An Authenticated Key Agreement Scheme for Securing In-Network Communication for Constrained Network Devices

Aline Zebaze Tsague
Department of Computer Engineering University of Buea,
Cameroon

Elie Tagne Fute
Department of Mathematics and Computer Science, University of Dschang
Department of Computer Engineering, University of Buea

ABSTRACT
This paperproposes using an ephemeral key-based encryption scheme derived from the Elliptic Curve Diffie-Hellman (ECDH) key exchange scheme to establish a secure communication link between constrained nodes of a hierarchical sensor network. Wireless sensor networks (WSN) consist of a collection of autonomous sensor nodes, interconnected via wireless links and deployed on a geographically limited environment. Regardless of the application for which a WSN can be deployed, security remains one of its main current challenges. The proposed approach is here applied not only to privacy and mutual authentication between the sensors and the base station, but also to the minimization of computational and communication overhead by employing the EC point multiplication from ECC while providing a strong security especially forward secrecy. Security analysis of the proposed scheme shows that it relies on use of short-term key-based encryption and also the intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP) and provides a good number of security properties. With the performance analysis performed, it's shown that the proposed scheme presents the merits of being purely autonomous and lightweight in terms of computational cost and number of communication passes necessary to run its operations.

General Terms
Authentication, Constrained Networks, Key Agreement, Security.

Keywords
Authentication, Ephemeral keys, Forward secrecy, In-network Communication, Security, Signature, Wireless Sensor Network.

1. INTRODUCTION
Based on their realism and concrete contribution in various domains, wireless sensor networks (WSN) have gain much interest whether for the researchers or the industry where they have been used for a diverse range of application scenario. Indeed, they are not novel any longer and are considered as the building blocks of the Internet of Things (IoT) as well as a revolutionary data gathering means. The purpose of a WSN is to gather a set of measures from the immediate environment of the sensors, such as temperature, radioactivity, gaz, atmospheric pressure, etc. in order to convey them to a processing station [1, 3, 5].

Several research works have focused on security issues in WSNs, especially regarding access control, authentication and key management [3, 8, 11-13, 22-24]. These works have led to the development of some efficient and trusted security schemes and protocols for WSN. Most of these security approaches rely on cryptographic primitives including ECC encryption and authentication as in [3, 9], Digital signature and Diffie-Hellman key exchange scheme as in [7, 11], Attribute-Based Encryption as in [12], Certificate-less public key cryptography as in [13], Signcryption and Bilinear pairings computations as in [4, 23], etc..

Through these various approaches, often used in combination, they offer a good number of security properties. Yet, a major consideration when designing a security scheme for WSN is to minimize energy consumption in terms of communicational and computational overhead. Moreover, given that most security protocols in WSN are application specific, security issues remain a current challenge, thus there is still room for improvement in the above-mentioned authentication and key agreement protocols [23, 30].

In this paper, an efficient and autonomous scheme is proposedfor securing in-network communication (among the sensor nodes) such that not only data privacy is guaranteed (confidentiality), but also nodes are sure with whom they share their data (authentication). The main contribution of this paper is to provide authenticated key agreement for a WSN based peer to peer systems. As a matter of fact, establishing authenticated key exchange in such environments require more planning than in client/server environments where authentication methods are server-based. Based on an initial pre-assigned hashed authentication key, the proposed scheme provides mutual authentication while using the ECDH to establish a shared secret key between any two nodes that can communicate directly within a cluster, from which a pairwise encryption key will be derived and used to encrypt their data before transmission. The authentication key is autonomously updated periodically by the nodes themselves using a lightweight method.

The rest of the paper is organized as follows. Section 2 presents and discusses related work. In section 3 the basic definition and properties of the Elliptic Curve Diffie-Hellman (ECDH) key agreement scheme is briefly described. Then the system model of the ECDH-based authenticated key agreement scheme is also briefly introduced. Section 4 presents the proposed security scheme for data communication in WSN. Section 5 reports security analysis of the proposed scheme and its performance features. Finally, section 6 concludes the work and gives future prospects.

2. RELATED WORK

The protection of data's privacy is an important concern in WSNs as the data involved in most WSN applications is sensitive and faces many security threats in open wireless network environment. Over the last few years, several cryptographic and key management mechanisms for efficient security have been proposed [6-9, 20, 27-29]. Indeed, two basic functions of cryptography are to preserve the privacy of communication between two entities and to provide authentication of one entity to another. Yet, cryptographic security mechanism operations demand a high level of computational time and memory resources, while sensor nodes have low memory and low computation capabilities. Elliptic curve cryptography has been popularly used over the years due to the fact that they provide smaller key sizes and higher security strength for each bit of the data. This section briefly reviews existing works in the area of authentication and key management for WSN.

Preetika et al. in [11] proposed using digital signature and Diffie-Hellman key exchange scheme to provide node authentication and to protect confidentiality of data using the Advanced Encryption Standard (AES) encryption algorithm. Their scheme enhances the security concern of key distribution among sensor nodes in order to solve the secured and secret key storage exposure problem. However, the computational overhead is increased with respect to the key sizes and the public key encryption algorithm used.

Portnoi and Shen in [12] proposed LOCATHE (Location-Enhanced Authenticated Key Exchange) a peer-to-peer protocol which combines location, user attributes, access policy and desired services as multi-factor authentication (MFA) factors to allow two parties to establish an encrypted, secure session and further performs mutual authentication with pre-shared keys, passwords and other authentication factors. The proposed protocol uses Attribute-Based Encryption (ABE)-encrypted broadcasts messages for inducing user (a peer) location and Elliptic-Curve Diffie-Hellman Ephemeral (ECDHE) scheme for the key exchange operations. Moreover, the authentication stage involves pairs of messages (a request and a response) plus one initial broadcast exchanged between two parties. Unfortunately, it entails high energy consumption due to the number of messages exchanged as well as the keys size, more if this is to be done regularly.

In [6], Tong et al. proposed a certificateless and anonymous authenticated key agreement scheme for Wireless body area networks (WBAN). In WBAN, sensor nodes collect the patient's (clients) physiological data and transmit it to a medical institution while considering the importance of privacy security and resource constraint. The proposed scheme is server-based and relies on a secure signature scheme from bilinear pairings and an identity-based authenticated key agreement protocol. However, the cost of pairing operations is still high for constrained sensor nodes. Also, the key agreement process not being performed by the communicating peers during authentication but at the registration phase entails the problem of key distribution as the need to be transmitted over a secured channel is posed.

In [24] an enhanced symmetric key-based authentication and key management protocol for IoT-based WSN is presented. The proposed protocol employs pseudodynamic identity and has the ability to counter user traceability, stolen verifier, and DoS attacks identified in the baseline protocol used. User anonymity is a property of authentication protocols where it is desired that the identities of communicating users must not be revealed. As for stolen verifier attack, it's a type of security threat where an adversary steals the data used for verification by the server in past or current sessions. The proposed protocol consists of three (03) main entities while the authentication process takes place in four (04) phases. Yet the authentication latency can be longer since it involves several parties, requiring several communication passes among them and yielding high communication cost. Moreover the key exchange process is not clearly addressed.

Kim and Song in [23] proposed an access control scheme for WSNs in the cross-domain context of the IoT using heterogeneous signcryption. The scheme allows an Internet user in a certificateless cryptography (CLC) environment to communicate a sensor node in an identity-based cryptography (IBC) environment with different system parameters. The signcryption scheme performs the signature and the encryption in one logical step but is derived from bilinear pairings. However, bilinear pairing computation is the most expensive operation in a signcryption scheme.

Vandana et al. in [16] present a reauthentication scheme used for securing mobile node in WSN. The proposed scheme ensures mobile node authentication for two (02) major cases: the first is when a mobile node moves to an adjacent cluster region. The secondcase refers to when the mobile node travels to a non-neighbor cluster region which may be several hops away from the initialposition of a mobile node.

Because the computation ability of wireless sensors nodes in WSNs is very limited, an efficient autonomous authenticated key agreement scheme for WSNs is here proposed.

3. PRELIMINARIES

The Elliptic Curve Diffie–Hellman Key Exchange (ECDH-KE) is an anonymous key agreement scheme, which enables two parties, each having an elliptic-curve public private key pair, to establish a shared secret over an insecure channel. ECDH-KE is very similar to the classical Diffie–Hellman Key Exchange (DHKE) algorithm, but it uses ECC point multiplication instead of modular exponentiations. ECDH-KE is based on the following property of EC points:

(a * G) * b = (b * G) * a where a and b are two secrets numbers belonging to the curve and G its generator point.

Considering an ECC elliptic curve with generator point G, the values of a and b are two private keys belonging to two users A and B wishing to exchange public key and possibly to establish a shared secret between them. This shared secret is obtained using the above EC point property as:

secret = (a * G) * b = (b * G) * a

The security of ECDH-KE relies on the intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP), a variant of the discrete logarithm problem, in which the cyclic group G is represented by the group $\langle P \rangle$ of points on an elliptic curve [14]. Let E be an elliptic curve over a finite field K. Suppose there are points $P, Q \in E(K)$ given such that $Q \in \langle P \rangle$. The ECDLP is to find the integer k, $0 \leq k \leq n-1$, such that $Q = [k]P$.

However, the Diffie-Hellman key exchange by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. In the proposed solution, authenticationis done the earliest possible so as avoid nodes to perform a lot of computations uselessly, before realizing probably that the correspondent is illegitimate. The

proposed system consists of sensor nodes organized into clusters, with each having a cluster head (CH). The CH is responsible for gathering data from the nodes in its cluster and relay these data to the sink (base station). Since the CH depletes its energy resources faster than the other nodes, it is assumed that there is an optimal algorithm to periodically reassign (rotatively) its role to different sensor nodes based on criteria such as residual energy, number of neighbours, and the distance from the base station.

4. PROPOSED SOLUTION

The proposed scheme allows two sensor nodes which are within their respective communication ranges to authenticate mutually while establishing a shared secret key in order to be sure with whom they are exchanging. The authentication and key exchange processes rely not on any third party for its execution, making it distributed and suitable for peer-to-peer systems.

The proposal consists of three phases: initialization, authentication and key update. During the initialization phase, necessary information are generated and preloaded into the sensor nodes before being deployed into the network. Such information includes node identity, elliptic curve domain parameters, authentication key, etc. In the authentication phase sensor nodes authenticate themselves while agreeing on a shared secret key using the ECDH scheme. This phase can be performed as many times as needed with the purpose of re-authentication. Reauthentication is highly required in WSNs and especially in mobile WSNs (MWSNs) to establish secure communications whenever the communication link changes [16, 25, 26]. Yet frequent reauthentication can cause significant energy consumption for the resource-constrained sensor nodes. Therefore, lightweight security mechanisms are required to handle frequent reauthentication in WSNs. In the key update phase, the nodes can update their authentication key by interacting with each other.

4.1 The initialization phase

This phase employs the pre-shared key method to establish the keys and other useful information which will be used by nodes for the authentication process. Recall that the pre-shared key method of authentication is used to enable a remote host to authenticate itself with a peer host by providing a secret key, which is known to both hosts. Any host that does not know the shared key cannot enter into negotiation. The key is pre-configured beforehand (eventually by the network manager or administrator). In this case, the keys are kept into the memory of the sensor as to avoid unauthorized access by third parties.

(a) The manager starts by determining which elliptic curve to use by selecting one from the list of Recommended Elliptic Curve Domain Parameters in [15]. Once selected, the chosen elliptic curve domain parameters are then preloaded into all the nodes. For example, elliptic curve domain parameters over Fp is defined in [15] as being a sextuple:

T = (p, a, b, G, n, h) consisting of an integer p specifying the finite field Fp, two elements a, b ϵ Fp specifying an elliptic curve E(Fp) defined by the equation:

$E : y2 \equiv x3 + a.x + b \pmod p$,

a base point G = (xG, yG) on E(Fp), a prime n which is the order of G, and an integer h which is the cofactor h = #E(Fp)/n.

(b) The manager then generates and preloads the authentication factors: an identity and an authentication key. An identity ID is being generated and assigned to every node. ID can be composed of characters representing the node's name, role or description. To each ID is associated a pseudonym Ps = {0, 1}* used to exchange information within the network. A node's ID also represents its public key and will be used as such during the key update phase. Both the ID and Ps are stored in the nodes. Afterwards the manager generates a random number Ka ϵ Fp as the master authentication key and computes its hash value using a hash function H, HKa = H(Ka). This hash value is then preloaded into all the nodes to be deployed.

4.2 The authentication phase

In this phase, any two nodes which are found within their communication ranges authenticate themselves to each other while agreeing on a shared secret value using the ECDH key agreement scheme as shown on Figure 2. This authenticated key agreement scheme enables each other to detect the earliest possible if its correspondent is legitimate or not. This avoids the nodes to waste their resources on useless computations and operations. Thus, two parties N1 and N2 wishing to establish a communication between them will perform the process as described by Algorithm1.

Algorithm1: ECDH-based Authenticated Key Agreement Scheme

1. N1 generates a random ECC key pair: {N1PrivKey, N1PubKey N1PrivKey * G} and computes its signature as SN1 = E(HI N1PrivKey). Then N1 sends (N1PubKey, SN1, Ps_{N1}) to N2 through insecure channel */* *N1PrivKey* ϵ F_p *is kept secret by N1; E is a signati generation algorithm */*

2. N2 generates a random ECC key pair: {N2PrivKey, N2PubKey = N2PrivKey * G} and computes its signature as SN2 = E(HKa, N2PrivKey). Then N2 sends (N2PubKey, SN2, Ps_{N2}) to N1 through an insecure channel */* *N2PrivKey* ϵ *Fp is kept secret by N2*

3. Upon receiving (N2PubKey, SN2, Ps_{N2}), N1 first verifies SN2 in order to authenticate N2 by performing HK = D(SN2, N2PubKey) and matching HK with HKa. If the matching fails then N2 is not authenticated and the process stops else N2 is authenticated and N1 can now compute the share secret as SK = N2PubKey*N1PrivKey */*D is a signature verification algorithm */*

4. Upon receiving (N1PubKey, SN1, Ps_{N1}), N2 first verifies SN1 in order to authenticate N1 by performing HK = D(SN1, N1PubKey) and matching HK with HKa. If the matching fails then N1 is not authenticated and the process stops else N1 is authenticated and N2 can now compute the share secret as SK = N2PubKey* N1PrivKey // *public chosen value*

5. Now both N1 and N2 have the same SK and know each other's identity

Once the two parties are successfully authenticated, they can now exchange messages securely either by using a symmetric key encryption scheme (with the symmetric key derived from the shared secret key SK using a key derivation function) or using ephemeral key-based encryption scheme derived from the ECDH scheme as described in section 4.4. Moreover as explained in section 5, Security analysis, both the making of keys (private, public and shared secret) and the ECC point multiplication takes less than one second to run. These properties make frequent reauthentication favorable with our method.

4.3 Key update phase

The key update phase as illustrated on Figure 3 is very essential given that it allows the nodes to change their authentication key autonomously and securely by themselves.

Before two nodes update their authentication key, they need to go through another authentication phase in order to make sure that the master authentication key is known and valid. To do this, two parties N1 and N2 use their shared secret key SK and HKa as described by Algorithm2.

Algorithm2: ECDH-based key update process

1. N1computes t_{N1} = G*HKa then encrypts the result using SK as CN1 $E(t_{N1}, SK)$ and sends CN1 to N2./* *E is an encryption algorithm* */

2. N2computes t_{N2} = G*HKa then encrypt the result using SKas CN2 = $E(t_{N2}, SK)$ and sends CN2 to N1. /* *E is an encryption algorithm* */

3. Upon receiving CN2, N1 decrypts it as t* = D(CN2, SK) and verifies if t* = t_{N1} holds. If the comparison is negative, N1 rejects N2 for not being authenticated. Otherwise N1 accepts N2 /**D is a decryption algorithm* */

4. Upon receiving CN1, N2decrypts it as t* = D(CN1, SK) and verifies if t* = t_{N2} holds. If the comparison is negative, N2 rejects N1 for not being authenticated. Otherwise N2 accepts N1 // *D is a decryption algorithm*

5. N1 and N2 then compute their new authentication key as NHKa = $H(t_{N1}) = H(t_{N2})$// *H is a hash function*

The two nodes N1 and N2 authenticate themselves to each other by demonstrating knowledge of the master authentication key and their shared secret before actually updating their authentication key. Obviously this stage is performed without any disclosure of sensitive information (the master key and the shared secret) thanks to encryption and EC point multiplication operations.

4.4 Ephemeral key-based encryption scheme

In this section, an ephemeral key-based encryption scheme derived from the ECDH key agreement described in section 4.2 is presented. An ephemeral key is used temporary and is typically generated for each execution of a key establishment process. In this case, the ephemeral key is symmetric and used more than once, within a single session (for efficiency purposes). The cluster head generates only one ephemeral key per session and encrypts it separately with each recipient's public key. After executing the authentication phase of section 4.2 above, the CH generates and sends the symmetric ephemeral key to its closest neighbors whom it has their public keys by performing the process as described by Algorithm3.

Algorithm3: Symmetric ephemeral key derivation

1. CH chooses a random private key **SessionPrivKey** ϵ F_p/* *temporary private key generation* */

2. CH computes the corresponding public key as **SessionPubKey = SessionPrivKey*G**/* *temporary public key generation* */

3. CH computes the session symmetric key as **SessionEphKey = SessionPrivKey*CHPrivKey**/**CHPrivKey is the CH's private key*/*

4. CH encrypts SessionEphKey using their public keys and sends it to its closest neighbours // *SessionEphKey is used for symmetric encryption within the cluster*

5. Each node then sends SessionEphKey to its neighbours which are not direct neighbours of the CH

Upon receiving the encrypted session ephemeral key from the CH, the direct neighbors of the CH first decrypt it using their respective private keys. Then they also send the ephemeral key to their own prospective neighbors (which are direct neighbors of the CH) by encrypting it with their respective public keys. Once shared with all the nodes of a cluster, the SessionEphKey can be used to encrypt any data to be sent within the cluster. This key can also be updated each time the authentication key is updated between nodes as described in section 5.3.

5. SECURITY ANALYSIS

This section provides security analysis of the proposed scheme. The security of the proposed solution mainly relies on the standards use to implement it: elliptic curve cryptography, digital signature and hash function. The security of ECC relies on the intractability of the discrete logarithmic problem for large prime numbers. The more the problem difficulty is, the better the security is. It also requires less complex computations as it employs EC point multiplication which is less costly as compared to exponentiation calculation and pairing-based operations. For example the implementation of 160-bit ECC on an Atmel AT-mega 128, which has an 8-bit 8 MHz CPU, shows that an ECC point multiplication takes less than one second [13]. In the case of the proposed scheme, both a 163-bit ECC and a 256-bit ECC have been implemented on an ESP8266 to which a DS18B20 Waterproof Digital Temperature Sensor is connected. This device was used to monitor the temperature within an enclosure used to store medicines in a medical center. The elliptic curve used for the ECDH computations were sect163k1 and secp256k1 respectively associated with a Koblitz curve [15]. The private keys are 163 bits and 256-bits (41 and 64 hex digits respectively) and are generated randomly. The public keys are 257 bits (65 hex digits), due to key compression. The result shows that the making of keys (private, public and shared secret) is done each in almost constant time within the range of 600ms and 615ms. Thus, the proposed scheme is computationally efficient and suitable for wireless sensor networks.

The security of the proposed scheme also depends on digital signature and hashing. Hashing (such as the SHA-3 family) algorithms provide special properties, such as resistance to collision, pre-image, and second pre-image attacks. These hash functions are also components for many important information security applications, including the generation and verification of digital signatures. Therefore, the proposal does not suffer from usual attacks based on cryptographic operations (also like identity theft and Man in the Middle attack).

Furthermore the preloaded information are done by a trusted party (network manager or administrator) and do not suffer from any form of attack. The master authentication key, which is used to authenticate the nodes during key agreement, is hashed before being preloaded into the nodes. Any node in possession of a public key cannot know its corresponding private key thanks to the public-key cryptosystems property stipulating "Knowledge of the algorithm plus one of the keys plus samples of ciphertexts must be insufficient to determine the other key". Similarly, any node in possession of a hash value cannot derive the value of the original information thanks to the pre-image and collision properties of hash functions. Therefore during the authentication process one needs to prove to the other that it knows a secret without revealing the secret itself.

The security properties of the proposed scheme can be described as follows:

Privacy: This property ensures that an attacker does not get any sensitive information (such as the identities of legitimate nodes and contents of messages) in authentication process.

None of such information is disclosed during the execution of the proposed scheme. As indicated in section 4, nodes' identities and pseudonym are generated under the supervision of a trusted party (network manager or administrator). Only the corresponding pseudonym to a node's identity is used during the authentication phase in section 4.2, the real identity being kept secret. Moreover no other information apart from the EC public keys is transmitted in clear form. The EC public keys have no stake if they are disclosed given that the secret keys used to compute them cannot be derived thanks to the intractability of the elliptic curve discrete logarithm problem. As for the authentication key Ka and the signature S, they are all protected by properties of hashing techniques and encryption. Indeed, Ka is stored in its hashed form and S is encrypted with the node's private key, which is chosen secretly.

Forward secrecy: This property ensures that in case the private key of a node is compromised, the adversary could not effectively generate the forward session key and the confidentiality of previous session keys is still fulfilled [21]. Since the authentication factor Ka is prepared and stored into the nodes beforehand, only legitimate nodes have possession of it. Therefore if the private key of a node is disclosed, the adversary cannot compromise the session key because the adversary cannot authenticate with other nodes, lack of the authentication key.

Mutual authentication: This property is used to demonstrate the legitimacy of the node's identity in the WSNs, so as to achieve the purpose of identifying and preventing illegal third parties from participating in communications. Nodes authenticate themselves during the authentication phase by demonstrating knowledge of Ka through signature. Because digital signature provides authentication by nature, the nodes can actually be sure of each other after successful verification of their signatures.

Non repudiation: The property ensures that a node cannot deny the validity of their signature or of an authenticated key establishment process. A node computes the signature information with another party for authentication; once the authentication is successful, the node cannot deny that he/she has established a shared secret with the other party.

Availability (of session key): Upon a successful mutual authentication process, a session key is established between the nodes within a cluster for secure subsequent communications. This session key is used to encrypt collected data while preserving them from unauthorized disclosure.

Furthermore, there are no assumptions of information being sent through a secure channel. This makes the security scheme look incomplete. The scheme proposed is as autonomous as much as possible to enable sensor nodes to be able to perform it themselves.

6. PERFORMANCE ANALYSIS

This section presents and discusses the performance evaluation and features of the proposed solution. Then it is compared with some existing solutions, that of Tong et al. [6], Portnoi and Shen [12] and of Vandana et al. [16].

The comparison starts by an evaluation in terms of number of communication passes needed by the considered schemes for the authentication and key update phases. This evaluation shows that the proposed scheme requires only two passes for each of the phases. Tong et al.'s scheme also requires two passes in the authentication phase like ours but its key update phase requires more passes because it involves a third party (a server) in this process. Regarding Vandana et al.'s scheme, four communication passes are required for each phase. As for Portnoi and Shen's scheme, the key update phase is not explicitly given but nodes can update their keys by reregistering. During their registration phase, a user registers with the service running the proposed protocol and exchanges security parameters such as the service's public key, the user's ABE (Attribute-Based Encryption) secret key and attributes, the seed and clock for the token authenticator algorithm, base point G for ECDHE, key-derivation function (KDF) salts, and a secret shared user key [12]. Thus this requires the exchange of at least two messages between the user and the service.

In order to evaluate the message sizes, the following parameter settingswere used,inspired from [16]: 2 bytes for an ID, 4 bytes for a MAC, 8 bytes for a timestamp, 8 bytes for a random number 16 bytes for a key size and 1 byte for a number of hops from source to destination. During the authentication phase, the proposed scheme requires each node to transmit a message which comprises 1key + 1MAC + 1ID = 16 + 4 +2 = 22 bytes. The signature here is obtained from a key reason why its size is considered as such. Tong et al.'s scheme requires a message from the client of 1key + 1MAC + 2ID + 1 + 1time stamp = 16 + 4 + 11 = 31 and another from the AP of 1MAC = 4 bytes, yielding a total of 31 + 4 = 35 bytes. As for Vandana et al.'s scheme, considering the case where a node authenticates within its initial cluster, there is one message from the node as 2ID + 1MAC + 1Key = 4 + 4 + 16 = 24 bytes. In this case the number of hops (i.e. the hop distance from node to the cluster head) is considered as 1. Portnoi and Shen's scheme for its part requires a message of 2keys + 2random number + 1time stamp = 32 + 16 + 8 = 56 bytes.

During the key update phase, the proposed scheme requires to transmit a message consisting of 1key = 16 bytes. Tong et al.'s scheme on its own requires 1authentication message + 1key = 35 + 16 = 51 bytes. As for Vandana et al.'s scheme, the pairwise session key is obtained after the node has been authenticated and is accompanied with a MAC for verification by the node. Thus, it requires the same message size as for the authentication phase, that is 24 bytes + 1MAC = 24 + 4 = 28 bytes. Finally Portnoi and Shen's scheme uses 2keys + 1random number + 1time stamp + attributes = 32 + 8 + 8 = 48 bytes, neglecting the size of the attributes. Table 1 presents the message sizes in various schemes during the authentication and key update phases. Figure 4 presents a comparison of the message sizes in bytes exchanged during both phases while figure 5 shows the variation of these message sizes when the authentication phase is being executed concurrently between different pairs of nodes.

Table 1: Message sizes during the Authentication and Key update phase for various schemes

Scheme	Authentication	Key update
Tong et al.(2017)	35 bytes	51 bytes
Portnoi and Shen (2016)	56 bytes	48 bytes
Vandana et al.(2019)	24 bytes	28 bytes
Our scheme	22 bytes	16 bytes

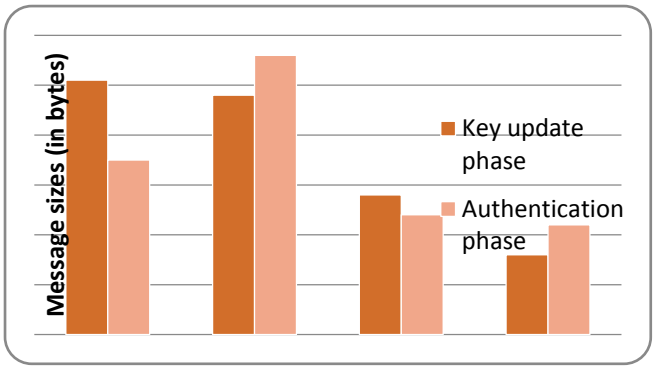

Figure 1: Message sizes exchanged during the authentication and Key update phases

As for computational cost, the different types of computations required for performing the authentication and key update phaseshave been evaluated. The results are presented in Table 2. In order to carry out this evaluation for the various schemes, the following notationshave been adopted:

- Ch: the cost of a hashing operation
- Ced: the cost of an encryption or decryption operation
- Cabe: the cost of an attribute-based encryption
- Cecm: the cost of an elliptic curve point multiplication operation
- Cadd: the cost of an elliptic curve point addition operation
- Cbp: the cost of a bilinear pairing operation

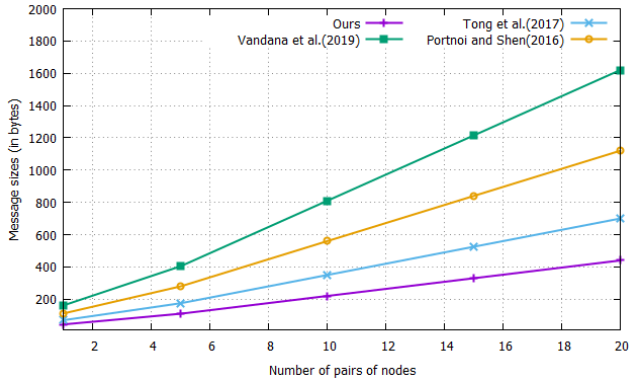

Figure 2: Evolution of message sizes exchanged during the authentication phase

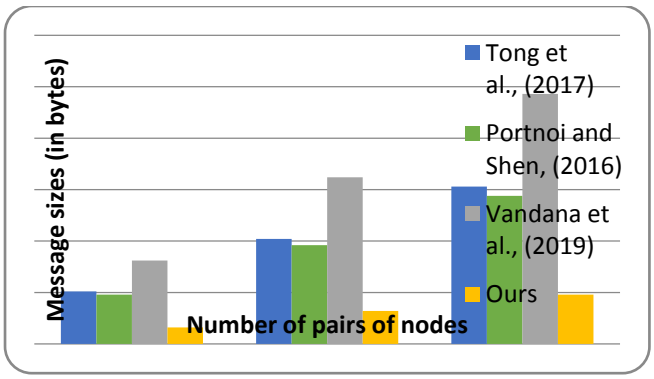

Figure 3: Message sizes exchanged during the key update phase

The evaluation shows that for the authentication phase, the proposed scheme has an advantage over the two others in that it does not perform any hashing operation. Yet the costs are relatively different from one scheme to another. For the key update phase the proposed scheme presents the best computational cost. It is worth highlighting that the key update phase for all three schemes rely on the same principle which requires that nodes should go through the authentication phase to make sure that the past session key is valid before they update the session key by re-registering.

Thus,the proposed scheme presents some performance features such as: efficiency, lightweight, autonomous and secured.

Table 2: Computational cost in the Authentication and Key update

Scheme	Authentication	Key update
Tong et al.	$2C_{ecm} + 1C_h + 1C_{ed}$	$4C_{ecm} + 1C_h + 1C_{ed} + 1C_{add} + 2C_{bp}$
Portnoi and Shen	$1C_{ecm} + 1C_h + 2C_{ed}$	$1C_{ecm} + 1C_{ed} + 1C_{abe}$
Vandana et al.	$4C_h + 2C_{ed}$	$4C_h + 3C_{ed}$
Ours	$2C_{ecm} + 2C_{ed}$	$1C_{ecm} + 1C_h + 2C_{ed}$

Since the various schemes use authenticated key agreement with different cryptographic primitives and pre-distribution of parameters, the energy consumption due to the computation is relativized. Thus, only energy consumption due to the communication is evaluated and compared. Moreover, given that communication is the major source of energy depletion in general, only the energy consumption comparison based on communication is consideredto highlight the efficiency of the proposed scheme. Based on the message sizes, energy consumption is computedby using the parameters from [16, 17]: 16.25 µJ per one byte transmission and 12.5 µJ per one byte reception. Figure 7 illustrates the energy consumption for various schemes during the authentication phase. During the authentication phase, our scheme requires each node to transmit and receive a message 22 bytes yielding a consumption of 22 * 16.25 = 357.5µJ and 22 * 12.5 = 275µJ for a total of 632.5µJ. Tong et al.'s scheme requires to transmit a message 31 bytes, yielding 31 * 16.25 = 503.75µJ. Reception uses 4 bytes, yielding 4 * 12.5 = 50µJ and a total consumption of 503.75 + 50 = 553.75µJ. Vandana et al.'s scheme on its own requires on the one hand to transmit 24 bytes as calculated previously, which yields 24*16.25 = 390µJ. On the other hand, reception takes 2ID + 1random number + 1Key = 4 + 8 + 16 = 28 bytes yielding 28*12.5 = 350µJ, for a total consumption of 390 + 350 = 740µJ. Finally, Portnoi and Shen's scheme requires to transmit and receive a message of 56 bytes, yielding a consumption of 56 * 16.25 = 910µJ and 56 * 12.5 = 700µJ for a total of 1610µJ.

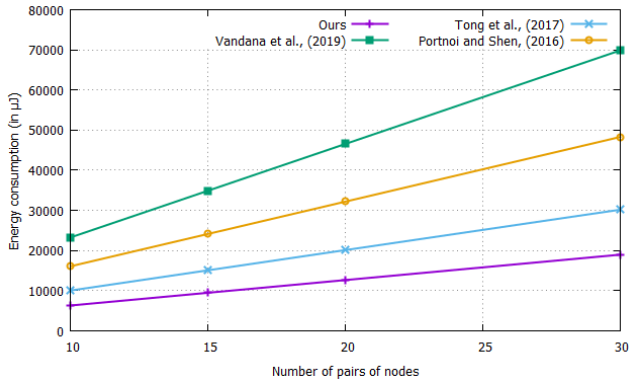

Figure 4: Energy consumption due to communication during authentication

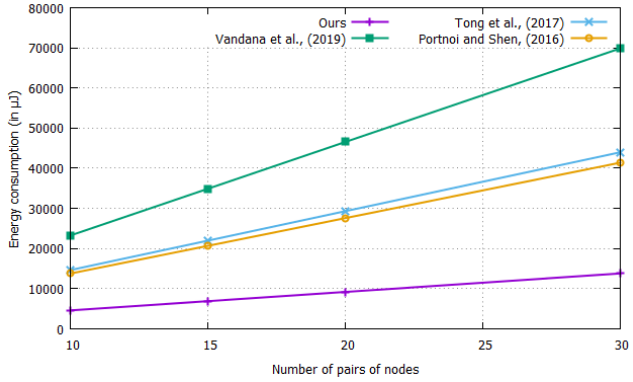

Figure 5: Energy consumption due to communication during key update

Similarly, figure 8 illustrates the energy consumption for various schemes during the key update phase. During the key update phase, our scheme requires to transmit and receive a message 16 bytes, yielding 16 * 16.25 = 260µJ and 16 * 12.5 = 200 µJ, for a total of 460µJ. Tong et al.'s scheme on its own requires to transmit 51 bytes, yielding 51 * 16.25 = 828.75µJ. Reception uses 16 bytes, yielding 16 * 12.5 = 200µJ for a total consumption of 1028.75µJ. Vandana et al.'s scheme requires on the one hand to transmit 24 bytes which yields 24*16.25 = 390µJ. The reception takes 24 + 4 = 28 bytes yielding 28*12.5 = 350µJ, for a total consumption of 390 + 350 = 740µJ. Portnoi and Shen's scheme requires each node to transmit a message of 16 bytes yielding 16 * 16.25 = 260µJ and receive a message of 32 bytes, yielding 32 * 12.5 = 400µJ for a total of 600µJ.

7. CONCLUSION AND FUTURE WORK

In this paper a lightweight and efficient security scheme for data communication within a wireless sensor networkhas been proposed. Based on the information provided beforehand to nodes, they can autonomously execute and provide the network with essential security features such as privacy, data integrity, non-repudiation and authenticated key agreement. The operations carried out on the nodes are simple and fast as compared to the level of security offered. Moreover, the information that flows during the scheme execution do not reveal nor might be used to derived any sensitive information. Thus, the proposed solution provides perfect forward secrecy, but requires the distribution of credentials (e.g. master authentication key) pre-deployment. However, it avoids computational and management overheads created by alternative solutions that provide exponentiation calculation, pairing-based operations, digital certificates and public key infrastructures in conventional IP networks. It's worth highlighting that the proposed scheme can be very efficiently implemented on sensor nodes since only few and lightweight operations are required such as EC point multiplication, hash function and symmetric encryption.

As future research directions, we plan to analyze and study the performance of the proposedscheme when new nodes join and/or leave the network, while employing aggregation techniques to minimize message sizes exchanged during the authenticated key agreement process. This will help reduce the storage resources and the energy consumption.

8. ACKNOWLEDGMENTS

Appreciations to the different experts who have contributed towards enrichment of this work.

9. REFERENCES

[1] G. Abdul-Salaam, A. H. Abdullah, M. H. Anisi, A. Gani, A. Alelaiwi, (2016). A comparative analysis of energy conservation approaches in hybrid wireless sensor networks data collection protocols. In Telecommunication Systems, Springer US, 61(1):159–179

[2] S. Aruna, L. M. Varalakshmi, (2013). Data Gathering Using Sink Mobility with Three Tier Security Scheme in Wireless Sensor Network. International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, 2(12)

[3] P. Kasyoka, M. Kimwele, S. M. Angolo, (2020). Multi-user broadcast authentication scheme for wireless sensor network based on elliptic curve cryptography. Engineering Reports. John Wiley & Sons, Ltd. https://doi.org/10.1002/eng2.12176

[4] S. Jebri, M. Abid, A. Bouallegue, (2018). LTAMA-algorithm: light and trust anonymous mutual authentication algorithm for IoT. In Proceedings of the IEEE 87th Vehicular Technology Conference (VTC Spring).

[5] M. Krol, (2016). Routing in Wireless Sensor Networks. PhD thesis, University of Grenoble, Grenoble, France

[6] L. Tong, Z. Yuhui, Z. Ti, (2017). Efficient Anonymous Authenticated Key Agreement Scheme for Wireless Body Area Networks. Security and Communication Networks, https://doi.org/10.1155/2017/4167549.

[7] C. Chen-Yang, L. Iuon-Chang, H. Shu-Yan, (2015). An RSA-like scheme for multiuser broadcast authentication in wireless sensor networks. International Journal of Distributed Sensor Networks. 11(9):743623.

[8] B. Muhammad, K. Shin-Gak, (2017). A Secure Key Agreement Protocol for Dynamic Group, Cluster Computing. The Journal of Networks, Software Tools and Applications" ISSN: 1386-7857 (Print) 1573-7543 (Online)

[9] H. Zhong, R. Zhao, J. Cui, X. Jiang, J. Gao, (2016). An improved ECDSA scheme for wireless sensor networks. International Journal of Future Generation Communication and Networking. 9(2):73-82. http://dx.doi.org/10.14257/ijfgcn.2016.9.2.08

[10] D. Manali, D. Aaradhana, (2013). Achieving Authentication and Integrity using Elliptic Curve Cryptography Architecture. International Journal of Computer Applications (0975 – 8887), 69 (24)

[11] J. Preetika, V. Manju, R.V. Pushpendra, (2015). Secure Authentication Approach Using Diffie-Hellman Key Exchange Algorithm for WSN. In 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pages 527-532

[12] M. Portnoi, C.C. Shen, (2016). Location-Enhanced Authenticated Key Exchange. 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, Hawaii, USA

[13] S. Seung-Hyun, W. Jongho, S. Salmin, B. Elisa, (2015). Effective Key Management in Dynamic Wireless Sensor Networks. IEEE Transactions on Information Forensics and Security, 10(2)

[14] C. Wang, G. Xu, J. Sun, (2017). An Enhanced Three-Factor User Authentication Scheme Using Elliptic Curve Cryptosystem for Wireless Sensor Networks. Sensors 17, 2946; https://doi.org/10.3390/s17122946

[15] Certicom Research (2010). Standards for Efficient Cryptography 2 (SEC 2): Recommended Elliptic Curve Domain Parameters, Version 2.0.

[16] M. Vandana, B. Ravindara, S. Yashwant, (2019). Reauthentication scheme for mobile wireless sensor networks. Sustainable Computing: Informatics and Systems, 23(2019): 158–166

[17] B. Kim, J. Song, (2017). An Efficient and Practical Mobile Node Reauthentication Scheme for Mobile Wireless Sensor Networks. Proceedings of the 3rd International Conference on Communication and Information Processing, pages 326–331

[18] Z. Quan1, T. Chunming, Z. Xianghan, R. Chunming, (2015). A secure user authentication protocol for sensor network in data capturing. Journal of Cloud Computing: Advances, Systems and Applications, 4:(6). DOI 10.1186/s13677-015-0030-z

[19] Z. Asim, I. A. K. M. Muzahidul, Z. Mahdi, A. M. Ishtiak, M. Nafees, B. Sabariah, K. Yoshiaki, K. Shozo, (2016). Clustering Analysis in Wireless Sensor Networks: The Ambit of Performance Metrics and Schemes Taxonomy. https://doi.org/10.1177%2F155014774979142

[20] Y. Chen, L. López, J.-F. Martínez, P. Castillejo, (2018). A Lightweight Privacy Protection User Authentication and Key Agreement Scheme Tailored for the Internet of Things Environment: LightPriAuth. Journal of Sensors, 2018:(7574238), pages 1-16. https://doi.org/10.1155/2018/7574238

[21] M. F. Fernandez, P. Caballero-Gil, C. Caballero-Gil, (2016). Authentication Based on Non-Interactive Zero-Knowledge Proofs for the Internet of Things. MDPI Journal Sensors, doi:10.3390/s16010075

[22] Q. Wang, W. Chen, L. Wang, (2019). A Sink Node Trusted Access Authentication Protocol for Mobile Wireless Sensor Network Using Block Cipher Algorithm Based on IoT. International Journal of Wireless Information Networks, https://doi.org/10.1007/s10776-019-00471-6

[23] M. Luo, Y. Luo, Y. Wan, Z. Wang, (2018). Secure and Efficient Access Control Scheme for Wireless Sensor Networks in the Cross-Domain Context of the IoT, Hindawi Security and Communication Networks, https://doi.org/10.1155/2018/6140978

[24] A. Ghani, K. Mansoor, S. Mehmood, S. A. Chaudhry, A. U. Rahman, M. N. Saqib, (2019). Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key, Int J Commun Syst. https://doi.org/10.1002/dac.4139

[25] B. Kim, J. Song, (2019). Energy-efficient and secure mobile node reauthentication scheme for mobile wireless sensor networks. Journal on Wireless Communications and Networking, 155 https://doi.org/10.1186/s13638-019-1470-9

[26] V. Mohindru, R. Bhatt, Y. Singh, (2019). Reauthentication scheme for mobile wireless sensor networks. Sustainable Computing: Informatics and Systems, 23: 158-166. https://doi.org/10.1016/j.suscom.2019.07.010

[27] M. Bilal, S.-G. Kang, (2017). An authentication protocol for future sensor networks. Journal of Sensors, 17:(979), pages 1–29.

[28] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila1, B. Stiller, (2015). Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications. IEEE Access: The journal of rapid open access publishing. 3:(2015). DOI 10.1109/ACCESS.2015.2474705

[29] A. P. Renold, B. G. Athi, (2019). Energy efficient secure data collection with path-constrained mobile sink in duty-cycled unattended wireless sensor network. Pervasive and Mobile Computing, 55 :1–12. https ://doi.org/10.1016/J.PMCJ.2019.02.002.

[30] M. El-hajj, A. Fadlallah, M. Chamoun, A. Serhrouchni, (2019). A Survey of Internet of Things (IoT) Authentication Schemes. Sensors 19:1141; doi:10.3390/s19051141