



HAL
open science

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

Fares Mezrag, Salim Bitam, Abdelhamid Mellouk

► **To cite this version:**

Fares Mezrag, Salim Bitam, Abdelhamid Mellouk. An efficient and lightweight identity-based scheme for secure communication in clustered wireless sensor networks. *Journal of Network and Computer Applications (JNCA)*, 2022, 200, pp.103282. 10.1016/j.jnca.2021.103282 . hal-04344311

HAL Id: hal-04344311

<https://hal.science/hal-04344311v1>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

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)

Preprint submitted to Journal of Network and Computer Applications

October 27, 2021

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.

21 Network security is the set of policies, mechanisms, and services that protect
22 a network from cyber-attacks and unauthorized access (Yousefpoor and Barati,
23 2019). Security in CWSN faces several challenges, especially when it comes to
24 applications requiring a high level of security, such as military, emergency re-
25 sponse, and healthcare (Benayache et al., 2019; Jain and Hussain, 2020). Sensor
26 devices are frequently deployed in hostile or even unsecured environments, which
27 make them subject to more cyber-attacks that can violate sensitive data and ad-
28 versely affect the performance of a network (Jiang et al., 2019; Boubiche et al.,
29 2021). Furthermore, wireless communications within the CWSN are insecure by
30 nature, and as a result, an adversary with a wireless device can easily listen in
31 on communications between legitimate nodes. Therefore, minimal security re-
32 quirements such as authentication, data confidentiality, and data integrity must
33 be assured. Also necessary to design a lightweight, efficient, and secure scheme
34 that considers the resource-constrained sensor nodes. In this context, a form of
35 public-key cryptography known as Identity-Based Cryptography (IBC) is consid-
36 ered a practical security solution for resource-constrained devices (Sogani and
37 Jain, 2019; Kim et al., 2019; Saeed et al., 2019; Hamouid et al., 2020; Mishra
38 et al., 2021; Kumar et al., 2021; Tseng et al., 2021). This is due to several fea-
39 tures, including the following:

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

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

48 Two main techniques have been used in the literature to implement IBC-based
49 schemes for sensor nodes: bilinear pairing-based and ECC-based. However, ac-
50 cording to recent implementation results on many WSN platforms, the time re-
51 quired to compute a single bilinear pairing is equal to the computation between
52 two to seven elliptic curve point multiplications (Shim, 2016). Therefore, IBC
53 schemes based on pairing are considered slow and increase a computation over-
54 head for sensor nodes compared with IBC schemes based on the elliptic curve.

55 *1.1. Contribution*

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

- 64 1. *No public key certificates are necessary*: The proposed scheme is designed
65 to use IBC. Consequently, our scheme provides easy management of public
66 keys compared to PKI-based cryptosystems, and there is no need to generate
67 and maintain public-key certificates.
- 68 2. *Elimination of bilinear pairing and MTP function*: According to our im-
69 plementation results on the WiSMote sensor device (See Table 4), the time

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

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

86 4. *Resource-efficiency*: IBAKAS is resource-efficient. Comparison with ex-
87 isting relevant schemes shows that IBAKAS decreases computational and
88 communication costs, save key storage space and reduces the energy con-
89 sumption on WiSMote sensor devices.

90 1.2. *Paper organization*

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

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

97 **2. Related Works**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

218 3. Preliminaries and system model

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

222 3.1. Elliptic Curve Cryptography

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

229 We consider \mathbb{F}_q a finite field of order q , where q is a large prime number.
230 E/\mathbb{F}_q represents an elliptic curve E over \mathbb{F}_q , which is given by the simplified
231 Weierstrass equation (Patil and Szygenda, 2012): $y^2 = x^3 + ax + b$, where $a,$
232 $b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$.

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

236 3.2. Computational problems

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

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

242 **3.3. Identity-Based Cryptography**

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

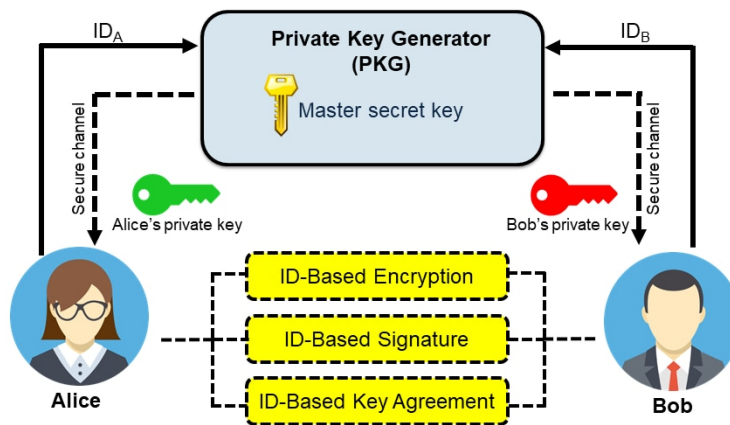


Figure 1: Identity-based cryptography concept.

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

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

275 *3.4. Network model*

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

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

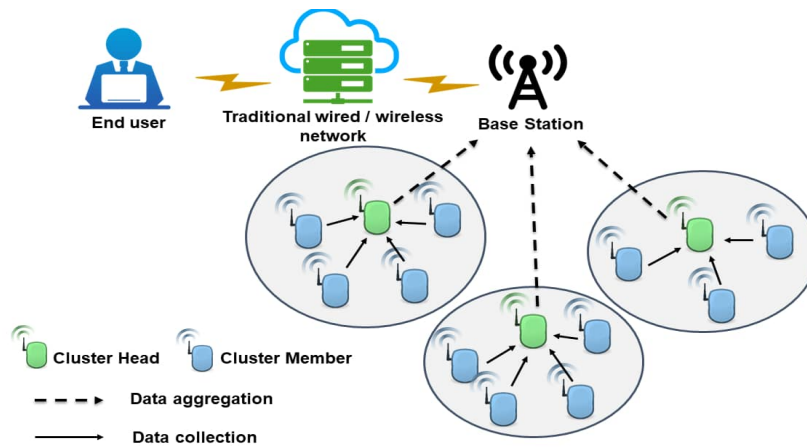


Figure 2: Network model.

288 3.5. Security properties of key agreement schemes

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

- 291 • **Known Session Key:** If an adversary has knowledge of some previous ses-
 292 sion keys, it cannot compromise other session keys.
- 293 • **Unknown Key Share:** A node ID_i cannot be forced to share a key with a
 294 node ID_j when ID_i believes that the key is shared with another node ID_k
 295 $\neq ID_j$.

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

304 3.6. *Cyber-attacks on CWSN*

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

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

- 321 • **MITM attack:** During this cyber-attack, an adversary can send forged mes-
322 sages to legitimate CH and CM nodes to control much of the data circulating
323 between them.
- 324 • **Replay attack:** The adversary attempts to retransmit previous messages
325 exchanged between CM and CH or between CH and BS to pretend that the
326 legitimate node sends the message again.
- 327 • **Sybil attack:** In this cyber-attack, a malicious node impersonates the iden-
328 tities of targeted legitimate nodes for the purpose of degrading the effective-
329 ness of several features such as data distribution.
- 330 • **HELLO flood attack:** The adversary with a high-powered antenna sends
331 a flood of HELLO message to sensor nodes. The remote node receiving
332 this message believes that the adversary as a neighbor and it is within the
333 range of communication. Hence it tries to send its messages directly to the
334 adversary leading to failure of messages transmission, and to disrupt the
335 network operation by prevent other messages to be exchanged.

336 **4. Proposed scheme**

337 In this section, we illustrate the proposed scheme, which is divided into two
338 main phases, namely System initialization phase and Key agreement phase. Table
339 1 lists the notations used in the proposed scheme. Below are the descriptions of
340 each phase.

341 *4.1. System initialization phase*

342 During this phase, two sub-phases are presented, the setup phase and the key
343 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

344 **Setup phase.** Given a security parameter k , the BS determines the tuple
 345 $\{\mathbb{F}_q, E/\mathbb{F}_q, \mathbb{G}, P\}$ where \mathbb{G} denotes a group with prime order q and the point P
 346 is the generator of \mathbb{G} . The BS picks a random number $x \in \mathbb{Z}_q^*$ as the master
 347 secret key, it thereafter computes the master public key $P_{pub} = xP$. Then, three
 348 hash functions are chosen: $H_0 : \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$, $H_1 : \{0, 1\}^* \times \mathbb{G}^2 \rightarrow \mathbb{Z}_q^*$
 349 and $H_2 : \{0, 1\}^* \times \{0, 1\}^* \times \mathbb{G}^3 \rightarrow \{0, 1\}^k$. finally, the system parameters
 350 $\{\mathbb{F}_q, E/\mathbb{F}_q, \mathbb{G}, P, P_{pub}, H_0, H_1, H_2\}$ 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:

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

351 4.2. *Mutual authentication and key agreement phase*

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

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

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

358 **Step 2.** Upon receiving the message M_1 , node B picks a random number $y_B \in \mathbb{Z}_q^*$
359 as its ephemeral secret key and computes both $T_B = (y_B + d_B)^2.P$ and the
360 value $\sigma_B = H_1(ID_B || T_B || d_B.W_A)$. Then, node B sends the message $M_2 =$
361 $(ID_B, T_B, W_B, \sigma_B)$ to node A through an insecure channel.

362 **Step 3.** Node A computes $\hat{\sigma}_B = H_1(ID_B || T_B || d_A.W_B)$ locally. Then, it verifies
363 the authenticity of node B by checking whether the condition $\hat{\sigma}_B \stackrel{?}{=} \sigma_B$.
364 If it holds, A authenticates B and then establishes the session key $sk =$
365 $H_2(ID_A || ID_B || T_A || T_B || K_A)$, where $K_A = (y_A + d_A)^2.T_B$. Furthermore,
366 node A computes $\sigma_A = H_1(ID_A || T_A || d_A.W_B)$ and then sends σ_A to node
367 B .

368 **Step 4.** Similarly, node B computes $\hat{\sigma}_A = H_1(ID_A || T_A || d_B.W_A)$ and compares
369 with received σ_A . If $\hat{\sigma}_A = \sigma_A$, node B authenticates A and establishes
370 the session key as $sk = H_2(ID_A || ID_B || T_A || T_B || K_B)$, where $K_B = (y_b +$
371 $d_b)^2.T_A$.

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:

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

372

373 **5. Security analysis of the proposed scheme**

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

376 *5.1. Formal security verification using AVISPA*

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

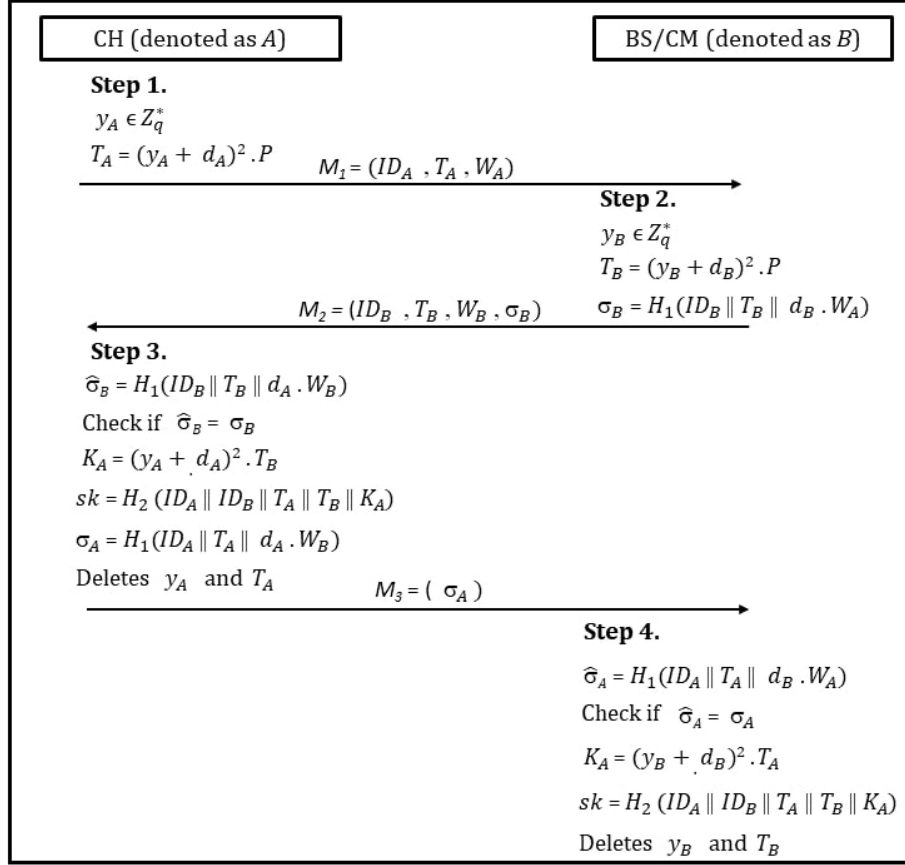


Figure 3: Mutual authentication and key agreement phase in the proposed scheme.

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

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

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

<pre> 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}) ∧ Sigma_b' := H(B.TB'.Mul(Db.WA')) 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 </pre>	<pre> 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 </pre>
---	---

Figure 4: The basic roles in HLPSL.

403

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

<pre> 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 </pre>	<pre> 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() </pre>
--	---

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

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

<pre> % 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 </pre>	<pre> 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 </pre>
--	---

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

408

409 **5.2. Informal security analysis**

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

- 413 • **Known Session Key:** In this proposal, the session key between CH and
414 CM is computationally dependent on ephemeral secrets (y_{CM}, y_{CH}) and
415 long-term private keys (d_{CM}, d_{CH}) . Each session has different ephemeral
416 secrets y_{CM} and y_{CH} . Due to difficulties of ECDLP, an adversary failed to
417 extract (y_{CM}, y_{CH}) from (T_{CM}, T_{CH}) , as well as (d_{CM}, d_{CH}) from $(W_{CM},$
418 $W_{CH})$. Thus, the compromised session key does not allow an adversary to
419 reveal other session keys. Therefore, our scheme could provide the Known
420 session key property.
- 421 • **Unknown Key Share:** The proposed IBAKAS satisfies this propriety since
422 both CH and CM compute the session key based on T_{CH} and T_{CM} validated
423 by their respective signatures σ_{CH} and σ_{CM} . Further, due to ECDLP, the
424 private keys of nodes cannot be derived from their public keys.
- 425 • **Perfect Forward Secrecy:** Suppose that an adversary has compromised
426 long-term private keys d_{CM} and d_{CH} . However, it cannot reveal previous
427 established session keys, since ephemeral secrets y_{CM} and y_{CH} are un-
428 known and renewed at every session. Moreover, an adversary is unable
429 to extract y_{CM} and y_{CH} from T_{CM} and T_{CH} , respectively, due to difficulties
430 of ECDLP. Therefore, the proposed scheme provides the perfect forward
431 secrecy.
- 432 • **Key Compromise Impersonation:** Suppose that the long-term private key
433 d_{CM} is disclosed to a malicious node (denoted as \mathcal{E}) who tries to imperson-

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

439 • **No Key Control:** Since both CH and CM choose random ephemeral se-
 440 crets y_{CH} and y_{CM} , respectively, neither entity can influence the random
 441 selection process. Thus, our scheme ensure no key control propriety.

442 • **MITM attack:** According to our scheme, $T_{CH} = (y_{CH} + d_{CH})^2 \cdot P$ and
 443 $T_{CM} = (y_{CM} + d_{CM})^2 \cdot P$ are exchanged with the σ_{CH} and σ_{CM} signatures.
 444 Once T_{CH} and T_{CM} are validated, CH and CM nodes compute the shared
 445 session key sk_{CH}^{CM} using the long-term private keys, d_{CH} and d_{CM} , and the
 446 ephemeral secret keys (random numbers), y_{CH} and y_{CM} . The MITM attack
 447 may occur in the proposed scheme if a malicious node extracts d_{CH} and
 448 d_{CM} from public values $(W_{CH}, W_{CM}) = (d_{CH} \cdot P, d_{CM} \cdot P)$, and then com-
 449 puts $d_{CH} \cdot d_{CM} \cdot P$. Due to the difficulties of CDHP, this computation is not
 450 possible. Thus, our scheme prevents MITM attack.

451 • **Replay attack:** As described in our scheme, messages M_1 and M_1 contain
 452 T_{CH} and T_{CM} , respectively. In addition, the message M_3 contains σ_{CH} ,
 453 which is calculated based on T_{CH} . Due to the dynamic nature of T_{CH} and
 454 T_{CM} , which are regularly updated, our scheme can reject all replayed mes-
 455 sages by checking T_{CH} and T_{CM} . Thus, the replay attack is prevented.

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

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

459 established between CH and CM or between CH and BS, the key is then used
460 to encrypt data sent between CH and CM or between CH and the BS, which
461 ensures data confidentiality and protects sensitive data from eavesdrop. Fur-
462 thermore, it is difficult for an adversary to discover the session key since it
463 is dynamic and is renewed at every session. Consequently, the proposed
464 IBAKAS can resist both eavesdropping and brute force attacks.

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

471 Comparing the security features of the proposed IBAKAS and existing authen-
472 tication and key agreement schemes (Mehmood et al., 2017; Harbi et al., 2019;
473 Saeed et al., 2019; Hamouid et al., 2020; Kumar et al., 2021) is provided in Table
474 2

475 6. Performance evaluation

476 In our performance study, we have implemented the IBAKAS scheme in Con-
477 tikiOS (Solapure et al., 2020), a lightweight operating system designed for WSN
478 and IoT devices. As well, IBAKAS and existing relevant schemes (Mehmood
479 et al., 2017; Harbi et al., 2019; Saeed et al., 2019; Hamouid et al., 2020; Kumar
480 et al., 2021) are tested using the Cooja network simulator (Solapure et al., 2020).
481 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.

482 Texas Instruments, 2007, 2021), which is equipped with MSP430F5437A MCU,
 483 256 KB of flash memory, 16 KB of SRAM, and CC2520 radio chip. For opera-
 484 tions on an elliptic curve, we used a lightweight asymmetric cryptographic library
 485 suitable for WSN and IoT devices, known as RELIC toolkit (Aranha et al., 2020),
 486 with 160-bit ECC to achieve the 80-bit level of security. Due to the SHA-1 hash
 487 function is broken, we applied the SHA256 hash function truncated to 20 bytes
 488 length.

489 6.1. Evaluation metrics and results

490 Four main metrics have been used to evaluate the performance of IBAKAS
 491 scheme , including the computation cost, the communication cost, the energy con-
 492 sumption and the key storage cost. The results obtained are also compared with
 493 existing authentication and key agreement schemes: ICMDS (Mehmood et al.,
 494 2017), MAKA (Harbi et al., 2019), AKAIoTs (Saeed et al., 2019), LSTR (Hamouid
 495 et al., 2020), and Kumar et al. (Kumar et al., 2021). It is clear that IBAKAS,
 496 AKAIoTs, and Kumar et al. are ECC-based schemes, while the others utilize a

497 pairing technique.

498 6.1.1. *Computation cost*

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

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

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

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

520 Table 5 shows the total computation time for IBAKAS and the cluster-based
521 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.

Schemes	Cluster-based	Pairing	Sensor node		
			Computation cost	Communication cost / Transmit	Communication cost / Receive
ICMDS (2017)	Y	Y	$1BP + 1HG + 1PM + 1H$	–	$2 \mathbb{Z}_q^* + (m + 2) G_1 $
MAKA (2019)	Y	Y	$1BP + 1HG + 4PM$	$ \mathbb{G}_2 + 2 G_1 + nonce $	$3 G_1 + nonce $
AKA-IoTs (2019)	N	N	$6EM + 1EA + 4H$	$ ID + 2 G + 2 \mathbb{Z}_q^* + nonce $	$ ID + 2 G + 2 \mathbb{Z}_q^* + nonce $
LSTR (2020)	Y	Y	$2BP + 1HG + 2PM + 1H$	$3 ID + G_1 + nonce $	$3 ID + G_1 + nonce $
Kumar et al. (2021)	N	N	$5EM + 2EA + 4H$	$ ID + 2 G + \mathbb{Z}_q^* $	$ ID + 2 G + \mathbb{Z}_q^* $
Proposed scheme	Y	N	$4EM + 3H$	$ ID + 2 G + \mathbb{Z}_q^* $	$ ID + 2 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

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

525 6.1.2. Communication cost

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

534 can easily determine the y-coordinate by computing the square root.(Shim, 2014).

535 The size of messages transmitted and received by the schemes are as follows:

- 536 • The ICMDS scheme requires a sensor node to receive $(P_{pub}, R, C, C_0, C_1, \dots$
537 $, C_m)$, where $P_{pub} \in \mathbb{G}_1$, $\{R, C\} \in \mathbb{Z}_q^*$ and $\{C_0, \dots, C_m\} \in \mathbb{G}_1$. Assuming
538 the number of CHs is $(m = 10)$, the size of the received message is $2|\mathbb{Z}_q^*| +$
539 $12|\mathbb{G}_1| = 2 \times 20 + 12 \times 21 = 292$ bytes. Note that the sensor node does not
540 transmit any message to the BS during the session key agreement. Thus,
541 there is no communication cost for transmitting messages.
- 542 • The MAKKA scheme requires a sensor node to transmit (PU, EM) , where
543 $EM \in \{|\mathbb{G}_1| + |\mathbb{G}_1| + |nonce|\}$ and $PU \in |\mathbb{G}_2|$. Additionally, it requires
544 a sensor to receive (P, EM) , where $EM \in \{|\mathbb{G}_1| + |\mathbb{G}_1| + |nonce|\}$ and
545 $P \in |\mathbb{G}_1|$. Therefore, the size of a transmitted message is $|\mathbb{G}_2| + 2|\mathbb{G}_1| +$
546 $|nonce| = 41 + 2 \times 21 + 2 = 85$ bytes. The size of a received message is
547 $3|\mathbb{G}_1| + |nonce| = 3 \times 21 + 2 = 65$ bytes.
- 548 • The LSTR scheme requires a sensor node to transmit $PDU \in \{|ID| + |ID| +$
549 $|ID| + |\mathbb{G}_1| + |nonce|\}$. In addition, LSTR requires a sensor node to receive
550 the same size message as it transmitted. Therefore, the size of a transmitted
551 message is $3|ID| + |\mathbb{G}_1| + |nonce| = 3 \times 2 + 21 + 2 = 29$ bytes. The size
552 of a received message is 29 bytes.

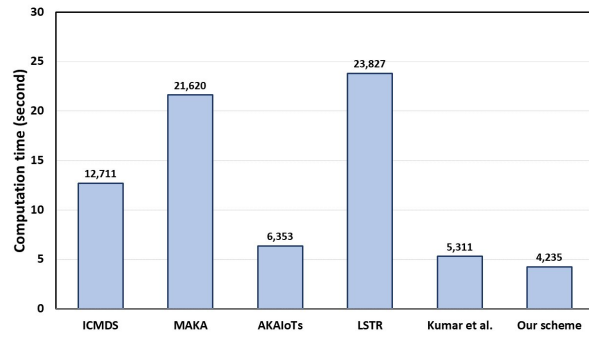
553 According to AKAIoTs (Saeed et al., 2019), Kumar et al. (Kumar et al., 2021),
554 and our scheme, we use the curve SECG-P160 over a 160-bit primary field. In this
555 curve, the size of an element in the group \mathbb{G} is 40 bytes and can be compressed
556 to 21 bytes. The size of messages transmitted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

- 558 • The AKAIoTs scheme requires a sensor node to transmit $(ID, Y, \sigma, nonce)$,
559 where $\sigma \in \{|\mathbb{Z}_q^*| + |\mathbb{Z}_q^*| + |\mathbb{G}|\}$ and $Y \in \mathbb{G}$. In addition, AKAIoTs requires
560 a sensor node to receive the same size message as it transmitted. Therefore,
561 the size of a transmitted message is $|ID| + 2|\mathbb{G}| + 2|\mathbb{Z}_q^*| + |nonce| = 2 +$
562 $2 \times 21 + 2 \times 20 + 2 = 86$ bytes. The size of a received message is 86 bytes.
- 563 • The Kumar et al. scheme requires a sensor node to transmit (ID, T, R, S) ,
564 where $\{T, R\} \in \mathbb{G}$ and $S \in \mathbb{Z}_q^*$. In addition, it requires a sensor node to
565 receive the same size message as it transmitted. Therefore, the size of a
566 transmitted message is $|ID| + 2|\mathbb{G}| + |\mathbb{Z}_q^*| = 2 + 2 \times 21 + \times 20 = 64$ bytes.
567 The size of a received message is 64 bytes.
- 568 • The proposed scheme requires a sensor node to transmit (ID, T, W, σ) , where
569 $\{T, W\} \in \mathbb{G}$ and $\sigma \in \mathbb{Z}_q^*$. In addition, the proposal requires a sensor node
570 to receive the same size message as it transmitted. Thus, the size of a trans-
571 mitted message is $|ID| + 2|\mathbb{G}| + |\mathbb{Z}_q^*| = 2 + 2 \times 21 + \times 20 = 64$ bytes. The
572 size of a received message is 64 bytes.

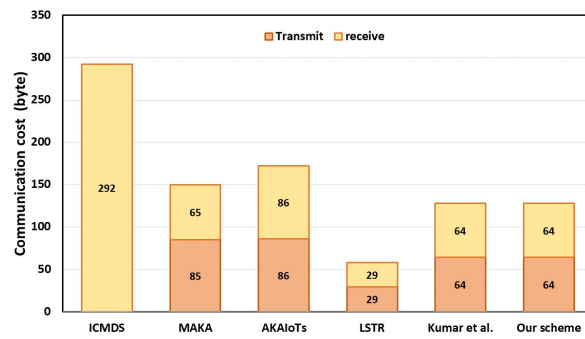
573 As shown in Figure 7(b), the obtained results demonstrate that the proposed IBAKAS
574 introduces a low communication cost than ICMDs, MAKA, and AKAIoTs. In
575 contrast, LSTR appears to offer better communication efficiency than our scheme.
576 However, as shown in Table 2, the LSTR scheme suffers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

579 6.1.3. *Energy consumption*

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



(a) Computation cost.



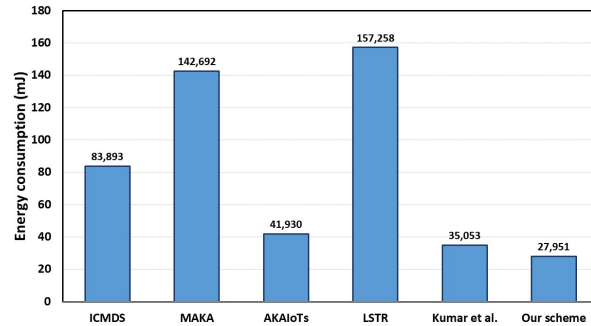
(b) Communication cost.

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

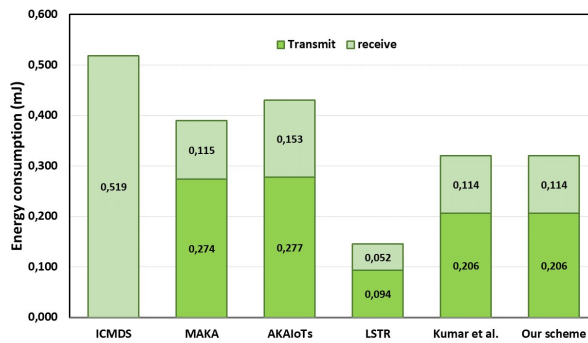
582 $V \times I_{tx/rx} \times U \times \frac{8}{dr}$, respectively. Where $W_{comp/tx/rx}$ represents the energy in mil-
583 lijoules (mJ), V is the voltage, I_c , denotes the current draw in CPU active mode,
584 $I_{tx/rx}$ denotes the current draw in transmitting/receiving mode, U is the size of
585 message in byte, t is the computation time in second and dr represents the data
586 rate. According to WiSMote sensor device, I_c , I_{tx} , I_{rx} are 2.2 mA, 33.6 mA and
587 18.5 mA respectively. In addition, the supply voltage is set to 3 Volts, and the data
588 rate is equal to 250 kbps (Texas Instruments, 2007, 2021).

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

600 Figure 9 illustrates the total estimated energy consumption according to the
601 number of clusters. Compared to ICMDS, MAKA, and LSTR schemes, IBAKAS
602 is energy efficient. Indeed, IBAKAS can reduce the total energy consumption by
603 66.68%, 80.41%, and 82.23% compared to ICMDS, MAKA, and LSTR, respec-
604 tively. The main reason for this improvement is that the computation affects the
605 energy consumption and the total computational time is considerably lower in the
606 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.

607 lower.

608 6.1.4. Key storage cost

609 Because sensor nodes are resource-constrained, key storage overhead is an
 610 important factor to consider. Figure 10 illustrates the amount of memory required
 611 to store long-term and ephemeral keys in a sensor node. Comparing to existing
 612 relevant schemes, IBAKAS is memory efficient and requires less memory space
 613 for storing keys. Indeed, in IBAKAS, ephemeral and long-term keys require only
 614 76 and 100 bytes, respectively. Therefore, the total size of key storage is 76 +

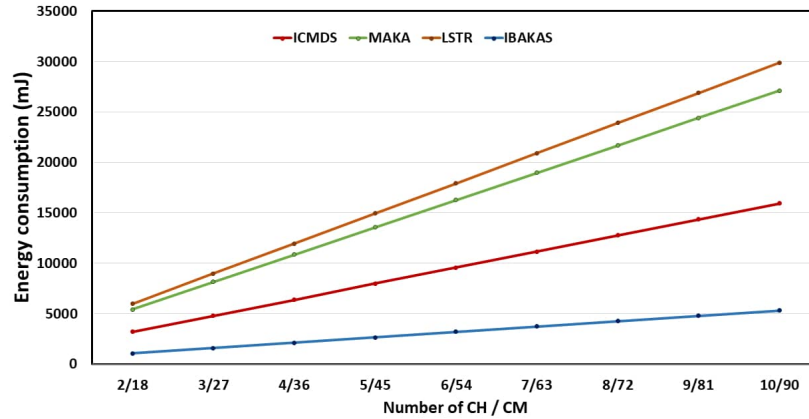


Figure 9: Total energy consumption according to a number of clusters.

615 100 = 176 bytes, which is equivalent to 1.07% (176 bytes from 16 KB) of SRAM
 616 memory. This percent is generally acceptable and satisfactory on the WiSMote
 617 sensor device.

618 7. Use cases

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

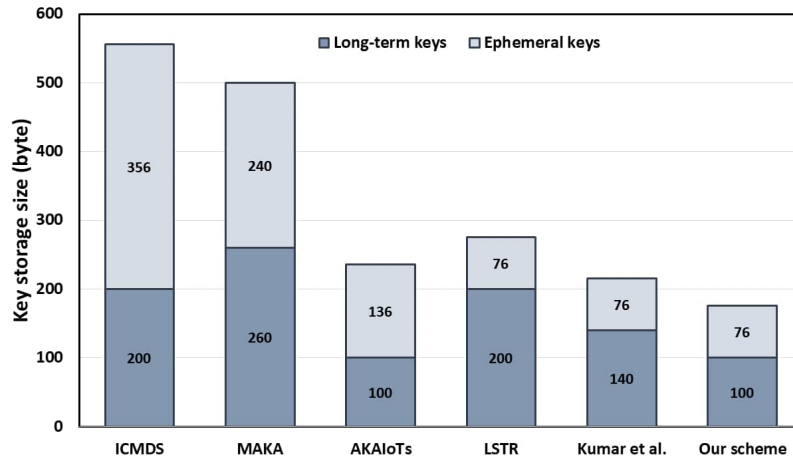


Figure 10: Key storage size required by a sensor node.

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

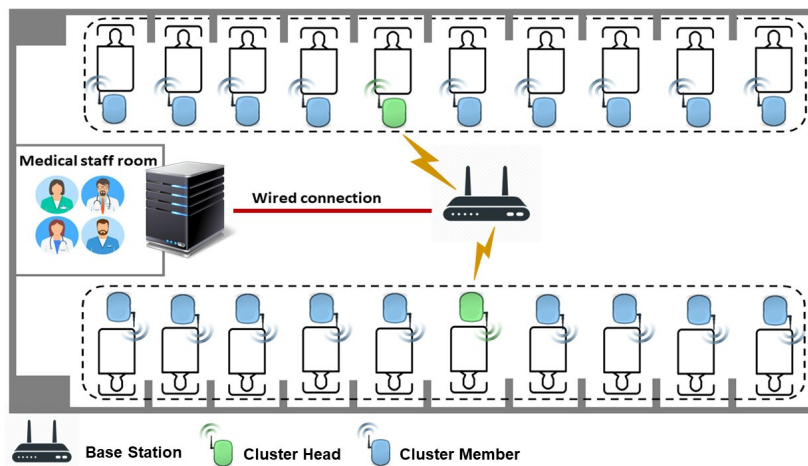


Figure 11: Patient's monitoring in the field hospital.

642 8. Conclusion

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

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

- 656 1. We aim to extend our scheme to support blockchain-based IoT in healthcare
657 applications. In this context, the extended version will be used to secure the
658 communication between IoT devices and blockchain nodes in order to pro-
659 tect the privacy of sensitive data such as Electronic Health Records (EHRs).
- 660 2. We will implement our scheme on real resource-constrained sensor devices.
- 661 3. We will validate our scheme using the Random Oracle Model (ROM).

662 **References**

- 663 Almajed, H.N., Almogren, A.S., 2019. Se-enc: A secure and efficient encoding
664 scheme using elliptic curve cryptography. *IEEE Access* 7, 175865–175878.
- 665 Aranha, D.F., et al., 2020. RELIC is an Efficient LIbrary for Cryptography. [On-
666 line]. Available:<https://github.com/relic-toolkit/relic>. Ac-
667 cessed April 2020.
- 668 Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J.,
669 Drielsma, P.H., Héam, P.C., Kouchnarenko, O., Mantovani, J., et al., 2005. The
670 avispa tool for the automated validation of internet security protocols and appli-
671 cations, in: *International conference on computer aided verification*, Springer.
672 pp. 281–285.
- 673 Benayache, A., Bilami, A., Barkat, S., Lorenz, P., Taleb, H., 2019. Msm: A mi-
674 croservice middleware for smart wsn-based iot application. *Journal of Network*
675 *and Computer Applications* 144, 138–154.

- 676 Blake-Wilson, S., Johnson, D., Menezes, A., 1997. Key agreement protocols and
677 their security analysis, in: Darnell, M. (Ed.), *Cryptography and Coding*, Springer
678 Berlin Heidelberg, Berlin, Heidelberg. pp. 30–45.
- 679 Boneh, D., Franklin, M., 2001. Identity-based encryption from the weil pairing,
680 in: *Annual international cryptology conference*, Springer, pp. 213–229.
- 681 Boubiche, D.E., Athmani, S., Boubiche, S., Toral-Cruz, H., 2021. Cybersecurity
682 issues in wireless sensor networks: Current challenges and solutions. *Wireless
683 Personal Communications* 117.
- 684 Castelluccia, C., Mykletun, E., Tsudik, G., 2005. Efficient aggregation of en-
685 crypted data in wireless sensor networks, in: *The Second Annual International
686 Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp.
687 109–117.
- 688 Chen, L., Kudla, C., 2003. Identity Based Authenticated Key Agreement Proto-
689 cols from Pairings, in: *16th IEEE Computer Security Foundations Workshop,
690 2003. Proceedings.*, pp. 219–233.
- 691 Dolev, D., Yao, A., 1983. On the security of public key protocols. *IEEE Transac-
692 tions on information theory* 29, 198–208.
- 693 Du, H., Wen, Q., Zhang, S., Gao, M., 2020. A new provably secure certificateless
694 signature scheme for internet of things. *Ad Hoc Networks* 100, 102074.
- 695 Dunkels, A., 2015. Platforms supported by Contiki-OS:
696 WiSMote Platform Specifications. [Online]. Available:
697 <https://github.com/contiki-os/contiki/>. Accessed March
698 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 Crypt-*
706 *ography*. 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.

- 721 Kar, J., Naik, K., Abdelkader, T., 2020. A secure and lightweight protocol for
722 message authentication in wireless sensor networks. *IEEE Systems Journal* ,
723 1–12doi:10.1109/JSYST.2020.3015424.
- 724 Kim, J.Y., Hu, W., Sarkar, D., Jha, S., 2019. Long-term secure management of
725 large scale internet of things applications. *Journal of Network and Computer*
726 *Applications* 138, 15–26.
- 727 Kumar, V., Ray, S., Dasgupta, M., Khan, M.K., 2021. A Pairing Free Identity
728 Based Two Party Authenticated Key Agreement Protocol Using Hexadecimal
729 Extended ASCII Elliptic Curve Cryptography. *Wireless Personal Communica-*
730 *tions* , 1–17.
- 731 Mehmood, A., Umar, M.M., Song, H., 2017. ICMDS: Secure Inter-Cluster
732 Multiple-key Distribution Scheme for wireless sensor networks. *Ad Hoc Net-*
733 *works* 55, 97–106.
- 734 Mezrag, F., Bitam, S., Mellouk, A., 2017. Secure routing in cluster-based wireless
735 sensor networks, in: *GLOBECOM 2017 - 2017 IEEE Global Communications*
736 *Conference*, pp. 1–6.
- 737 Mezrag, F., Bitam, S., Mellouk, A., 2019. IDSP: A New Identity-Based Secu-
738 rity Protocol for Cluster-Based Wireless Sensor Networks, in: *2019 IEEE 30th*
739 *Annual International Symposium on Personal, Indoor and Mobile Radio Com-*
740 *munications (PIMRC)*, pp. 1–6. doi:10.1109/PIMRC.2019.8904276.
- 741 Mishra, S., Yaduvanshi, R., Dubey, K., Rajpoot, P., 2021. Ess-ibaa: Efficient,
742 short, and secure id-based authentication algorithm for wireless sensor network.
743 *International Journal of Communication Systems* 34, e4764.

- 744 Oliveira, L.B., Aranha, D.F., Gouvêa, C.P., Scott, M., Câmara, D.F., López,
745 J., Dahab, R., 2011. Tinyabc: Pairings for authenticated identity-based non-
746 interactive key distribution in sensor networks. *Computer Communications* 34,
747 485 – 493. Special Issue of *Computer Communications on Information and*
748 *Future Communication Security*.
- 749 Patil, H.K., Szygenda, S.A., 2012. *Security for Wireless Sensor Networks using*
750 *Identity-Based Cryptography*. Auerbach Publications.
- 751 Saeed, M.E.S., Liu, Q.Y., Tian, G., Gao, B., Li, F., 2019. AKAIoTs: Authen-
752 ticated Key Agreement for Internet of Things. *Wireless Networks* 25, 3081–
753 3101.
- 754 Shamir, A., 1984. Identity-based cryptosystems and signature schemes, in: *Work-*
755 *shop on the theory and application of cryptographic techniques*, Springer, pp.
756 47–53.
- 757 Sharma, G., Bala, S., Verma, A.K., 2017. Pf-ibs: pairing-free identity based dig-
758 ital signature algorithm for wireless sensor networks. *Wireless personal com-*
759 *munications* 97, 1185–1196.
- 760 Shen, L., Ma, J., Liu, X., Wei, F., Miao, M., 2017. A Secure and Efficient ID-
761 Based Aggregate Signature Scheme for Wireless Sensor Networks. *IEEE Inter-*
762 *net of Things Journal* 4, 546–554.
- 763 Shim, K.A., 2014. S2DRP: Secure implementations of distributed reprogramming
764 protocol for wireless sensor networks. *Ad Hoc Networks* 19, 1–8.
- 765 Shim, K.A., 2016. A survey of public-key cryptographic primitives in wireless
766 sensor networks. *IEEE Communications Surveys Tutorials* 18, 577–601.

767 Sogani, A., Jain, A., 2019. Energy aware and fast authentication scheme using
768 identity based encryption in wireless sensor networks. *Cluster Computing* 22,
769 10637–10648.

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

773 Texas Instruments, 2007. CC2520 2.4 Ghz IEEE 802.15.4/Zig-
774 Bee RF Transceiver Data sheet. [Online]. Available:
775 <http://www.ti.com/product/CC2520>. Accessed March 2021.

776 Texas Instruments, 2021. MSP430F5437 online data sheet. [Online]. Available:
777 <https://www.ti.com/product/MSP430F5437A>. Accessed March
778 2021.

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

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

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

786 Yousefpoor, M.S., Yousefpoor, E., Barati, H., Barati, A., Movaghar, A., Hossein-
787 zadeh, M., 2021. Secure data aggregation methods and countermeasures against
788 various attacks in wireless sensor networks: A comprehensive review. *Journal*
789 *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.