and decryption of data. The session key establishment protocol is particularly important for those applications of WSNs which frequently exchange confidential data through insecure channels, for instance, environmental monitoring and ocean reading. In these applications, the data collected by the sensor nodes is useful for many research and business purposes. Different research organizations and businesses pay money to the deployment agencies of large scale sensor networks and obtain data from them. Thus, the data collected within the network is valuable and confidential in these applications. Since the data is available only to "authorized" users who have paid for the data, **user authentication** is another security requirement for such applications. These authorized users, after successful authentication, issue queries to the sensor nodes to access data of their interest. Therefore, a secure mechanism becomes highly desirable in these applications that allows sensor nodes to establish a session key with the users (to encrypt and decrypt confidential data) while facilitating all the necessary authentications (to know who their counterparts are). On the other hand, designing a secure and efficient security protocol for resource constrained sensor nodes is a challenging task.

In this paper, we propose an efficient and secure ID-based one-pass session key establishment protocol between an outside user and a sensor node which combines user authentication with session key establishment. ID-based cryptography [2] replaces a user's public key with his unique public identifier (ID), such as email address. The corresponding private key is generated by a private key generator (PKG), a trusted third party. ID-based cryptography removes the need for certificate transmission and verification to obtain the public key, and hence reduces the transmission and the processing costs of the security schemes.

A. One-Pass Key Establishment

High computation and communication cost of secure two-pass key establishment protocols makes them expensive for low-power WSN applications (where low computation and communication cost is critical), for instance, the authenticated DiffieHellman protocol named as Station-to-Station protocol [3]. To satisfy the resource constraints of sensor nodes, a session key establishment protocol with high se-

 ISBN: 978-1-61208-144-1

curity and a minimum amount of computation and number of passes is required. A secure one-pass key establishment protocol is an attractive alternative for them. In a one-pass key establishment protocol only one message transmission is required for the establishment of key, i.e., only the sender (initiator) of the protocol generates an ephemeral private key and transmits its public part, called the ephemeral public key, to the receiver (responder). Both parties then compute a shared session key using their own private keys and ephemeral keys. In one-pass key establishment protocols, the reduced number of exchanged messages lessens the transmission and processing costs because only one message is transmitted and processed.

Besides the reduced cost, another advantage of a one-pass key establishment protocol is its use for off-line communications, explained as follows: The sender computes its shared session key, encrypts the message m (any confidential message) using the computed session key and sends both the ephemeral public key and the ciphertext of m to the receiver. The receiver can compute the same shared key any time using the sender's ephemeral public key and decrypt the message. The receiver needs not to be necessarily on-line. This feature is particularly useful for applications where only one entity is on-line, for instance, email. This feature can also be utilized in a WSN environment where the only message sent by the user for key establishment can be combined with the encrypted user query to provide query privacy. The details are discussed in Section III.D.1.

Contribution. The main contribution of this paper is a new secure and efficient ID-based one-pass key establishment protocol for WSNs. To the best of our knowledge, this is the first ID-based one-pass key establishment protocol which does not require any pairing computation. Lack of pairing operation makes our scheme computationally efficient and, hence, suitable for resource constrained sensor nodes. Scalability is another attractive feature of the proposed protocol which is required for WSN environment. Other than performance, the provable security is also an aspect of the proposed protocol. Although the details of formal security analysis including security model and security proof are omitted in this paper due to space limitations, they will be a part of the extended version of this paper.

Organization. Section II presents an overview of the related work. Section III describes the proposed scheme in detail. Section IV and Section V give the security analysis and the performance evaluation of the proposed protocol, respectively. Finally, a brief conclusion is given in Section VI.

II. Related Work

A. Session Key Establishment in Wireless Sensor Networks

This section briefly reviews the work related to the session key establishment in WSNs. A public key cryptography based hybrid authenticated key establishment protocol between a sensor node and a security manager is proposed by Huang et al. [4]. Their protocol exploits the difference in capabilities between the sensor nodes and the security manager. Like an outside user, a security manager is a powerful device (compared to a sensor node) which establishes a session key with a sensor node for subsequent use. In the beginning of the protocol, both parties exchange their certificates signed by a certification authority to extract the public keys of each other. However, the knowledge of the corresponding private keys is only proved after the complete run of the protocol on both sides. An adversary can exploit this fact and repeat this protocol with the sensor node by replaying a valid certificate, resulting into Denial of Service (DoS) attack. Before a sensor node detects the replayed certificate, it would have performed expensive computations and communications wasting its resources, particularly battery power. Later on Tian et al. [5] detected another serious security attack against Huang et al.'s protocol. They showed in [5] that a security manager (user in our case) easily learns the long-term private key of a sensor node after having one normal run of the protocol with the sensor node.

Kim et al. [6] propose an ID-based key establishment protocol from pairing-based cryptography which aims to reduce the communication cost of [4]. Being ID-based, their protocol replaces the public keys by the IDs, eliminating the need of exchange of certificates. This protocol reduces the communication cost but increases the overall computation cost of the protocol due to the expensive pairing computation. Like [4], this protocol also experiences a delayed user authentication (again by the proof of private key knowledge) on the sensor node's side, causing a DoS attack. An attempt to reduce the computation cost of [6] is made by Zhang et al. in [7]. They propose another version using pairing-based cryptography. Compared with Kim et al.'s protocol, their contribution is to scale down the number of point multiplication operations on a sensor node under the same communication complexity as in [6]. However, their protocol does not authenticate the security manager at all which enables any one to establish a session key with the sensor node. Yasmin et al. [8] propose an authentication framework describing user authentication and session key establishment for WSNs using ID-based cryptography. However, they did not provide any concrete scheme for the establishment of session key between the user and the sensor node. Our proposed protocol can be integrated into their authentication framework to provide a concrete scheme for user authentication and session key establishment.

B. ID-based Key Establishment

This section lists the work related to the ID-based key establishment. In recent years, a few ID-based one-pass key establishment protocols [9], [10], [11], [12] have been designed for traditional networks. However, none of these

 ISBN: 978-1-61208-144-1

schemes is efficient as all of these require pairing computations. Extensive use of pairings makes these schemes quite slow and computationally expensive, particularly for resource constrained sensor nodes consuming considerable resources on them. Other related work includes the pairing-free ID-based two-pass authenticated key establishment schemes, for instance, [13], [14], [15] using the same ID-based setup as used in our scheme. However, as mentioned earlier, the secure two-pass key establishment schemes consume more resources on sensor nodes in terms of computation and communication overheads than one-pass schemes.

III. The Proposed Session Key Establishment Protocol

In this section, we present our proposed ID-based one-pass authenticated key establishment protocol by introducing the four phases: *System Initialization*, *Private Key Generation*, *User Registration* and *Key Establishment*. The first two phases are performed once, before the deployment of the sensor network. In an ID-based cryptosystem, a private key generator (PKG) computes the private keys corresponding to *ID*s. In WSNs the base station, a resourceful device, is considered as trustworthy. In our scheme, the base station plays the role of *PKG* and computes the private keys for sensor nodes and users.

A. System Initialization

In this phase, the *Setup* algorithm runs on the base station (before deployment) and generates the system parameters, including master public key (*mpk*), and the corresponding master secret key (*msk*) by using a security parameter k. This algorithm performs the following steps:

(a) Specify q, p, E/F_p, P and $\mathbb{G}$ where
- q is a large prime number and p is the field size,
- E/F_p is an elliptic curve E over a finite field F_p,
- P is a base point of order q on the curve E and
- $\mathbb{G}$ is a cyclic group of order q under the point addition "+" generated by P.

(b) For *msk* $s \in_R \mathbb{Z}_q^*$, compute *mpk* as $P_{PKG} = sP$.
(c) Choose one hash function H: $\{0,1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$.
(d) Choose one key derivation function χ: $\mathbb{G} \rightarrow \{0,1\}^k$.
(e) Output system parameters $\{q, p, E/F_p, P, \mathbb{G}, P_{PKG}, H, \chi\}$ and keep s secret.

B. Private Key Generation

In this phase, the *Extract* algorithm runs on the base station (before deployment) and computes the private keys of all sensor nodes corresponding to their *ID*s. This algorithm takes *msk* and a sensor node's *ID* as input and generates a private key corresponding to that *ID* using the well known Schnorr signature. For a sensor node I with identity ID_i, this algorithm performs the following steps:

(a) For $r_i \in_R \mathbb{Z}_q^*$, compute $R_i = r_i P$ and $c_i = H(ID_i, R_i)$.
(b) Compute private key as $s_i = c_i s + r_i$.
(c) Output (s_i, R_i) where s_i is secret while R_i is public.

Here the private key s_i is the Schnorr signature on the ID of the node signed with the private key of the PKG. *ID*s, corresponding private keys and system parameters are stored on sensor nodes before deployment. Hence, every sensor node i stores $\{ID_i, s_i, R_i\}$ and system parameters.

C. User Registration

This phase is repeated every time when a new user is registered with the system. In this phase, the *Extract* algorithm runs on the base station and computes the private key for a user U corresponding to his identity ID_u in the same way as computed for sensor nodes in the *Private Key Generation* phase. The base station, who runs this algorithm, sends the private key to the user via a secure channel. Hence, every user U gets $\{ID_u, s_u, R_u\}$ and system parameters.

D. Key Establishment: One-Pass Authenticated Session Key Establishment

Whenever a user wants to access data from sensor nodes, he establishes a session key with the sensor node in his range after successfully authenticating himself to the sensor node. Whether the user query is processed by a single sensor node or a set of sensor nodes is related to the topic of *query processing in wireless sensor networks* and is not addressed in our paper. We now describe our ID-based one-pass session key establishment protocol between a user U and a sensor node I. Fig. 1 describes the steps of the protocol.

(a) The user U chooses at random $t \in \mathbb{Z}_q^*$ as ephemeral key and computes $y = ts_u$ and $L = yP$. U signs the ephemeral public key L together with ID_u, ID_i and TS and sends $[L, ID_u, ID_i, TS, Sig_{s_u}(L, ID_u, ID_i, TS)]$ to the sensor node I in his range. Here TS is the current time stamp to avoid a replay attack and $Sig_{s_u}(L, ID_u, ID_i, TS)$ is a signature signed by U using his private key s_u. Computing y from L is the so-called *Elliptic Curve Discrete Logarithm* (ECDL) problem, which is intractable.

(b) The sensor node I first checks the time stamp TS to avoid the verification of a replayed message. If this is a fresh message, I verifies the signature $Sig_{s_u}(L, ID_u, ID_i, TS)$. Successful signature verification implies the message is actually sent by the user U and is fresh. Hence, I accepts the message, otherwise the protocol is terminated at this stage. Next the sensor node I computes the shared secret $K_{i,u}$ as
$K_{i,u} = s_i L$ (=$s_i t s_u P$)
and deletes L.

(c) The user U computes the same shared secret $K_{u,i}$ as
$S_i = c_i P_{PKG} + R_i$ where $c_i = H(ID_i, R_i)$
$K_{u,i} = yS_i$ (=$t s_u s_i P$)
U then deletes L, t and y.
The both parties then compute the shared session key as $SK = \chi(K_{u,i}) = \chi(K_{i,u}) = \chi(ts_u s_i P)$, where χ is the key derivation function.

User U		**Sensor Node** I
$t \in_R \mathbb{Z}_q^*$		
$y = ts_u$		
$L = yP$	$\xrightarrow{L, ID_u, ID_i, TS, Sig_{s_u}(L, ID_u, ID_i, TS)}$	
$c_i = H(ID_i, R_i)$		
$S_i = c_i P_{PKG} + R_i$		$Ver(L, ID_u, ID_i, TS)$
$K_{u,i} = yS_i$		$K_{i,u} = s_i L$
$\left\{ \begin{array}{l} = ys_iP \\ = ts_us_iP \end{array} \right\}$		$\left\{ \begin{array}{l} = s_iyP \\ = s_its_uP \end{array} \right\}$
$SK_u = \chi(K_{u,i})$		$SK_i = \chi(K_{i,u})$
	$SK = SK_u = SK_i = \chi(ts_us_iP)$	
		$E = (SK_i \oplus ID_u \oplus ID_i)$
	$\xleftarrow{E'}$	
$E = Dec_{SK_u}(E')$		$E' = Enc_{SK_i}(E)$
$E \stackrel{?}{=} (SK_u \oplus ID_u \oplus ID_i)$		

Figure 1. Authenticated One-Pass Session Key Establishment Protocol with Key Confirmation

However, there is no guarantee that at the end of the secure run of the protocol both parties compute the key. Indeed, in any key establishment protocol, the sender of the last message cannot make sure whether or not its last message is received by the other party. The user may successfully finish the protocol with a key output. Although the adversary is not able to learn the computed key, the sensor node might not receive the user's message and consequently might not be able to compute the key. The assurance against this scenario is achieved via an *authenticated key establishment protocol with key confirmation (AKC)*. This is usually achieved by adding a key confirmation message to the authenticated key establishment protocol after the key has been established. Hence, after both parties establish the session key, the *Key Establishment* algorithm proceeds as follows:

(d) After key computation, the sensor node I performs the following steps:
   i) Computes the XOR of its computed key SK_i with ID_u and ID_i as follows: $E = (SK_i \oplus ID_u \oplus ID_i)$.
   ii) Encrypts E with SK_i using a secure symmetric encryption algorithm, i.e., $E' = Enc_{SK_i}(E)$ and sends E' to U.

(e) After U receives E', he performs the following steps:
   i) Decrypts E' using his computed key SK_u to obtain E, i.e., $E = Dec_{SK_u}(E')$.
   ii) Checks whether $E \stackrel{?}{=} (SK_u \oplus ID_u \oplus ID_i)$.

Successful verification implies that both parties have computed the shared session key. As user does not expect to receive any message from the sensor node to compute the key, he does not need to send a key confirmation message to the sensor node.

1) Authentication, Key Establishment and Query Privacy: To obtain sensor nodes data, the user first authenticates himself to his nearby sensor node, establishes a session key with it and then sends his query to it. The sensor node, after successful user authentication and session key establishment, processes the received user query, encrypts the query results and sends them back to the user. For privacy reasons, the user query needs to be encrypted in some situations [16] since users may not be willing to disclose their areas of interests. Due to the one-pass key establishment, query privacy can also be provided by the proposed protocol as follows: the user computes his shared session key, encrypts his query using computed session key and sends his signed ephemeral public key to the sensor node together with his encrypted query in a single message. The sensor node first authenticates the user by verifying the signature. If the signature verification fails, the protocol terminates here. Otherwise, the sensor node computes the same shared session key, decrypts the user query, processes it and sends the encrypted query results back to the user. Thus, only a single message is exchanged for authentication, key establishment and encrypted query transmission achieving transmission efficiency.

2) ID-based Signature: To sign the ephemeral public key, any secure ID-based signature (IBS) scheme with the same ID-based parameters can be used, for instance, the secure ***BNN-IBS*** [17] scheme proposed by Bellare et al., whose security is proved under the discrete logarithm problem. There are also some other secure variants of ***BNN-IBS***, e.g., ***vBNN-IBS*** [18] and ***SLL-IBS*** [19] which can be used. ***vBNN-IBS*** has already been used in WSNs to provide broadcast authentication.

3) Distributing The Public Information ID_i and R_i: One possible question might be how a user can obtain ID_i and R_i, the public information of a sensor node I. As the user is equipped with a resourceful device, it can store the ID_i and R_i pairs of the sensor nodes in user's range. In 160-bit ECC settings, the size of the ID_i and R_i pair is about

25 bytes. For say 5000 sensor nodes in user's range, the total storage required will be about 125KB. This is an acceptable storage overhead on a resourceful user device to provide security with efficiency on resource constrained sensor nodes. User can also obtain ID_i and R_i pair from the base station via any other means e.g., Internet, before making a query to I. Note the difference that here ID_i and R_i are two identity elements of I and not the public keys as in the traditional public key crypto system where public keys are verified using the signed certificates. Here, if some one tries to use fake ID_i and R_i pair, he would not be able to generate corresponding private key s_i which is generated using msk. Generating such a valid triplet without msk would be equivalent to forging Schnorr signature. The ID-based two-pass key establishment schemes mentioned in Section II.B and the ID-based signature schemes mentioned in Section III.D.2 all use the same ID-based setup. This setup allows to construct efficient pairing-free ID-based schemes while handling the problem of public keys/certificates.

IV. Security Analysis

The security of our protocol is formally analyzed using the reductionist proof technique under the standard Computational Diffie-Hellman (CDH) assumption. The CDH assumption assumes that for a large security parameter k (e.g., $k \geq 160$) it is intractable to compute abP given $\langle P, aP, bP \rangle$, where P is a random generator of $\mathbb{G}$ and a, b are uniformly selected at random from $\mathbb{Z}_q$. By assuming that the CDH assumption holds in $\mathbb{G}$ we show that the proposed protocol is secure in the ID-eCk model [10]. Due to space limitation, the security model and rigorous proof have to be omitted in this paper. The detailed security analysis including security proof will be a part of the extended version of this paper. In this section, we informally discuss five security attributes appertaining to the proposed protocol.

Authentication. The proposed protocol provides the required authentication. There is only one message exchanged and that is sent by the user. Authentication of that single message is achieved by the verification of signature signed by the user. It is infeasible for an adversary to sign a message on behalf of a user without knowing user's private key. Successful signature verification by the sensor node I proves the fact that the ephemeral public key is actually sent by a legitimate user U. On the other side, $S_i (= s_i P)$ computed from I's public information assures the user that the session key is, in fact, established with I. Only the sensor node I with the valid corresponding private key s_i can compute the same session key. Authentication avoids the chances of the adversary mounting a man-in-the-middle attack.

Key Confidentiality. After the successful key establishment between a sensor node and a user, the only information available to the adversary is the public parameters and the ephemeral public key $L (= ts_u P)$. However, he cannot compute the user U's private key s_u and/or ephemeral private key t from L since we assume there is no polynomial time algorithm to solve the ECDL problem. Furthermore, he cannot compute the shared secret $ts_u s_i P$ because it requires the knowledge of private keys of both the sensor node and the user. Hence, the key is computable only by the user U and the sensor node I.

Key Compromise. The random value for ephemeral private key t is separately generated for each session. Therefore, the established session key is computationally different for different sessions. A session key established between a compromised sensor node and a user would not enable an adversary to compute or learn any other session key established between any other legitimate sensor node and a user. Furthermore, it would not enable an adversary to learn the user's private key s_u from L due to the intractability of ECDL problem. In fact, the proposed protocol guarantees that the communication between an uncompromised sensor node and a user cannot be exposed, irrespective of the number of other nodes that are compromised.

Key Confirmation. In the proposed protocol, the key confirmation message E' provides the explicit key confirmation. The sensor node computes E' and sends it to the user so that the user can be assured that the sensor node has received the user's ephemeral public key and successfully computed the session key. However, the user does not expect to receive any message from the sensor node for key establishment, as he can compute the same session key by himself. Hence, the user does not need to send a key confirmation message back to the sensor node.

Replay Attack. In a replay attack, an adversary replays the previous successful user request to either establish a session key with the sensor node or to waste sensor node resources by the request verification. In the proposed protocol, because of user's signed message, the adversary will not be able to authenticate successfully and establish a key. Furthermore, the time stamp TS provides freshness. The sensor node checks time stamp before the signature verification to avoid the verification of a replayed request message. Depending on the transmission delay imposed by the communication channel between the user and the sensor node, the sensor node sets a time threshold leaving a potential attacker little time to mount a replay attack.

V. Performance Comparison

In this section, we evaluate the performance of our proposed protocol in two ways: firstly, by comparing it with the existing session key establishment protocols for WSNs in Tables I and II, and secondly, by comparing it with the other ID-based one-pass session key establishment protocols in Table III. The factors used to evaluate the performance

 ISBN: 978-1-61208-144-1

Table I
COMPUTATION COST COMPARISON WITH THE EXISTING SESSION KEY ESTABLISHMENT PROTOCOLS FOR WSNS

	Key Establishment Cost		**User Authentication Cost**		**Time (s)**
	User	*Sensor Node*	*Sensor Node*		*Sensor Node*
Huang et al. [4]	$4M+3H$	$3M+3H$	Signed certificate ver. (***ECDSA***)	$2M$	1.60
Kim et al. [6]	$2P+1M+1E+2H$	$3M+1E+2H$	Implicit ver.	NA	2.24
Zhang et al. [7]	$2P+1M+4H$	$2M+1E+3H$	Does not support	NA	1.92
Our scheme	$3M+1H$	$1M$	Signature ver. (***vBNN-IBS***)	$3M$	1.28

are the number of complex cryptographic operations including pairing, point multiplication and exponentiation operations (computation overhead), total number of messages exchanged in each protocol run (communication overhead) and the memory requirements (storage overhead). Since the sensor nodes are more resource-constrained than the users devices, we pay more attention to the efficiency of the protocol on the sensor node side than on the user side.

For 80-bit security, in an efficient and optimized implementation on a standard MICA2 sensor node, one pairing computation takes 1.90s [20] and one point multiplication takes 0.32s [21]. Note that if the basic operation in $\mathbb{G}$ is denoted multiplicatively ($*$) instead of additively ($+$), the point multiplication in $\mathbb{G}$ is then called exponentiation correspondingly and thus takes 0.32s. However, the exponentiation in the target group $\mathbb{G}_T$ (in the settings of pairing [22]) takes more time than exponentiation (or point multiplication) in $\mathbb{G}$ because of the fact that it computes arithmetic in $\mathbb{G}_T$ which is operated in a field much bigger than the field in which $\mathbb{G}$ is defined. In usual implementations of pairing, 1 exponentiation in $\mathbb{G}_T$ costs about equal to 4 exponentiations in a multiplicative group [22] or 4 point multiplications in an additive group. The overheads of hash operation and arithmetic operations in $\mathbb{Z}_q^*$ are very small compared to the above mentioned expensive cryptographic operations. Thus, we only consider the expensive cryptographic operations for performance analysis. In all tables, P denotes one pairing computation, H denotes one hash evaluation, M denotes one point multiplication or exponentiation in $\mathbb{G}$ and E denotes one exponentiation in target group $\mathbb{G}_T$.

A. Session Key Establishment for Wireless Sensor Networks

This section compares the proposed protocol with the existing session key establishment protocols for WSNs. Tables I and II show the comparison results.

1) Computation Overhead: In WSNs scenario, it is highly desirable for a security protocol to have low computational overhead on resource constrained sensor nodes. In Huang et al.'s key establishment protocol [4], the computation overhead on a sensor node is the verification of a signed certificate to extract user's public key and the computations of 3 point multiplications to compute session key. The user authentication, however, is achieved via key confirmation messages. For comparison purpose, we assume that the certificate verification requires the verification of an ***ECDSA*** signature. The ***ECDSA*** signature is considered more efficient for sensor nodes than RSA signature because of shorter key and signature sizes. ***ECDSA*** requires 2 point multiplications as expensive operations to verify a signature. Hence, the total computation overhead of Huang et al's protocol is 5 point multiplications. Kim et al.'s protocol [6] requires sensor nodes to compute 3 point multiplications and 1 exponentiation in $\mathbb{G}_T$. Zhang et al.'s protocol [7] brings down the computation cost of [6] by one point multiplication without providing user authentication. Our proposed protocol requires a sensor node to compute only 1 point multiplication to compute the session key and one signature verification to authenticate the user. For comparison with [4] and [6], we assume that the secure and efficient ID-based signature scheme ***vBNN-IBS*** [18] is used in our protocol for user authentication which requires 3 point multiplications for signature verification. It is clear from Table I that the overall computational load of the proposed protocol is still lower than the computational loads of both [4] and [6] and the key computation cost is lower than the key computation cost of [7]. At the same time, the proposed protocol has stronger security, as we shall discuss in Section V.A.6.

2) Time Consumption: We now compare the estimated total computation time taken by a sensor node to authenticate a user and derive a session key. The results of this time analysis are also given in Table I. Huang et al.'s protocol [4] requires a sensor node to compute 5 point multiplications and therefore, takes about 1.60s on it. Kim et al.'s protocol [6], on the other hand, computes 3 point multiplications and 1 exponentiation in $\mathbb{G}_T$. Considering the fact that the exponentiation in $\mathbb{G}_T$ costs four times than one point multiplication, the estimated computation time is about 2.24s for their protocol. Zhang et al.'s protocol [7] requires a sensor node to compute 2 point multiplications and 1 exponentiation in $\mathbb{G}_T$ for key computation (this protocol does not provide user authentication) and consumes about 1.92s on a sensor node. Considering the ID-based signature scheme ***vBNN-IBS*** [18], the total estimated computation time for the proposed protocol is about 1.28s for 4 point multiplications. This implies that compared with the protocols proposed by Huang et al., Kim et al. and Zhang et al., our protocol reduces the total computation time for key establishment and user authentication on a sensor node by 20%, 33%, and 43%, respectively, without mentioning that Huang et al.'s and Zhang et al.'s protocols are quite weak in security

 ISBN: 978-1-61208-144-1

Table II
COMMUNICATION COST COMPARISON WITH THE EXISTING SESSION KEY ESTABLISHMENT PROTOCOLS FOR WSNS

	Messages Exchanged	
	Key Establishment	*Key Confirmation*
Huang et al. [4]	4	2
Kim et al. [6]	3	1
Zhang et al. [7]	3	NA (Does not support)
Our scheme	1	1

(Refer to Section V.A.6). In addition, note that our protocol also improves the performance of a user by 25%, 82%, and 75% over Huang et al.'s, Kim et al.'s and Zhang et al.'s solutions, respectively. As improving the efficiency of the user side is not our focus in this paper, we do not discuss this issue in detail.

3) Communication Overhead: To achieve network resource efficiency and minimum latency, the number of message exchanges between the sensor node and the user should be as small as possible. Huang et al.'s protocol [4] and Kim et al.'s protocol [6] exchange 6 and 4 messages, respectively, for key establishment and user authentication. The key confirmation messages are compulsory to provide user authentication in their protocols. Zhang et al.'s protocol [7] exchanges 3 messages for the key establishment. The proposed protocol exchanges only 1 message for both key establishment and user authentication. Hence, the proposed protocol causes very low communication overhead than the other three protocols for WSNs as shown in Table II.

4) Storage Overhead: The storage overhead of the proposed protocol is similar to the other protocols which is not very high. The proposed protocol does not require sensor nodes to store any user credentials (IDs, public keys, certificates etc.) for the verification of a user's legitimacy and so provides storage efficiency. The only storage overhead is the sensor node's ID, corresponding ID-based key and the system parameters.

5) Scalability: Since the overheads of the proposed protocol do not increase with the network size, it supports large scale deployment of WSNs. New sensor nodes and outside users can be added to the WSN easily at any time. Preloaded with ID, secret key and public parameters, the new sensor node can establish a key with any legitimate user after user authentication. The new users simply need to register themselves to the base station and get their private keys and system parameters. ID-based cryptography relieves sensor nodes from storing any users specific information to authenticate them, and consequently eliminates the restriction on the number of outside users.

6) Analysis - Performance Versus Security: As discussed earlier, Huang et al.'s protocol [4] is not secure since user can learn a sensor node's private key after one run of the protocol with that node. This is a severe security attack against a key establishment protocol which cannot be tolerated, no matter how efficient a protocol is. Another drawback is the DoS attack caused by the delayed user authentication. On the other hand, Zhang et al.'s protocol [7] does not support user authentication at all allowing any adversary to establish a session key and obtain sensor nodes data. Hence, these two protocols lack the required security. Kim et al.'s protocol [6] also suffers from the DoS attack caused by the delayed user authentication wasting sensor node's resources. The proposed protocol authenticates a user at the first step by the verification of a signed user's ephemeral public key and the time stamp. Furthermore, it is not possible for any participant or any adversary to learn any participant's private key.

However, an adversary can cause a sensor node to verify a fake signature in the proposed protocol wasting its resources. To see how devastating this attack is as compared to the DoS attack in Kim et al.'s protocol, we assume the secure and efficient ID-based signature scheme ***vBNN-IBS*** [18] for signature generation in our protocol. To detect a fake user request sent by an adversary, a sensor node will perform 3 point multiplications in the proposed protocol and 3 point multiplications and 1 exponentiation in $\mathbb{G}_T$ in Kim et al.'s protocol. Before a user is authenticated, 4 messages will be exchanged in Kim et al.'s protocol while only 1 message will be exchanged in the proposed protocol as after receiving the first message from the user the sensor node can find out the fake request and terminate the protocol. Thus, DoS attack in the proposed protocol is much less devastating than in Kim et al's protocol saving both the communication and the computation costs. Hence, the proposed protocol provides better performance versus security than the existing session key establishment protocols for the WSNs.

B. ID-based One-Pass Session Key Establishment

In this section, by comparing our protocol with the existing ID-based one-pass session key establishment protocols, we show that the significant efficiency improvement achieved by the proposed protocol is its very low computation overhead. Note that the existing protocols are not originally claimed for WSNs. What we are discussing here is that these protocols do not suit WSNs due to their low performances. Table III compares our protocol with the existing protocols by listing the key establishment costs for both sides of each protocol. Compared to the existing protocols, the proposed protocol is computationally efficient on the

Table III
COMPARISON WITH THE EXISTING ID-BASED ONE-PASS KEY ESTABLISHMENT PROTOCOLS

	Key Establishment Cost		**Time (s)**
	User	*Sensor Node*	*Sensor Node*
Benit et al. [9]	$1P+2M+1H$	$1P+1H$	1.90
Okamoto et al. (*II*) [11]	$1P+3M+2H$	$1P+1M+2H$	2.22
Wang [12]	$1P+3E+2H$	$1P+2E+2H$	4.46
Gorantla et al. [10]	$1P+2M+1H$	$1P+1M+1H$	2.22
Our scheme	$3M+1H$	$1M$	0.32

 ISBN: 978-1-61208-144-1

sensor node's side requiring only one point multiplication but no pairing computation. One pairing computation on a standard MICA2 sensor node takes 1.90s versus 0.32s for a point multiplication on the same node and therefore, consumes resources equal to 6 point multiplications. Due to the lack of pairing computations on both sides, our proposed protocol provides much better performance than the existing protocols. Table III also shows the estimated time that a sensor node consumes if the existing protocols are applied in WSNs. It is clear from Table III that our protocol is almost 6 times faster than Benit et al.'s protocol [9], which is the best existing ID-based one-pass key establishment protocol in terms of efficiency on the sensor node side. Moreover, if we also count the user authentication (signature verification) cost mentioned in Section V.A.2, the proposed protocol still outperforms all the existing protocols with the total time of 1.28s. Note that not all the existing protocols include user authentication, for instance, Benit et al's protocol. Hence, the proposed protocol is the first most suitable ID-based one-pass session key establishment protocol for WSNs.

VI. Conclusion

In this paper, we propose a new secure and efficient ID-based one-pass key establishment protocol for WSNs. To the best of our knowledge, this is the first ID-based one-pass authenticated key establishment protocol without pairing. Lack of pairing computation makes it much more efficient for sensor nodes than the existing ID-based one-pass key establishment protocols. At the same time, it enjoys all the desirable security properties for session key establishment protocols. The security and efficiency analysis shows that the proposed protocol performs better than the existing ID-based one-pass key establishment protocols and the key establishment protocols for WSNs.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] A. Shamir, "Identity-based Cryptosystems and Signature Schemes," in *Advances in Cryptology - CRYPTO 1984*. Springer-Verlag, 1985, pp. 47–53.

[3] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Design Codes and Cryptograhpy*, vol. 2, no. 2, pp. 107–125, 1992.

[4] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang, "Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks," in *Proc. WSNA'03*. ACM, 2003, pp. 141–150.

[5] X. Tian, D. Wong, and R. Zhu, "Analysis and improvement of an authenticated key exchange protocol for sensor networks," *Communications Letters*, vol. 9, no. 11, pp. 970 – 972, 2005.

[6] Y. Kim, H. Lee, J. Park, L. Yang, and D. Lee, "Key establishment scheme for sensor networks with low communication cost," in *Autonomic and Trusted Computing*, ser. LNCS, vol. 4610. Springer Berlin / Heidelberg, 2007, pp. 441–448.

[7] L.-P. Zhang and Y. Wang, "An ID-Based Key Agreement Protocol for Wireless Sensor Networks," in *Proc. Int. Conf. Information Science and Engineering*. IEEE Computer Society, 2009, pp. 2542–2545.

[8] R. Yasmin, E. Ritter, and G. Wang, "An Authentication Framework for Wireless Sensor Networks using Identity-Based Signatures," in *Proc. Int. Conf. CIT'10*. IEEE Computer Society, 2010, pp. 882–889.

[9] W. Benits Jr and R. Terada, "An IBE Scheme to Exchange Authenticated Secret Keys," Cryptology ePrint Archive, Report 2004/071, 2004, http://eprint.iacr.org/.

[10] M. C. Gorantla, C. Boyd, and J. M. González Nieto, "ID-based One-pass Authenticated Key Establishment," in *Proc. AISC*. Australian Computer Society,, 2008, pp. 39–46.

[11] T. Okamoto, R. Tso, and E. Okamoto, "One-Way and Two-Party Authenticated ID-Based Key Agreement Protocols Using Pairing," in *Proc. MDAI' 05*, ser. LNCS, vol. 3558. Springer-Verlag, 2005, pp. 122–133.

[12] Y. Wang, "Efficient Identity-Based and Authenticated Key Agreement Protocol," Cryptology ePrint Archive, Report 2005/108, 2005, http://eprint.iacr.org/.

[13] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[14] Y.-M. Tseng, "An Efficient Two-Party Identity-Based Key Exchange Protocol," *Informatica*, vol. 18, no. 1, pp. 125–136, 2007.

[15] R. Zhu, G. Yang, and D. Wong, "An Efficient Identity-Based Key Exchange Protocol with KGS Forward Secrecy for Low-Power Devices," in *WINE'05*, ser. LNCS, vol. 3828. Springer Berlin/Heidelberg, 2005, pp. 500–509.

[16] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," in *Proc. SECON'07*, 2007, pp. 203–212.

[17] M. Bellare, C. Namprempre, and G. Neven, "Security Proofs for Identity-Based Identification and Signature Schemes," in *Proc. EUROCRYPT'04*, ser. LNCS, vol. 3027. Springer, 2004, pp. 268–286.

[18] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: Identity-based Multi-user Broadcast Authentication in Wireless Sensor Networks," *Computer Communications*, vol. 31, no. 4, pp. 659–667, 2008.

[19] D. Galindo and F. D. Garcia, "A Schnorr-Like Lightweight Identity-Based Signature Scheme," in *Proc. AFRICACRYPT'09*, vol. 5580. Springer, 2009, pp. 135–148.

[20] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.

[21] D. F. Aranha, R. Dahab, J. López, and L. B. Oliveira, "Efficient Implementation of Elliptic Curve Cryptography in Wireless Sensors," *Advances in Mathematics of Communications*, vol. 4, no. 2, pp. 169–187, 2010.

[22] L. Chen, P. Morrissey, and N. P. Smart, "Pairings in Trusted Computing," in *Proc. Pairing'08*, ser. LNCS, vol. 5209. Springer Berlin/Heidelberg, 2008, pp. 1–17.

 ISBN: 978-1-61208-144-1