An efficient and lightweight identity-based scheme for secure communication in clustered wireless sensor networks

Fares Mezrag, Salim Bitam, Abdelhamid Mellouk

Version of Record: https://www.sciencedirect.com/science/article/pii/S1084804521002782
Manuscript_b8df7591dc2ee898d287bf556b87ef39

An efficient and lightweight identity-based scheme for secure communication in clustered wireless sensor networks

Fares Mezrag[a], Salim Bitam[b], Abdelhamid Mellouk[c,*]

[a]*Department of Computer Science, University of M'Sila, Algeria*
[b]*Department of Computer Science, University of Biskra, Algeria*
[c]*University of Paris-Est Creteil, LISSI/TincNET, F-94400, Vitry sur Seine, France*

Abstract

Clustered Wireless Networks (CWSNs) are typically deployed in unsecured or even hostile areas, making them vulnerable to many cyber-attacks and security threats that adversely affect their performance. Furthermore, the design of an efficient cryptographic scheme for CWSN is challenging due to the dynamic nature of the network and resource-constrained sensor devices. The paper presents a new identity-based authentication and key agreement scheme for CWSNs called IBAKAS, which combines Elliptic Curve Cryptography (ECC) and Identity-Based Cryptography (IBC) to provide mutual authentication and establish secret session keys over insecure channels. IBAKAS achieves all desirable security properties of key agreement and prevents specific cyber-attacks on CWSN. Moreover, the formal security of the proposed scheme is verified using the AVISPA tool. Comparison with existing relevant schemes shows that the proposed scheme decreases

*Corresponding author
Email addresses: fares.mezrag@univ-msila.dz (Fares Mezrag), s.bitam@univ-biskra.dz (Salim Bitam), mellouk@u-pec.fr (Abdelhamid Mellouk)

computational and communication overheads, saves keys storage space and prolongs the network lifetime by reducing the energy consumption of the sensor node.
Keywords: Cluster-Based WSN, Identity-Based Cryptography, Elliptic curve, Mutual authentication, Key agreement, AVISPA

1. Introduction

Wireless sensor networks (WSNs) are considered one of the emerging technologies that have attracted wide attention from industry and academia due to their ability to use them in many applications, such as military, healthcare , and industrial control. WSN is made up of many tiny devices called sensor nodes which are deployed in a monitored area. These nodes can wirelessly communicate and exchange data between them without using fixed network infrastructure. However, WSN is usually characterized by the resource-constrained nature of sensor devices such as processing, energy, storage space, and bandwidth. Besides the limited energy nature, recharging or replacing batteries is considered a difficult task in sensors deployed in an inaccessible environment. Therefore, this issue would adversely affect the network lifetime.

To extend the wireless sensor network lifetime by reducing the energy consumption of a sensor node, clustering mechanism was proposed (Fanian and Rafsanjani, 2019; Yousefpoor et al., 2021; Mezrag et al., 2017). In a CWSN, a whole network is partitioned into groups called clusters. Each has one Cluster Head (CH) and several sensor nodes known as Cluster Members (CMs). The CH is responsible for aggregating data gathered from all CMs and then transmits the result to Base Station (BS). The latter serves as a gateway for transmitting data to the end-user over a traditional wired or wireless network.

Network security is the set of policies, mechanisms, and services that protect a network from cyber-attacks and unauthorized access (Yousefpoor and Barati, 2019). Security in CWSN faces several challenges, especially when it comes to applications requiring a high level of security, such as military, emergency response, and healthcare (Benayache et al., 2019; Jain and Hussain, 2020). Sensor devices are frequently deployed in hostile or even unsecured environments, which make them subject to more cyber-attacks that can violate sensitive data and adversely affect the performance of a network (Jiang et al., 2019; Boubiche et al., 2021). Furthermore, wireless communications within the CWSN are insecure by nature, and as a result, an adversary with a wireless device can easily listen in on communications between legitimate nodes. Therefore, minimal security requirements such as authentication, data confidentiality, and data integrity must be assured. Also necessary to design a lightweight, efficient, and secure scheme that considers the resource-constrained sensor nodes. In this context, a form of public-key cryptography known as Identity-Based Cryptography (IBC) is considered a practical security solution for resource-constrained devices (Sogani and Jain, 2019; Kim et al., 2019; Saeed et al., 2019; Hamouid et al., 2020; Mishra et al., 2021; Kumar et al., 2021; Tseng et al., 2021). This is due to several features, including the following:

- IBC provides basic security requirements at a low cost regarding computational overhead, storage space, and energy consumption. Therefore, this feature makes IBC suitable for devices with limited resources such as sensor nodes.
- Compared with symmetric key cryptography, the key distribution in IBC is uncomplicated and easier to manage.

- Unlike traditional public-key infrastructure (PKI), A public key in IBC is self-authenticated, and a digital certificate is not required.

Two main techniques have been used in the literature to implement IBC-based schemes for sensor nodes: bilinear pairing-based and ECC-based. However, according to recent implementation results on many WSN platforms, the time required to compute a single bilinear pairing is equal to the computation between two to seven elliptic curve point multiplications (Shim, 2016). Therefore, IBC schemes based on pairing are considered slow and increase a computation overhead for sensor nodes compared with IBC schemes based on the elliptic curve.

1.1. ***Contribution***

This paper proposes an efficient and lightweight identity-based authentication and key agreement scheme for CWSN called IBAKAS. The preliminary version of this scheme is published in IEEE PIMRC 2019 (Mezrag et al., 2019). IBAKAS depends on ECC and IBC, provides mutual authentication and establishes a session key between two communicating parties over a public channel. The session key can be established between CH and CM or between CH and BS, and it is used for secure data transmission. The main properties of the proposed IBAKAS are as follows:

1. *No public key certificates are necessary*: The proposed scheme is designed to use IBC. Consequently, our scheme provides easy management of public keys compared to PKI-based cryptosystems, and there is no need to generate and maintain public-key certificates.

2. *Elimination of bilinear pairing and MTP function*: According to our implementation results on the WiSMote sensor device (See Table 4), the time

required to compute a single bilinear pairing is equal to the computation of seven elliptic curve point multiplications (EM). Furthermore, the computation overhead of one Map-To-Point function (MTP) is more than an EM. Therefore, pairing computations and MTP are computationally expensive and not suitable for resource-constrained sensor devices. Our scheme does not require any pairing computation and MTP function in order to establish session keys.

3. *Formal and informal security analysis*: The formal security of the proposed IBAKAS is verified using AVISPA tool. The simulation results show that IBAKAS is safe and resistant to passive and active cyber-attacks, including eavesdropping, MITM and replay attacks, and it achieves security goals, such as confidentiality and mutual authentication. Moreover, IBAKAS achieves all the desirable security properties of the authenticated key agreement described in (Blake-Wilson et al., 1997). A comparison of security features with the existing relevant schemes is also provided in this research activity (See Table 2).

4. *Resource-efficiency*: IBAKAS is resource-efficient. Comparison with existing relevant schemes shows that IBAKAS decreases computational and communication costs, save key storage space and reduces the energy consumption on WiSMote sensor devices.

1.2. ***Paper organization***

The remainder of this paper is organized as follows. In section 2, we discuss related works and we describe preliminary knowledge and the system model in section 3. Section 4 illustrates the phases of our proposed scheme (IBAKAS), then

the security analysis and the performance results are presented in section 5 and 6, respectively. The section 7 describes two examples of application scenarios. The last section concludes this work with a summary and future research directions.

2. Related Works

In recent years, several identity-based schemes have been proposed in the literature for securing WSNs. In this section, we review and critically analyze these schemes.

The authors assumed in (Mehmood et al., 2017) that a CH is an important node in the network, and it is more vulnerable to cyber-attacks than other sensor nodes. Thus, they proposed a public key-based scheme called Inter-Cluster Multiple Key Distribution Scheme (ICMDS), which focuses on securing CHs and makes data routing unreadable by intermediate nodes. Furthermore, ICMDS is based on pairing operations to secure inter-cluster communication. However, the authenticity of nodes is provided with involving the BS, where this way is not preferred in WSN environments. Moreover, ICMDS is vulnerable to cyber-attacks such as replay attack and cluster head impersonation attack, and it suffers from a lack of mutual authentication between sensor nodes. In addition, a public key can be intercepted by a malicious node, and therefore the communication between nodes is exposed to Man-In-The-Middle (MITM) attack.

To overcome security weaknesses of ICMDS (Mehmood et al., 2017), an enhanced scheme was introduced in (Harbi et al., 2019) called a Mutual Authentication and session Key Agreement (MAKA). The proposed scheme uses a pairing over elliptic curves in order to introduce a session key agreement and to achieve mutual authentication between CH and CMs. Furthermore, MAKA is designed to

secure all communications in the network rather than securing inter-cluster communication. However, MAKA applies asymmetric encryption/decryption operations, and it uses large-size messages. Such factors are considered unsuitable for resource-constrained node because they require high computation and communication costs. The authors assumed that all sensor nodes share a master secret key k. T_{min} is regarded as a required time by a sensor node to compute its private key using the key k. If T_{min} is expired, each sensor node deletes k. Note that if an adversary physically compromises any legitimate sensor node before T_{min}, it can access the key k. Thus, all private keys can be discovered by an adversary. Consequently, it is able to decrypt all exchanged messages and to generate a digital signature for any legitimate sensor node.

In (Saeed et al., 2019), the authors have proposed AKAIoTs: an identity-based authentication key agreement scheme for WSN-IoT based on elliptic curves and Diffie-Hellman (DH) Key exchange. The proposed scheme is used to secure data transmission between sensor nodes and a cloud server in IoTs. Regarding the security aspect, the authors have verified that AKAIoTs is secure in the random oracle model. AKAIoTs ensures several security properties of key agreement. Besides, it can prevent specific cyber-attacks such as eavesdropping and replay attacks. However, to establish a shared key, a sensor node requires six point multiplications, which are considered expensive for a resource-constrained node.

In (Kar et al., 2020), the authors presented MA-IDOOS: an ID-based security scheme for WSN, which used an ID-Based Online/Offline digital Signatures (IBOOS). In MA-IDOOS, the authors focused on securing the communication between sensor nodes and BS. To that end, they exploited a bilinear pairing over elliptic curves to achieve message authentication and protect data integrity. To

ensure end-to-end confidentiality, the authors used the homomorphic encryption scheme proposed in (Castelluccia et al., 2005). According to the experimental results, the proposed protocol provides a good resilience to active and passive attacks. However, MA-IDOOS suffers from a lack of mutual authentication. Additionally, the authors use SHA-1 as a hash function, which is considered broken and no longer secure. MA-IDOOS requires high computational and communication overheads. Therefore, this issue would adversely affect the network lifetime.

A secure data aggregation scheme was introduced in (Zhong et al., 2018). The authors used a combination of a homomorphic encryption and an identity-based signature schemes to enhance the security in heterogeneous CWSN. The proposed scheme includes five algorithms: *Setup*, *Private key extraction*, *Encrypt-Sign*, *Verify-Aggregate-Sign*, and *Verify-Decrypt*. The BS runs the first algorithm to generate its master private key and publish the system parameters across the entire network. In the *Private key extraction* process, the BS generates private keys for both CHs and CMs using the BS's master private key. Next, each CM needs to *Encrypt-Sign* algorithm for encrypting and signing its sensed data. Then, the result is sent to the corresponding CH. The signature generation in the *Encrypt-Sign* algorithm is based on the CM's private key. Using the *Verify-Aggregate-Sign* algorithm, the CH verifies all signatures received from its CMs by batch signature verification, aggregates all encrypted data, and signs the aggregated ciphertext using the CH's private key. The result is forwarded to BS. In the last algorithm, the BS first checks the aggregated ciphertext through batch signature verification. Then, the BS decrypts the aggregated ciphertext.

Regarding the security aspect, the proposed scheme achieves data confidentiality and integrity. Moreover, it can resist specific cyber-attacks such as replay

and eavesdropping attacks. However, the recoverable sensing data is inefficient in the proposed scheme due to large-sized of aggregated messages.

In (Hamouid et al., 2020), the authors proposed a Lightweight and Secure Tree-Based Routing (LSTR) for WSN, which ensures a trade-off between resource efficiency and security. The design of LSTR aims at using a tree structure where the root is a BS, and the tree leaves are sensor nodes. The routing tree is constructed to connect each sensor node to the BS through the short and secure path. To secure the communication among sensors nodes, the authors adopted an ID-based authenticated key agreement scheme (Chen and Kudla, 2003) which is based on bilinear pairing. LSTR ensures confidentiality and authenticity of messages. It further prevents specific cyber-attacks, including eavesdropping, Sybil, key compromising, and impersonation attacks. Based on the presented experimental results, LSTR requires low communication and storage costs. However, its computational overhead is considerable.

An ID-based security scheme was proposed for WSNs in (Kumar et al., 2021). The proposed scheme is designed to introduce an authenticated key agreement to establish a secret session key between two sensor nodes. Moreover, the authors used hexadecimal extended ASCII-ECC to encrypt/decrypt a user's identity. However, the proposed scheme is inefficient in terms of computational cost. Thus is not suitable for devices with limited resources.

In (Shen et al., 2017), the authors proposed an Identity-Based Aggregate Signature (IBAS) scheme for heterogeneous WSN by adopting an identity-based signature with a bilinear pairing. The authors assume that the network model of IBAS consists of three components, including BS, CH, and CM. The CH acts as an aggregator, a special node with a more powerful resource. The CMs of the

same cluster send their signatures to the corresponding CH. The latter aggregates the signatures received into a single signature called the aggregated signature. Then the result is forwarded to the BS for verification. IBAS scheme comprises six algorithms, including *setup*, *Key-Generation*, *Signing*, *Verification*, *Aggregation*, and *Agg-Verification*. The BS runs the *setup* algorithm to obtain the master secret key and initialize the system parameters. In addition, the BS generates private keys for both CHs and CMs using the *Key-Generation* algorithm. The CMs run the *Signing* algorithm to generate their signatures, while CHs run the *Verification* algorithm to check the signatures received. The *Aggregation* and *Agg-Verification* algorithms are used to generate the aggregate signatures and verify them, respectively. IBAS ensures data integrity and authentication while reducing communication and storage costs. However, data confidentiality is not ensured. Consequently, the proposed scheme is vulnerable to eavesdropping attack.

A Key Management scheme was proposed for heterogeneous CWSN in (Yuan et al., 2020). The authors adopt the Pairing-Free Identity-Based Signature (PF-IBS) (Sharma et al., 2017) and the ECC encryption algorithm (Almajed and Almogren, 2019) to ensure the security of the key establishment process between CH and CMs, as well as between CH and BS. The proposed scheme can resist various cyber-attacks, and it further provides several security requirements such as authentication, data confidentiality, and data integrity. However, it suffers from a lack of mutual authentication between sensor nodes. Furthermore, the authors use the BS as a reference to generate and send session keys to sensor nodes. This leads to generating high traffic, causing network congestion. The proposed scheme is inefficient in terms of storage cost. Additionally, all exchanged messages are encrypted using asymmetric cryptography. This makes more computation cost.

3. Preliminaries and system model

In next subsections, we briefly introduce an overview of ECC, some computational problems and the IBC. We further present our network model, security properties of key agreement, and cyber-attacks on CWSN.

*3.1. **Elliptic Curve Cryptography***

ECC is a public-key cryptography algorithm based on elliptic curves over a finite field. It has attracted much attention as a means of security for resource-constrained environments. This cryptosystem provides the same level of protection as the RSA cryptosystem but with shorter key sizes. Thus, ECC involves less computational overhead (Du et al., 2020). In the following, the basics of ECC are given.

We consider $\mathbb{F}_q$ a finite field of order q, where q is a large prime number. $E/\mathbb{F}_q$ represents an elliptic curve E over $\mathbb{F}_q$, which is given by the simplified Weierstrass equation (Patil and Szygenda, 2012): $y^2 = x^3 + ax + b$, where a, $b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$.

Given a point P on $E/\mathbb{F}_q$ and a scalar k, the point multiplication (also known as the scalar multiplication), kP, is calculated by adding P to itself k times. The result of kP is a different point on the same elliptic curve.

*3.2. **Computational problems***

Elliptic Curve Discrete Logarithm Problem (ECDLP): Given two points $P, Q \in \mathbb{G}$, it is difficult to find $k \in \mathbb{Z}_q^*$ where $Q = kP$ (Hankerson et al., 2004).

Computational Diffie Hellman problem (CDHP): Given the points $P, aP, bP \in \mathbb{G}$ where $a, b \in \mathbb{Z}_q^*$ are unknown, the computation of abP is hard in $\mathbb{G}$ (Hankerson et al., 2004).

3.3. *Identity-Based Cryptography*

IBC is an extension of public-key cryptography introduced in (Shamir, 1984). In such cryptosystem, an entity's public key is derived from its identity. A third party, known as a Private Key Generator (PKG), is responsible for issuing the corresponding private key. The generation of a private key is based on an entity's identity and a master secret key. The latter is known only to PKG. After the generation process, PKG sends a private key to an entity through a secure channel. Figure 1 illustrates the concept of IBC. Several asymmetric schemes are available in the IBC, including Identity-Based Signature (IBS), Identity-Based Encryption (IBE), and Identity-Based Key Agreement (IBKA). The first IBS scheme is proposed by Shamir (Shamir, 1984), which is based on the RSA cryptosystem. While in (Joux, 2000), Joux proposed IBKA scheme allowing the establishment of a session secret key between three entities using a pairing concept. After this, Boneh and Franklin proposed the first IBE scheme in (Boneh and Franklin, 2001) using a pairing over elliptic curves.

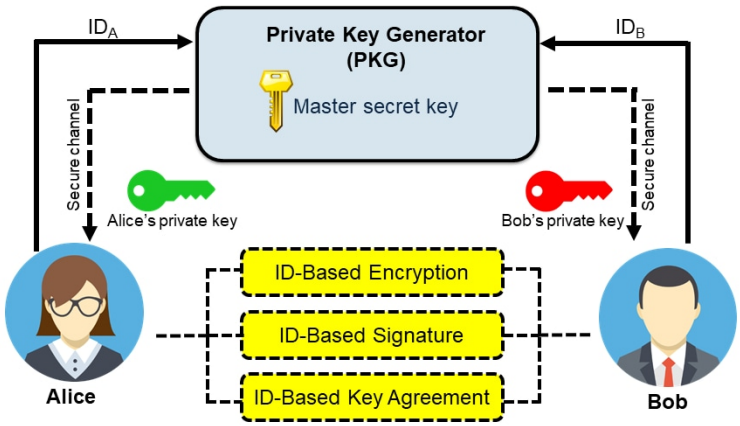


Figure 1: Identity-based cryptography concept.

In the literature, IBC is suitable for devices with limited resources, such as sensor nodes. This is due to the fact that the IBC provides easy management of public keys compared to PKI-based cryptosystems, and there is no need to generate and maintain public key certificates. Consequently, IBC requires low computational and communication overhead. However, IBC is vulnerable to the key escrow problem where the security of the whole network depends on the PKG. Therefore, the PKG must be an unconditionally trusted entity. However, it may be challenging to provide such a feature in many scenarios (Oliveira et al., 2011). Fortunately, in the CWSN scenario, the BS who plays the role of the network deployer is trustworthy. It is considered a laptop-class device with physical protection as assumed in the subsection 3.4. Thus, the BS can act as a PKG. Moreover, to solve the problem of key escrow, all sensor nodes' long-term private keys are issued by BS.

According to IBC requirements, long-term private keys must be delivered to the sensor nodes through secure channels. However, in the CWSN scenario, such channels do not exist between the BS and sensor nodes. Therefore, this issue is eliminated by preloading each sensor node with the corresponding long-term private key before deployment.

*3.4. **Network model***

In our work, the network model is composed of a single BS and hundreds of sensor nodes (Up to 300 nodes). Here, sensor nodes are resource-constrained and homogeneous in their capabilities and functionalities. The BS is assumed to be reliable and trustworthy and is responsible for configuring the nodes before deploying the network. Additionally, all sensor devices are distributed at random. Upon deployment, the BS is static, as are all the sensor nodes. To achieve energy-

efficient, a whole network is organized into clusters using a dynamic clustering method presented in (Jerbi et al., 2016). The cluster number is equal to 10% of the number of distributed nodes. In each cluster, there is a single CH and 9 CMs. The CHs aggregate data sensed from their CMs and transmit the result to the BS. The latter serves as a gateway for transmitting data to the end-user over a traditional wired or wireless network. The network model is given in Figure 2.

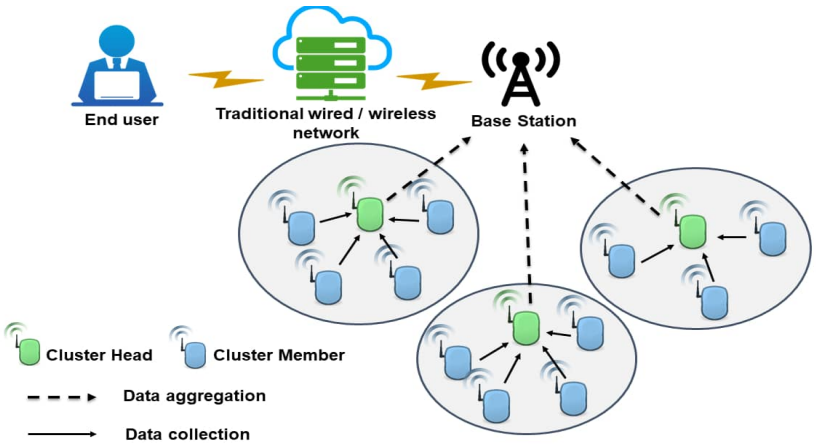


Figure 2: Network model.

*3.5. **Security properties of key agreement schemes***

According to Blake-Wilson et al. (Blake-Wilson et al., 1997), key agreement schemes should achieve the following security properties.

- **Known Session Key:** If an adversary has knowledge of some previous session keys, it cannot compromise other session keys.
- **Unknown Key Share:** A node ID_i cannot be forced to share a key with a node ID_j when ID_i believes that the key is shared with another node ID_k $\neq ID_j$.

- **Perfect Forward Secrecy:** if the long-term private key of one or more sensor nodes are compromised, an adversary will not be able to compromise previous established session secret keys.
- **Key Compromise Impersonation:** When an adversary compromises long-term private keys for node ID_i, he/she can impersonate ID_i to other nodes, but cannot impersonate other nodes to ID_i.
- **No Key Control** A session key shouldn't be a preselected by either of participating nodes.

3.6. Cyber-attacks on CWSN

For good protection in CWSNs, following cyber-attacks need to be resisted by our scheme.

- **Eavesdropping Attack:** In this cyber-attack, the adversary is limited to listen to traffic being exchanged between nodes for the purpose of obtaining data.
- **Brute force attack:** To decrypt the exchanged messages in the data transmission, an adversary tries to uncover the correct secret keys of nodes by testing many potential keys.
- **False data injection attack:** A malicious node sends random false data to targeted CH in order to falsify the result of aggregation, therefore, the CH accepts the data sent by malicious node and aggregates them. Thus, the final result is necessarily wrong.
- **Selective forwarding attack:** If a malicious node becomes CH, it selectively forwards some messages coming from neighboring nodes, and drops the others. The choice of messages is based on certain criteria (e.g. content of the messages, identity of the source node) or in a random manner.

- **MITM attack:** During this cyber-attack, an adversary can send forged messages to legitimate CH and CM nodes to control much of the data circulating between them.
- **Replay attack:** The adversary attempts to retransmit previous messages exchanged between CM and CH or between CH and BS to pretend that the legitimate node sends the message again.
- **Sybil attack:** In this cyber-attack, a malicious node impersonates the identities of targeted legitimate nodes for the purpose of degrading the effectiveness of several features such as data distribution.
- **HELLO flood attack:** The adversary with a high-powered antenna sends a flood of HELLO message to sensor nodes. The remote node receiving this message believes that the adversary as a neighbor and it is within the range of communication. Hence it tries to send its messages directly to the adversary leading to failure of messages transmission, and to disrupt the network operation by prevent other messages to be exchanged.

4. Proposed scheme

In this section, we illustrate the proposed scheme, which is divided into two main phases, namely System initialization phase and Key agreement phase. Table 1 lists the notations used in the proposed scheme. Below are the descriptions of each phase.

4.1. System initialization phase

During this phase, two sub-phases are presented, the setup phase and the key extraction phase. Both are performed by the BS prior to network deployment.

Table 1: List of notations

Notation	Description
BS, CH, CM	Base Station, Cluster Head and Cluster Member
ID_i	Identity of a node
P	A generator of group $\mathbb{G}$
q	A prime order of group $\mathbb{G}$
x	Master secret key
P_{pub}	Master public key
d	ID-based long-term private key
W	ID-based long-term public key
y, T	Ephemeral secret and public keys
sk	Secret session key

Setup phase. Given a security parameter k, the BS determines the tuple $\{\mathbb{F}_q, E/\mathbb{F}_q, \mathbb{G}, P\}$ where $\mathbb{G}$ denotes a group with prime order q and the point P is the generator of $\mathbb{G}$. The BS picks a random number $x \in \mathbb{Z}_q^*$ as the master secret key, it thereafter computes the master public key $P_{pub} = xP$. Then, three hash functions are chosen: $H_0 : \{0,1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$, $H_1 : \{0,1\}^* \times \mathbb{G}^2 \rightarrow \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}^3 \rightarrow \{0,1\}^k$. finally, the system parameters $\{\mathbb{F}_q, E/\mathbb{F}_q, \mathbb{G}, P, P_{pub}, H_0, H_1, H_2\}$ are published while x is kept only in the BS.

Key extraction phase. This phase takes as input a master secret key, a node's identity ID_i and system parameters. The output is a long-term private/public key pair (d_i,W_i). The details are described as follows:

• The BS picks a random number r_i, then it computes $R_i = r_i.P$.

• The BS computes a long-term private key $d_i = (r_i + H_0(ID_i||R_i)x) \bmod q$. Then, it computes a long-term public key $W_i = R_i + H_0(ID_i||R_i).P_{pub}$. Next, each sensor node i is preloaded with R_i, d_i and W_i. Here, we mention that the nodes can validate their private/public key by checking whether the equation $d_i.P = R_i + H_0(ID_i||R_i).P_{pub}$ is correct. We have:

$$\begin{aligned} d_i.P &= (r_i + H_0(ID_i||R_i)x).P \\ &= r_i.P + H_0(ID_i||R_i)x.P \\ &= R_i + H_0(ID_i||R_i).P_{pub}. \end{aligned}$$

*4.2. **Mutual authentication and key agreement phase***

As shown in Figure 3, the authentication and key agreement between CH (denoted as A) and CM/BS (denoted as B) consists of four steps. We assume that nodes A and B serve as an initiator and a responder, respectively.

Step 1. Node A picks a random number $y_A \in \mathbb{Z}_q^*$ as its ephemeral secret key and

computes the ephemeral public key $T_A = (y_A + d_A)^2.P$. Thereafter, it sends the message $M_1 = (ID_A, T_A, W_A)$ to node B through an insecure channel.

Step 2. Upon receiving the message M_1, node B picks a random number $y_B \in \mathbb{Z}_q^*$ as its ephemeral secret key and computes both $T_B = (y_B + d_B)^2.P$ and the value $\sigma_B = H_1(ID_B||T_B||d_B.W_A)$. Then, node B sends the message $M_2 = (ID_B, T_B, W_B, \sigma_B)$ to node A through an insecure channel.

Step 3. Node A computes $\widehat{\sigma}_B = H_1(ID_B||T_B||d_A.W_B)$ locally. Then, it verifies the authenticity of node B by checking whether the condition $\widehat{\sigma}_B \stackrel{?}{=} \sigma_B$. If it holds, A authenticates B and then establishes the session key $sk = H_2(ID_A||ID_B||T_A||T_B||K_A)$, where $K_A = (y_A + d_A)^2.T_B$. Furthermore, node A computes $\sigma_A = H_1(ID_A||T_A||d_A.W_B)$ and then sends σ_A to node B.

Step 4. Similarly, node B computes $\widehat{\sigma}_A = H_1(ID_A||T_A||d_B.W_A)$ and compares with received σ_A. If $\widehat{\sigma}_A = \sigma_A$, node B authenticates A and establishes the session key as $sk = H_2(ID_A||ID_B||T_A||T_B||K_B)$, where $K_B = (y_b + d_b)^2.T_A$.

Both A and B establish the same session key sk = $H_2(ID_A||ID_B||T_A||T_B||K)$, where $K = K_A = K_B$. For correctness we have:

$$\begin{aligned} K_A &= (y_A + d_A)^2.T_B \\ &= (y_A + d_A)^2.T_B \\ &= (y_A + d_A)(y_B + d_B)P \\ &= (y_B + d_B)(y_A + d_A)P \\ &= (y_B + d_B)^2.T_A \\ &= K_B \end{aligned}$$

5. Security analysis of the proposed scheme

This section evaluates the proposed scheme using both formal and informal security analyses.

*5.1. **Formal security verification using AVISPA***

In this section, we provide a formal analysis of our proposed scheme by using software called Automated Validation of Internet Security Protocols and Applications (AVISPA) (Armando et al., 2005; Vigano, 2006). The purpose of such software is first, to analyze automatically whether our scheme is safe and resistant to passive and active cyber-attacks, including eavesdropping, MITM and replay attacks. Second, AVISPA verifies whether our scheme achieves security goals, such as confidentiality and mutual authentication. AVISPA tool provides a formal language called HLPSL (High-Level Protocol Specification Language)

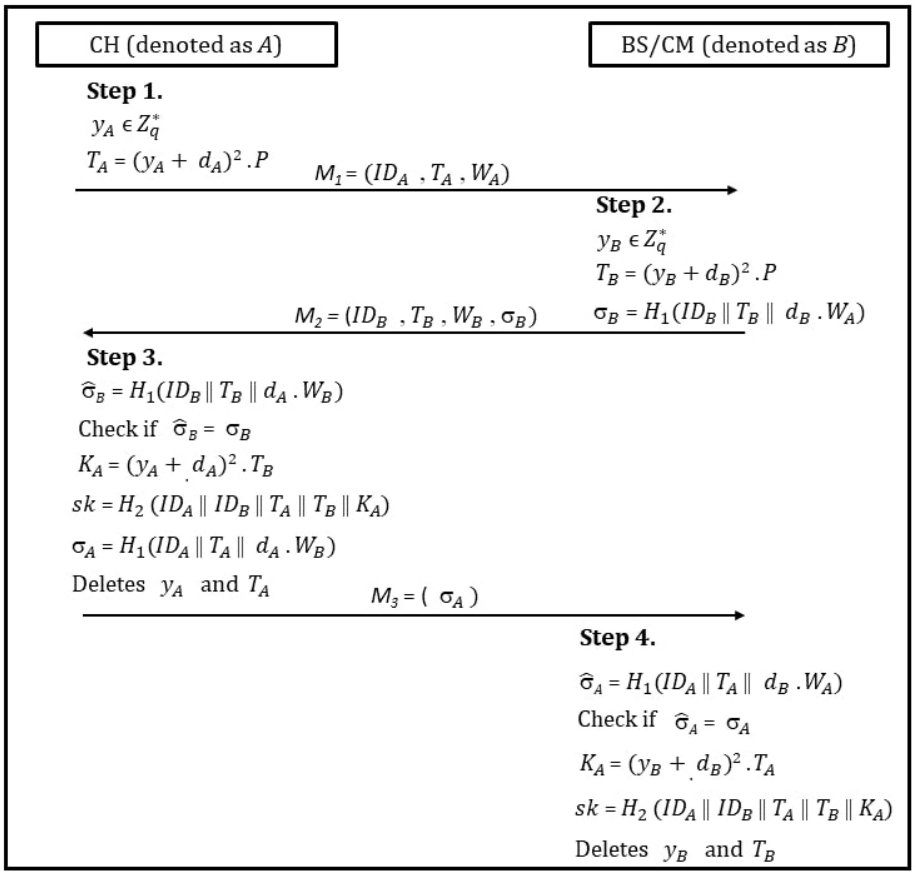


Figure 3: Mutual authentication and key agreement phase in the proposed scheme.

to specify cryptographic protocols. In addition, AVISPA tool has four back-ends, including OFMC (On-the-fly Model-Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker), and TA4SP (Tree Automata-based Protocol Analyzer). These back-ends are used to analyze and verify the security properties such as authentication and secrecy of keys. The HLPSL is role-based, which defines two main types of roles: (1) *the basic roles*, illustrate the actions of the entities participating; (2) *the composed roles*, describe the different scenarios in which basic roles are involved. Furthermore, HLPSL

supports the Dolev-Yao threat model (Dolev and Yao, 1983), which allows an attacker to intercept, modify, and replay messages transmitted over a public network channel. The specification code of HLPSL is automatically translated in Intermediate Format (IF) using the HLPSL2IF translator. Then, the AVISPA sends the IF specifications to the back-ends, analyzing whether the scheme is safe or not from intruders.

1. **Specification of our scheme:** We have implemented IBAKAS in HLPSL for the authentication and key agreement phases. Figure 4 illustrates the detailed specifications of the basic roles for CH (denoted by node_A) and CM/BS (denoted by node_B). The composed roles, which consist of session, environment, and goals, are shown in Figure 5.

```
role node_A (A, B: agent,
             Add,Mul,H : hash_func,
             Snd, Rcv: channel(dy))

played_by A
def=
   local State  : nat,
         Ya, Yb, P, Da, Db: text,
         TA,TB, WA, WB, Sigma_a, Sigma_b, KA,KB, SK: text

init  State := 0
transition
1. State   = 0 /\ Rcv(start) =|> State' := 1 /\ Ya' := new()
/\ TA' := Mul(Add(Ya'.Da).P) /\ WA':=Mul(Da.P)
/\ Snd(A.TA'.WA')
/\ secret(Ya',sec_ya,{A,B})
2. State   = 1 /\ Rcv(B.TB'.WB'.Sigma_b') =|>
State' := 2 /\ Sigma_a' := H(A.TA.Mul(Da.WB'))
/\ KA' := Mul(Add(Da.Ya).TB')
/\ SK' := H(A.B.TA.TB.KA')
/\ Snd(Sigma_a')
/\ witness(A,B,auth_node_a,Sigma_a')
/\ request(A,B,auth_node_b,Sigma_b')

end role
```

```
role node_B (B, A: agent,
             Add,Mul,H : hash_func,
             Snd, Rcv: channel(dy))

played_by B
def=
   local State  : nat,
         Ya, Yb, P, Da, Db : text,
         TA, TB, WA,WB, Sigma_a,  Sigma_b,KA, KB, SK: text

init  State := 0
transition
1. State   = 0 /\ Rcv(A.TA'.WA') =|>  State' := 1
/\ Yb' := new()
/\ TB' := Mul(Add(Yb'.Db).P) /\ WB':=Mul(Db.P)
/\ Sigma_b' := H(B.TB'.Mul(Db.WA'))
/\ Snd(B.TB'.WB'.Sigma_b')
/\ secret(Yb',sec_yb,{A,B})
/\ witness(B,A,auth_node_b,Sigma_b')
2. State   = 1 /\ Rcv(Sigma_a') =|>
State' := 2 /\ KA' := Mul(Add(Db.Yb).TA)
/\ SK' := H(A.B.TA.TB.KA)
/\ request(B,A,auth_node_a,Sigma_a')

end role
```

Figure 4: The basic roles in HLPSL.

2. **Verification results:** Figure 6 presents the verification results of IBAKAS

```
role session(A, B: agent,
             Add, Mul, H: hash_func)
def=
local SA, SB, RA, RB: channel (dy)
 composition
  node_A(A, B, Add, Mul, H, SA, RA)
 /\  node_B(B, A, Add, Mul, H, SB, RB)
end role
role environment()
def=
const a, b: agent,
     add,mul,h: hash_func,
      sec_ya,sec_yb,auth_node_a,auth_node_b:
protocol_id
```

```
intruder_knowledge = {a, b, mul, add, h}
composition
session(a,b,add,mul,h)
/\ session(i,b,add,mul,h)
/\ session(a,i,add,mul,h)
end role
goal
 secrecy_of sec_ya, sec_yb
 authentication_on auth_node_a
 authentication_on auth_node_b
end goal
environment()
```

Figure 5: The role specification in HLPSL, for session, environment and goal.

under OFMC and CL-AtSe back-ends. These results indicate that security goals such as confidentiality and mutual authentication are satisfied. Thus, IBAKAS is safe and resistant to cyber-attacks such as MITM and replay attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/ibakas.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 16 nodes
  depth: 4 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/ibakas.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analyses   : 0 states
  Reachable  : 0 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Figure 6: Verification results of our scheme in OFMC and CL-AtSe back-ends.

5.2. *Informal security analysis*

In this subsection, we describe how the informal security properties of the IBAKAS scheme are achieved. Furthermore, we analyze the effectiveness of the IBAKAS scheme against CWSN cyber-attacks.

- **Known Session Key:** In this proposal, the session key between CH and CM is computationally dependent on ephemeral secrets (y_{CM}, y_{CH}) and long-term private keys (d_{CM}, d_{CH}). Each session has different ephemeral secrets y_{CM} and y_{CH}. Due to difficulties of ECDLP, an adversary failed to extract (y_{CM}, y_{CH}) from (T_{CM}, T_{CH}), as well as (d_{CM}, d_{CH}) from (W_{CM}, W_{CH}). Thus, the compromised session key does not allow an adversary to reveal other session keys. Therefore, our scheme could provide the Known session key property.
- **Unknown Key Share:** The proposed IBAKAS satisfies this propriety since both CH and CM compute the session key based on T_{CH} and T_{CM} validated by their respective signatures σ_{CH} and σ_{CM}. Further, due to ECDLP, the private keys of nodes cannot be derived from their public keys.
- **Perfect Forward Secrecy:** Suppose that an adversary has compromised long-term private keys d_{CM} and d_{CH}. However, it cannot reveal previous established session keys, since ephemeral secrets y_{CM} and y_{CH} are unknown and renewed at every session. Moreover, an adversary is unable to extract y_{CM} and y_{CH} from T_{CM} and T_{CH}, respectively, due to difficulties of ECDLP. Therefore, the proposed scheme provides the perfect forward secrecy.
- **Key Compromise Impersonation:** Suppose that the long-term private key d_{CM} is disclosed to a malicious node (denoted as $\mathcal{E}$) who tries to imperson-

ate CH to CM to obtain the session key sk_{CH}^{CM}. However, node $\mathcal{E}$ cannot compute $\sigma_{CH} = (ID_{CH}||T_{CH}||d_{CH}.W_{CM})$ without knowing the long-term private key d_{CH}. Therefore, $\mathcal{E}$ cannot be authenticated as legitimate CH, and CM rejects the session key establishment. Consequently, our scheme provides the key compromise impersonation resilience.

- **No Key Control:** Since both CH and CM choose random ephemeral secrets y_{CH} and y_{CM}, respectively, neither entity can influence the random selection process. Thus, our scheme ensure no key control propriety.
- **MITM attack:** According to our scheme, T_{CH} = $(y_{CH} + d_{CH})^2.P$ and T_{CM} = $(y_{CM} + d_{CM})^2.P$ are exchanged with the σ_{CH} and σ_{CM} signatures. Once T_{CH} and T_{CM} are validated, CH and CM nodes compute the shared session key sk_{CH}^{CM} using the long-term private keys, d_{CH} and d_{CM}, and the ephemeral secret keys (random numbers), y_{CH} and y_{CM}. The MITM attack may occur in the proposed scheme if a malicious node extracts d_{CH} and d_{CM} from public values (W_{CH}, W_{CM}) = ($d_{CH}.P$, $d_{CM}.P$), and then computes $d_{CH}.d_{CM}.P$. Due to the difficulties of CDHP, this computation is not possible. Thus, our scheme prevents MITM attack.
- **Replay attack:** As described in our scheme, messages M_1 and M_1 contain T_{CH} and T_{CM}, respectively. In addition, the message M_3 contains σ_{CH}, which is calculated based on T_{CH}. Due to the dynamic nature of T_{CH} and T_{CM}, which are regularly updated, our scheme can reject all replayed messages by checking T_{CH} and T_{CM}. Thus, the replay attack is prevented.

After a successful session key establishment between CM and CH, IBAKAS will resist following cyber-attacks.

- **Eavesdropping and brute force attacks:** Once a session key has been es-

tablished between CH and CM or between CH and BS, the key is then used to encrypt data sent between CH and CM or between CH and the BS, which ensures data confidentiality and protects sensitive data from eavesdrop. Furthermore, it is difficult for an adversary to discover the session key since it is dynamic and is renewed at every session. Consequently, the proposed IBAKAS can resist both eavesdropping and brute force attacks.

- **False data injection attack, Selective forwarding, Sybil and Hello flood attacks:** The best way of preventing such cyber-attacks is by ensuring the authenticity of messages between CH and CM or between CH and BS. To this end, and based on the session key sk, a sending node can compute a Message Authentication Code $MAC_{sk}(message)$ as digital signature. Using the same session key a receiving node can verify $MAC_{sk}(message)$.

Comparing the security features of the proposed IBAKAS and existing authentication and key agreement schemes (Mehmood et al., 2017; Harbi et al., 2019; Saeed et al., 2019; Hamouid et al., 2020; Kumar et al., 2021) is provided in Table 2

6. Performance evaluation

In our performance study, we have implemented the IBAKAS scheme in ContikiOS (Solapure et al., 2020), a lightweight operating system designed for WSN and IoT devices. As well, IBAKAS and existing relevant schemes (Mehmood et al., 2017; Harbi et al., 2019; Saeed et al., 2019; Hamouid et al., 2020; Kumar et al., 2021) are tested using the Cooja network simulator (Solapure et al., 2020). The performance was measured on the WiSMote sensor device (Dunkels, 2015;

Table 2: Comparison of security features of our scheme and existing ID-based schemes

Schemes	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
ICMDS (2017)	No	No	No	No	No	No	No	No	Yes	Yes	Yes	No	No	No
MAKA (2019)	Yes	No	No	No	No	No	Yes	Yes	Yes	No	Yes	No	Yes	Yes
AKAIoTs (2019)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
LSTR (2020)	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Kumar et al. (2021)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No
Proposed scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

F1: Mutual authentication, F2: Known Session Key, F3: Unknown Key Share, F4: Perfect Forward Secrecy, F5: Key Compromise Impersonation, F6: No Key Control, F7: MITM attack resistance, F8: Replay attack resistance, F9: Eavesdropping attack resistance, F10: brute force attack resistance, F11: False data injection resistance, F12: Selective forwarding attack resistance, F13: Sybil attack resistance, F14: Hello flood attack resistance.

Texas Instruments, 2007, 2021), which is equipped with MSP430F5437A MCU, 256 KB of flash memory, 16 KB of SRAM, and CC2520 radio chip. For operations on an elliptic curve, we used a lightweight asymmetric cryptographic library suitable for WSN and IoT devices, known as RELIC toolkit (Aranha et al., 2020), with 160-bit ECC to achieve the 80-bit level of security. Due to the SHA-1 hash function is broken, we applied the SHA256 hash function truncated to 20 bytes length.

*6.1. **Evaluation metrics and results***

Four main metrics have been used to evaluate the performance of IBAKAS scheme , including the computation cost, the communication cost, the energy consumption and the key storage cost. The results obtained are also compared with existing authentication and key agreement schemes: ICMDS (Mehmood et al., 2017), MAKA (Harbi et al., 2019), AKAIoTs (Saeed et al., 2019), LSTR (Hamouid et al., 2020), and Kumar et al. (Kumar et al., 2021). It is clear that IBAKAS, AKAIoTs, and Kumar et al. are ECC-based schemes, while the others utilize a

pairing technique.

6.1.1. ***Computation cost***

Given that the BS is a powerful device, in this paper we focus only on the computational costs required by constrained sensor nodes. The computational cost of IBAKAS is evaluated and compared with ICMDS, MAKA, AKAIoTs, LSTR, and Kumar et al. schemes, based on the number of cryptographic operations computed. Table 3 presents the obtained results.

According to our experimental results using the WiSMote sensor device, the computation times of required cryptographic operations in IBAKAS and existing relevant schemes are listed in Table 4. As seen in this Table, the MTP function and pairing-related operations are computationally expensive.

Figure 7(a) illustrates the computation time (in seconds) required by a sensor node. The proposed IBAKAS takes 4.235 seconds, this result is considered the lowest computational time compared to existing authentication and key agreement schemes. The reason is that in IBAKAS, a sensor node (CH or CM) executes neither pairing operations nor MTP function. Moreover, IBAKAS requires a small number of cryptographic operations. As shown in Table 3, each sensor node executes only 4 point multiplications and 3 one-way hash functions to achieve an authentication and establish a single session key.

Considering a network containing m CHs and n CMs, the total computational cost associated with m CHs is $m \times 10(4EM + 3H)$ and the total computational cost associated with n CMs is $n \times (4EM + 3H)$. Thus, the total computational cost for our scheme is $(n + 10m)(4EM + 3H)$.

Table 5 shows the total computation time for IBAKAS and the cluster-based schemes, including ICMDS, MAKA, and LSTR. In this comparison, the number

Table 3: Comparison of computation and communication costs on sensor nodes to establish a single session key.

			Sensor node		
Schemes	**Cluster-based**	**Pairing**	**Computation cost**	**Communication cost / Transmit**	**Communication cost / Receive**
ICMDS (2017)	Y	Y	$1BP + 1HG + 1PM + 1H$	$-$	$2\|\mathbb{Z}_q^*\| + (m+2)\|\mathbb{G}_1\|$
MAKA (2019)	Y	Y	$1BP + 1HG + 4PM$	$\|\mathbb{G}_2\| + 2\|\mathbb{G}_1\| + \|nonce\|$	$3\|\mathbb{G}_1\| + \|nonce\|$
AKAIoTs (2019)	N	N	$6EM + 1EA + 4H$	$\|ID\| + 2\|\mathbb{G}\| + 2\|\mathbb{Z}_q^*\| + \|nonce\|$	$\|ID\| + 2\|\mathbb{G}\| + 2\|\mathbb{Z}_q^*\| + \|nonce\|$
LSTR (2020)	Y	Y	$2BP + 1HG + 2PM + 1H$	$3\|ID\| + \|\mathbb{G}_1\| + \|nonce\|$	$3\|ID\| + \|\mathbb{G}_1\| + \|nonce\|$
Kumar et al. (2021)	N	N	$5EM + 2EA + 4H$	$\|ID\| + 2\|\mathbb{G}\| + \|\mathbb{Z}_q^*\|$	$\|ID\| + 2\|\mathbb{G}\| + \|\mathbb{Z}_q^*\|$
Proposed scheme	Y	N	$4EM + 3H$	$\|ID\| + 2\|\mathbb{G}\| + \|\mathbb{Z}_q^*\|$	$\|ID\| + 2\|\mathbb{G}\| + \|\mathbb{Z}_q^*\|$

Table 4: Computation time of cryptographic operations on WiSMote sensor device

Operation	**Notation**	**Computation time (seconds)**
Bilinear pairing	BP	8.142
Pairing-based point multiplication	PM	2.974
MTP function	HG	1.582
Elliptic curve point multiplication	EM	1.049
Elliptic curve point addition	EA	0.007
Hash function	H	0.013

of clusters varies from 2 to 10. Each cluster contains 9 CMs. Based on Table 5, we demonstrate that the IBAKAS scheme is lightweight and offers better computation efficiency compared to ICMDS, MAKA and LSTR schemes.

Table 5: Total computational time comparison (Unit: seconds)

Network size	CH	CM	ICMDS	MAKA	LSTR	IBAKAS
20	2	18	483.018	821.560	905.426	160.93
30	3	27	724,527	1232,340	1358,139	241.395
40	4	36	966,036	1643,120	1810,852	321,860
50	5	45	1207,545	2053,900	2263,565	402,325
60	6	54	1449,054	2464,680	2716,278	482,790
70	7	63	1690,563	2875,460	3168,991	563,255
80	8	72	1932,072	3286,240	3621,704	643,720
90	9	81	2173,581	3697,020	4074,417	724,185
100	10	90	2415,090	4107,800	4527,130	804,650

6.1.2. ***Communication cost***

We assume that $|ID|$ and $|nonce|$ are each 2 bytes in size. In the schemes ICMDS (Mehmood et al., 2017), MAKA (Harbi et al., 2019) and LSTR (Hamouid et al., 2020), we use the pairing-friendly curve BN-P158 over a 158-bit primary field. According to this curve, the size of an element in the groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ is respectively equal to 40 bytes, 80 bytes, and 240 bytes. However, for better performance, the size of an element in $\mathbb{G}_1$ and $\mathbb{G}_2$ should be compressed to 21 bytes and 41 bytes, respectively. During the compression process, only x-coordinate and a single bit of y-coordinate are transmitted, rather than both. The receiver

can easily determine the y-coordinate by computing the square root.(Shim, 2014). The size of messages transmitted and received by the schemes are as follows:

- The ICMDS scheme requires a sensor node to receive $(P_{pub}, R, C, C_0, C_1, ..., C_m)$, where $P_{pub} \in \mathbb{G}_1$, $\{R, C\} \in \mathbb{Z}_q^*$ and $\{C_0, ..., C_m\} \in \mathbb{G}_1$. Assuming the number of CHs is ($m = 10$), the size of the received message is $2|\mathbb{Z}_q^*| + 12|\mathbb{G}_1| = 2 \times 20 + 12 \times 21 = 292$ bytes. Note that the sensor node does not transmit any message to the BS during the session key agreement. Thus, there is no communication cost for transmitting messages.
- The MAKA scheme requires a sensor node to transmit $(\textit{PU}, \textit{EM})$, where $\textit{EM} \in \{|\mathbb{G}_1| + |\mathbb{G}_1| + |nonce|\}$ and $\textit{PU} \in |\mathbb{G}_2|$. Additionally, it requires a sensor to receive $(P, \textit{EM})$, where $\textit{EM} \in \{|\mathbb{G}_1| + |\mathbb{G}_1| + |nonce|\}$ and $P \in |\mathbb{G}_1|$. Therefore, the size of a transmitted message is $|\mathbb{G}_2| + 2|\mathbb{G}_1| + |nonce| = 41 + 2 \times 21 + 2 = 85$ bytes. The size of a received message is $3|\mathbb{G}_1| + |nonce| = 3 \times 21 + 2 = 65$ bytes.
- The LSTR scheme requires a sensor node to transmit $\textit{PDU} \in \{|\textit{ID}| + |\textit{ID}| + |\textit{ID}| + |\mathbb{G}_1| + |nonce|\}$. In addition, LSTR requires a sensor node to receive the same size message as it transmitted. Therefore, the size of a transmitted message is $3|\textit{ID}| + |\mathbb{G}_1| + |nonce| = 3 \times 2 + 21 + 2 = 29$ bytes. The size of a received message is 29 bytes.

According to AKAIoTs (Saeed et al., 2019), Kumar et al. (Kumar et al., 2021), and our scheme, we use the curve SECG-P160 over a 160-bit primary field. In this curve, the size of an element in the group $\mathbb{G}$ is 40 bytes and can be compressed to 21 bytes. The size of messages transmitted and received by the schemes are as follows:

- The AKAIoTs scheme requires a sensor node to transmit $(ID, Y, \sigma, nonce)$, where $\sigma \in \{|\mathbb{Z}_q^*| + |\mathbb{Z}_q^*| + |\mathbb{G}|\}$ and $Y \in \mathbb{G}$. In addition, AKAIoTs requires a sensor node to receive the same size message as it transmitted. Therefore, the size of a transmitted message is $|ID| + 2|\mathbb{G}| + 2|\mathbb{Z}_q^*| + |nonce| = 2 + 2 \times 21 + 2 \times 20 + 2 = 86$ bytes. The size of a received message is 86 bytes.

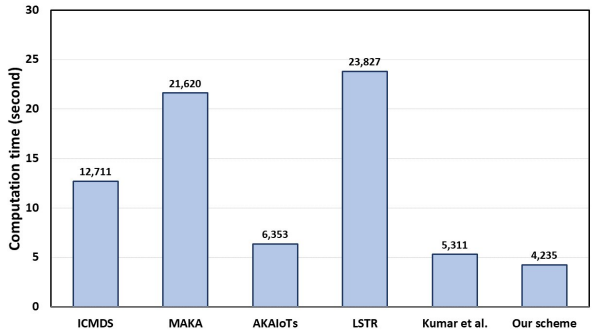
- The Kumar et al. scheme requires a sensor node to transmit (ID, T, R, S), where $\{T, R\} \in \mathbb{G}$ and $S \in \mathbb{Z}_q^*$. In addition, it requires a sensor node to receive the same size message as it transmitted. Therefore, the size of a transmitted message is $|ID| + 2|\mathbb{G}| + |\mathbb{Z}_q^*| = 2 + 2 \times 21 + \times 20 = 64$ bytes. The size of a received message is 64 bytes.

- The proposed scheme requires a sensor node to transmit (ID, T, W, σ), where $\{T, W\} \in \mathbb{G}$ and $\sigma \in \mathbb{Z}_q^*$. In addition, the proposal requires a sensor node to receive the same size message as it transmitted. Thus, the size of a transmitted message is $|ID| + 2|\mathbb{G}| + |\mathbb{Z}_q^*| = 2 + 2 \times 21 + \times 20 = 64$ bytes. The size of a received message is 64 bytes.

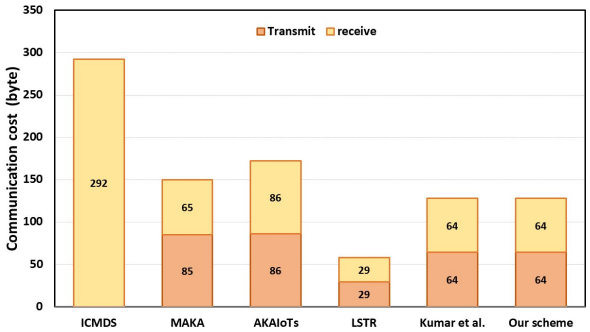
As shown in Figure 7(b), the obtained results demonstrate that the proposed IBAKAS introduces a low communication cost than ICMDS, MAKA, and AKAIoTs. In contrast, LSTR appears to offer better communication efficiency than our scheme. However, as shown in Table 2, the LSTR scheme suffers from a lack of security features, such as mutual authentication, Unknown key share, and key-compromise impersonation resilience.

6.1.3. ***Energy consumption***

To evaluate the energy consumption associated with computation and communication, we use the equations (Shim, 2014) $W_{comp} = V \times I_c \times t$ and $W_{tx/rx} =$



(a) Computation cost.



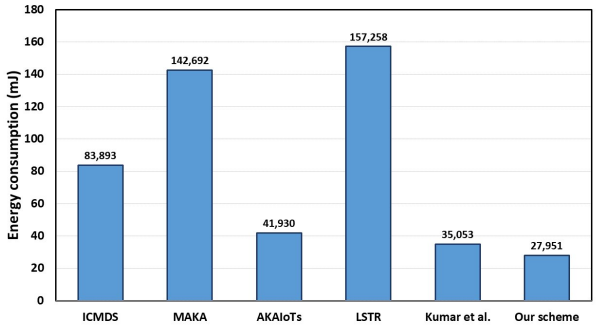
(b) Communication cost.

Figure 7: Computation and communication costs required by a sensor node to establish one session key.

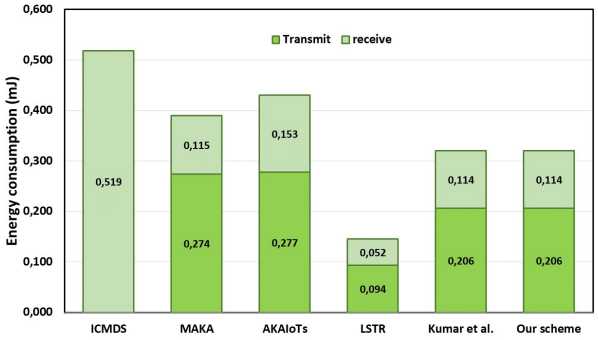
$V \times I_{tx/rx} \times U \times \frac{8}{dr}$, respectively. Where $W_{comp/tx/rx}$ represents the energy in millijoules (mJ), V is the voltage, I_c, denotes the current draw in CPU active mode, $I_{tx/rx}$ denotes the current draw in transmitting/receiving mode, U is the size of message in byte, t is the computation time in second and dr represents the data rate. According to WiSMote sensor device, I_c, I_{tx}, I_{rx} are 2.2 mA, 33.6 mA and 18.5 mA respectively. In addition, the supply voltage is set to 3 Volts, and the data rate is equal to 250 kbps (Texas Instruments, 2007, 2021).

Figure 8 illustrates the energy consumed by a sensor node for (a) the computation process and (b) the transmission/reception of messages. From Figure 8(a), IBAKAS is energy efficient during the computation process and consumes less energy than existing relevant schemes. The main reason is that W_{comp} can be derived from computation time. Since the computation affects the energy consumption and the computational time is lower in IBAKAS, the energy consumption is also lower. From Figure 8(b), IBAKAS consumes less energy than ICMDS, MAKA, and AKAIoTs. However, it has a higher energy consumption than LSTR. This is mainly due to the correlation between the size of transmitted/received messages U and the energy consumption $W_{tx/rx}$. Thus, The larger the message size, the more energy is consumed.

Figure 9 illustrates the total estimated energy consumption according to the number of clusters. Compared to ICMDS, MAKA, and LSTR schemes, IBAKAS is energy efficient. Indeed, IBAKAS can reduce the total energy consumption by 66.68%, 80.41%, and 82.23% compared to ICMDS, MAKA, and LSTR, respectively. The main reason for this improvement is that the computation affects the energy consumption and the total computational time is considerably lower in the IBAKAS scheme, as shown in Table 5. Thus, the total energy consumption is also



(a) Energy consumption for computation.



(b) Energy consumption for communication.

Figure 8: Energy consumed by a sensor node to establish one session key.

lower.

6.1.4. ***Key storage cost***

Because sensor nodes are resource-constrained, key storage overhead is an important factor to consider. Figure 10 illustrates the amount of memory required to store long-term and ephemeral keys in a sensor node. Comparing to existing relevant schemes, IBAKAS is memory efficient and requires less memory space for storing keys. Indeed, in IBAKAS, ephemeral and long-term keys require only 76 and 100 bytes, respectively. Therefore, the total size of key storage is $76 +$

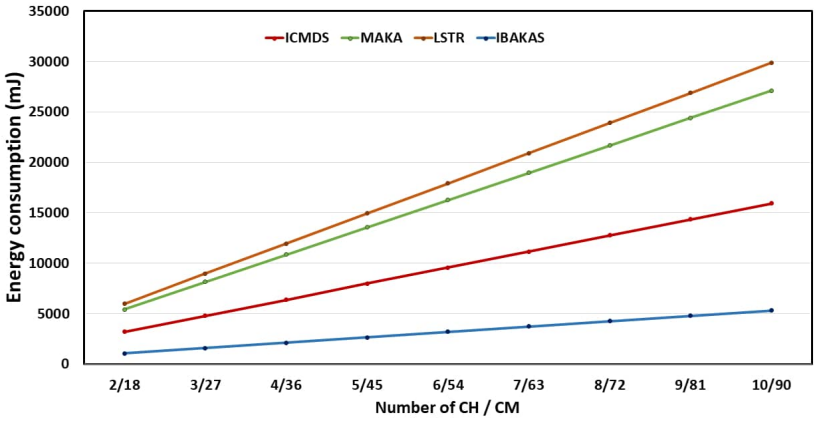


Figure 9: Total energy consumption according to a number of clusters.

$100 = 176$ bytes, which is equivalent to 1.07% (176 bytes from 16 KB) of SRAM memory. This percent is generally acceptable and satisfactory on the WiSMote sensor device.

7. Use cases

This section presents two use cases to our scheme, including military and healthcare applications, which require a high security level. Our scheme can be useful in the military field where sensor nodes are used to monitor a critical border area between two countries in order to provide information concerning the number and the nature of the enemy (persons or vehicles). Sensor nodes deployed in the target area are camouflaged to keep from being detected by the enemy. Additionally, they are equipped with thermal sensors in order to read the thermal signatures of moving objects. The gathering data from sensor nodes helps the military information analysis service to classify those moving objects and intervene in the event of cross-border infiltration.

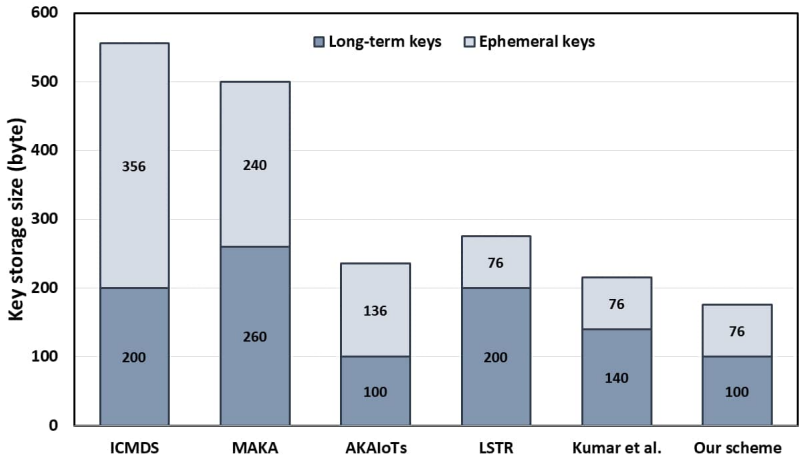


Figure 10: Key storage size required by a sensor node.

In the healthcare field, the proposed scheme can be applied inside a field hospital for monitoring patients injured on a battlefield or in case of disasters. Indeed, our scheme keeps the medical personnel continuously informed about the state of a patient to intervene and take the necessary measures in the event of deterioration in the health state of a patient. The field hospital contains several dozen patients' beds. Each one is equipped with a WiSMote device and several medical sensors placed on the patient's body, such as airflow (breathing), body temperature, pulse, blood pressure, and patient position (accelerometer). Patients' beds can dynamically be grouped into clusters. Each having one bed acts as CH, and several beds act as CMs. The CHs can perform aggregation medical data collected from their CMs and forward the result directly to BS. The latter serves as a gateway to transmit medical data to the healthcare server located in the medical staff room over a wired connection. Figure 11 illustrates the proposed architecture. .

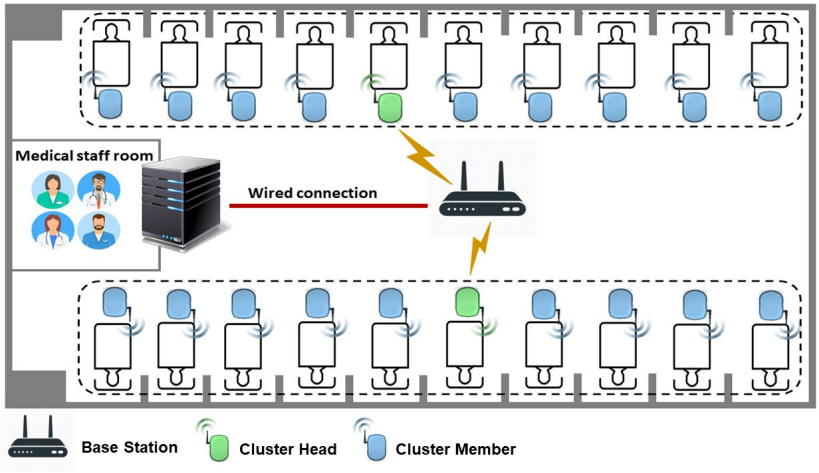


Figure 11: Patient's monitoring in the field hospital.

8. Conclusion

In this paper, we propose an Identity-Based Authentication and Key Agreement Scheme (IBAKAS) for CWSN. With our design, we aim to achieve the best possible balance between security and lightness. In the proposed scheme, IBC is used, which doesn't require public key infrastructure or complicated certificate management. Furthermore, instead of expensive bilinear pairing and MTP function, IBAKAS uses elliptic curves to achieve more computational and energy efficiency. We verified the formal security of the proposed scheme using the AVISPA tool. In addition, the detailed informal security analysis showed that our scheme achieves all the desirable security properties and prevents various cyber-attacks in CWSN. Compared with existing relevant schemes, IBAKAS decreases computation and communication costs, saves keys storage space, and prolongs the network lifetime by reducing the consumed energy on a sensor node.

As a future work, IBAKAS will be extended with more research:

1. We aim to extend our scheme to support blockchain-based IoT in healthcare applications. In this context, the extended version will be used to secure the communication between IoT devices and blockchain nodes in order to protect the privacy of sensitive data such as Electronic Health Records (EHRs).

2. We will implement our scheme on real resource-constrained sensor devices.

3. We will validate our scheme using the Random Oracle Model (ROM).

References

Almajed, H.N., Almogren, A.S., 2019. Se-enc: A secure and efficient encoding scheme using elliptic curve cryptography. IEEE Access 7, 175865–175878.

Aranha, D.F., et al., 2020. RELIC is an Efficient LIbrary for Cryptography. [Online]. Available:`https://github.com/relic-toolkit/relic`. Accessed April 2020.

Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., Drielsma, P.H., Héam, P.C., Kouchnarenko, O., Mantovani, J., et al., 2005. The avispa tool for the automated validation of internet security protocols and applications, in: International conference on computer aided verification, Springer. pp. 281–285.

Benayache, A., Bilami, A., Barkat, S., Lorenz, P., Taleb, H., 2019. Msm: A microservice middleware for smart wsn-based iot application. Journal of Network and Computer Applications 144, 138–154.

Blake-Wilson, S., Johnson, D., Menezes, A., 1997. Key agreement protocols and their security analysis, in: Darnell, M. (Ed.), Crytography and Coding, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 30–45.

Boneh, D., Franklin, M., 2001. Identity-based encryption from the weil pairing, in: Annual international cryptology conference, Springer, pp. 213–229.

Boubiche, D.E., Athmani, S., Boubiche, S., Toral-Cruz, H., 2021. Cybersecurity issues in wireless sensor networks: Current challenges and solutions. Wireless Personal Communications 117.

Castelluccia, C., Mykletun, E., Tsudik, G., 2005. Efficient aggregation of encrypted data in wireless sensor networks, in: The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 109–117.

Chen, L., Kudla, C., 2003. Identity Based Authenticated Key Agreement Protocols from Pairings, in: 16th IEEE Computer Security Foundations Workshop, 2003. Proceedings., pp. 219–233.

Dolev, D., Yao, A., 1983. On the security of public key protocols. IEEE Transactions on information theory 29, 198–208.

Du, H., Wen, Q., Zhang, S., Gao, M., 2020. A new provably secure certificateless signature scheme for internet of things. Ad Hoc Networks 100, 102074.

Dunkels, A., 2015. Platforms supported by Contiki-OS: WiSMote Platform Specifications. [Online]. Available: `https://github.com/contiki-os/contiki/`. Accessed March 2021.

699 Fanian, F., Rafsanjani, M.K., 2019. Cluster-based routing protocols in wireless
700 sensor networks: A survey based on methodology. Journal of Network and
701 Computer Applications 142, 111–142.

702 Hamouid, K., Othmen, S., Barkat, A., 2020. LSTR: Lightweight and Secure
703 Tree-Based Routing for Wireless Sensor Networks. Wireless Personal Com-
704 munications , 1–23.

705 Hankerson, D., Vanstone, S., Menezes, A.J., 2004. Guide to Elliptic Curve Cryp-
706 tography. Springer, New York, NY.

707 Harbi, Y., Aliouat, Z., Refoufi, A., Harous, S., Bentaleb, A., 2019. Enhanced
708 authentication and key management scheme for securing data transmission in
709 the internet of things. Ad Hoc Networks 94, 101948.

710 Jain, U., Hussain, M., 2020. Securing wireless sensors in military applications
711 through resilient authentication mechanism. Procedia Computer Science 171,
712 719–728.

713 Jerbi, W., Guermazi, A., Trabelsi, H., 2016. O-leach of routing protocol for
714 wireless sensor networks, in: 2016 13th international conference on computer
715 graphics, imaging and visualization (CGiV), IEEE. pp. 399–404.

716 Jiang, J., Han, G., Wang, H., Guizani, M., 2019. A survey on location privacy
717 protection in wireless sensor networks. Journal of Network and Computer Ap-
718 plications 125, 93–114.

719 Joux, A., 2000. A one round protocol for tripartite diffie–hellman, in: Interna-
720 tional algorithmic number theory symposium-Springer, pp. 385–393.

Kar, J., Naik, K., Abdelkader, T., 2020. A secure and lightweight protocol for message authentication in wireless sensor networks. IEEE Systems Journal , 1–12doi:10.1109/JSYST.2020.3015424.

Kim, J.Y., Hu, W., Sarkar, D., Jha, S., 2019. Long-term secure management of large scale internet of things applications. Journal of Network and Computer Applications 138, 15–26.

Kumar, V., Ray, S., Dasgupta, M., Khan, M.K., 2021. A Pairing Free Identity Based Two Party Authenticated Key Agreement Protocol Using Hexadecimal Extended ASCII Elliptic Curve Cryptography. Wireless Personal Communications , 1–17.

Mehmood, A., Umar, M.M., Song, H., 2017. ICMDS: Secure Inter-Cluster Multiple-key Distribution Scheme for wireless sensor networks. Ad Hoc Networks 55, 97–106.

Mezrag, F., Bitam, S., Mellouk, A., 2017. Secure routing in cluster-based wireless sensor networks, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp. 1–6.

Mezrag, F., Bitam, S., Mellouk, A., 2019. IDSP: A New Identity-Based Security Protocol for Cluster-Based Wireless Sensor Networks, in: 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–6. doi:10.1109/PIMRC.2019.8904276.

Mishra, S., Yaduvanshi, R., Dubey, K., Rajpoot, P., 2021. Ess-ibaa: Efficient, short, and secure id-based authentication algorithm for wireless sensor network. International Journal of Communication Systems 34, e4764.

Oliveira, L.B., Aranha, D.F., Gouvêa, C.P., Scott, M., Câmara, D.F., López, J., Dahab, R., 2011. Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. Computer Communications 34, 485 – 493. Special Issue of Computer Communications on Information and Future Communication Security.

Patil, H.K., Szygenda, S.A., 2012. Security for Wireless Sensor Networks using Identity-Based Cryptography. Auerbach Publications.

Saeed, M.E.S., Liu, Q.Y., Tian, G., Gao, B., Li, F., 2019. AKAIoTs: Authenticated Key Agreement for Internet of Things. Wireless Networks 25, 3081–3101.

Shamir, A., 1984. Identity-based cryptosystems and signature schemes, in: Workshop on the theory and application of cryptographic techniques, Springer, pp. 47–53.

Sharma, G., Bala, S., Verma, A.K., 2017. Pf-ibs: pairing-free identity based digital signature algorithm for wireless sensor networks. Wireless personal communications 97, 1185–1196.

Shen, L., Ma, J., Liu, X., Wei, F., Miao, M., 2017. A Secure and Efficient ID-Based Aggregate Signature Scheme for Wireless Sensor Networks. IEEE Internet of Things Journal 4, 546–554.

Shim, K.A., 2014. S2DRP: Secure implementations of distributed reprogramming protocol for wireless sensor networks. Ad Hoc Networks 19, 1–8.

Shim, K.A., 2016. A survey of public-key cryptographic primitives in wireless sensor networks. IEEE Communications Surveys Tutorials 18, 577–601.

Sogani, A., Jain, A., 2019. Energy aware and fast authentication scheme using identity based encryption in wireless sensor networks. Cluster Computing 22, 10637–10648.

Solapure, S.S., Kenchannavar, H.H., Sarode, K.P., 2020. Issues faced during rpl protocol analysis in contiki-2.7. ICT Systems and Sustainability: Proceedings of ICT4SD 2019, Volume 1 1077, 477.

Texas Instruments, 2007. CC2520 2.4 Ghz IEEE 802.15.4/ZigBee RF Transceiver Data sheet. [Online]. Available: http://www.ti.com/product/CC2520. Accessed March 2021.

Texas Instruments, 2021. MSP430F5437 online data sheet. [Online]. Available: https://www.ti.com/product/MSP430F5437A. Accessed March 2021.

Tseng, Y.M., Chen, J.L., Huang, S.S., 2021. A lightweight leakage-resilient identity-based mutual authentication and key exchange protocol for resource-limited devices. Computer Networks , 108246.

Vigano, L., 2006. Automated security protocol analysis with the avispa tool. Electronic Notes in Theoretical Computer Science 155, 61–86.

Yousefpoor, M.S., Barati, H., 2019. Dynamic key management algorithms in wireless sensor networks: A survey. Computer Communications 134, 52 – 69.

Yousefpoor, M.S., Yousefpoor, E., Barati, H., Barati, A., Movaghar, A., Hosseinzadeh, M., 2021. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. Journal of Network and Computer Applications , 103118.

790 Yuan, E., Wang, L., Cheng, S., Ao, N., Guo, Q., 2020. A key management scheme
791 based on pairing-free identity based digital signature algorithm for heteroge-
792 neous wireless sensor networks. Sensors 20, 1543.

793 Zhong, H., Shao, L., Cui, J., Xu, Y., 2018. An efficient and secure recoverable
794 data aggregation scheme for heterogeneous wireless sensor networks. Journal
795 of Parallel and Distributed Computing 111, 1 – 12.