People’s Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

MOHAMED KHIDER UNIVERSITY OF BISKRA

Exact Sciences, Natural Sciences and Life Sciences Faculty

Department of Computer Sciences

Order Number: ....

Serial Number: ....

Thesis

Presented as a requirement for the degree of

Doctor in computer science

by

Fares MEZRAG

THEME

Identity-Based Security for Clustered Wireless Sensor Networks

Board of Examiners:

Pr. Sadek Labib Terrissa	University of Biskra	Chair
Pr. Salim Bitam	University of Biskra	Supervisor
Pr. Abdelhamid Mellouk	University of Paris-Est Créteil	Co-supervisor
Pr. Nasreddine Lagraa	University of Laghouat	Examiner
Dr. Noureddine Chikouche	University of M’Sila	Examiner
Dr. Mohamed Faouzi Zerarka	University of Biskra	Examiner

Academic year 2021/2022

Acknowledgements

First of all, I thank **ALLAH**, the almighty, for giving me the patience and the will to carry out this work.

I would like to express my gratitude to my supervisor, **Pr. Salim Bitam** for giving me the opportunity to work under his direction. Moreover, I would like to thank him for his patience, motivation, professional guidance, and academic support.

I am incredibly grateful to my co-supervisor, **Pr. Abdelhamid Mellouk** for his helpful remarks and enlightening advice.

I wish to thank the board of examiners, **Pr. Sadek Labib Terrissa**, **Pr. Nasreddine Lagraa**, **Dr. Noureddine Chikouche**, and **Dr. Mohamed Faouzi Zerarka** for accepting to evaluate this thesis.

Finaly, I dedicate this work to my parents, my wife, my two little girls: **Rital** and **Ilaf**, and my family. I am most grateful to them for their patience, understanding, and encouragement.

Abstract

The security of CWSN is challenged by several factors, particularly when it comes to applications requiring a high level of security, such as military surveillance, emergency response, and healthcare services. Indeed, the sensors are considered resource-constrained and are not tamper-resistant. They are usually deployed in hostile or even insecure environments, making them vulnerable to cyber-attacks that can compromise sensitive data and adversely affect the good operation of the network. Moreover, wireless communications within the CWSN are insecure by nature. As a result, an adversary with a wireless device can easily listen in on communications between legitimate nodes. Therefore, basic 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 constrained resources of sensor nodes, particularly the available energy. The main objective of this thesis is to overcome the security issues associated with the CWSNs. In this context, we propose three efficient security schemes called HCBS, IDSP, and IBAKAS. Our proposals are built upon a perfect trade-off between three core elements: (i) a good level of security, particularly against the various cyber-attacks that target CWSNs, (ii) resource efficiency, (iii) ease of distribution and management of cryptographic keys between sensor nodes. HCBS is tested on the TOSSIM simulator using the MicaZ platform. While, IDSP and IBAKAS are tested using a Cooja simulator and the WiSMote platform. According to obtained results, our proposals are secure, efficient, and suitable for CWSNs compared to recent related schemes.

Keywords : Cluster-Based WSN, Identity-based cryptography, Elliptic curve cryptography, Bilinear pairing, Mutual authentication, Key distribution.

ملخص

يتعرض أمن CWSN لتحديات عدة ، لا سيما عندما يتعلق الأمر بالتطبيقات التي تتطلب مستوى عالٍ من الأمان ، مثل المراقبة العسكرية والاستجابة للطوارئ وخدمات الرعاية الصحية. في الواقع ، تعتبر المستشعرات محدودة الموارد وغير مقاومة للعبث. عادة ما يتم نشرها في بيئات معادية أو حتى غير آمنة ، مما يجعلها عرضة للهجمات الإلكترونية التي يمكن أن تعرض البيانات الحساسة للخطر وتؤثر سلبًا على الأداء الجيد للشبكة. علاوة على ذلك ، الاتصالات اللاسلكية داخل CWSN تعتبر غير آمنة بطبيعتها. نتيجة لذلك ، يمكن لخصم يملك جهاز لاسلكي أن يستمع بسهولة إلى الاتصالات بين العقد الشرعية. لذلك ، يجب ضمان المتطلبات الأمنية الأساسية مثل المصادقة وسرية البيانات و سلامة البيانات. ضروري أيضًا لتصميم مخطط خفيف ، فعال وآمن يأخذ في الاعتبار الموارد المحدودة لعقد المستشعر، لا سيما الطاقة المتاحة. الهدف الرئيسي من هذه الأطروحة هو التغلب على المشاكل الأمنية المرتبطة بـ CWSNs. في هذا السياق ، نقترح ثلاثة أنظمة أمان فعالة تدعى HCBS ، IDSP و IBAKAS. مقترحاتنا مبنية على مقايضة مثالية بين ثلاثة عناصر أساسية: (1) مستوى جيد من الأمن ، لا سيما ضد الهجمات الإلكترونية التي تستهدف CWSNs ، (2) كفاءة الموارد ، (3) سهولة توزيع و إدارة مفاتيح التشفير بين عقد الاستشعار. تم اختبار HCBS على المحاكي TOSSIM باستخدام المنصة MicaZ. بينما تم اختبار IDSP و IBAKAS باستخدام المحاكي Cooja و المنصة WiSMote. وفقًا للنتائج التي تم الحصول عليها، فإن مقترحاتنا آمنة وفعالة ومناسبة لـ CWSNs مقارنةً بالمخططات الحديثة ذات الصلة.

الكلمات الرئيسية: شبكة المستشعرات اللاسلكية العنقودية ، التشفير القائم على الهوية ، تشفير المنحنى الإهليلجي ، الاقتران الخطي ، المصادقة المتبادلة ، توزيع المفاتيح.

Résumé

La sécurité de CWSN est mise à l'épreuve par plusieurs facteurs, en particulier lorsqu'il s'agit d'applications nécessitant un haut niveau de sécurité, telles que la surveillance militaire, les interventions d'urgence et les services de santé. En effet, les capteurs sont considérés comme ayant des ressources limitées et ne sont pas inviolables. Ils sont généralement déployés dans des environnements hostiles voire non sécurisés, ce qui les rend vulnérables aux cyber-attaques qui peuvent compromettre des données sensibles et nuire au bon fonctionnement du réseau. De plus, les communications sans fil au sein du CWSN ne sont pas sécurisées par nature. En conséquence, un adversaire avec un appareil sans fil peut facilement écouter les communications entre les nœuds légitimes. Par conséquent, les exigences de sécurité de base telles que l'authentification, la confidentialité des données et l'intégrité des données doivent être assurées. Il est également nécessaire de concevoir un schéma léger, efficace et sécurisé qui prend en compte les ressources limitées des nœuds de capteurs, en particulier l'énergie disponible. L'objectif principal de cette thèse est de surmonter les problèmes de sécurité associés aux réseaux CWSN. Dans ce contexte, nous proposons trois schémas de sécurité efficaces appelés HCBS, IDSP et IBAKAS. Nos propositions reposent sur un compromis parfait entre trois éléments essentiels : (i) un bon niveau de sécurité, notamment contre les différentes cyber-attaques qui ciblent les CWSN, (ii) l'efficacité des ressources, (iii) la facilité de distribution et de gestion des clés cryptographiques entre les nœuds capteurs. HCBS est testé sur le simulateur TOSSIM à l'aide de la plateforme MicaZ. Tandis qu'IDSP et IBAKAS sont testés à l'aide d'un simulateur Cooja et de la plate-forme WiSMote. Selon les résultats obtenus, nos propositions sont sécurisées, efficaces et adaptées aux CWSN par rapport aux schémas connexes récents.

Mots-clés : WSN basé sur les clusters, cryptographie basée sur l'identité, cryptographie sur les courbes elliptiques, appariement bilinéaire, authentification mutuelle, distribution de clés.

Contents

List of Figures

List of Tables

List of Publications

Journals

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

- *Authors*: Fares Mezrag, Salim Bitam, and Abdelhamid Mellouk.

- *Journal*: Journal of Network and Computer Applications.

- *Year*: 2022.

Conferences

- *Title*: **Secure routing in cluster-based wireless sensor networks.**

- *Authors*: Fares Mezrag, Salim Bitam, and Abdelhamid Mellouk.

- *Conference*: GLOBECOM 2017-2017 IEEE Global Communications Conference.

- *Location*: Singapore.

- *Year*: 2017.

- *Title*: **IDSP: A New Identity-Based Security Protocol for Cluster-Based Wireless Sensor Networks.**

- *Authors*: Fares Mezrag, Salim Bitam, and Abdelhamid Mellouk.

- *Conference*: 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC).

- *Location*: Istanbul, Turkey

- *Year*: 2019.

List of Abbreviations

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AKAIoTs	Authenticated Key Agreement for Internet of Things
ASCII	American Standard Code for Information Interchange
AVISPA	Automated Validation of Internet Security Protocols and Applications
BDHP	Bilinear Diffie-Hellman Problem
BF-IBE	Boneh Franklin-Identity-Based Encryption
BS	Base Station
CA	Certificate Authority
CDHP	Computational Diffie Hellman problem
CH	Cluster Head
CL-AtSe	Constraint-Logic-based Attack Searcher
CM	Cluster Member
CN	Central Node
CPU	Central Processing Unit
CWSN	Cluster-Based Wireless Sensor Network
DH	Diffie-Hellman
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie–Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
EDD	End-to-End Delay

HCBS	Hybrid Cryptography-Based Scheme
IBAKAS	Identity-Based Authentication and Key Agreement Scheme
IBC	Identity-Based Cryptography
IBE	Identity-Based Encryption
IBKA	Identity-Based Key Agreement
IBS	Identity-Based Signature
ICMDS	Inter-Cluster Multiple key Distribution Scheme
IDSP	Identity-Based Security Protocol
IPv6	Internet Protocol version 6
LEACH	Low-Energy Adaptive Clustering Hierarchy
LPM	Low Power Mode
LR-WPAN	Low Rate-Wireless Personal Area Networks
LSTR	Lightweight and Secure Tree-Based Routing
MAC	Medium Access Control
MAC	Message Authentication Code
MA-IDOOS	Message Authentication-IDentity base Online/Offline Signature
MAKA	Mutual Authentication and session Key Agreement
MCU	Micro-Controller Unit
MITM	Man-In-The-Middle
MMH	Multilinear Modular Hashing
MTP	Map To Point
OFMC	On-the-Fly Model-Checker
OSI	Open Systems Interconnection
PBC	Pairing-Based Cryptography
PHY	Physical
PKC	Public-Key Cryptography
PKG	Private Key Generator
PKI	Public Key Infrastructure
QoS	Quality of service
ROM	Random Oracle Model
RSA	Rivest–Shamir–Adleman

RX	Radio reception
SDTS	Secure Data Transmission Scheme
SecLEACH	Secure LEACH
SET-IBOOS	Secure and Efficient data Transmission-Identity Based Online/Offline Signature
SET-IBS	Secure and Efficient data Transmission-Identity Based Signature
SHA	Secure Hash Algorithm
SKC	Symmetric-Key Cryptography
SOK	Sakai, Ohgishi and Kasahara
SRAM	Static Random Access Memory
TDMA	Time Division Multiple Access
TOSSIM	TinyOS SIMulator
TX	Radio transmission
UDGM	Unit Disk Graph Medium
WSN	Wireless Sensor Network

General introduction

Wireless sensor networks (WSNs) are considered 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 consists of many small devices known as sensor nodes deployed throughout the monitored area. Nodes on this network can communicate wirelessly and exchange data without requiring a fixed network infrastructure. On the other hand, WSNs are typically characterized by the resource-constrained nature of sensor devices, such as processors, energy, storage space, and bandwidth. Besides the limited energy nature, recharging or replacing batteries is considered a difficult task for sensors deployed in an inaccessible environment. This issue would, therefore, adversely affect the lifetime of the network.

To overcome this problem, researchers have investigated the cluster-based WSN (CWSN) architecture to maximize node lifetime as well as to reduce bandwidth consumption [30, 110, 86, 68]. 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.

Problem statement

The concept of network security describes a set of rules and configurations that protect networks from cyber-attacks and unauthorized access, as well as ensuring the integrity and confidentiality of data [108]. The security of CWSN is challenged by several factors, particularly when it comes to applications requiring a high level of security, such as the military, emergency response, and healthcare services [9, 45]. The sensors are usually deployed in hostile or even insecure environments, making them vulnerable to cyber-attacks that can compromise sensitive data and adversely affect the performance of a network [14, 47]. Moreover, wireless communications within the CWSN are insecure by nature. 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 resource-constrained sensor nodes.

Cryptography is considered a security countermeasure to protect communication in open networks such as WSN. The security requirements, including data confidentiality, data integrity, authentication, and non-repudiation, are ensured by cryptography. Selecting appropriate cryptographic primitives is vital in WSNs. Thus, cryptography-based schemes used in WSN should consider resource-constrained sensor nodes. Additionally, these schemes should be evaluated regarding energy consumption, processing time, code size, and data size [94].

Cryptographic techniques can be divided into two categories: Symmetric-Key Cryptography (SKC) and Public-Key Cryptography (PKC). Using SKC offers good performance in terms of computational overhead and energy consumption. However, it does not support non-repudiation, and the distribution of keys is complex. PKC addresses these issues. It provides a more flexible interface and eliminates the need for key pre-distribution and pairwise key sharing. For resource-constrained sensor nodes, it is well known that PKC is computationally expensive [114, 94]. However, recent papers [15, 72, 37, 31, 61, 85, 75, 62, 94] showed that it is feasible to apply PKC in WSN using elliptic curves. In PKC, a public key must be authenticated. It will be necessary to provide a way of verifying the validity of the public key. Otherwise, the communication between nodes is exposed to MITM attacks. Traditional solutions, including Public Key Infrastructure (PKI), which aims to ensure public key authentication, use public key certificates (also known as digital certificates). Such certificates are signed and issued by a trusted entity called Certificate Authority (CA). Unfortunately, PKI is impractical to apply in WSN, as it would introduce overhead and complexity of certificate operations, including distribution, storage, and certificate verification.

An extension of public-key cryptography called Identity-Based Cryptography (IBC), allows an entity's public key to be easily derived from its known identity information . Private key corresponding is issued by a trusted third party called Private Key Generator (PKG). IBC is suitable for WSNs since the BS can serve as a PKG. Sensor nodes' identities and their private keys are generated by the BS, and the corresponding keys are embedded into the nodes before use in the field. Therefore, a secret channel is not required for key setup. Consequently, only the identities of the sensors are exchanged without sending extra public keys and certificates. The identity length is much shorter than the public key and its certificate. This saves energy during the communication between the sensor nodes. Also, the public key is self-authenticated without needing a digital certificate.

Goals and contributions

The main objective of this thesis is to research the security related issues surrounding the cluster-based WSNs as well as develop new security mechanisms for thier. The proposed approaches must take into consideration the limited resources of sensor nodes and address limitations of previous methods to improve the CWSN security. In this context, we propose three contributions.

- Contribution 1 proposes a secure version of LEACH called Hybrid Cryptography-Based Scheme for secure data communication in cluster-based WSN (HCBS). The proposed protocol is based on symmetric and asymmetric (ECC) cryptography primitives. HCBS guarantees the most important security requirements and it resists attacks that LEACH is vulnerable.
- Contribution 2 introduces s an IBC-protocol based on pairing to enhance the security in CWSN, called Identity-based Security Protocol (IDSP).
- Contribution 3 presents an identity-based authentication and key agreement scheme for CWSNs called IBAKAS, which combines ECC and IBC to provide mutual authentication and establish secret session keys over insecure channels.

Dissertation outline

The chapters of this thesis are transcriptions of our research papers that have been published in scientific journals or conference proceedings. There are six chapters in total, divided evenly into two main parts: Background and contributions. The background part provides an overview of WSN and analyzes WSN security, including security constraints, requirements, and potential cyber-attacks that may target WSN. In addition, the part reviews the related works proposed in the literature. The contributions part describes our proposed techniques to enhance the security of CWSNs. The dissertation is organized as follows:

- Chapter 1 defines the principal concepts associated with WSNs, such as architecture and real-world applications. It also highlights the constraints and challenges in WSNs.
- Chapter 2 discusses WSN security, including security constraints, security requirements, and possible cyber-attacks aimed at WSNs. The chapter also reviews cryptographic notions and primitives.
- Chapter 3 reviews existing PKC-based schemes in wireless sensor networks, including IBC schemes based on pairings, IBC schemes based on ECC, and schemes based on ECC.
- Chapter 4 presents the first contribution.
- Chapter 5 presents the second contribution.
- Chapter 6 presents the third contribution.

Background

Chapter 1: Wireless Sensor Networks: An Overview

Chapter 2: Security and Cryptography in WSN: A Background

Chapter 3: PKC-Based Security in WSNs: A Review

Chapter 1

Wireless Sensor Networks: An Overview

Chapter Overview: *This chapter presents an overview of wireless sensor networks, including the principal concepts associated with WSNs, such as architecture and real-world applications. It also highlights the constraints and challenges in wireless sensor networks, which are barriers to facilitating the widespread deployment of WSN technology in real-world domains.*

1.1 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. Moreover, WSNs constitute a special case of ad hoc networks in which no fixed infrastructure is required for device management, and each sensor node can act as a host or a router that cooperates with other nodes.

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. Each sensor node communicates directly with other nodes that are within its transmission range. The communication with the remote node or out of transmission range is done through other nodes that route data to the destination. This process is ensured by a routing protocols. The final destination of this data is a base station (BS) which represents downstream of all data coming from the sensor nodes. The BS also serves as a gateway for transmitting data to the end-user over a traditional wired or wireless network. The typical WSN is illustrated in Figure 1.1.

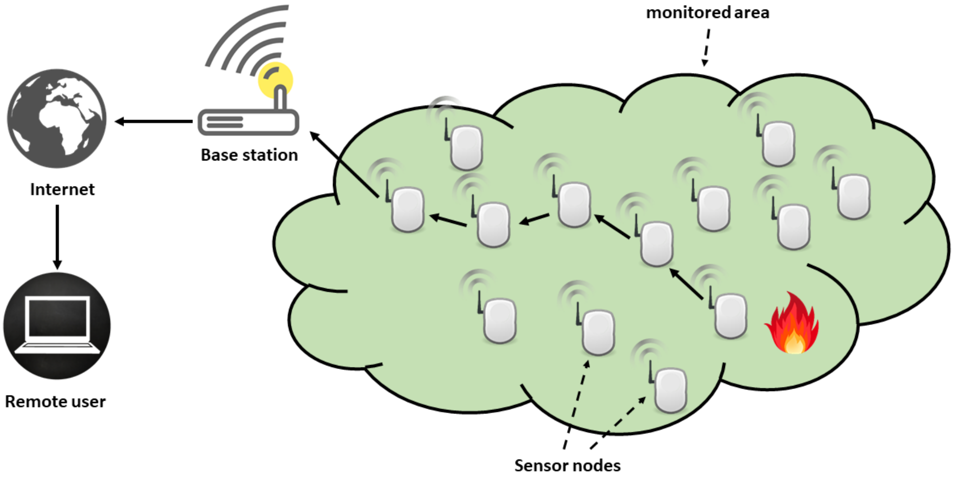


FIGURE 1.1: Example of a wireless sensor network

1.2 Hardware Architecture of Sensor Node

A sensor node comprises four basic components: a sensing unit, a processing unit, a transceiver unit, and a power unit [2]. Furthermore, location-finding system, power generator, and mobilizer may also be included as application-dependent modules. The physical structure of a sensor node is shown in Figure 1.2

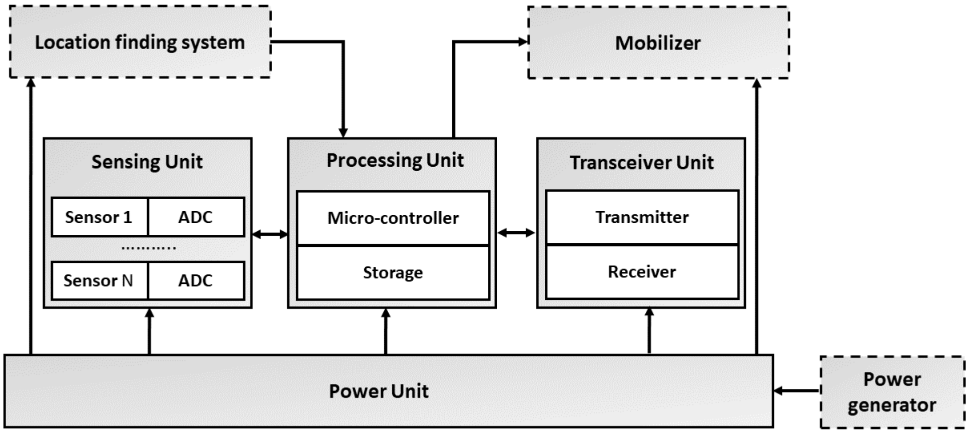


FIGURE 1.2: Physical structure of a sensor node

- **Sensing Unit:** The sensing unit consists of one or more sensors and Analog-to-Digital Converters (ADCs). A sensor is a hardware device that measures physical data based on the observed phenomenon, such as temperature, humidity, or pressure. By means of an ADC, the analog signals produced by the sensors are converted into digital signals and then transmitted to the processing unit.

- **Processing Unit:** The processing unit contains a micro-controller (MCU) and small storage memory. This unit is the core component of a sensor node where it can process the sensed data and execute communication protocols, allowing the sensor node to collaborate with the other nodes in the network.
- **Transceiver unit:** All communications between the sensor nodes are conducted wirelessly through a transceiver unit, which performs all data transmissions and receptions.
- **Power Unit:** One of the most important components of a sensor node is the power unit [2]. It is responsible for providing the supply voltage to all components of a sensor node.

Many manufacturers such as MEMSIC, UC Berkeley, Arago Systems, and Zolertia have developed commercial hardware platforms for sensor nodes. We mention Micaz, Tmote Sky, WiSMote, Z1, and RE-Mote (Figure 1.3). Table 1.1 summarizes the characteristics of these sensor nodes' platforms.

(a) MicaZ (b) Tmote Sky (c) WiSMote (d) Z1 (E) RE-Mote

FIGURE 1.3: Commercial hardware platforms for sensor nodes.

1.3 Architectures of WSNs

WSNs can be categorized into two types of network architectures, including a flat architecture and a cluster-based architecture [86].

TABLE 1.1: Characteristics of some existing commercial platforms for sensor nodes.

Platform	Manufacturer	MCU	SRAM	Flash	Transceiver	Battery
MicaZ [70]	MEMSIC	Atmel ATmega128L	4 KB	128 KB	CC2420	2xAA
Tmote Sky [10]	UC Berkeley	TI MSP430F1611	10 KB	48 KB	CC2420	2xAA
WisMote [25]	Arago Systems	TI MSP430F5437A	16 KB	256 KB	CC2520	2xAA
Z1 [117]	Zolertia	TI MSP430F2617	8 KB	92 KB	CC2420	2xAA
RE-Mote [116]	Zolertia	ARM Cortex-M3	32 KB	512 KB	CC2538 CC1200	2xAA

1.3.1 Flat architecture

Except for the base station node, all the other sensor nodes in the flat architecture are identical and have the same computing and energy capacities. Additionally, a sensor node can act as a host or a router that cooperates with other nodes. Indeed, each sensor node communicates directly with other nodes that are within its transmission range. The communication with the base station or the remote node out of transmission range is done through other nodes that route data to the destination. This type of communication is called multi-hop communication. Figure 1.4 illustrates an example of a WSN' flat architecture.

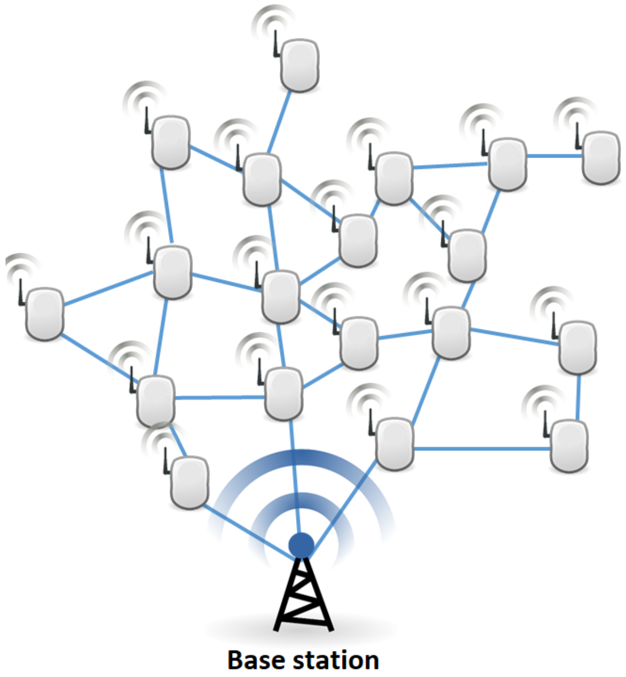


FIGURE 1.4: Example of a flat architecture.

1.3.2 Cluster-based architecture

The idea consists of partitioning a whole network into groups called clusters. Each has one CH and several sensor nodes known as CMs. The CH collects data from all CMs belonging to its cluster, aggregates, and transmits results directly to BS as shown in Figure 1.5. The data aggregation and processing in CH significantly reduce the total number of messages sent to BS. Thus, clustering is an effective way to reduce the total energy consumption of WSN [30, 110]. This hierarchy can be classified into two types: the first is the physical hierarchy, in which the choice of the CH is dependent on the physical capabilities of the sensor node (high level of resource capacity). Here, we are referring to heterogeneous WSNs. In the second type, the sensor nodes used are physically identical. CH selection depends on a variety of factors, such as the distance to the base station or residual energy.

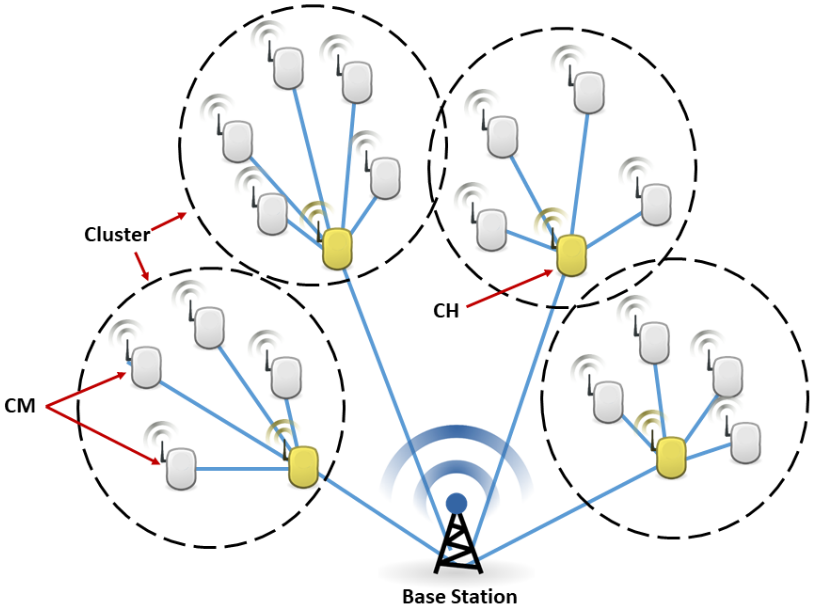


FIGURE 1.5: Example of a cluster-based architecture.

1.4 WSN Applications

Hardware miniaturization technologies, the expansion of the range of sensors types available (thermal and optical, etc.), and the use of wireless communication mediums make sensor networks cover many applications. In the following, we present examples of WSN applications:

1.4.1 Military Applications

The military field encompasses various applications, including tracking friendly or enemy forces and battlefield monitoring. For instance, sensor nodes are used to monitor a critical border area between two countries 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 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.

1.4.2 Healthcare Applications

In the healthcare field, WSN can be applied inside a field hospital for monitoring patients injured on a battlefield or in case of disasters. Indeed, WSN 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 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.

1.4.3 Home automation

Home automation has different tasks such as fire alarm systems, video surveillance, heating system, intrusion detection alarm, and managing domestic appliances. In these applications, sensors are incorporated in various domestic devices to meet a resident's needs, allowing easier control of these devices locally or remotely by the internet. Moreover, a resident obtains all the information necessary regarding the sate of his home.

1.4.4 Environmental Applications

It is possible to use sensor networks to monitor environmental changes in order to determine the values of parameters at a given location (such as temperature, atmospheric pressure, etc.). Indeed, the deployment of sensor nodes in nature can help detect events such as forest fires, storms, or floods. Consequently, more rapid and efficient responses are possible. Likewise, the deployment of sensor nodes in urban areas can assist in detecting pollution and analysis of air quality.

1.4.5 Industrial Applications

A wide range of industrial applications utilizes WSNs, reducing human interaction at industrial locations and improving safety and reliability [73]. WSNs have two primary uses in industries: safety hazards detection and equipment failures detection [56] During the equipment failures detection, sensors are installed inside the equipment and measure parameters such as temperature or vibration. These measurements are used to predict when a machine is likely to malfunction and take preventative measures. In the safety hazards detection, many industries use sensor networks to ensure compliance with regulations and keep employees safe [112].

1.5 Technologies based on IEEE 802.15.4 standard

IEEE 802.15.4 standard [1] is defined and developed for short-range wireless technologies and aims to operate data communication devices in Low Rate Wireless Personal Area Networks (LR-WPANs). This standard offers low-cost, low data rate communication, short-range, and low-power in sensor networks. Therefore, WSN applications are a target for the 802.15.4 standard [82]. Based on the OSI model, this standard uses only the first two layers: the physical (PHY) and medium access control (MAC) layers. The upper layers of the OSI model are separately defined by other architectures such as ZigBee [3], 6LoWPAN [91], WirelessHART [57], and

TABLE 1.2: Key IEEE 802.15.4-based WSN standards [82].

Standard	**Topology**	**Battery life (days)**	**Max data rate**	**Range (m)**
ZigBee	star, mesh	100–1000+	250 kbps	10–100
6LoWPAN	mesh	100–365+	250 kbps	10–100
WirelessHART	star, mesh	760+	250 kbps	10–100
ISA100.11a	star, mesh	1000+	250 kbps	10–100

ISA100.11a [7]. A technical comparison of IEEE 802.15.4-based WSN standards is presented in Table 1.2.

1.5.1 ZigBee

The ZigBee standard was developed by the ZigBee Alliance [3], which is an association of companies collaborating to enable reliable, cost-effective, low power, wirelessly networked monitoring and control products based on an open global standard. ZigBee is a short-range wireless technology based on the IEEE 802.15.4 standard. ZigBee protocol stack consists of four layers: PHY, MAC, the network, and the application layers [33]. PHY and MAC layers are defined by the IEEE 802.15.4 standard, while the ZigBee specification determines the remaining layers. ZigBee nodes are unable to communicate with IP-based networks without an 802.15.4/IP gateway. Therefore, ZigBee is well suited for WSN applications that do not require the use of IP-based communications. Furthermore, ZigBee system cannot communicate with networks having different standards.

1.5.2 6LoWPAN

6LoWPAN [91] is an adaptation layer specified and standardized by the IETF (Internet Engineering Task Force) working group to address the issue of integrating WSNs on the Internet. 6LoWPAN refers to IPv6 over Low power Wireless Personal Area Networks. It renders possible the communication of IPv6 packets over IEEE 802.15.4 enabled WSNs and provides a lightweight solution compliant with IP-based standards. In fact, 6LoWPAN features such as header compression and packet fragmentation reduce the communication costs and fit IPv6 packets within IEEE 802.15.4 frames where the network's MTU (Maximum Transmission Unit) is about 127 bytes as opposed to IPv6's minimum MTU of 1280 bytes. In addition, 6LoWPAN can compress a typical 40-byte IPv6 header to 2 bytes [33].

1.5.3 WirelessHART

WirelessHART [57] is an 802.15.4-based wireless communications standard that HART Communication Foundation developed as an open standard. This technology is used in industrial applications such as process measurement and control applications [82]. A WirelessHART system consists of several main components, including a network manager, gateways, access points, and field devices. At industrial plants, field devices are used for data acquisition and routing, and access points are used to transfer the acquired data to gateways. Furthermore, the network manager is responsible for configuring the network, scheduling communication between devices, managing routes, and monitoring network health. The WirelessHART standard can support star, mesh, and a combination of star-mesh network topologies. Nevertheless, WirelessHART operates as a standalone system, as with ZigBee. Therefore, WirelessHART cannot communicate with networks with different standards unless a particular gateway is employed [50].

1.5.4 ISA100.11a

ISA 100.11a [7] is an open standard for wireless communication networks was defined by the International Society of Automation (ISA), a U.S. non-profit organization, which addresses the disadvantages of wired communications in existing industrial control systems. ISA 100.11a is based on 802.15.4, characterized by low-power, low-rate wireless networking. Moreover, it is compliant with 6LoWPAN to handle IPV6 traffic,and compatible with other communications standards.ISA 100.11a belongs to a family of wireless standards for industrial automation that address the needs of industrial applications. Moreover, it can support star, star-mesh, or mesh topologies. ISA 100.11a system consists of field devices, gateways, and handheld devices. Sensor data is collected by field devices, and some of them provide routing functionality as well. The gateways are responsible for interfacing the WSN with the user application as well as supporting interoperability with different standards. In addition, handheld devices facilitate device installation, configuration, and maintenance.

1.6 Challenges in WSNs

In this section we will briefly describe the main challenges in WSNs [2, 82, 78, 53]. These include resource constraints, sensor platform price, scalability, dynamic network topology, node failure, environment conditions, unreliable wireless communication, prone to node failures, and security.

1.6.1 Resource constraints

Sensor nodes are typically tiny, with limited resources such as computational ability, storage capacity, communication bandwidth, and battery-mounted. The limited energy supplies of the sensor nodes in the network impose lifetime constraints on the WSN. The problem of limited resources can be addressed by using them efficiently. Indeed, The implementation of energy-efficient protocols is required to maximize the network lifetime. We mention, for instance, energy-aware routing on the network layer and energy-saving mode on the MAC layer. Efficient use of limited memory in sensors is required by taking into account the memory-consuming issues such as routing tables, data replication, and security.

1.6.2 Sensor platform price

Due to their high price, the available sensor platforms on the market cannot be widely adopted. Furthermore, producing cheaper and disposable sensor platforms makes a challenge [78].

1.6.3 Scalability

Theoretically, the number of sensor nodes deployed in a network to monitor a phenomenon can be in the order of thousands or even millions. Even so, the current real-world WSNs are capable of accommodating tens to hundreds of nodes [78]. Consequently, it is necessary to prove that the available theoretical solutions are suitable for large-scale WSNs.

1.6.4 Dynamic network topology

There are three possible scenarios responsible for changing the sensor network topology and making it unstable.

- *Node mobility:* The sensor network's topology may change frequently and be unstable. This is due to the mobility of network nodes, which are equipped with moving objects (mobilizers) that facilitate their movement within the network.

- *Node failure:* The high probability of failure of a sensor node means that the network's topology must change dynamically. In addition, factors such as obstructions and interference may disrupt communication between neighboring nodes and cause links to be broken. Thereby requiring costly reorganizations.

- *Deployment new nodes:* New sensor nodes can also be deployed to an existing network to address shortcomings and replace broken down or destroyed nodes.

In the previously mentioned scenarios, the sensor network must reconfigure quickly and with a reduced energy cost.

1.6.5 Environment conditions

Typically, WSNs are deployed in open areas and are not constantly monitored while left unattended. Due to this, sensor nodes are easy to access physically by anyone. Additionally, WSNs operate in harsh and even hostile environments, such as extreme weather conditions and natural disasters, leading to their failure and performance degradation.

1.6.6 Unreliable wireless communication

Communication in WSN is unreliable due to error-prone wireless medium with high bit error rates and variable link capacity. Consequently, a WSN must be reliable to function correctly, according to the application's requirements. Sensed data should be delivered reliably to the BS.

1.6.7 Prone to node failures

In wireless networks, nodes are commonly subject to unexpected failures for a variety of reasons, such as running out of energy or being damaged. In addition, communication between two nodes can be permanently interrupted. This requires WSNs to be robust to node failures. Therefore, WSNs can achieve improved fault tolerance by deploying more nodes than are required.

1.6.8 Security

Due to the nature of deployment of sensor devices that are usually in unprotected or even hostile areas, most WSN applications require a high-security level to provide the basic security requirements and to make these applications invulnerable to different cyber-attacks, preventing an intruder from disrupting the good operation of the network by taking control of sensor nodes. Another critical issue that affects the use of WSN is communications security. By its nature, the communications over wireless channels are insecure, and through this vulnerability, the intruder can eavesdrop and alter the messages exchanged between nodes. Moreover, the resource-constrained nature of sensor devices makes it impractical to apply conventional security schemes in WSN, which require a high overhead of computation, communication, and memory storage. Therefore, security in WSN is considered a challenging task.

1.7 Conclusion

Wireless sensor networks represent an emerging technology that has attracted researchers and developers in many fields through their particular characteristics. However, sensor networks are subject to a variety of challenges and limitations that researchers must consider. This chapter has described the main concepts related to wireless sensor networks, such as architecture, applications, and main facing challenges and limitations. The next chapter will present a detailed view of security and cryptography in wireless sensor networks.

Chapter 2

Security and Cryptography in WSN: A Background

Chapter Overview: *In this chapter, several background concepts are presented that are useful for the lecture of this thesis. The first part of the chapter discusses wireless sensor networks security, including security constraints, security requirements, and potential cyber-attacks that may target wireless sensor networks. The second part of the chapter reviews the cryptographic notions and primitives, as well as the computational assumptions used in this thesis.*

2.1 Security constraints

WSNs are subject to several constraints that make the security schemes proposed for Ad hoc networks inapplicable at their level. Therefore, these schemes need to be adapted to the characteristics of this particular type of wireless network. Developing reliable security schemes in WSN requires an understanding of the following constraints [14, 106].

2.1.1 Resource limitations

For any security approach to be implemented, a certain amount of resources are required, including processing power, data storage, code space, and energy to power the sensor. Unfortunately, these resources are very limited in a tiny wireless sensor.

- *Limited amount of energy:* The most significant constraint to wireless sensor technology is energy. We assume that sensor nodes cannot be easily replaced or recharged upon deployment into a sensor network. Therefore, it is imperative to conserve battery power in the field to prolong the life of each sensor node and the entire sensor network. It is also necessary to consider the energy impact of adding security codes to a sensor node when implementing cryptographic functions or schemes.

- *Limited processing capability:* Sensor nodes micro-controllers are slow and unable to perform certain arithmetic operations. This makes it impossible to perform very complex security schemes. operations.
- *Limited storage capability:* Sensor devices have an extremely limited amount of memory (Just a few kilobytes). Consequently, any security scheme designed for sensor networks should consume as little memory as possible.

2.1.2 Unreliable communications

Unreliable communications certainly pose a threat to sensor security. The security of a sensor network relies significantly on security schemes, which in turn, depend on wireless communication. Indeed, the communications nature over wireless channels are insecure, and through this, any transmission can easily be intercepted, altered, or re-transmitted by an adversary. The malicious node can also negatively affect data packets by causing collisions and interference in the communication channel. Furthermore, wireless communications involve high energy costs (one transmitted bit is equivalent to approximately 1,000 micro-controller operations [42]). Due to this, complex security schemes that entail an exchange of numerous messages between sensor nodes are not feasible.

2.1.3 The Unguarded Environment

Sensor nodes may be deployed in an open, hostile, and unattended environment, making them susceptible to physical attacks. Consequently, sensor nodes may be compromised or destroyed by adversaries. Indeed, an adversary can control a node in the network after deployment and physically damage it. Consequently, a sensor node becomes non-functional. Moreover, critical information, such as cryptographic keys can be retrieved from a captured node. The adversary can also modify the data acquired by the sensor node, thus performing various attacks. Even though most sensor networks contain a BS, its role is generally limited to data collection and query distribution, and no monitoring is performed.

2.2 Security Requirements

Due to the nature of deploying sensor devices that are usually in unprotected or even hostile areas, most WSN applications require a high-security level. This is to provide the basic security requirements and make these applications invulnerable to different cyber-attacks, preventing an intruder from disrupting the good operation of the network and taking control of sensor nodes. Basic security requirements for WSNs include [21, 14, 106]:

2.2.1 Data confidentiality

Data confidentiality represents one of the basic security requirements in sensor networks. The service guarantees that the information has not been disclosed and that only authorized parties have access to it. Cryptographic techniques are the typical countermeasure to confidentiality threats.

2.2.2 Data integrity

This requirement refers to ensuring that messages are not modified while being routed through the network, either intentionally or accidentally. In such a case, the receiver can verify that the message received matches the message sent by the sender.

2.2.3 Authentication

It must always be possible to verify the sender's identity of any message exchanged in the network. Consequently, the confidentiality and integrity of the exchanged messages cannot be guaranteed if communication with the correct node is not assured. A weakened authentication process could allow an adversary to gain access to the network and inject false information.

2.2.4 Availability

Network services should be available even if there are cyber-attacks or malfunction in a part of the network. In fact, WSNs are service-oriented networks, meaning they are specially designed to provide a well-defined, often quite critical, service. Therefore, even if a sensor network is the attack target, it must resist as much as possible and maintain the availability of its resources and services.

2.2.5 Data freshness

Data freshness indicates that the data is recent, and it ensures that an adversary cannot re-transmit old messages. In order to address this security requirement, a nonce, or another time-related counter, can be added to the packet.

2.2.6 Non-repudiation

In cybersecurity, non-repudiation refers to the ability to verify that the sender and receiver are in fact the parties who claim to have sent or received the message. Therefore, non-repudiation of data origin proves that data was sent. Non-repudiation of their arrival, on the other hand, confirms they were received.

2.3 Cyber-attacks in WSN

Network security is the set of policies, mechanisms, and services that protect a network from cyber-attacks and unauthorized access [109]. Security in WSN faces several challenges, especially when it comes to applications requiring a high level of security, such as military, emergency response, and healthcare [9, 45]. 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 [47, 14]. Cyber-attacks against WSN can be classified into three categories: *attacks on confidentiality*, *attacks on reliability of traffic data*, and *attacks on availability*. Figure 2.1 shows the classification of cyber-attacks in WSN.

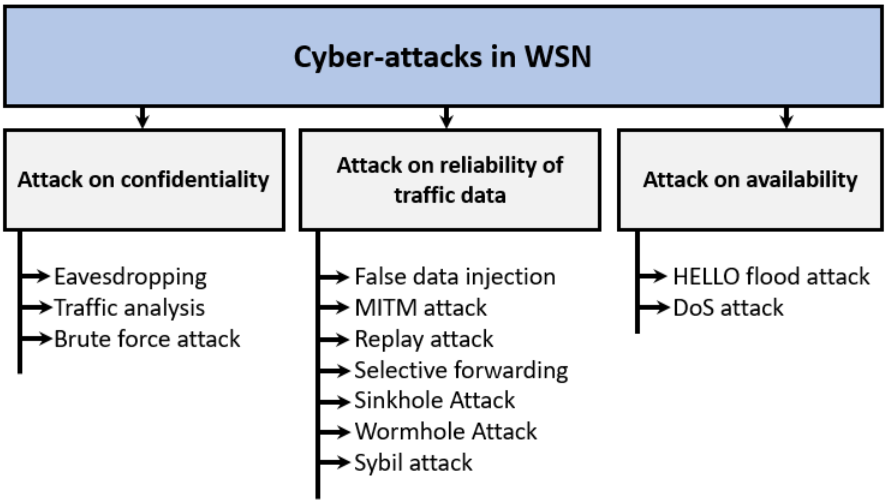


FIGURE 2.1: Taxonomy of security attacks in WSN.

2.3.1 Attacks on confidentiality

It concerns the passive adversary, which can eavesdrop on messages exchanged between sensor nodes through a communication channel. Thus, it can recover the content of messages circulating in the network. In this case, if the exchanged messages are encrypted, the adversary attempts to guess by testing many potential keys until the correct secret key is discovered.

- **Eavesdropping:** This cyber-attack is the easiest to implement among all attacks against data confidentiality. The adversary can listen to messages being exchanged between nodes. Consequently, he can capture strategic information that may be used to launch more harmful attacks.

- **Traffic analysis:** The adversary can determine sensor nodes with special and important roles in the network by traffic analysis. For instance, an increase in the number of messages exchanged between sensors indicates the presence of specific activities and events that should be monitored. Additionally, the adversary can determine the CH nodes without understanding the contents of the messages [21, 14].

- **Brute force attack:** To decrypt messages exchanged during data transmission, an adversary tests a large number of potential keys (guessing) to discover the correct keys.

2.3.2 Attacks on reliability of traffic data

The adversary can try to inject erroneous data into the network, replay previous messages and alter messages transmitted by sensor nodes in order to falsify the final result.

- **False data injection attack:** In the aggregation process, a malicious node transmits random false data to the targeted CH in order to falsify the aggregation result. As a result, the CH accepts the data transmitted by the malicious node and aggregates it. Thus, the final result is necessarily wrong, as shown in Figure 2.2. The countermeasure to this attack is node authentication [11] or end-to-end encryption [105], which prevents injecting fake data packets or modifying packet contents.

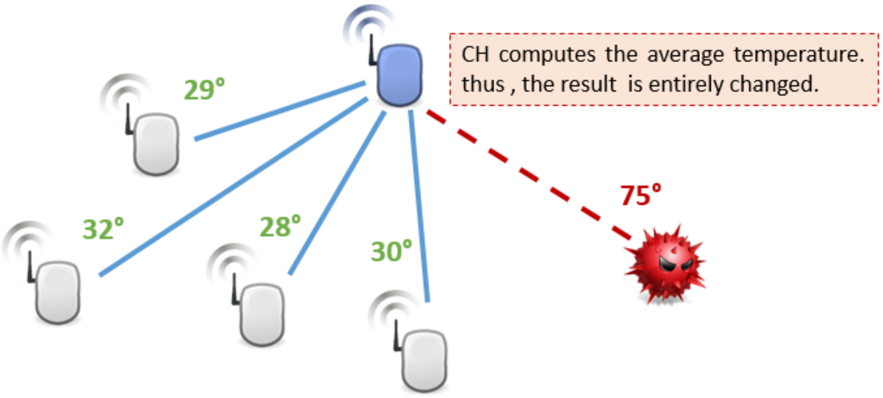

CH computes the average temperature.
thus , the result is entirely changed.
29°
32°
28°
30°
75°

FIGURE 2.2: Example of a false data injection attack.

- **MITM attack:** Its name implies that the adversary is placed between the transmitter and the receiver. In the case of WSNs, the attacker is a malicious node that is inserted between two legitimate nodes that communicate. The malicious node controls communication between the two victims while the parties believe they are directly communicating. In the literature, the man-in-the-middle attack is used to violate security requirements such as authentication, integrity, and non-repudiation [69].

- **Replay attack:** This is a classic attack. It involves retransmitting old messages several times within the network. Such an attack can, for example, be used to retransmit the traffic within a network of military sensors deployed in order to capture the movements of enemy troops. A replay attack could therefore cause combat units to be misdirected [14].

- **Selective forwarding Attack:** An adversary inserts or compromises nodes in a network so that they refuse to forward certain packets from neighbouring nodes. Usually, packets are chosen based on certain criteria such as the packet's content, the sender's address, or at random. According to the Figure 2.3, a malicious node transmits all packets except those that originate from a sensor node 1, based on the originating address.

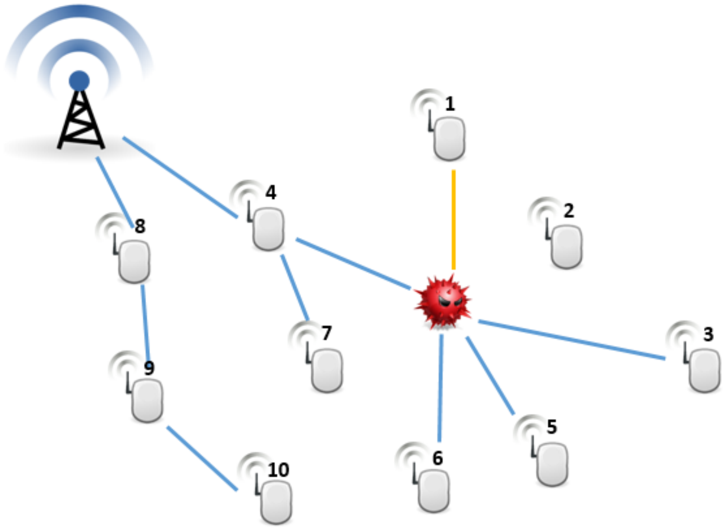


FIGURE 2.3: Example of a selective forwarding attack.

- **Sinkhole Attack:** In this attack, all traffic from a specific area is diverted by a malicious node in the false routing metric. Therefore, neighboring nodes believe a high-quality path exists and start forwarding the packets to the malicious node. The process of gathering traffic is called a sinkhole attack. It aims at harming the data at the collection point, compromising the integrity and reliability of the data sent by the network's sensor nodes [20].

- **Wormhole Attack:** This attack involves the adversary using a low-latency channel to route traffic between two malicious nodes which are not physically close to one another. Such a low-latency tunnel will likely increase the probability of being selected as an active path. The wormhole attack can launch a sinkhole attack where all traffic is routed via a malicious node, considering it the node closest to the BS, as shown in Figure 2.5.

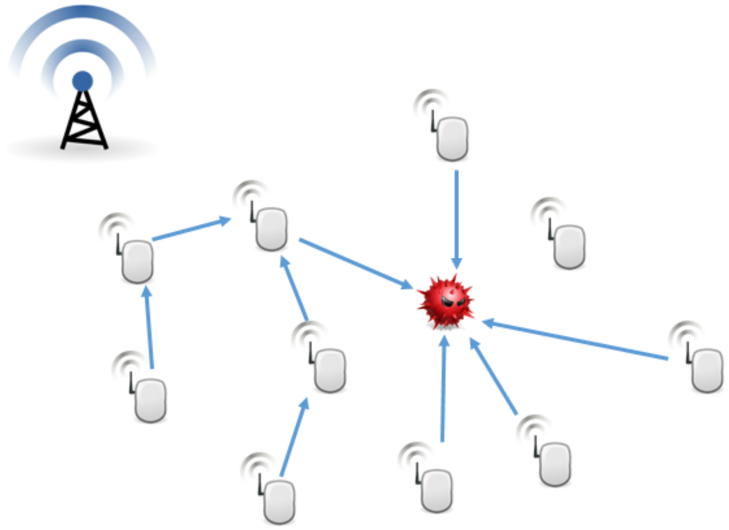


FIGURE 2.4: Example of a sinkhole attack.

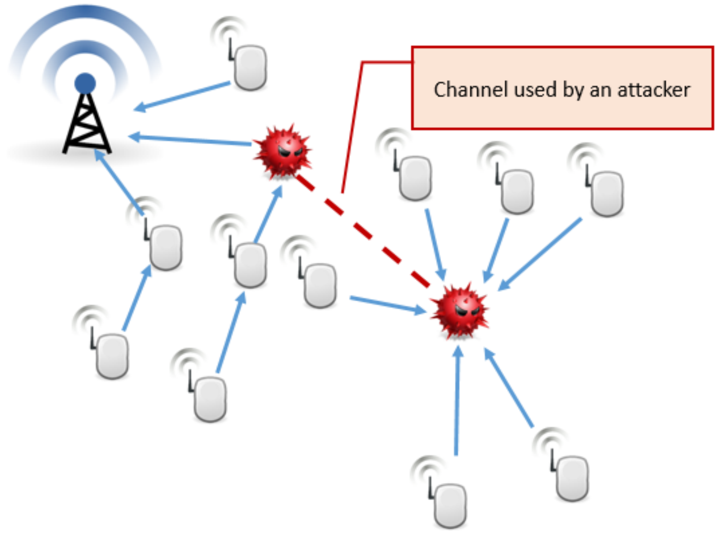


FIGURE 2.5: Example of a wormhole attack.

- **Sybil Attack:** In this attack, an adversary can degrade the effectiveness of several features such as the distribution of data, aiming to change the data integrity and routing mechanisms. Indeed, a malicious node collects multiple identities in the network, either by manufacturing or by theft of the identity of legitimate nodes. Consequently, the malicious node can exploit these multiple identities in order to be selected as the CH, or to create routing paths in order to benefit itself.

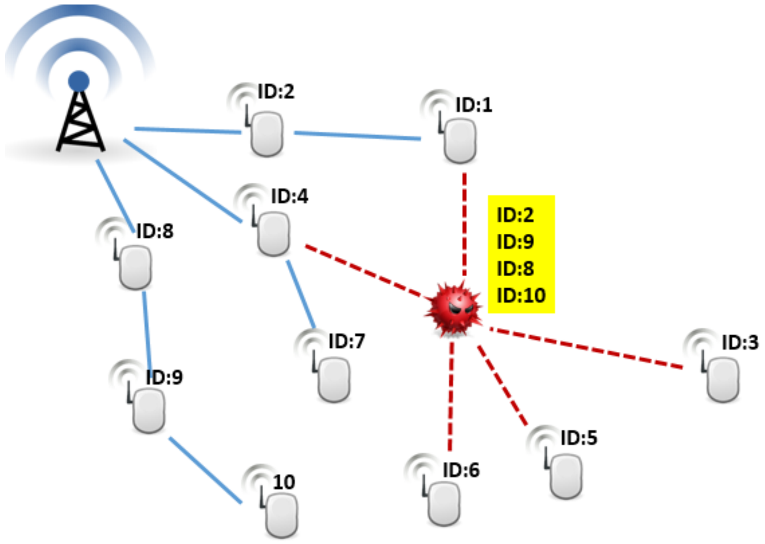


FIGURE 2.6: Example of a sybil attack.

2.3.3 Attacks on availability

The adversary may attempt to make WSN services unavailable by conducting specific security attacks.

- **Denial of Service (DoS):** The Denial of Service (DoS) attack targets the availability of network services, which can have serious consequences, especially for WSNs applications where nodes are resource-constrained. DoS attack can be performed by internal or external malicious and compromised nodes to the network. In this attack, the adversary acts as the attack manager, controlling many nodes, whether malicious or compromised nodes, to send a high volume of dummy generated messages to the target node, such as CH. This makes the target node exhaust all its resources and then is unavailable. As a result, the targeted node cannot provide services to legitimate nodes.

- **HELLO Flood Attack:** Sending HELLO packets is a common technique for discovering neighbors. Indeed, a sensor node assumes that the reception of such a packet implies the presence of the transmitter in its communication range. Therefore, a laptop-class adversary with a high-powered antenna sends a flood of HELLO messages to sensor nodes. The remote node receiving this message believes that the adversary is a neighbor and within the range of communication. Hence it tries to send its messages directly to the adversary leading

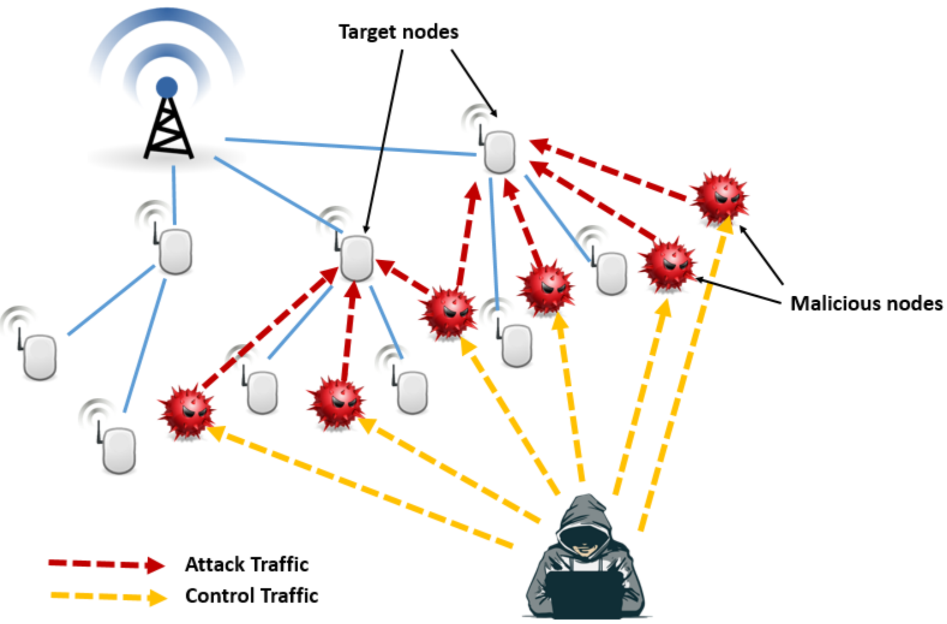


FIGURE 2.7: Example of a Denial of Service (DoS) attack.

to failure of messages transmission and disrupting the network operation by preventing other messages from being exchanged.

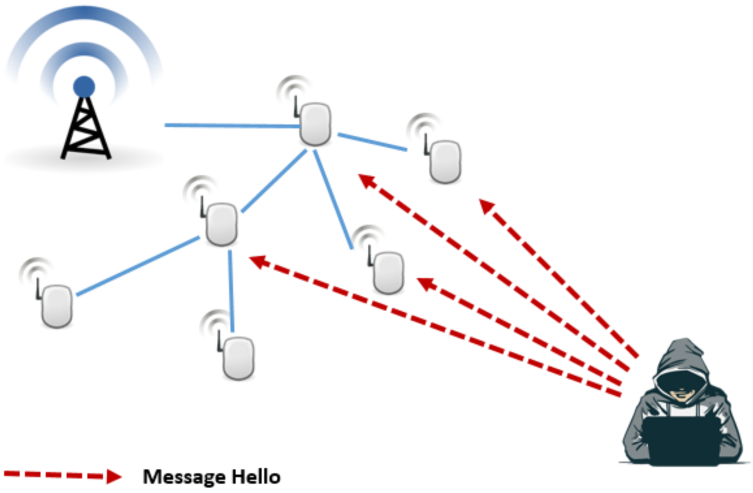


FIGURE 2.8: Example of a Hello flood attack.

Table 2.1 summarizes the security attacks in WSN as well as the corresponding countermeasures and the affected security requirements.

2.4 Cryptographic techniques in WSNs

Cryptography is considered a security countermeasure to protect communication

TABLE 2.1: Summary of cyber-attacks in WSNs and corresponding countermeasures.

Attacks	**Compromised requirements**	**Countermeasures**
Eavesdropping	Confidentiality	- Encrypt data which has paramount importance [69].
Traffic analysis	Confidentiality	- Randomized communications [21] - Encrypt data which has paramount importance [69].
Brute force	Confidentiality	- Use strong encryption and key generation algorithms unbreakable within a reasonable running time [69].
False data injection	Authentication Integrity	- End-to-end encryption [105] - Neighbor authentication [11]
MITM	Authentication Confidentiality Integrity non-repudiation	- Use a strong authentication methods such as digital certificates and zero-knowledge [69].
Replay	Freshness	- Timestamp and nonce options are added to messages [101].
Selective forwarding	Authentication Integrity Availability	- Authentication, IDS, multi-path routing [21].
Sinkhole	Authentication Integrity Confidentiality Non-repudiation Availability	- Giving each node a certificate [55] - Geographic routing [54]
Wormhole	Authentication Integrity Confidentiality Non-repudiation Availability	- Neighbor authentication [101] - Geographic routing [54] - Packet leash mechanism [44]
Sybil	Authentication non-repudiation	- Neighbor authentication, ID-based encryption, and symmetric-key techniques [55, 48]
DoS	Availability	- Encryption algorithms, Monitoring [55, 93]
Hello flood	Availability	- Neighbor authentication [54] - Bidirectional link verification mechanism[32] - Signal strength [21]

in open networks such as WSN. The security requirements, including data confidentiality, data integrity, authentication, and non-repudiation, are ensured by cryptography. Selecting appropriate cryptographic primitives is vital in WSNs. Thus, cryptography-based schemes used in WSN should consider resource-constrained sensor nodes. Additionally, these schemes should be evaluated regarding energy consumption, processing time, code size, and data size [94].

Cryptographic techniques can be divided into two categories: Symmetric-Key Cryptography (SKC) and Public-Key Cryptography (PKC). Using SKC offers good performance in terms of computational overhead and energy consumption. However, it does not support non-repudiation, and the distribution of keys is complex. PKC addresses these issues. It provides a more flexible interface and eliminates the need for key pre-distribution and pairwise key sharing. For resource-constrained sensor nodes, it is well known that PKC is computationally expensive [114, 94]. However, recent papers [15, 72, 37, 31, 61, 85, 75, 62, 94] showed that it is feasible to apply PKC in WSN using elliptic curves.

2.4.1 Symmetric-key cryptography

SKC is also known as secret-key cryptography. The sender and the receiver only share a secret key at the beginning of the process, and then they can encrypt and decrypt messages between them using that key. Advanced Encryption Standard (AES) [40] is one of the most well-known symmetric algorithms.

We define a SKC scheme in the following Definition 2.1.

Definition 2.1 (SKC scheme)**.** A SKC scheme with the input security parameter k, is defined by a pair of two deterministic algorithms (Enc, Dec) as follows:

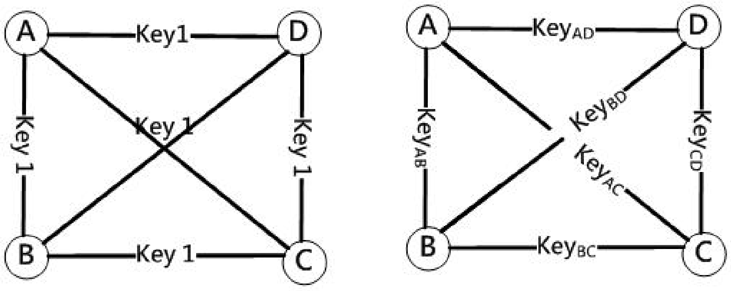
- $Enc(K, M) \to C$. This is the encryption algorithm that takes as input a key K of the set of keys $\mathcal{K}$, a message M from the set of plaintexts $\mathcal{M}$ and outputs an encrypted message C from the set of ciphertexts $\mathcal{C}$.

- $Dec(K, C) \to M$. This is the decryption algorithm that takes as input the same secret key K, the ciphertext C and outputs the message M.

It is required that the equation $Dec(K, Enc(K, M)) = M$ holds for every $K \in \mathcal{K}$ and $M \in \mathcal{M}$.

To make SKC work in WSNs, a shared-key distribution is necessary. We can divide key distribution methods into three categories:

- ***A Global key (or network key)***: A simple technique of distributing keys is to use the same key for all sensor nodes (See Figure 2.9 (a)). In this case, all sensor nodes are pre-loaded with the global key, enabling any two nodes to communicate securely. However, If an adversary gains access to the global key, he can control the entire network.
- ***A pairwise key***: Another technique is using a secret key shared by two nodes (See Figure 2.9 (b)). If a node is compromised, the other nodes will not be affected. Using this technique makes the entire network will not be susceptible to attack. A WSN containing n nodes will require each node to store $n-1$ keys, and the entire network will need to store $n(n-1)/2$. Using this technique requires a large amount of memory if n is a large number.
- ***A group key***: Is a secret key that is shared between a group of nodes. If an adversary reveals the group key, he can easily compromise the all group of nodes.



(a) Global key. **(b)** Fully pairwise keys.

FIGURE 2.9: Cases of using a global key and pairwise key.

2.4.2 Public-key cryptography

PKC or asymmetric cryptography allows two parties to exchange data over an insecure channel while ensuring data confidentiality, non-repudiation, and authenticity. Unlike symmetric encryption, which relies on sharing a secret key between two communicating entities, PKC relies on a pair of keys to protect exchanged data.

Such key pair comprises a public key and a private key related by a mathematical equation. Solving this mathematical equation breaks a hard mathematical problem such as the Discrete Logarithm Problem (DLP). Each communicating entity shares its public key. In contrast, the corresponding private key must remain confidential. Figure 2.10 illustrates the encryption and signature processes in PKC.

ECC and RSA are mature public-key cryptographic algorithms that the academic community has researched for many years. Rivest, Shamir, and Adleman designed RSA in 1977 [83]. While, Koblitz and Miller independently proposed ECC in 1985 [59], [74].

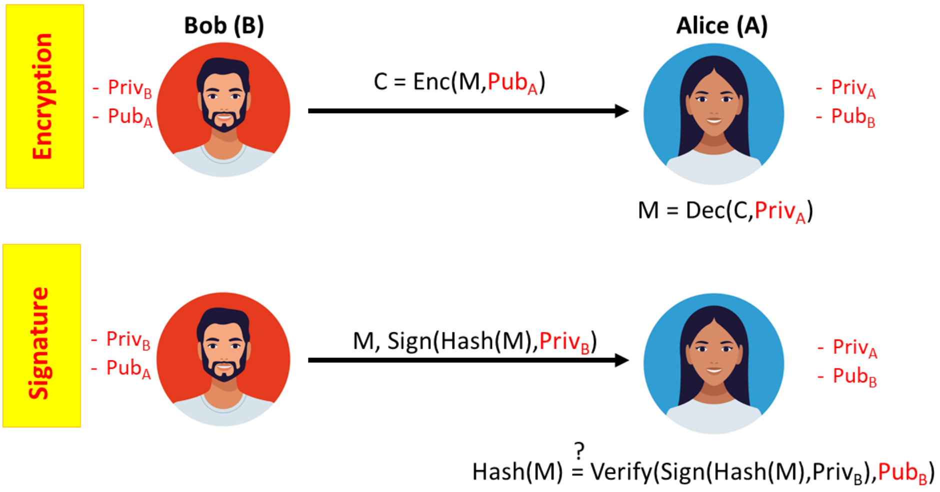


FIGURE 2.10: Public key cryptography.

Public-key infrastructure: In PKC, a public key must be authenticated, i.e., it will be necessary to provide a way of verifying the validity of the public key. Otherwise, the communication between nodes is exposed to a MITM attack. Public Key Infrastructure (PKI) solutions, which aim to ensure that the public key is authenticated, use public key certificates (also known as digital certificates). Such certificates are issued by a trusted entity called Certificate Authority (CA). With PKI, an entity's public key is signed by the CA. The digital certificate contains the public key and signature of the CA. When communicating, the recipient needs to know the sender's digital certificate and the CA's public key to authenticate the sender and verify the message. Digital certificates can be stored in the recipient in advance or retrieved on the fly from CA or a centralized certificate repository.

Unfortunately, PKI is impractical to apply in WSNs, as it would introduce overhead and complexity of certificate operations, including distribution, storage, and certificate verification [94].

2.5 Elliptic Curves Cryptography

ECC is a public-key cryptography algorithm based on elliptic curves over a finite field. Such a cryptosystem can be used for asymmetric operations such as key exchange on a channel non-secure or asymmetric encryption. ECC has attracted much attention as a means of security for resource-constrained environments. It provides the same level of protection as the RSA cryptosystem but with a smaller key size. For example, a 160-bit ECC key provides security equivalent to a 1024-bit RSA key [67]. This helps reduce computing time, save energy and save memory [24]. Table 2.2 shows a comparison of the key sizes between RSA and ECC while ensuring the same level of security. In the following, the basics of ECC are given.

TABLE 2.2: RSA and ECC key length equivalence for the same security level [60]

Security level	80	112	128	192	256
RSA key length (bits)	1024	2048	3072	7680	15360
ECC key length (bits)	160	224	256	384	512

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 [79]: $y^2 = x^3 + ax + b$, 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.

2.5.1 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 $Q = kP$ [36].

- *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 $\mathbb{G}$ [36].

2.5.2 Well-known ECC schemes

This section describes well-known ECC schemes, including a key agreement (ECDH), public-key encryption (ECIES), and a digital signature (ECDSA). Suppose that E is the elliptic curve over a finite field $\mathbb{F}_q$ and G represents the generator point on the curve.

- **ECDH**: It is a key exchange mechanism based on elliptic curves helping two parties establish a shared secret key (Pairwise Key) through an open and insecure channel. Alice secretly selects an integer k_A (as private key) and computes the point $Q_A = k_A.G$ (as public key) which will be sent to Bob. In turn, Bob secretly chooses k_B and computes $Q_B = k_B.G$ that will be sent to Alice. Both parties compute the shared key $sk = k_A.Q_B = k_B.Q_A = k_A.k_B.G$.

- **ECIES:** It is a public-key encryption scheme based on ECC. The scheme is designed to be semantically secure in the presence of an adversary capable of launching chosen-plaintext and chosen-ciphertext attacks. Please refer to [16] for further details. Next, we demonstrate how to exchange a message using ECIES. To send an encrypted message to Bob using ECIES, Alice needs the following information:

  - The cryptography suite to be used, including Key Derivation Function (KDF), a Message Authentication Code (MAC), and a symmetric encryption scheme such as AES.
  - The Bob's public key $Q_B = k_B.G$, where $k_B \in \mathbb{Z}_q^*$ is the Bob's private key (randomly selected).
  - shared information S_1 and S_2.

  In Table 2.3, the steps that Alice and Bob must follow to encrypt and decrypt the message m. Note that $x(r.Q_B)$ returns the x-coordinate of an elliptic curve point $r.Q_B$

- **ECDSA:** It is a public-key digital signature scheme based on elliptical curves. Suppose Alice wants to send a signed message to Bob. First, Alice must create a private key $k_A \in \mathbb{Z}_q^*$ and a public key $Q_A = k_A.G$. Bob requires Q_A to verify

TABLE 2.3: ECIES scheme: encryption/decryption.

To encrypt a message m by Alice	**To decrypt a ciphertext $R\Vert c\Vert d$ by Bob**
1 - Picks a random number $r \in \mathbb{Z}_q^*$	1 - Computes $S = x(k_B.R)$
2 - Computes $R = r.G$	2- Computes $k_E \Vert k_M = KDF(S\Vert S_1)$
3 - Computes $S = x(r.Q_B)$	3 - Computes $\overline{d} = MAC(k_M, c\Vert S_2)$
4 - Computes $k_E\Vert k_M = KDF(S\Vert S_1)$	4 - Outputs failed if $\overline{d} \neq d$
5 - Computes $c = E(k_E, m)$	5 - If it holds, Bob decrypts the message:
6 - Computes $d = MAC(k_M, c\Vert S_2)$	$m = E^{-1}(k_E, c)$
7 - Send $R\Vert c\Vert d$ to Bob	

the authenticity of Alice's signature. In Table 2.4, the steps that Alice and Bob must follow to sign and verify Alice's signature.

TABLE 2.4: ECDSA scheme: signature/verification.

To sign a message m by Alice	**To verify Alice's signature by Bob**
1 - Computes $z = HASH(m)$	1 - Verify that $r \in \mathbb{Z}_q^*$ and $s \in \mathbb{Z}_q^*$
2 - Picks a random number $k \in \mathbb{Z}_q^*$	2- Computes $z = HASH(m)$
3 - Computes $\overline{X} = x(k.G)$	3 - Computes $u_1 = z.s^{-1} \mod q$
4 - Computes $r = \overline{X}$ mod q	4 - Computes $u_2 = rs^{-1} \mod q$
5 - If $r = 0$, go back to step 2	5 - Computes a point $X = u_1 \times G + u_2 \times Q_A$
6 - Computes $s = k^{-1}(z + r.k_A) \mod q$	6 - Computes $\overline{X} = x(X)$
7 - If $s = 0$, go back to step 2.	7 - The signature is valid if $r \equiv \overline{X}$ mod q
8 - Send (m, r, s)	

2.6 Pairing-Based Cryptography

PBC is an emerging trend in cryptography that is strictly related to ECC. The main idea behind PBC is to create a mapping between two cryptographic groups using a pairing function (denoted as $\hat{e}$). Indeed, the pairing is a mathematical concept introduced by mathematicians Weil and Tate, which is defined as follows: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ denote two additive groups of order q, and $\mathbb{G}_T$ is a multiplicative group of order q.

The bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ if it satisfies the following properties [18]:

- *Bilinearity*: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and all $a, b \in \mathbb{Z}_q^*$.
- *Non-degeneracy*: there exists $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ such that $\hat{e}(P, Q) \neq 1$.

• *Computability*: for all $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, an efficient algorithm exists to compute $\hat{e}(P, Q)$.

For their security, most pairing-based schemes depend on the hardness of the following problem:

- *Bilinear Diffie-Hellman Problem (BDHP)*: Given the points $P, aP, bP, cP \in \mathbb{G}_1$ and $\acute{P}, b\acute{P}, c\acute{P} \in \mathbb{G}_2$ where $a, b, c \in \mathbb{Z}_q^*$ are unknown, the computation $\hat{e}(P, \acute{P})^{abc} \in \mathbb{G}_T$ is hard in $\mathbb{G}_T$ [18].

Several pairing-based schemes were proposed for securing constrained devices such as sensor nodes in literature. However, those schemes are inefficient when they require two or more pairing operations. Moreover, according to recent implementation results on many sensor platforms, the time required to compute a single bilinear pairing is equal to the computation between two to seven elliptic curve point multiplications [94].

2.7 Identity-based cryptography

IBC is an extension of public-key cryptography introduced in [89]. 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 as shown in Figure 2.11. 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 [89], which is based on the RSA cryptosystem. While in [49], 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 [13] using a pairing over elliptic curves.

2.7.1 Identity-based encryption

Suppose Alice wants to send an encrypted message to Bob as shown in Figure

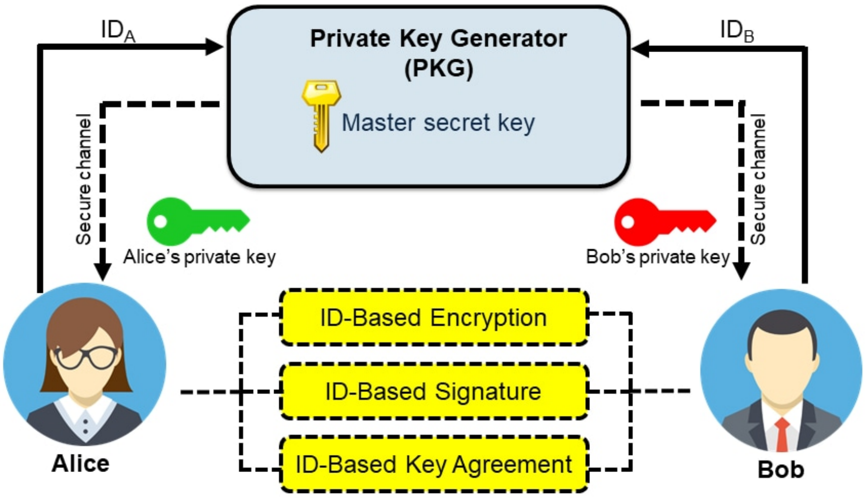


FIGURE 2.11: Identity-based cryptography concept.

2.12. First, the PKG picks a random number $s \in \mathbb{Z}_q^*$ as the master secret key, it after that computes the master public key $P_{pub} = s.G$ where G represents the generator point. P_{pub} is published while s is kept only in the PKG. To encrypt the message M, Alice uses Bob's identity id_B and P_{pub}. Alice then sends the ciphertext C to Bob. Upon receiving C, Bob obtains his private key $Priv_B$ from the PKG and uses it to decrypt the ciphertext C.

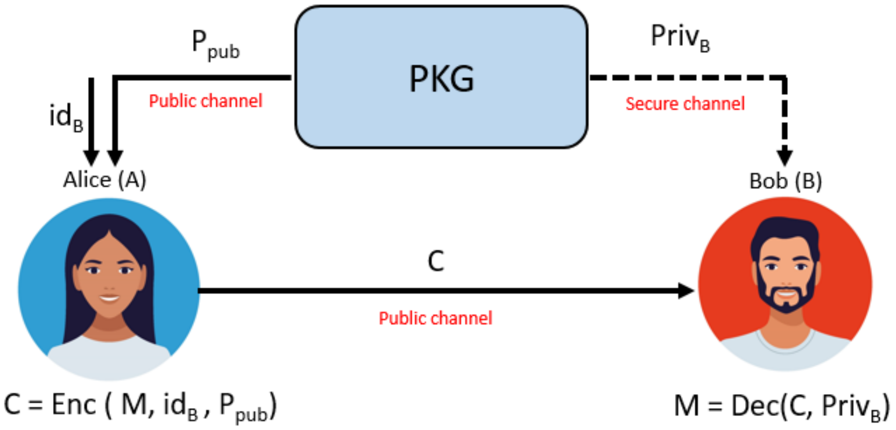


FIGURE 2.12: Identity-based encryption.

2.7.2 Identity-based signature

Suppose Alice wants to send a signed message to Bob. As illustrated in Figure 2.13, the basic IBS scheme operates as follows:

- The PKG generates the master secret key and the master public key (s, P_{pub}).

- Alice obtains her private key $Priv_A$ from the PKG.
- Using $Priv_A$, Alice signs the message M and sends it to Bob along with the signature σ.
- Upon receiving M and σ, Bob uses id_A and P_{pub} to check whether σ is a genuine signature on M. If it holds, Bob returns *Accept*. Otherwise, he returns *Reject*.

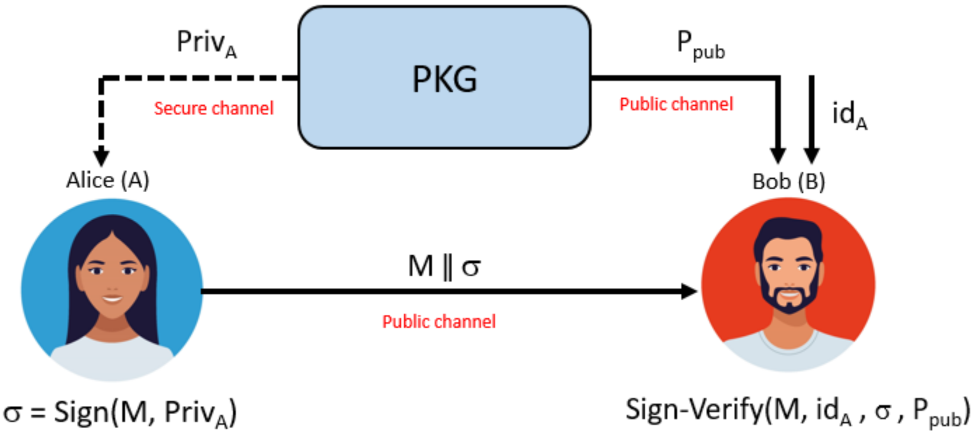


FIGURE 2.13: Identity-based signature.

2.7.3 IBC and WSNs

In the literature, IBC is suitable for devices with limited resources, such as sensor nodes [96, 58, 85, 35, 75, 61, 102]. 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. Indeed, the PKG generates and escrows the entity's private key. Upon compromising the PKG, it can impersonate any entity in the network. Therefore, the PKG must be an unconditionally trusted entity. However, it may be challenging to provide such a feature in many scenarios [77]. Fortunately, in the WSN scenario, the BS who plays the role of the network deployer is trustworthy. It is considered a laptop-class device with physical protection. 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, private keys must be delivered to the sensor nodes through secure channels. However, in the WSN 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.

Two cryptosystems are used in the literature to implement IBC-based schemes for sensor nodes: PBC and ECC (pairing-free). However, as mentioned in section 2.6, IBC schemes based on the pairing are computationally expensive when they require two or more pairing operations. The resource consumption might exceed the resource capability of resource-constrained devices. In addition, most of the pairing-based cryptosystems require a special hash function called the Map-To-Point hash function (MTP) for converting an entity's identity to a point on the elliptic curve. This function is also computationally expensive for small sensor devices. Thus, IBC schemes used the pairing are not suitable for resource-constrained sensor devices. IBC schemes implemented over ECC consider the perfect choice for sensor devices since they do not require any pairing computation and MTP function. Thus, IBC schemes based on ECC decrease computational overhead and prolong the network lifetime by reducing the energy consumption of the sensor device.

2.8 Conclusion

Security in WSN is considered a challenging task which faces this type of network. The dynamic nature of WSN and its deployment in open areas make these networks vulnerable to different kinds of cyber-attacks that can adversely affect their functioning. Moreover, the resource-constrained nature of sensor devices makes it impractical to apply conventional security schemes in WSN, which require a high overhead of computation, communication, and increased energy consumption. Cryptography is considered a security technique that protects communication in open networks such as WSN. The requirements of security are ensured by cryptography.

SKC-based solutions are highly efficient in terms of computation overhead and energy consumption. The distribution of keys, however, presents a significant challenge. Furthermore, such solutions do not achieve a good balance between resilience and the storage of cryptographic keys. First efforts to apply PKC in sensor networks proved that RSA is not feasible due to its large key size. In addition, its cryptographic primitives are costly in terms of computation.

Further investigation in this area has shown that ECC is a more suitable PKC method in resource-constrained devices due to its small key sizes and faster execution times. Nevertheless, to use PKC methods in WSNs, a public key must be authenticated. Using a PKI with certificates and digital signatures is not feasible in sensor networks. As a result, a PKC extension called IBC was developed. This chapter proposes the use of the IBC method in WSN. Compared to existing PKI-based solutions, IBC offers several advantages. It can provide easy management of public keys than PKI-based cryptosystems, and there is no need to generate and maintain public key certificates. Consequently, IBC requires low computational and communication overhead.

Chapter 3

PKC-Based Security in WSNs: A Review

Chapter Overview: *The objective of this chapter is to review existing PKC-based schemes in wireless sensor networks, including IBC schemes based on pairings, IBC schemes based on ECC, and schemes based on ECC. The chapter discusses the existing PKC-based schemes and highlights their shortcomings. Furthermore, it compares the different PKC-based schemes according to a number of criteria.*

3.1 Introduction

Comparing PKC with SKC method, PKC provides a simple solution, robust security, good scalability, and immediate message authentication. In addition, the studies conducted in this context have indicated that ECC is a more suitable PKC method in resource-constrained devices than RSA. This is due to ECC small key sizes and faster execution times. Therefore, several PKC-based schemes have been proposed in the literature for securing WSNs. In this chapter, we review and critically analyze the recent PKC-based schemes. We class PKC-based schemes for WSNs into three categories: IBC schemes based on pairings, IBC schemes based on ECC, and ECC-based schemes. At the end of the chapter, we compare the reviewed schemes based on several criteria. To better understand the notations used in the studied schemes, we list the notations and their corresponding descriptions in Table 3.1.

TABLE 3.1: List of notations

Notation	Description
$\mathbb{F}_q$	a finite field of order q
$E/\mathbb{F}_q$	Elliptic curve over $\mathbb{F}_q$
$\mathbb{Z}_q^*$	Multiplicative group of an integer modulo q
P	Generator point
id_j	Identity of a node j
msk	Master secret key
P_{pub}	Master public key
$\hat{e}$	Bilinear pairing function
t	Time stamp
H_{mtp}	Map to point hash function
h	Hash function
$Priv$	Identity-based private key
$\oplus$	XOR operation

3.2 IBC schemes based on pairings

Huang et al. proposed in [66] SET-IBS: a secure and efficient data transmission protocol for CWSNs based on identity-based settings. In this protocol, Identity-Based Signature (IBS) scheme [41] is applied to send messages as a way to guarantee the authenticity of the node sending the message. The authors also use a homomorphic encryption scheme [17] for end-to-end data encryption. The IBS scheme adopted for CWSNs consists of four algorithms: setup, key extraction, signature signing, and signature verification.

- *Setup*. The BS (serves as a PKG) determines the pairing parameters $\{E/\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}\}$. The BS picks a random number $msk \in \mathbb{Z}_q^*$ as the master secret key, it thereafter computes the master public key $P_{pub} = msk.P$. Then, two hash functions H_{mtp} and h are chosen. The BS Preloads each sensor node with the system parameters $param = \{E/\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, H_{mtp}, h, msk, P_{pub}\}$.
- *Key extraction*. A sending node with identity id_j generates its private key $Priv_j$, based on its identity id_j, msk and, a time stamp t_j, where $Priv_j = msk.H_{mtp}(id_j||t_j)$.
- *Signature signing*. A sensor node with identity id_j generates its signature σ_j using the encrypted message C_j, the time stamp t_j, and its private key $Priv_j$. Indeed, the sensor node picks a random number $\alpha_j \in \mathbb{Z}_q^*$ and computes $\theta =$

$\hat{e}(P,P)^{\alpha_j}$. The sensor node further computes the digital signature $\langle \sigma_j, c_j \rangle$, where $c_j = h(C_j||t_j||\theta_j)$ and $\sigma_j = c_j.Priv_j + \alpha_j.P$

- *Verification*. Upon receiving the message, a sensor node computes: $\grave{\theta}_j = \hat{e}(\sigma_j, P)\hat{e}(H_{mtp}(id_j||t_j), -P_{pub})^{c_j}$. If $h(C_j||t_j||\theta_j) = h(C_j||t_j||\grave{\theta}_j) = c_j$, the sensor node considers the received message authentic. Otherwise, it will reject the message.

SET-IBS includes an initialization phase prior to network deployment. During this phase, the BS generates and preloads pairing settings as well as secret keys (e.g., the master secret key and the homomorphic encryption key) in all sensor nodes. Following the termination of the initialization process, SET-IBS operates in rounds during communication, similar to other LEACH-like protocols. Each round consists of a setup phase to form clusters, followed by a steady-state phase for data transmission from the sensor node to the BS. The authors adopt $\langle id||t \rangle$ as a public key. The security in SET-IBS depends on the hardness of the Bilinear Diffie-Hellman Problem (BDHP). SET-IBS provides confidentiality, authentication, integrity, non-repudiation, and freshness of messages. In addition, it can prevent various cyber-attacks, including Hello flood, selective forwarding, and sinkhole attacks.

Limitations: The master secret key is distributed in all nodes in the SET-IBS protocol. This creates the key escrow problem, where any node can generate other nodes' private keys. Therefore, the security of the whole network to be threatened if one node has been compromised. Additionally, the SET-IBS protocol adds extra computational and storage loads to the sensor nodes, which have severe resource constraints. Furthermore, SET-IBS uses SHA-1 as a cryptographic hash algorithm, where this latter is no longer secure.

In [43] Shuaiqi Hu proposed a hierarchical key management scheme for WSN, which is based on ID-Based Encryption (IBE) and Diffie-Hellman (DH) algorithms. The communication overhead is reduced in [43] by managing a flat structure of WSN in a clustered manner. The design of the proposed scheme allows some sensor nodes to work as CHs. While the BS and the other sensor nodes negotiate to establish a session keys with cluster heads by using Boneh-Franklin IBE (BF-IBE) scheme [13] and

DH key exchange [22]. The session keys can be used later for secure BS-CH and CH-CM communications, and for data encryption. The key negotiation process consists of three basic steps: broadcast, parameter calculation, and parameter exchange. The details of each step are described as follows:

- *Broadcast.* The CH (denoted as A) broadcasts its identity id_A to the BS and cluster members (denoted as B).
- *Parameter calculation.* The node A chooses a secret integer $x_A \in \mathbb{Z}_q^*$ and generates $Y_A = x_A.P \bmod q$. Similarly, the node B chooses a secret integer $x_B \in \mathbb{Z}_q^*$ and generates $Y_B = x_B.P \bmod q$.
- *Parameter exchange.* The node B uses the IBE scheme to encrypt Y_B and id_B. The public key used for the encryption process is id_A. Upon receiving the encrypted message, the node A uses $Priv_A$ and the IBE scheme to decrypt the message and obtain Y_B and id_B. In contrast, the node A encrypts Y_A using id_B and sends the encrypted result to the node B. Both parties compute and store the symmetric key $K_{AB} = x_A.Y_B \bmod q = x_B.Y_A \bmod q = x_A.x_B.P \bmod q$.

Regarding the security aspect, the proposed scheme prevents the intruder to threaten the whole network by using the compromised nodes. Moreover, it has the ability to prevent specific cyber-attacks, including eavesdropping and MITM attacks.

Limitations: The use of BF-IBE scheme [13] to encrypt/decrypt a public key and insert the latter in the exchanged messages, increases the energy consumption, the communication and computation overhead. Indeed, to perform a single BF-IBE encryption/decryption operation, a node must compute: two bilinear pairings, one MTP function, one pairing-based exponentiation, and one scalar multiplication. Thus, encryption/decryption using BF-IBE are computationally expensive and unsuitable for resource-constrained devices.

Qin et al. [81] designed a novel identity-based security scheme for cluster-based WSN by adopting BF-IBE scheme [13]. The main idea of this scheme is to employ multiple PKGs in CWSN instead of single PKG. For this end, the authors assumed that each CH serves as PKG in its cluster to improve the security so that the adversary cannot compromise the entire network. The proposed scheme is divided into four main phases, namely initialization, clustering, parameter distribution and data collection. The details of each phase are described below.

- *Initialization*. Initially, the BS determines the parameters of the system $param = \{E/\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, H_{mtp}, r, P_{pub}\}$, where $r \in \mathbb{Z}_q^*$ is a random number. Each sensor device stores *param*. The BS also computes private keys $Priv_j = msk.H_{mtp}(id_j)$, and then preloads them into the sensor nodes.
- *Clustering*. The BS divides the entire network into n clusters by selecting n CHs. The authors assume that CHs have significantly more resources than other sensor nodes. then, the BS generates a set of keys:$\langle K1, K2, K3, ..., Kn\rangle$. K_i is considered a group key that can be shared between all CMs in the same cluster. K_i is distributed to each CH_i and the relationship between K_i and CH_i is stored in the list by the BS. Each CH serves as PKG in its cluster which chooses a random number $msk_{CH} \in \mathbb{Z}_q^*$ as a master secret key and generates its private key $Priv_{CH} = msk_{CH}.H_{mtp}(id_{CH})$. The cluster public key can be computed as $P_{CH} = msk_{CH}.P$.
- *Parameter distribution*. During this phase, the CH_i distributes P_{CH} and K_i to its CMs. At the beginning, CH_i computes $g_j = h(\hat{e}(H_{mtp}(id_j), P_{pub})^r)$ for every CM_j in its cluster, and chooses a random number φ to construct a polynomial $F(x) = \varphi(\varphi e)^{\prod_{j \in \Lambda}(x - g_j)}$ where e equals 2.718, and Λ denotes the set of CMs located in the cluster of CH_i. Afterwards, CH_i sets the encrypted message $C = (U \| V \| F(x))$, where $U = P_{CH} \oplus K_i$ and $V = \varphi \oplus K_i$. The encrypted message C is broadcast by CH_i to all its CMs. Upon receiving C, every CM_j computes $g_j = h(\hat{e}(H_{mtp}(id_j), P_{pub})^r)$ and substitutes the g_j value into the polynomial $F(x)$ as: $F(g_j) = \varphi$. Then CM_j gets $K_i = V \oplus \varphi$, and $P_{CH} = K_i \oplus U$.
- *Data collection*. The sensed data are encrypted and transmitted from CMs to CH. In this case, the authors encrypt the data using the IBE scheme. The result is then encrypted again using a symmetric key. K_i. For encryption, CM_j generates a random number t and calculates $g_{CH} = \hat{e}(H_{mtp}(id_{CH}), P_{CH})^t$. The ciphertext is $C = E_{K_i}(id_j \| W \| F)$, where $W = t.P$ and $F = m \oplus h(g_{CH})$. Upon receiving C, the CH_i decrypts ciphertext using K_i. Then, CH_i can obtain the plaintext $m = F \oplus h(\hat{e}(Priv_{CH}, W))$.

The proposed scheme can prevent certain cyber-attacks, such as Hello Flood, Sinkhole and Sybil attacks. Although the adversary compromises the group key K_i, he cannot obtain the plaintext data. The adversary requires access to both $Priv_{CH}$ and K_i keys to decrypt the ciphertext data.

Limitations: The proposed scheme is resource-inefficient for sensor nodes since it requires costly operations, including pairing computations, MTP hash function, and pairing-based exponentiation. If an adversary compromises the CH and obtains msk_{CH}, he will compromise the cluster security and generate the CMs' private keys. Thus an adversary makes the key escrow problem. Additionally, the BS has been used as reference to check the validity and authenticity of new node. This can result in generating a lot of messages that cause network congestion.

Mehmood et al. proposed in [68] a public key-based scheme called Inter-Cluster Multiple key Distribution Scheme (ICMDS) for WSNs. The authors assumed that a CH is an essential node in the network, and it is more vulnerable to cyber-attacks than other sensor nodes. Therefore, they focused on authentication and key management in clustered wireless networks. To secure inter-cluster communication, the BS encrypts a session key and transmits it to the CHs. The session key generation process is divided into four phases: setup, encryption setup, encryption, and decryption.

- *Setup*. The BS determines the system parameters $\{E/\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, H_{mtp}, h\}$. Next, it computes $P_{pub} = msk.P$ and delivers it to CHs. Then, each CH_i computes $Priv_i = msk.H_{mtp}(id_i)$.
- *Encryption setup*. The BS selects a random number $r \in \mathbb{Z}_q^*$ and computes $R = r.P$. The BS also computes $x_i = \hat{e}(r.H_{mtp}(id_i), P_{pub})$ and $(a_0.P, a_1.P, ..., a_m.P) = (P_0, P_1, ..., P_m)$ where m denotes the number of CHs. The reference [68] explains how to calculate $a_0, a_1, ..., a_m$ to achieve $\sum\limits_{j=0}^{m} a_j x_i^j = 0$.
- *Encryption*. The BS generates a session key $S_k \in \{0,1\}^*$ and selects $R \in \mathbb{Z}_q^*$ and $D \in \mathbb{G}_1$ two random numbers. Next, the BS calculates $T = (R, S_k \oplus h(D), D + RP_0, RP_1, ..., RP_m) = (R, C, D, C_0, C_1, ..., C_m)$. Then, it broadcast T to CHs.
- *Decryption*. To decrypt the corresponding session key, each CH computes the following:

1) $\hat{e}(Priv_i, R) = \hat{e}(msk.H_{mtp}(id_i), r.P) = \hat{e}(r.H_{mtp}(id_i), P_{pub}) = x_i$
2) $C_0 + \sum_{j=1}^{m} x_i^j C_j = D + R(a_0 + a_1 x_i + a_2 x_i^2 ... + a_m x_i^m)P = D$
3) $C \oplus h(D) = S_k$

The proposed ICMDS makes the contents of messages unreadable for the intermediate nodes. Moreover, the CH of each cluster mainly performs the security function.

Limitations: The proposed 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. To decrypt the session key, each CH computes one pairing operation, *m* pairing-based exponentiation, and *m* point multiplications (*m* denoted the number of CHs). Therefore, CH's computational overhead is high, and it may exceed CH's limited resource capability. Moreover, the CH receives and stores *T*, which is very large. Because of this, CH incurs high communication and storage costs.

An attempt to overcome security weaknesses of ICMDS [68] is made by Harbi et al. in [37]. An enhanced scheme was introduced 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. The proposed MAKA consists of five phases: initialization, key generation, node registration, node authentication, and session key agreement. The first phase is executed by The BS in offline mode to determine the pairing parameters. During the registration phase, the communication nodes register with the BS to prepare for authentication. Both the CH and the CM are mutually authenticated in the authentication phase. It is also necessary for a CH to authenticate at the BS. Following successful authentication, session keys are established between CH and CM as well as between CH and BS in the final phase of the scheme. The session keys are used to secure intra-cluster and BS-CHs communications. The proposed MAKA is secure against several cyber-attacks such as replay, cluster head impersonation, Sybil, and eavesdropping attacks.

Limitations: MAKA scheme 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, an adversary can decrypt all exchanged messages and impersonate any legitimate sensor node.

Kar et al. [51] presented MA-IDOOS: an ID-based security scheme for WSN, which exploited a bilinear pairing over elliptic curves to achieve message authentication and protect data integrity. In MA-IDOOS, the network model consists of multiple wireless sensors that constitute a clustered WSN. Each cluster contains a single Central Node (CN) and many sensor nodes. Sensor nodes are resource-constrained devices that encrypt, sign, and send the data to the CN. The latter acts as a CH, which receives the encrypted data from the sensor node and verifies the node's signatures. After verification, the data will be transmitted to another cluster (the next CN) or the destination node within the same cluster. The computational capability of the CN is high compared to the sensor node.

In MA-IDOOS, the authors focused on using an ID-Based Online/Offline digital Signatures (IBOOS) between sensor nodes and CNs. All heavy computations, such as pairing operations, are performed by CM. A sensor node runs lightweight computations, such as scalar multiplication and addition operations. The proposed MA-IDOOS provides a good resilience to active and passive cyber-attacks.

Limitations: MA-IDOOS suffers from a lack of mutual authentication between CN and sensor nodes. The PKG transmits the private keys to sensor nodes through a secure channel to the node following the Datagram Transport Layer Security (DTLS) protocol to prevent passive attack [107]. Moreover, sensor nodes use the homomorphic encryption scheme [17] to encrypt the sensed data. These factors add extra computational and communication loads to a sensor node. Additionally, the authors use

SHA-1 as a hash function, which is considered broken and no longer secure.

Hamouid et al. [35] 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. In the network model, the authors define three roles for sensor nodes:

- *Regular node*. The nodes in the network act as regular sensor nodes, which collect and transmit sensed data to the BS.
- *Relay node*. All of the non-leaf nodes in the tree act as relays (forwarders) for the data received from the child nodes. The data traffic coming from the source nodes is forwarded hop by hop from the child to the parent until it reaches the root node (BS).
- *Aggregator node*. The node that has more than one child aggregates the data of its children using a suitable aggregation function [84]. Then it forwards the result to its parent nodes.

To secure the communication among sensors nodes, the authors adopted an ID-based authenticated key agreement scheme [19] 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.

Limitations: The proposed LSTR requires a considerable computational overhead. furthermore, LSTR suffers from a lack of mutual authentication between sensor nodes during session key establishment process.

Shen et al. [92] 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, which is a particular node with a more powerful resource. On the other hand, CMs are resource-constrained devices. 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. All expensive computations, such as pairing operations, are performed by CH and BS (during Verification, Aggregation, and Agg-verification algorithms). A sensor node runs lightweight computations during the signing algorithm, just two scalar multiplications. Thus the IBAS scheme requires a low computational overhead and storage costs in CM. Moreover, IBAS ensures data integrity and prevents the adversary from injecting an invalid signature to generate a valid aggregation signature.

Limitations: The proposed scheme does not ensure data confidentiality. Consequently, it is vulnerable to eavesdropping attacks. Furthermore, the proposed scheme is cannot be applied in homogeneous WSNs with limited resources.

3.3 IBC schemes based on ECC

Huang et al. presented in [66] an enhanced version of SET-IBS protocol, called SET-IBOOS, which is proposed to reduce the computational overhead in SET-IBS by using an Identity-Based Online/Offline Signature (IBOOS) scheme []. The proposed SET-IBOOS is a paring free which combines identity-based settings with ECC. This protocol is similar to the previous SET-IBS concerning the rounds, the phases and the steps. The only difference is that in SET-IBOOS, the signature generation process is divided into offline signature generation and online signature generation. CH takes the responsibility of generating the offline signature $\widehat{\sigma_j}$ and transmits it to sensor nodes in its cluster. Each sensor node generates the online signature $\widetilde{\sigma_j}$ based on the encrypted data C_j and $\widehat{\sigma_j}$. Indeed, the IBOOS scheme employed by the proposed SET-IBOOS protocol comprises five algorithms: setup, key extraction, offline signing, online signing, and signature verification.

- *Setup*. The BS (serves as a PKG) determines the system parameters $\{E/\mathbb{F}_q, \mathbb{G}, P\}$. The BS picks a random number $msk \in \mathbb{Z}_q^*$ as the master secret key, it thereafter computes the master public key $P_{pub} = msk.P$. Then, a hash function H is chosen. The BS Preloads each sensor node with id_j the system parameters $param = \{E/\mathbb{F}_q, \mathbb{G}, P, H, r_j, msk, P_{pub}\}$, where $r_j \in \mathbb{Z}_q^*$ is randomly selected.
- *Key extraction*. A node with id_j computes the private key $Priv_j = (R_j, s_j)$, where: $R_j = r_j.P$ and $s_j = r_j + H(R_j||id_j).msk \bmod q$
- *Offline signing*. For each cluster member node, the CH generates an offline signature $\widehat{\sigma_j} = P^{-t_j}$, and then transmits it to the concerned node. $\widehat{\sigma_j}$ is a negative exponentiation value and t_j is a time stamp.
- *Online signing*. A node with id_j computes the online signature $\langle \sigma_j, z_j \rangle$, where:
  $h_j = H(C_j||id_j)$
  $z_j = \widehat{\sigma_j} + h_j.s_j \bmod q$
  $\sigma_j = \widehat{\sigma_j}.P$
  Then, node with id_j sends the $(id_j, t_j, R_j, \sigma_j, z_j, C_j)$ to its destination.
- *Verification*. A receiving node verifies authenticity by checking the following:
  $z_i.P \stackrel{?}{=} \sigma_j + h_j.R_j + (h_j.H(R_j||id_j))P_{pub}$
  If it holds, a receiving node regards the message as genuine. Otherwise, the node rejects the message.

Concerning the security aspect, the proposed protocol offers the same level of security as the SET-IBS protocol. Additionally, the security in SET-IBOOS depends on the hardness of the discrete logarithm problem (ECDLP).

Limitations: In the SET-IBOOS protocol, the master secret key is distributed to all nodes. This creates the key escrow problem, where any node can generate other nodes' private keys. Therefore, when one node is compromised, the security of the entire network is at risk. Although the authors indicate that this protocol reduces the computational overheads in SET-IBS, it is still computationally expensive for CH. Indeed, a CH is considered a resource-constrained sensor device. In the SET-IBOOS protocol, a CH generates offline signatures and transmits them to all cluster members. Therefore, the resource consumption might exceed the resource capability of a CH. The proposed SET-IBOOS also uses SHA-1 as a cryptographic hash algorithm, which is no longer secure.

Saeed et al. [85] introduced 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 the sensor nodes and the cloud server in IoTs. The proposed AKAIoTs scheme involves three main entities:

- *The BS*. The authors assume that the BS is a reliable and powerful device. It also functions as a PKG. The task of the PKG is to initialize all system parameters and extract sensor/server private keys corresponding to their identities.
- *The sensor node*. It charges for sending sensed data to the cloud server via the Internet. The sensor node should be registered with the BS and preloaded with a private key and public parameters before sending data. Additionally, the sensor node establishes a shared session key with the cloud server. Data is encrypted using this key.
- *The cloud server*. It serves as a database server, which stores the data coming from the sensor node and provides them to the users. The cloud server should be registered with the BS to obtain its private key and system parameters. Then it generates a shared key with the sensor node to decrypt the received data.

The authors have verified that AKAIoTs is secure in the random oracle model regarding the security aspect. The random oracle model is a mathematical model that proves the security of cryptographic schemes [8]. AKAIoTs further ensures several security properties of key agreement. Besides, it can prevent specific cyber-attacks such as eavesdropping and replay attacks.

Limitations: The sensor node requires six-point multiplications to establish a single shared session key with the cloud server. This is considered expensive for a resource-constrained node. Additionally, the authors did not specify the communication parameters (such as the frequency, data rate, and bandwidth) between the sensor nodes and the cloud server.

A secure data aggregation scheme was introduced by Zhong et al. [115]. 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.

Limitations: This scheme is unsuitable for homogeneous networks with limited resources since batch signature verification and encrypted data aggregation produce heavy and expensive computations. They might exceed the resources capability of a sensor device. Moreover, the recoverable sensing data is inefficient in the proposed scheme due to large-sized of aggregated messages.

Kumar et al. [61] proposed an identity-based security scheme for WSNs. The proposed scheme introduces an authenticated key agreement to establish a secret session key between two sensor nodes. Moreover, the authors used the extended

ASCII table to encrypt/decrypt a user's identity before the session key establishment process. The extended ASCII table was used only for English uppercase and lowercase letters. Moreover, the PKG converts each letter (from 52 letters) into two elliptic points using the extended ASCII table and the scalar multiplication with the generator point. Then the PKG uploads the table and the points in all sensor nodes. The following example explains the identity encryption/decryption process. Bob wants to establish the session key with Alice. First, he selects the corresponding points to each identity's letter.

$''B'' \rightarrow (0x42)_{16} \rightarrow (4,2)_{16} \rightarrow (4,2)_{10} \rightarrow (X_1 = 4.P, Y_1 = 2.P)$

$''o'' \rightarrow (0x6\text{f})_{16} \rightarrow (6,\text{f})_{16} \rightarrow (6,15)_{10} \rightarrow (X_2 = 6.P, Y_2 = 15.P)$

$''b'' \rightarrow (0x62)_{16} \rightarrow (6,2)_{16} \rightarrow (6,2)_{10} \rightarrow (X_3 = 6.P, Y_3 = 2.P)$

Then Bob computes:

$(Z_1 = X_1 + P_{pub}, W_1 = Y_1 + P_{pub})$

$(Z_2 = X_2 + P_{pub}, W_2 = Y_2 + P_{pub})$

$(Z_3 = X_3 + P_{pub}, W_3 = Y_3 + P_{pub})$

Bob will send its encrypted id_B to Alice: $id_B = [\{Z_1, W_1\}, \{Z_2, W_2\}, \{Z_3, W_3\}]$

Upon receiveng id_B, Alice computes:

$(X_1 = Z_1 - P_{pub}, Y_1 = W_1 - P_{pub})$

$(X_2 = Z_2 - P_{pub}, Y_2 = w_2 - P_{pub})$

$(X_3 = Z_3 - P_{pub}, Y_3 = W_3 - P_{pub})$

Then Alice performs the reverse process. She determines each identity's letter using its corresponding elliptic points(X_j, Y_j) and the extended ASCII table.

$(X_1, Y_1) \rightarrow (4,2)_{10} \rightarrow (4,2)_{16} \rightarrow (0x42)_{16} \rightarrow ''B''$

$(X_2, Y_2) \rightarrow (6,15)_{10} \rightarrow (6,\text{f})_{16} \rightarrow (0x6\text{f})_{16} \rightarrow ''o''$

$(X_3, Y_3) \rightarrow (6,2)_{10} \rightarrow (6,2)_{16} \rightarrow (0x62)_{16} \rightarrow ''b''$

Limitations: In the proposed scheme, the sensor node contains the ASCII codes for 52 English letters (lowercase and uppercase) as well as two points for each letter, which results in 104 elliptic points. Therefore, the scheme is inefficient in terms of storage. Further, the node's identity is exchanged in the form of elliptic points, which adds additional communication load to the sensor node. The use of ASCII code for identity encryption and decryption is insecure. Indeed, an adversary can know the ASCII codes of English letters, the generator point, and the master public key. Consequently, he can generate the elliptic points for each identity letter.

Erdong et al. [111] proposed a key management scheme for heterogeneous CWSN. The authors adopt the Pairing-Free Identity-Based Signature (PF-IBS) [90] and the ECC encryption algorithm [4] 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.

Limitations: The proposed scheme 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.

3.4 Schemes based on ECC

Boudia et al. [71] Designed a secure data aggregation scheme in CWSN. The network is divided into many clusters, and each cluster contains many CMs and two CHs. Through hop-by-hop verification, the proposed scheme provides end-to-end privacy and allows early detection of attacks, thus reducing the need to rely solely on the BS for verification. For encrypting and verifying data, the authors used ECC-based Homomorphic encryption and hash-based authentication codes, respectively. According to the proposed scheme, each sensor node transmits encrypted data to two CHs. Afterwards, both CHs communicate with each other to verify data, and finally, both CHs send messages to BS.

Limitations: The proposed scheme requires more communication overhead as each sensor node sends messages to 2 CHs. Moreover, The scheme has an additional verification process and aggregation operations at CHs, and thus it requires more computation cost.

A secure scheme called Secure Data Transmission Scheme (SDTS) was proposed by Harbi et al. [38] to improve communication security in CWSNs and prevent information leakage. SDTS is based on ECC and achieves several security requirements, including confidentiality, integrity, and authentication. According to the authors, the network is composed of a single BS and a number of homogeneous sensor nodes. The BS is assumed to be a robust and reliable device, and sensor nodes are grouped into clusters. Each cluster has a single CH and several cluster members (CMs). CHs aggregate the encrypted data from cluster members (CMs) and forward the result to the BS.

Limitations: The proposed SDTS scheme has the following security vulnerabilities:

- Since the nature of the public channel is insecure, an adversary could obtain the exchanged public keys for CH, CM, and BS, which would make them vulnerable to MITM attacks.
- Lack of mutual authentication makes a malicious node impersonate a legitimate CH and communicate with CMs to receive the data.
- All messages are exchanged without any nonce. Thus, the sent data can be intercepted and replayed several times by an adversary.

Elhoseny et al. [28] proposed an encryption scheme for secure the data transmission in CWSNs. To generate the CMs node and the BS's public and private keys, the authors used ECC. Further, they employed homomorphic encryption to allow each CH to aggregate the encrypted data from its CMs without decryption. This information is then forwarded to the BS. The encryption key at each CM is 176-bits and is created by combining k_i, id_i, and D_{i-CH}, where:

- K_i (148-bits): represents the shared key between CM_i and the BS, which is generated based on Diffie–Hellman key establishment protocol.
- Id_i (13-bits): represents the identity of the CMi node.
- D_{i-CH} (15-bits): represents the distance between the CM_i node and its CH.

The encryption key is also used by the BS to decrypt the ciphertext of CM_i. The experimental results demonstrate that the proposed method significantly improves network performance in terms of communication overhead, memory requirements, and energy consumption compared to other approaches. Furthermore, it is resistant to many types of cyber-attacks, such as passive attacks and brute force attacks.

Limitations: The proposed scheme does not guarantee data integrity and mutual authentication between CM and CH. Consequently, a malicious node masquerades as CM and sends false data to CH for the purpose of altering the aggregated result.

3.5 Comparison

Table 3.2 shows the comparison between the different PKC-based schemes studied in this chapter according to the following criteria:

- The type of public-key cryptographic used in designing the security scheme.
- The network structure on which the security scheme is based.
- Does the security scheme provide mutual authentication between two communicating parties before exchanging data?
- Is the security scheme vulnerable to the key escrow problem?
- Is the security scheme vulnerable to the public key authenticated problem?
- Cryptographic primitives used in the implementation of the security scheme.
- The vulnerability to cyber-attacks as well as the inefficiencies in computation, communication, and storage costs.

3.6 Conclusion

We presented in this chapter a survey and analysis of PKC-based schemes recently proposed in the literature for achieving security in WSNs. Furthermore, we provided a comparison between the studied schemes according to several criteria. In the following chapters, we present our proposed methods for improving the security and the performance of WSN, taking into consideration the advantages and limitations of the related works.

TABLE 3.2: Comparison between different PKC-based schemes in WSNs .

Schemes	**PKC**	**Network Structure**	**MA**	**KEP**	**PKAP**	**Cryptographic Primitives**	**Vulnerabilities / inefficiency**
SET-IBS [66]	IBC with Pairing	Clustered/Homogeneous	No	Yes	No	- Homomorphic encryption - Identity-based signature	- High computational cost
Hu [43]	- IBC with Pairing - ECC	Clustered/Homogeneous	No	No	No	- Identity-based encryption - DH key agreement	- High computational cost
Qin et al. [81]	IBC with Pairing	Clustered/Homogeneous	No	Yes	No	- Identity-based encryption	- High computational cost
ICMDS [68]	IBC with Pairing	Clustered/Homogeneous	No	Yes	No	- Identity-based encryption - Session key generation	- High computational cost - High communication and storage costs - Replay and CH impersonation attacks
MAKA [37]	IBC with Pairing	Clustered/Homogeneous	Yes	Yes	No	- Identity-based encryption - Identity-based key agreement	- High computational communication costs
MA-IDOOS [51]	IBC with Pairing	Clustered/Heterogeneous	No	No	No	- Identity-based signature - Homomorphic encryption - DTLS protocol	- High computational communication costs
LSTR [35]	IBC with Pairing	Tree/Homogeneous	No	No	No	- Identity-based key agreement	- High computational cost
IBAS [92]	IBC with Pairing	Clustered/Heterogeneous	No	No	No	- Identity-based signature	- Eavesdropping attack
SET-IBOOS [66]	IBC with ECC	Clustered/Homogeneous	No	Yes	No	- Homomorphic encryption - Identity-based signature	- High computational cost attack
AKAIoTs [8]	IBC with ECC	Flat/Homogeneous	Yes	No	No	- Identity-based signature - Identity-based key agreement	- Adds extra computational load to sensor nodes
Zhong et al. [115]	IBC with ECC	Clustered/Heterogeneous	No	No	No	- Homomorphic encryption - Identity-based signature	- Heavy and expensive computation
Kumar et al. [61]	IBC with ECC	Flat/Homogeneous	Yes	No	No	- Identity-based key agreement - ASCII code-based encryption	- ASCII-based encryption is insecure - High communication and storage costs
Erdong et al. [111]	IBC with ECC	Clustered/Heterogeneous	No	No	No	- ECC-based encryption - Identity-based signature	- High traffic causing network congestion - Inefficient in terms of storage and computation
Boudia et al. [71]	ECC	Clustered/Homogeneous	No	No	No	- Homomorphic encryption - Hash-based authentication	- Requires more computation and communication costs
SDTS [38]	ECC	Clustered/Homogeneous	No	No	Yes	- Shared Key establishment - Homomorphic encryption - Digital signature	- MITM and Replay attacks - Impersonates a legitimate CH - Adds extra load to CH in terms of computation, communication and storage.
Elhoseny et ał. [28]	ECC	Clustered/Homogeneous	No	No	Yes	- Homomorphic encryption	- False data injection attack - Does not guarantee the data integrity

MA: Mutual Authentication, **KEP**: Key Escrow Problem, **PKAP**: Public Key Authentication Problem.

Contributions

Chapter 4: HCBS: Hybrid Cryptography-Based Scheme for secure data communication in CWSNs

Chapter 5: IDSP: A New Identity-Based Security Protocol for CWSNs

Chapter 6: IBAKAS: A new Identity-Based Authentication and Key Agreement Scheme for CWSNs

Chapter 4

HCBS: Hybrid Cryptography-Based Scheme for secure data communication in CWSNs

Chapter Overview: *This chapter proposes a new security protocol based on the well-known LEACH routing protocol, Hybrid Cryptography-Based Scheme, for secure data communication in cluster-based WSN (HCBS). As a multi-constrained criteria approach, HCBS is built on a combination of the cryptography primitives to establish the shared keys, encrypt the sensed data, and compute MAC values. After a set of tests on the TOSSIM simulator, the results obtained showed that our proposal achieves good performance in energy consumption, loss rate, and end-to-end delay. In addition, HCBS guarantees a high level of security.*

4.1 Introduction

Sensor nodes communicate with the BS through wireless communications. BS represents the downstream of all data coming from the sensor nodes. Nodes communicate directly with other nodes within their range of transmission. Communicating with a remote node or a node out of transmission range is done through other nodes that route data to the destination. The routing mechanisms ensure this process, which is based on multi-hop communication. A variety of routing protocols have been developed for WSNs. They are generally divided into network structure, and protocol operation [52]. The network structure is classified as flat, hierarchical, and location-based. Protocol operation is classified into negotiation, multi-path, query, QoS, and coherent based routing. The cluster-based routing protocol is an effective way to reduce the total energy consumption of the WSN. The idea is to form a set of clusters of sensor nodes. Each CH collects data from all the sensor nodes belonging to its cluster, aggregates, and transmits the result directly to BS. Data aggregation in CH significantly reduces the number of messages sent to BS.

One of the most critical issues of the WSN is the security in cluster-based routing protocols. This issue did not have a great deal of attention since most of these protocols have been developed for the efficient routing of information; however, the security aspect has been neglected [34]. Indeed, cluster-based routing protocols are vulnerable to attacks threatening the reliability of data traffic. For example, LEACH [39] is a considered the most popular clustered routing protocol that rely fundamentally on elected CHs for data aggregation and routing. Like most protocols for WSN, LEACH is vulnerable to several cyber-attacks [54], such as spoofing and replay, attacks involving CH are the most damaging [113]. If a malicious node becomes CH, it can launch several attacks to disrupt the network operation. Note that the malicious node can attack non-CH nodes and inject erroneous information into the network. Thus, LEACH needs to ensure the confidentiality, integrity, freshness, and authentication of the originating node of the transmitted message.

This chapter proposes a secure version of LEACH called Hybrid Cryptography-Based Scheme for secure data communication in cluster-based WSN (HCBS). This protocol is based on symmetric and asymmetric cryptographic primitives. HCBS guarantees the most important security requirements and it resists attacks that LEACH is vulnerable. The rest of the chapter is organized as follows. Section 4.2 describes our proposed protocol (HCBS), then security analysis and evaluation results are presented in Sections 4.3 and 4.4, respectively. The last Section concludes this work.

4.2 HCBS: The proposed protocol

To explain the proposed HCBS, the following assumptions are specified:

- The studied area is a WSN, in which sensor nodes are homogeneous in processing capacity, communication, energy, and storage.

- The BS is assumed to be a powerful device, reliable, and responsible for configuring all the nodes before deployment of the WSN.
- In this study, an attacker is supposed to be passive or active during the operation of the network. BS and the sensor nodes are not mobile.

The notations used in this chapter are summarized in Table 4.1.

TABLE 4.1: The notations used in the HCBS protocol.

Notation	Description
BS	Base Station
CH_i	Cluster-Head
CM_j	Cluster-Member
ID_i	Node identity
K_{Gp}	Global key
K_y^x	Pairwise key shared between two entities x and y
K_x^+	Public key of entity x
K_x^-	Private key of entity x
$A\|\|B$	Concatenation A with B
$nonce$	Random number used only once to ensure that old communications cannot be reused
$MAC_K(M)$	Message authentication code M with key K
$E_K(M)$	Symmetric encryption function of the message M by using a key K

4.2.1 An overview

The proposed protocol is a secure version of LEACH, consisting of three phases: initialization, set-up, and steady-state. During the first phase, each sensor node is preloaded with a key K_{G0} and pairs: (K_{SN}^-, K_{SN}^+) and (K_{BS}^-, K_{BS}^+). K_{G0} is considered as a global key of round 0. In other words, this key will be used during the set-up phase for round 0. It is shared between all deployed nodes and BS. The global key is generally used to encrypt messages (for example, the message broadcast by the BS upon announcement of a new round) or compute a MAC. As a security measure, this key is renewed during the formation of clusters. In the set-up phase, CHs are elected, and clusters are formed where each CH broadcasts an announcement to neighboring nodes, inviting them to be CMs in its cluster in a secure manner. Additionally, the pairwise keys K_{CHi}^{BS} and K_{CMj}^{CHi} are established using the ECDH scheme. These pairwise keys are used in the steady-state phase to secure communications between BS, CHs, and CMs, as shown in Figure 4.1.

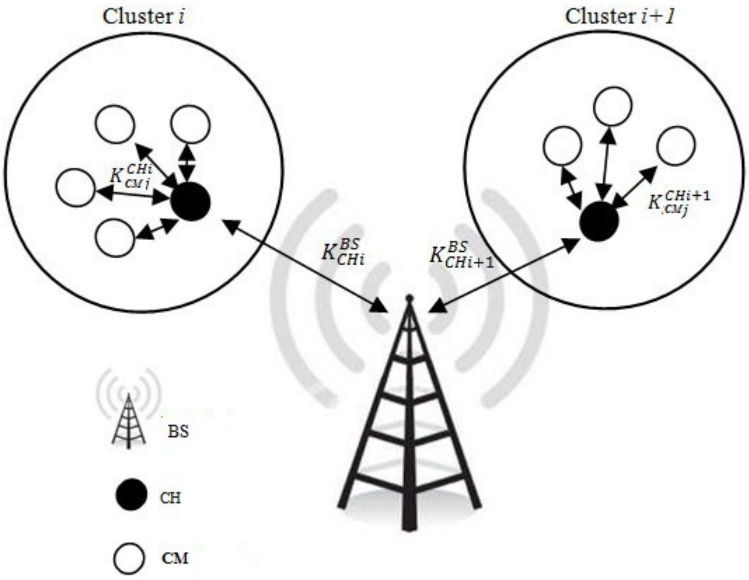


FIGURE 4.1: Different secure links during the steady-state phase.

The final phase is steady-state, in which the data collected by the CMs is transmitted to the CHs, which will, in turn, transmit them to the BS. At each round, only the set-up and steady-state phases are repeated. Algorithm 1 describes the two phases. The messages exchanged in HCBS are encrypted, so their confidentiality is guaranteed. A MAC and a nonce are added to the sent message, ensuring authentication, data integrity, and freshness.

4.2.2 Detailed description of HCBS protocol steps

In this section, we discuss HCBS protocol in more detail.

• **Initialization:** The management of this phase is the responsibility of BS:

- BS computes a set of keys $S = \{K_{G0}, K_{G1}, ..., K_{Gp}, ..., K_{Gn}\}$. To do this, it generates the key k_{p+1} and calculates the following key K_{Gp} using keyed one-way hash function H. The set S is stored in the BS.

$$\boxed{H(k_{p+1}) = k_p, 0 \leq p < n}$$

- BS generates ECC key pairs $(K_{SNj}^{-}, K_{SNj}^{+})$ for each sensor node, then it generates the BS ECC key pair (K_{BS}^{-}, K_{BS}^{+}).
- Each sensor node is preloaded by ECC key pairs $(K_{SNj}^{-}, K_{SNj}^{+})$ and (K_{BS}^{-}, K_{BS}^{+}).
- BS selects a K_{G0} key out of the set S and preloads it in all sensor nodes. After the

deployment of the WSN, each node can use the key to encrypt and decrypt messages or calculate a MAC. Note that the key K_{Gp} can be considered as a global key that will be used during the set-up phase of the round p.

• **Set-up:** This phase begins with the announcement of a new round by the BS. The latter encrypts its identity ID_{BS}, a *nonce* and a threshold value T using K_{Gp}. It generates a MAC value and broadcasts this information to all sensor nodes. It should be noted that in LEACH, the threshold T represents the probability of a node becoming CH. The sensor node becomes CH if it generates a value less than T [39]. Only legitimate nodes that have K_{Gp} key can decrypt T and verify the validity of the BS's MAC value. In the event a sensor node becomes CH, it encrypts its ID_{CHi}, generates a MAC value, and transmits the information to the BS (line **5** - Algorithm 1). The BS receives the message from CH_i and if the MAC value is valid, the CH_i and the BS establish the shared key K_{CHi}^{BS} (line **6** - Algorithm 1) using ECDH scheme. In the next step, the CH_i encrypts the notification message *adv* and the public key K_{CHi}^{+} and generates a MAC value using K_{Gp}. Then, it broadcasts this information to nodes that are within its transmission range (line **8** - Algorithm 1). The sensor nodes receive *adv*, and only the legitimate node can decrypt *adv* and decide which cluster it will belong to. Next, the legitimate node sends a message *Join_req* and its public key K_{CMj}^{+} to CH_i chosen (line **12** - Algorithm 1). Upon receiving the message *Join_req*, CH_i and CM_j establish the shared key K_{CMj}^{CHi} (lines **9** and **13** - Algorithm 1). After the clusters are formed, the BS sends the global key of the next round K_{Gp+1} to CHs. In turn, the latter sends K_{Gp+1} to its CMs. In addition, each CH creates a TDMA schedule and assigns a time slot for each CM to transmit data. Time slots and K_{Gp+1} are encrypted with K_{CMj}^{CHi} (line **10** - Algorithm 1).

• **Steady-state:** During this phase, a node CM_j sends to its CH_i the sensed data in an encrypted manner. The CH_i collects data from all its CMs and transmits it to BS after the aggregation (lines **17, 20** and **21** - Algorithm 1). The BS decrypts the data and verifies the validity of the MAC. In the positive case, it accepts the aggregated data. Otherwise, it refuses them.

Algorithm 1 HCBS Pseudo Code Protocol

```
**i** : *sensor node;* **N** : *number of nodes in network*
**for** $i = 1$ to $N$ **do**
    /** Set-up phase**/
    /*Compute shared key between BS and CHi*/
    **if** $(i = CH)$ **then**
        $CH_i \rightarrow BS : E_{K_{Gp}}(ID_{CHi}||nonce)||ID_{BS}||MAC_{K_{Gp}}(ID_{CHi}||nonce)$
        $K^{BS}_{CHi} = ECDH(K^{-}_{CHi}, K^{+}_{BS})$
        /*Formation of clusters and compute shared key between CHi and CMj */
        $CH_i \rightarrow Broadcast : E_{K_{Gp}}(ID_{CHi}||adv||nonce||K^{+}_{CHi})||MAC_{K_{Gp}}(ID_{CHi}||adv||nonce)$
        $K^{CHi}_{CMj} = ECDH(K^{-}_{CHi}, K^{+}_{CMj})$
        $CHi \rightarrow CMj : E_{K^{CHi}_{CMj}}(ID_{CHi}||K_{Gp+1}||(Timeslot(CM_j))||nonce)||ID_{CMj}|| MAC_{K^{CHi}_{CMj}}(ID_{CHi}||K_{Gp+1}||nonce)$
    **else**
        $CM_j \rightarrow CH_i : E_{K_{Gp}}(ID_{CMj}||Join\_req||nonce||K^{+}_{CMj})||ID_{CHi}|| MAC_{K_{Gp}}(ID_{CMj}||Join\_req||nonce)$
        $K^{CHi}_{CMj} = ECDH(K^{-}_{CMj}, K^{+}_{CHi})$
    **end if**
    /**Steady-State phase**/
    /*Aggregate and send data to BS */
    **if** $(i = CM)$ **then**
        $CM_j \rightarrow CHi : E_{K^{CHi}_{CMj}}(ID_{CMj}||data||nonce)||ID_{CHi}||MAC_{K^{CHi}_{CMj}}(ID_{CMj}||nonce)$
    **end if**
    **if** $(i = CH)$ **then**
        $data\_agg = Data\_Aggregation(data)$
        $CHi \rightarrow BS : E_{K^{BS}_{CHi}}(ID_{CHi}||data\_agg||nonce)||ID_{BS}||MAC_{K^{BS}_{CHi}}(ID_{CHi}||data\_agg||nonce)$
    **end if**
**end for**
```

4.3 Security analysis

HCBS provides network confidentiality by encrypting all messages. In addition, it can guarantee freshness, integrity, and authentication of the originating node of the transmitted message by adding a nonce and a MAC to the sent message. HCBS also resists several attacks against LEACH:

4.3.1 Eavesdropping attack

The attack allows the adversary to listen to transmissions. Therefore, it recovers the contents of the messages circulating within the network. To prevent such an attack, HCBS encrypts all messages. Moreover, encryption keys are renewed periodically to strengthen security.

4.3.2 Data modification and insertion

In order to falsify the result of aggregation, an adversary can alter messages transmitted by CMs. The CH will accept the altered data and aggregate it. Consequently, the final result will be incorrect. HCBS is protected against this attack since messages are authenticated using a message authentication code (MAC).

4.3.3 MITM attack

HCBS protects against MITM attacks. Because, during the steady-state phase, the sender node encrypts both its public key and its identity before transmitting them. Only the legitimate node, which has the decryption key, can read the message through this process. Additionally, the sent message is protected by a MAC computed over the identity of the sender node and a nonce to ensure message authentication.

4.3.4 Replay attack

Since each transmitted message includes a random value called nonce, the previous messages intercepted by an intruder cannot be re-injected. The message with a different nonce than the expected nonce will be rejected. Additionally, the intruder cannot alter the value of the nonce since the messages are encrypted and protected by MAC.

4.3.5 HELLO flood

The HCBS ensures the confidentiality, authentication, and integrity of both messages *adv* and *join_req*, allowing only legitimate nodes to participate in the formation of clusters.

4.3.6 Selective forwarding

The proposed HCBS prevents a malicious node from becoming a CH. Additionally, the keys are renewed proactively to prevent key compromises, reducing the chances of this attack happening.

4.3.7 Sybil attack

Cryptographic techniques, such as encryption and mutual authentication, can help prevent an intruder from launching a Sybil attack against a WSN. In HCBS, all messages exchanged during the set-up, and steady-state phases are encrypted and authenticated.

4.4 Evaluation and simulation results

The following section discusses the performance evaluation of HCBS by comparing it with SecLEACH [76]. The evaluation is based on three metrics:

- Energy consumption efficiency. This is an important performance metric that directly influences the lifetime of the entire network. Here, we examine the impact of the HCBS protocol on the energy consumed by comparing it with the SecLEACH protocol. We compute the average energy consumed in each node as an evaluation metric.
- Loss rate. It represents the ratio of the number of packets lost to the number of packets sent. The following formula defines the loss rate:

$$Loss\ rate = \frac{Loss\ packets}{Sent\ packets} * 100$$

- End-to-End Delay (EED). The data sensed by CMs are not sent directly to the BS but are aggregated by CHs. Note, that the latter must wait until all CMs have completed sending their data before moving on to the aggregation phase. Consequently, the criterion we use is the average EED of all packets passing through the network.

$$Average\ EED = \frac{\sum packet\ delivery\ time}{Number\ of\ packets\ received}$$

4.4.1 Simulation parameters

HCBS and SecLEACH protocols have been implemented using NesC [103] language and simulated in TinyOS SIMulator (TOSSIM) [64]. We encrypt our data with the AES algorithm [40] with a key size of 128 bits. We calculated the MAC value using MMH algorithm[100], which provides a 32 bit MAC. We use ECDH, an asymmetric key agreement scheme. This scheme is implemented using TinyECC 2.0 [65], a configurable ECC library designed specifically for WSNs. The simulations have been conducted with different network topologies. Four networks with different sizes were considered: 50, 100, 150, and 200 nodes. The nodes are homogeneous and deployed randomly. In each network, the simulation time is 500 seconds, and the number of malicious nodes is five nodes. The number of CHs in a network is fixed at 10%. Each metric represents the average of five simulation results for given network size. The noise model we use in our simulations is Meyer Heavy [63]. Table 4.2 summarizes the parameters used in our simulations.

TABLE 4.2: Simulation parameters

Parameter	Value
Network size	200 nodes
Simulation time	500 seconds
Area	100*100 m
Distribution of nodes	Randomly
Noise model	Meyer Heavy
Sensor platform	MicaZ
Chip radio	CC2420
Initial energy	20 joules
Dissipated energy by a node for a 1-bit transmission	e_{tx} = 0.209 µJ
Dissipated energy by a node for 1-bit reception	e_{rx} = 0.226 µJ
Malicious nodes	5 nodes

4.4.2 Results and interpretations

Figure 4.2 illustrates the average energy consumption for communication during the set-up and steady-state phases based to different size network. It is computed as:

$$E = \frac{\sum_{i=1}^{k} ((\text{The size of sent messages} \times e_{tx}) + (\text{The size of the received messages} \times e_{rx}))}{k\text{: Network Size}}$$

According the characteristics of the CC2420 transceiver used in MicaZ platform [99], the unit communication costs e_{rx} and e_{rx} are equal to 0.209 µJ and 0.226 µJ, respectively. Based on Figure 4.2, we observe that HCBS offers better results in terms of energy consumption for communication than SecLEACH. This is due to the fact that the total size of messages exchanged in the HCBS protocol is less compared to SecLEACH.

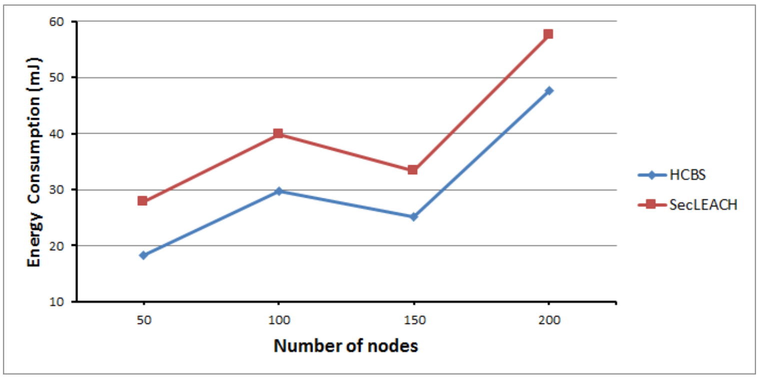


FIGURE 4.2: Energy consumption for communication.

Figure 4.3 depicts the average energy consumption for computation (cryptographic primitives) for different sizes of networks. It is observed that the energy consumption of HCBS is higher than that of SecLEACH. This is due to the asymmetric cryptographic primitives used by HCBS during its set-up phase, where the MicaZ sensor node requires 50.82 mJ to establish a single shared key using the ECDH scheme [65]. However, the energy consumption of HCBS is generally acceptable and satisfactory on the MicaZ sensor device, although it is not as good as SecLEACH.

Figure 4.4 illustrates the simulation results in terms of packet loss rate in the steady-state phase. In both protocols, a tolerable packet loss rate has been observed. According to the number of nodes deployed, the packet loss rate ranges between 3.04 and 4.64 percent. There are two reasons for this: noise in communications and

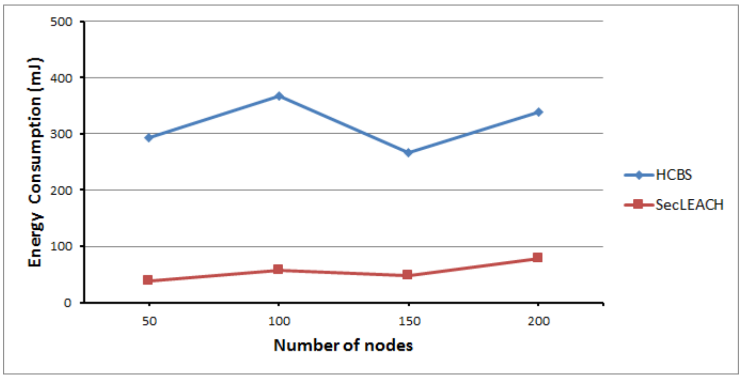


FIGURE 4.3: Energy consumption for computation.

the authentication mechanism implemented in SecLEACH and HCBS, eliminating any packet whose source is not authenticated. Compared to SecLEACH, the average packet loss rate in HCBS is lower.Because the SecLEACH's network connectivity is variable and unstable, dependent on the size of the preloaded keyrings in the sensor nodes, in the absence of a shared key, CMs cannot communicate with their CHs.

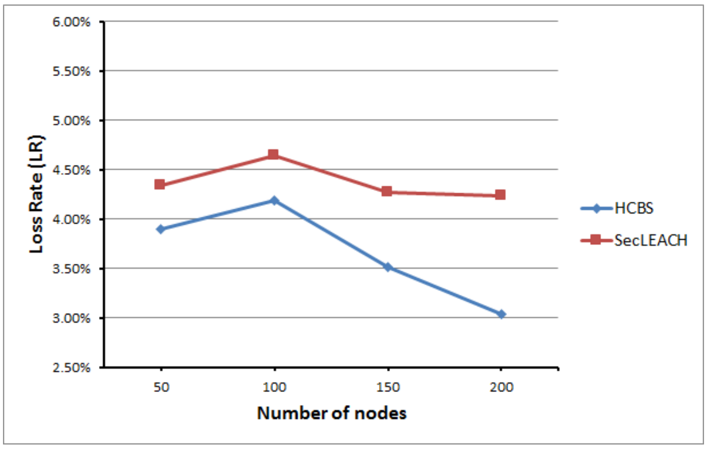


FIGURE 4.4: Packet loss rate.

Figure 4.5 shows the average EED calculated for all sensor nodes in the network during the steady-state phase. According to Figure 3, the EED is not dependent on the number of nodes deployed in the network. Indeed, a node can transmit its information in just one hop. The CM must only transmit sensed data to its CH, which will then aggregate the data and transmit the result to the BS. Furthermore, the average EED achieved by HCBS is smaller than that of SecLEACH. This is due to the smaller size of the packets sent in the HCBS protocol.

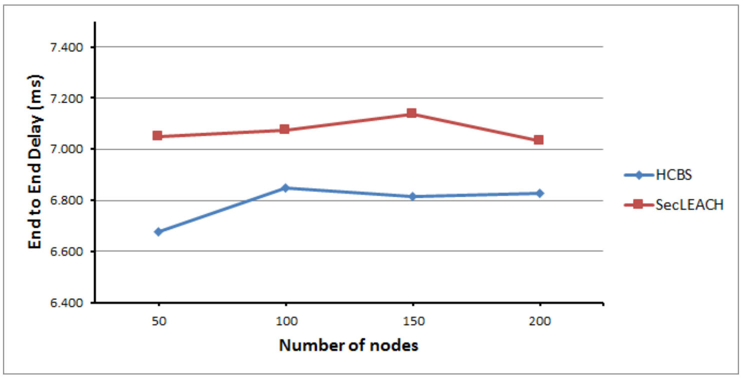


FIGURE 4.5: End-to-end delay.

4.5 Conclusion

This chapter focuses on the security of cluster-based routing protocols for WSN by proposing a secure version of LEACH called the Hybrid Cryptography-Based Scheme for secure data communication in cluster-based WSN (HCBS). The proposed protocol relies on ECC to facilitate the key distribution between the communication nodes. Additionally, HCBS is based on symmetric cryptography to guarantee mutual authentication, data integrity, and data confidentiality during set-up and steady-state phases. The HCBS protocol enhances the security of the LEACH, ensuring the basic security requirements and resisting common attacks against LEACH. Our evaluation of HCBS is based on simulations and comparisons to SecLEACH in terms of energy consumption for computation and communication, loss rate, and end-to-end delay. Overhead generated by HCBS is almost as low as that generated by SecLEACH. HCBS also allows stabilizing the EED even as the number of nodes in the network increases. In the next chapter, we propose a new IBC scheme based on pairing to enhance the security in homogeneous clustered WSNs.

Chapter 5

IDSP: A New Identity-Based Security Protocol for CWSNs

Chapter Overview: *This chapter proposes a lightweight and secure version of the LEACH protocol called IDSP. The security of data routing from CMs to the BS is achieved based on an ID-based authenticated key agreement with a bilinear pairing. IDSP reduces the number of necessary keys to a signal-ephemeral global key and one long-term private key preloaded in the sensor nodes. These keys are then used to compute pairwise shared keys with other nodes without the requirement of digital certificates. As a result, IDSP provides both node authenticity and confidentiality during data transmission at a low cost and without involving the BS. IDSP has been tested in the Cooja network simulator with the WiSMote platform, and the results showed that IDSP is lightweight and secure.*

5.1 Introduction

Since the PKI introduces complicated certificate operations, such as distribution, storage, and certificate verification, IBC is the alternative solution to overcome this issue. This chapter proposes an IBC-protocol based on pairing to enhance the security in CWSN, called Identity-based Security Protocol (IDSP). Indeed, the IDSP protocol requires one pairing computation and one MTP function to establish a single shared key. This reduces the computation cost on resource-limited devices compared with existing relevant schemes. Several factors are considered in the design of our protocol, including:

- A good level of security is essential, especially in the event of cyber-attacks against the CWSN.

- Save energy and memory on sensor devices with limited resources.
- Incorporate a lightweight and uncomplicated key exchange mechanism into the proposed protocol.

The rest of the chapter is organized as follows. Section 5.2 gives details of the proposed IDSP protocol. The security analysis is discussed in Section 5.3. The performance evaluation is done in Section 5.4. Finally, Section 5.5 concludes the chapter.

5.2 IDSP: IDentity-based Security Protocol

This section aims to design and propose a secure version of the LEACH protocol called IDSP. The proposed protocol secures data communication in cluster-based WSNs by utilizing identity-based cryptography. To help BS or CMs establish pairwise shared keys with CHs, IDSP adopts the SOK scheme (outlined in subsection 5.2.1). The reason we chose SOK is its low bandwidth requirements. Only nodes identities must be exchanged to establish the shared key rather than the exchange of the public keys. During data transmission, pairwise shared keys are used to authenticate as well as encrypt the data. Additionally, the IDSP protocol uses a global key (*net*) during the clustering network. The purpose of using the *net* key is mutual authentication and encryption. *net* is considered a globally shared key between BS and all network nodes. Each round, *net* is renewed in order to enhance security. The notations used in the proposed protocol are listed in Table 5.1.

TABLE 5.1: The notations used in the IDSP protocol.

Notation	**Description**
id	Node identity
$\hat{e}(\ )$	Pairing function
H	MTP function
s	Master secret key
net_i	Global key
K_y^x	Pairwise key shared between two nodes x and y
d_x	Private key of a node x
$nonce$	Random number that is only used once.
$MAC_K(M)$	Message authentication code with the key K
$E_K(M)$	Symmetric encryption of the message M using a key K

5.2.1 Sakai, Ohgishi and Kasahara (SOK) Key sharing scheme

SOK [87] is an identity-based key exchange scheme based on cryptographic pairings on elliptic curves that enables two sensor nodes to compute the shared key. This key can then be used for symmetric cryptography. For instance, assume we have two sensor nodes x and y. Each node has a private key generated by the BS and wishes to compute a shared key K_x^y. Private keys can be calculated such that $d_x = s.Q_x$ and $d_y = s.Q_y$ for the sensor nodes x and y, respectively, where:

- s is a master secret key, known only to BS.
- $Q_x = H(id_x)$ and $Q_y = H(id_y)$. Note that id_x and id_y are the identities of nodes x and y respectively.
- H is a MTP function.

As a first step, nodes x and y exchange id_x and id_y. Subsequently, x will compute $\hat{e}(d_x, Q_y)$ and y will compute $\hat{e}(d_y, Q_x)$.

$\hat{e}(d_x, Q_y) = \hat{e}(s.Q_x, Q_y) = \hat{e}(Q_x, Q_y)^s = \hat{e}(Q_x, sQ_y) = \hat{e}(Q_x, d_y)$

As result, x and y compute the same value which will be considered as the shared key K_x^y.

5.2.2 Network model

Throughout this work, we consider that the studied network is a WSN. Sensor nodes here are resource-constrained devices with homogenous capabilities and functionalities. In contrast, the BS is considered reliable and is responsible for configuring the nodes before the network is deployed. As well, all sensor nodes are randomly distributed. All nodes, including the BS, are not mobile.

5.2.3 Threat model

Based on the Dolev-Yao attack model [23], we assume the adversary performs the following cyber-attacks.

- The adversary can obtain private information by listening to data transmissions over public channels.
- There is the potential for the adversary to inject false data in the message or to change or delete important information.

- The adversary can intercept data and then resend it maliciously in order to misdirect the receiving node or disrupt its services.
- The adversary cannot physically access information stored in sensor nodes.

5.2.4 The proposed IDSP protocol

In this section, we present our proposed IDSP protocol. It consists of three phases: initialization, clustering, and data transmission.

Initialization

The BS bootstraps offline sensor nodes with initial keying material prior to network deployment.

- The BS generates the Public Parameters (PP) which are $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, H)$. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups with order q. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map. $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ is cryptographic hash function.
- For each node, the BS computes $d_{ID_i} = s.Q_{ID_i}$ as a private key, where $Q_{ID_i} = H(ID_i)$. At each node, the BS uploads the corresponding private key and the public parameters. In contrast, the master secret key s is kept only at the BS.
- the BS uses a keyed one way hash function F to compute the set of keys $\beta = \langle net_{n-1}, ..., net_1, net_0 \rangle$. It is worth noting that β is stored at BS only.

$$\boxed{F(net_{n-1}) = net_{n-2}}$$

- BS selects net_0 key from a set β, then preloads this key in all sensor nodes in order to use it in network clustering process.

Clustering

The IDSP protocol operates similarly to LEACH, and its operation is based on rounds. Figure.5.1 presents the clustering phases. Initially, each node determines whether it is a CH of the current round through probability. When the node becomes a CH in the current round, it establishes the shared key K_{CH}^{BS} using id_{BS} and d_{CH}. After that, the CH encrypts its identity and sends it to the BS. Upon arrival, the BS decrypts the received message and verifies its authenticity by examining the MAC value received from the CH. If the MAC value is valid, the BS authenticates the CH and establishes the same shared key K_{CH}^{BS}.

Next, the CH broadcasts an advertisement message (*adv*) to the neighbor nodes to join its cluster as well as establish the shared keys. Each sensor node may receive several advertisement messages from CHs, and it will choose one CH to join. This is done by sending a confirmation message (*join_req*) to the respective CH. The choice of CH is based on the strength of the signal received from the advertisement message. The CH having the strongest adv signal is chosen. Afterwards, both CH and CM compute the shared key K_{CM}^{CH}.

After completing the formation of groups, the BS sends net_{i+1} to all CHs. Each CH, in turn, forwards this key to its CM. Note that net_{i+1} will be used during the next round. Additionally, the CH establishes a TDMA schedule and assigns each CM a time slot by which the CM transmits data to the CH. It should be noted that the TDMA schedule eliminates collisions during the communication of sensor nodes. This could be achieved by allowing each sensor to send or receive data packets in a time slot allocated to it.

Data transmission

As shown if Figure 5.2, this phase involves two steps: the CMs sending data to the CH and the CH sending data to the BS. The CMs can sleep to conserve energy. CH must keep its receiver open in order to receive data from its CMs. To ensure the security of data transmission, the CH and the CM use a shared secret key K_{CM}^{CH} to authenticate each other and encrypt messages. Having collected data from all member nodes, the CH aggregates the data and forwards the result to the BS. The latter will validate the received data using the shared key K_{CH}^{BS}.

5.3 Security analysis

Based on the assumptions mentioned in the threat model, we provide an informal security analysis of the IDSP protocol.

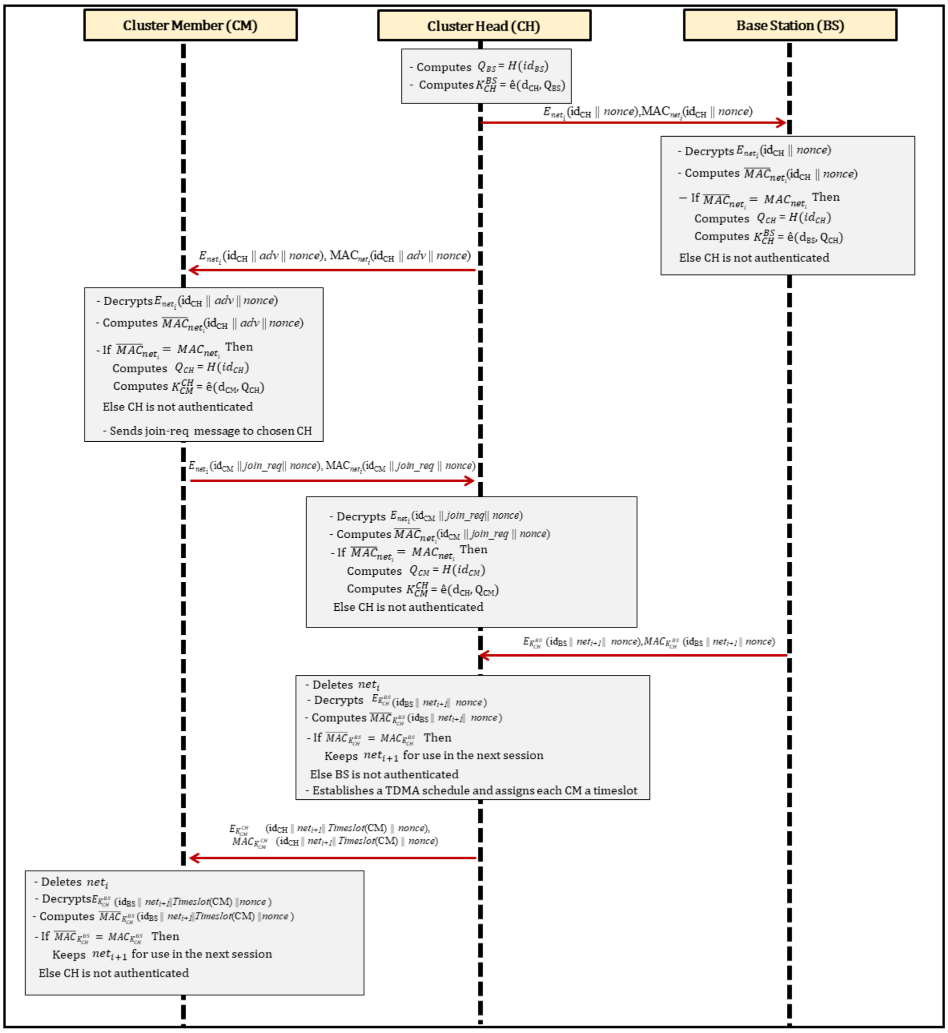

Cluster Member (CM)
Cluster Head (CH)
Base Station (BS)
- Computes $Q_{BS} = H(id_{BS})$
- Computes $K_{CH}^{BS} = \hat{e}(d_{CH}, Q_{BS})$
$E_{net_i}(id_{CH} \| nonce)$,$MAC_{net_i}(id_{CH} \| nonce)$
- Decrypts $E_{net_i}(id_{CH} \| nonce)$
- Computes $\overline{MAC}_{net_i}(id_{CH} \| nonce)$
– If $\overline{MAC}_{net_i} = MAC_{net_i}$ Then
Computes $Q_{CH} = H(id_{CH})$
Computes $K_{CH}^{BS} = \hat{e}(d_{BS}, Q_{CH})$
Else CH is not authenticated
$E_{net_i}(id_{CH} \| adv \| nonce)$, $MAC_{net_i}(id_{CH} \| adv \| nonce)$
- Decrypts $E_{net_i}(id_{CH} \| adv \| nonce)$
- Computes $\overline{MAC}_{net_i}(id_{CH} \| adv \| nonce)$
- If $\overline{MAC}_{net_i} = MAC_{net_i}$ Then
Computes $Q_{CH} = H(id_{CH})$
Computes $K_{CM}^{CH} = \hat{e}(d_{CM}, Q_{CH})$
Else CH is not authenticated
- Sends join-req message to chosen CH
$E_{net_i}(id_{CM} \| join\_req \| nonce)$, $MAC_{net_i}(id_{CM} \| join\_req \| nonce)$
- Decrypts $E_{net_i}(id_{CM} \| join\_req \| nonce)$
- Computes $\overline{MAC}_{net_i}(id_{CM} \| join\_req \| nonce)$
- If $\overline{MAC}_{net_i} = MAC_{net_i}$ Then
Computes $Q_{CM} = H(id_{CM})$
Computes $K_{CM}^{CH} = \hat{e}(d_{CH}, Q_{CM})$
Else CH is not authenticated
$E_{K_{CH}^{BS}}(id_{BS} \| net_{i+1} \| nonce)$,$MAC_{K_{CH}^{BS}}(id_{BS} \| net_{i+1} \| nonce)$
- Deletes net_i
- Decrypts $E_{K_{CH}^{BS}}(id_{BS} \| net_{i+1} \| nonce)$
- Computes $\overline{MAC}_{K_{CH}^{BS}}(id_{BS} \| net_{i+1} \| nonce)$
- If $\overline{MAC}_{K_{CH}^{BS}} = MAC_{K_{CH}^{BS}}$ Then
Keeps net_{i+1} for use in the next session
Else BS is not authenticated
- Establishes a TDMA schedule and assigns each CM a timeslot
$E_{K_{CM}^{CH}}(id_{CH} \| net_{i+1} \| Timeslot(CM) \| nonce)$,
$MAC_{K_{CM}^{CH}}(id_{CH} \| net_{i+1} \| Timeslot(CM) \| nonce)$
- Deletes net_i
- Decrypts$E_{K_{CH}^{BS}}(id_{BS} \| net_{i+1} \| Timeslot(CM) \| nonce)$
- Computes $\overline{MAC}_{K_{CH}^{BS}}(id_{BS} \| net_{i+1} \| Timeslot(CM) \| nonce)$
- If $\overline{MAC}_{K_{CH}^{BS}} = MAC_{K_{CH}^{BS}}$ Then
Keeps net_{i+1} for use in the next session
Else CH is not authenticated

FIGURE 5.1: Clustering phase.

5.3.1 Privacy-preserving

In our protocol's clustering and data transmission phases, the sensed data and the node's identity are encrypted using symmetric encryption. Furthermore, encryption keys are generated dynamically and are renewed at each round. As a result, privacy is preserved as an adversary cannot obtain the data and the node's identity without the encryption key.

5.3.2 Mutual authentication

In the IDSP scheme, the CM and the CH are mutually authenticated during clustering and data transmission phases. Indeed, a MAC and a nonce are included in

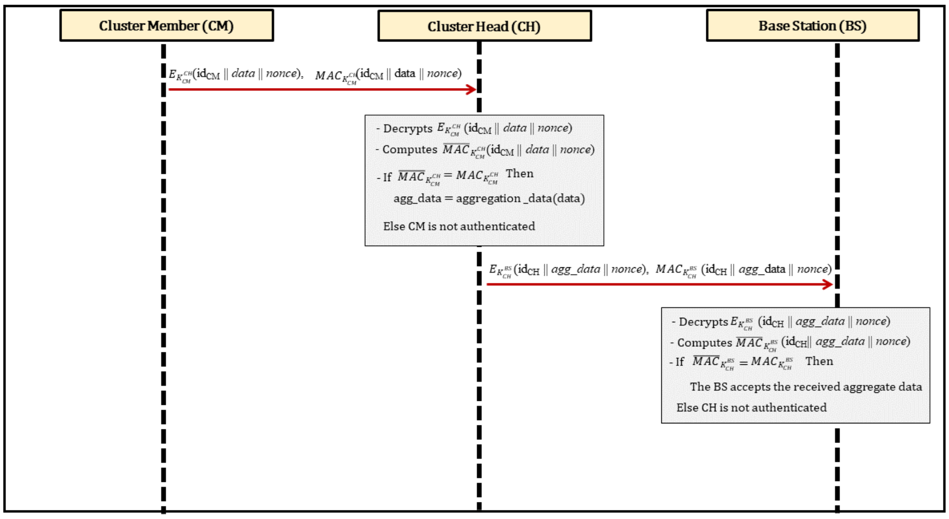

Cluster Member (CM)
Cluster Head (CH)
Base Station (BS)
$E_{K_{CM}^{CH}}(\mathrm{id}_{\mathrm{CM}} \| data \| nonce)$, $MAC_{K_{CM}^{CH}}(\mathrm{id}_{\mathrm{CM}} \| \mathrm{data} \| nonce)$
- Decrypts $E_{K_{CM}^{CH}}(\mathrm{id}_{\mathrm{CM}} \| data \| nonce)$
- Computes $\overline{MAC}_{K_{CM}^{CH}}(\mathrm{id}_{\mathrm{CM}} \| data \| nonce)$
- If $\overline{MAC}_{K_{CM}^{CH}} = MAC_{K_{CM}^{CH}}$ Then
agg_data = aggregation _data(data)
Else CM is not authenticated
$E_{K_{CH}^{BS}}(\mathrm{id}_{\mathrm{CH}} \| agg\_data \| nonce)$, $MAC_{K_{CH}^{BS}}(\mathrm{id}_{\mathrm{CH}} \| \mathrm{agg\_data} \| nonce)$
- Decrypts $E_{K_{CH}^{BS}}(\mathrm{id}_{\mathrm{CH}} \| agg\_data \| nonce)$
- Computes $\overline{MAC}_{K_{CH}^{BS}}(\mathrm{id}_{\mathrm{CH}} \| agg\_data \| nonce)$
- If $\overline{MAC}_{K_{CH}^{BS}} = MAC_{K_{CH}^{BS}}$ Then
The BS accepts the received aggregate data
Else CH is not authenticated

FIGURE 5.2: Data transmission phase.

the transmitted message to provide authentication, integrity, and freshness. The adversary cannot be authenticated as a legitimate node (CM or CH) because he/she cannot compute a MAC value without the shared key or global key.

5.3.3 MITM attack

IDSP protects against this cyber-attack by providing authentication node-to-node and securing the wireless channel. This means that all messages exchanged are encrypted and protected by MAC values.

5.3.4 Cluster head impersonation

Although a malicious node may claim to be a CH, each CM should authenticate the CH before sending the data. A malicious node cannot compute the correct MAC values without using shared or global keys. This prevents it from being authenticated. Therefore, the proposed protocol resists a CH impersonation.

5.3.5 Replay attack

With IDSP protocol, messages are always exchanged freshly, and old messages that an attacker has intercepted cannot be replayed. The reason for this is adding a nonce to the transmitted message. Additionally, sent messages are encrypted and authenticated, preventing an attacker from altering a nonce value.

5.3.6 Selective forwarding attack

A malicious node that becomes CH selectively forwards some messages from neighboring nodes and drops the rest. The IDSP protocol prevents malicious nodes from becoming CHs since all communications with forged MAC values are rejected. Furthermore, the keys are regularly renewed to prevent key compromise.

5.3.7 Brute force attack

The adversary attempts to discover the correct secret keys by testing many potential keys to decrypt the exchanged messages. The adversary will find it difficult to discover the shared and global keys in the proposed IDSP since they are dynamic and are renewed at every round.

5.3.8 DoS attack

DoS attacks involve the adversary sending fake messages to disrupt the services of a targeted node. In the IDSP protocol, the received MAC is verified upon receiving the transmitted message, and only data from authentic nodes are processed. Therefore, the proposed scheme is secure against DoS attacks.

5.4 Experiments, evaluation and simulation results

Based on our performance analysis, the IDSP protocol has been implemented in Contiki OS [26], a lightweight operating system designed for WSNs and IoT devices. In this analysis, we examine the following metrics: energy consumption and key storage requirement. In addition, we analyze the IDSP protocol as compared to the Hu scheme [43] discussed in chapter 3. This choice depends on the scope of the relevance. Cooja's network simulator [29] is used to test IDSP and Hu schemes. The performance was measured on WiSMote platform [25, 97, 98] which is equipped with MSP-430F5437 MCU, 256 KB of flash memory, 16 KB of SRAM and CC2520 radio chip. To perform MTP and pairing operations on elliptic curve, we used a lightweight asymmetric cryptographic library and suitable for WSN and IoT devices, called RELIC toolkit [5], with specifying 160-bit ECC in order to provide the

80-bit security level. We applied the MMH algorithm[100] to compute MAC values. We also applied AES algorithm (16 bytes length) for symmetric encryption. To model the lossy environment, we adopted the Unit Disk Graph Medium (UDGM) with distance loss as the propagation model. Note that the UDGM is included with Cooja simulator and takes into account two configurable parameters, including a Transmission Range (TR) and an Interference Range (IR). In our simulation, TR and IR are 100 m and 120 m, respectively. It is well-known that a WSN is not a secure environment, where it is subject to malicious attacks. During the experiment process, we have injected 05 malicious nodes to disrupt the network operation. The parameters of our simulation are given in Table 5.2.

TABLE 5.2: Simulation parameters

Parameter	Configuration
Operating system / Simulator	Contiki OS / Cooja
Network size	100 nodes
Nodes distribution	Randomly distributed
Simulation time	1500 s
Radio interface	CC2520 2.4 GHz (IEEE 802.15.4)
Sensor node type	WiSMote platform
Propagation model	UDGM with distance loss
Transmission range	100 m
Interference range	120 m
Data rate	250 kbps
Initial energy	20 Joules
Number of CHs	10% of the network
Number of malicious nodes	05

5.4.1 Energy consumption

To estimate the energy consumption, we first obtain the time spent in each power state, including CPU active mode, CPU sleep mode (also known as LPM: Low Power Mode), radio transmission (TX), and radio listen (RX). This task is performed by using an included system in Contiki OS called PowerTrace[27]. PowerTrace provides power state time in ticks, and in order to obtain the time in seconds, it should be divided (time) by 32768, which is the number of ticks per second. Based on the current values of the WiSMote platform [97],[98], and after obtaining the power states times, we calculate the energy consumption using the following formula:

$$E_{(mJ)} = \frac{(T_c * 2.2 + T_l * 0.00169 + T_x * 33.6 + T_r * 18.5) * 3}{32768}$$

- T_c, T_l, T_x and T_r are the times spent in CPU, LPM, TX and RX respectively.
- According to WiSMote platform, the current values for CPU, LPM, TX and RX are 2.2 mA, 0.00169 mA, 33.6 mA and 18.5 mA respectively [97],[98]. Moreover, the supply voltage is 3V.

Figure 5.3 illustrates the total energy consumed in our protocol and the Hu scheme. According to the comparison in Figure 5.3, the Hu scheme consumes more energy than IDSP. This is primarily due to the fact that IDSP uses the smallest bandwidth possible (only the identities are exchanged) in the establishment of a shared key. Furthermore, the Hu scheme uses the BF-IBE scheme to encrypt/decrypt public keys. Additionally, encrypted public keys are included within the exchanged message, increasing computation and communication overhead. Thus, the energy consumption increases.

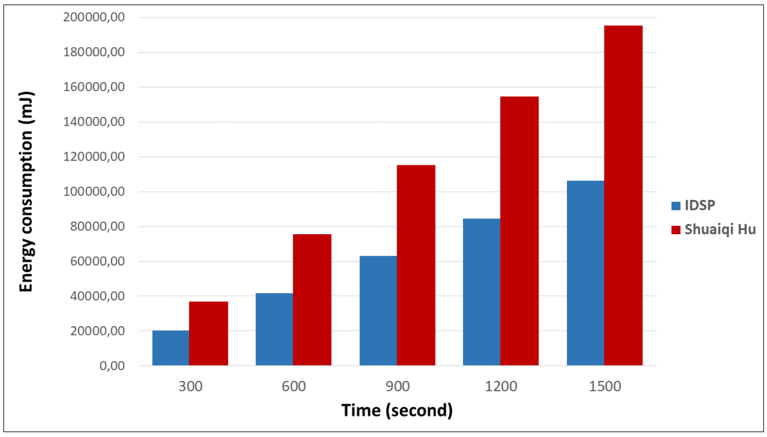


FIGURE 5.3: Total energy consumption.

5.4.2 Key storage requirement

Figure 5.4 depicts the average amount of memory necessary to store cryptographic keys in both CH and CM. The IDSP protocol requires 353 bytes for CH and 154 bytes for CM. In the Hu scheme, CH and CM require 399 bytes and 198 bytes, respectively. It is noteworthy that the average amount of memory necessary for storing the cryptographic keys is lower with IDSP than Hu. This is due to the size and number of keys used. Further details are given in Table 5.3. The value m in the two tables refers to the number of CMs in each cluster.

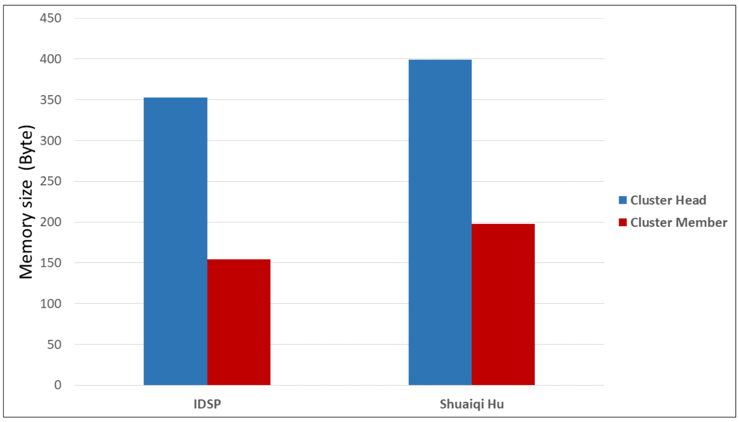


FIGURE 5.4: The average memory size required by a sensor node to store cryptographic keys.

TABLE 5.3: Key storage requirement

IDSP protocol

Key	Size (byte)	CH	CM
Identity-Based Cryptography Keys			
ID (as public key)	2	1	1
ID-based private key	120	1	1
Symmetric Cryptography Keys			
Global key	16	1	1
Shared key	16	m	1

Hu scheme

Key	Size (byte)	CH	CM
Elliptic Curve Cryptography Keys			
ECC public key	40	1	1
ECC private key	20	1	1
Identity-Based Cryptography Keys			
ID (as public key)	2	1	1
ID-based private key	80	1	1
Master public key	40	1	1
Symmetric Cryptography Keys			
Shared key	16	m	1

5.5 Conclusion

This research activity is primarily focused on proposing a secure version of the LEACH protocol called IDSP. The proposed protocol secures data communication in cluster-based WSNs by utilizing IBC a bilinear pairing. Indeed, the BS serves as PKG which generates sensor nodes' identities and the corresponding private keys and then embeds the private keys in the nodes prior to its use in the field, and no secret channel is needed for shared key establishment. Thus, only the identities of BS, CHs and CMs are exchanged without sending public keys and their certificates. This results in energy saving for the communication. Our simulation results showed that

IDSP is performing well in terms of energy consumption and key storage requirement. Furthermore, it is resistant to specific cyber-attacks, such as replay, selective forwarding and MITM attacks. In the next chapter, we propose a new IBC scheme based on ECC to secure the communication in homogeneous clustered WSNs.

Chapter 6

IBAKAS: A new Identity-Based Authentication and Key Agreement Scheme for CWSNs

Chapter Overview: *This chapter aims to present a new identity-based authentication and key agreement scheme for CWSNs referred to as IBAKAS. This scheme uses both ECC and IBC to provide mutual authentication and establish secret session keys over insecure channels. The IBAKAS achieves all of the desired security properties of key agreements and prevents specific cyberattacks against the CWSN. Furthermore, AVISPA is used to verify the formal security of the proposed scheme. Comparing the proposed scheme with existing relevant schemes, the IBAKAS scheme decreases computational, and communication overhead, requires less storage space for keys and prolongs the lifetime of the network by reducing the energy consumption of the sensor node.*

6.1 Introduction

A network's security is determined by policies, mechanisms, and services that protect against unauthorized access and cyber-attacks [108]. The security of CWSN is subject to several challenges, particularly when it comes to applications that require a high level of security, such as emergency response, military and medical services [9, 45]. Sensing devices are frequently placed in untrusted or hostile environments, making them more susceptible to cyber-attacks that may compromise sensitive data and impair network performance [47, 14]. In addition, the wireless communication within the CWSN is inherently insecure, and as a result, an adversary with wireless can monitor communications among legitimate nodes. Consequently, minimum security requirements, such as confidentiality, authentication, and integrity, should be met. Additionally, it is necessary to develop a lightweight, efficient, and secure scheme that considers the constraints of resource-constrained sensor nodes. In this context, we propose IBAKAS: an identity-based key agreement and authentication scheme for CWSNs. Accordingly, IBAKAS relies on IBC and ECC to establish a secret session key and achieve mutual authentication between two parties communicating through an insecure channel. The secret session key is used for secure data transmission between CH and CM or between CH and BS. 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 6.4), 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 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 [12]. A comparison of security features with the existing relevant schemes is also provided in this research activity (See Table 6.2).

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.

The remainder of this chapter is organized as follows. In Section 6.2, we describe the system model. Section 6.3 illustrates the phases of our proposed scheme (IBAKAS). The security analysis and the performance results are presented in Sections 6.4 and 6.5, respectively. The Section 6.6 describes two examples of application scenarios. The last section concludes this work with a summary.

6.2 System model

This section presents our network model and security properties of key agreement.

6.2.1 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 [46]. 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 6.1.

6.2.2 Security properties of key agreement schemes

According to Blake-Wilson et al. [12], key agreement schemes should achieve the following security properties.

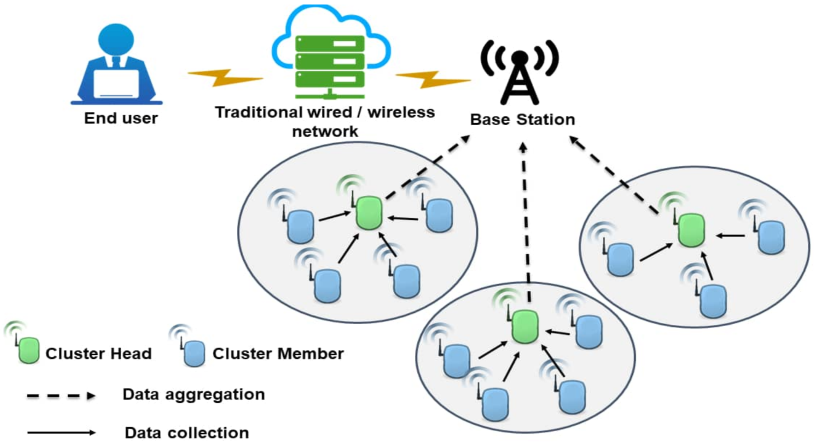


FIGURE 6.1: Network model.

- **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** Either participating nodes shouldn't preselect a session key.

6.3 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 6.1 lists the notations used in the proposed scheme. Below are the descriptions of each phase.

6.3.1 System initialization

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 6.1: List of notations used in the IBAKAS scheme

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. 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: $H_0 : \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_q^*$, $H_1 : \{0,1\}^* \times \mathbb{G}^2 \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}^3 \to \{0,1\}^k$. 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. 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:

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

6.3.2 Mutual authentication and key agreement

As shown in Figure 6.2, 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:

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

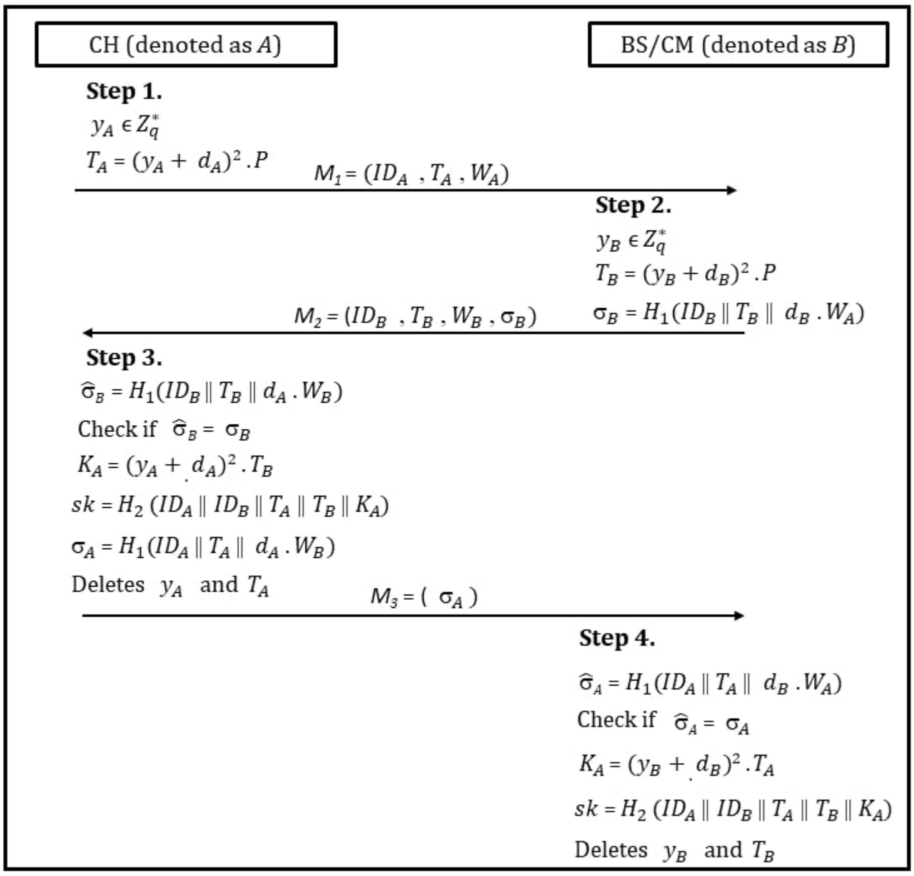

CH (denoted as A)
BS/CM (denoted as B)
Step 1.
$y_A \in Z_q^*$
$T_A = (y_A + d_A)^2 . P$
$M_1 = (ID_A , T_A , W_A)$
Step 2.
$y_B \in Z_q^*$
$T_B = (y_B + d_B)^2 . P$
$\sigma_B = H_1(ID_B \| T_B \| d_B . W_A)$
$M_2 = (ID_B , T_B , W_B , \sigma_B)$
Step 3.
$\widehat{\sigma}_B = H_1(ID_B \| T_B \| d_A . W_B)$
Check if $\widehat{\sigma}_B = \sigma_B$
$K_A = (y_A + d_A)^2 . T_B$
$sk = H_2 (ID_A \| ID_B \| T_A \| T_B \| K_A)$
$\sigma_A = H_1(ID_A \| T_A \| d_A . W_B)$
Deletes y_A and T_A
$M_3 = (\sigma_A)$
Step 4.
$\widehat{\sigma}_A = H_1(ID_A \| T_A \| d_B . W_A)$
Check if $\widehat{\sigma}_A = \sigma_A$
$K_A = (y_B + d_B)^2 . T_A$
$sk = H_2 (ID_A \| ID_B \| T_A \| T_B \| K_A)$
Deletes y_B and T_B

FIGURE 6.2: Mutual authentication and key agreement phase in the proposed scheme.

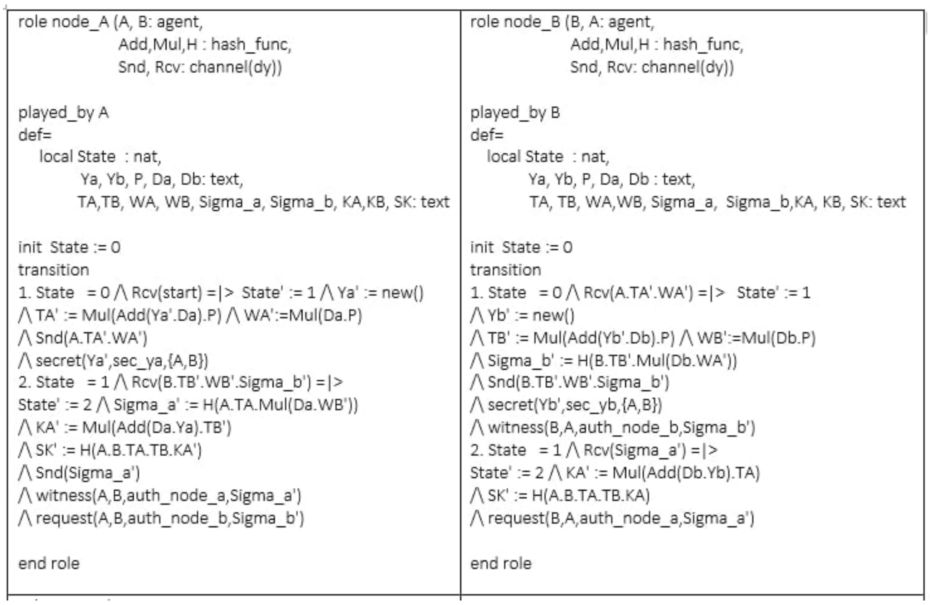
6.4 Security analysis of the proposed scheme

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

6.4.1 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) [6, 104]. 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) 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 [23], 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 6.3 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 6.4.

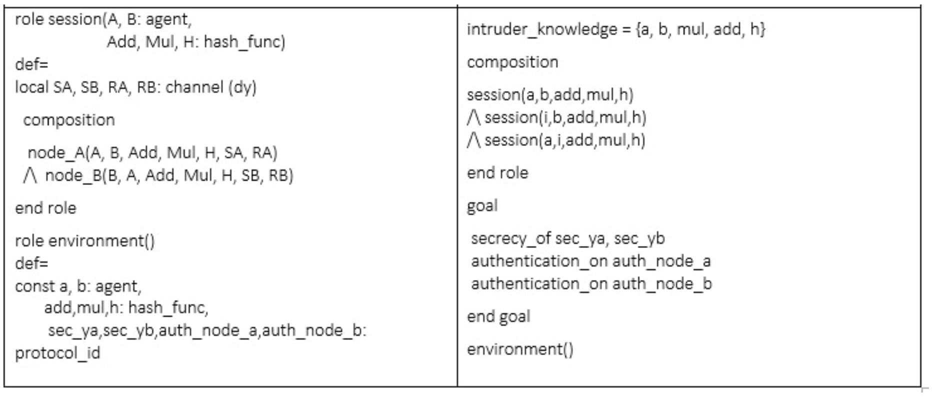


```
role node_A (A, B: agent,
             Add,Mul,H : hash_func,
             Snd, Rcv: channel(dy))

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 6.3: The basic roles in HLPSL.

2. **Verification results:** Figure 6.5 presents the verification results of IBAKAS under OFMC and CL-AtSe back-ends. These results indicate that security goals

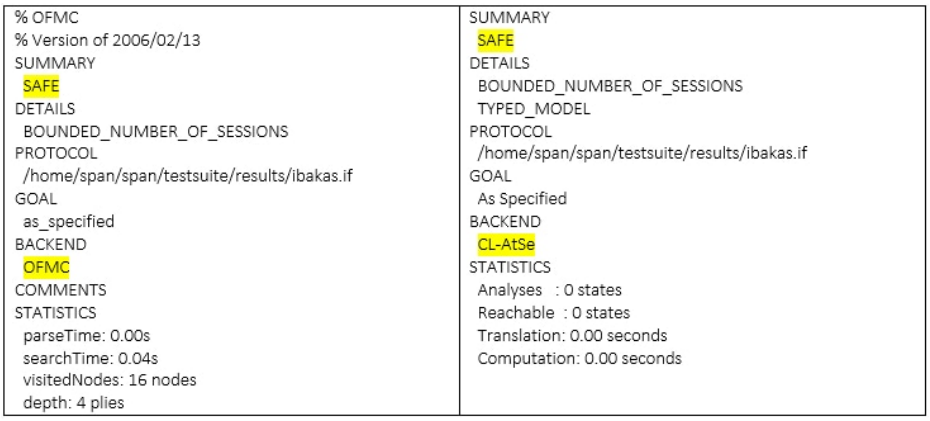


```
role session(A, B: agent,
            Add, Mul, H: hash_func)
def=
local SA, SB, RA, RB: channel (dy)

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

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

6.4.2 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 impersonate 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 established 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 [68, 37, 85, 35, 61] is provided in Table 6.2

6.5 Performance evaluation

In our performance study, we have implemented the IBAKAS scheme in Contiki OS [26], a lightweight operating system designed for WSN and IoT devices. As well, IBAKAS and existing relevant schemes [68, 37, 85, 35, 61] are tested using the Cooja network simulator [29]. The performance was measured on the WiSMote sensor device [25, 97, 98], which is equipped with MSP430F5437A MCU, 256 KB of flash memory, 16 KB of SRAM, and CC2520 radio chip. For operations on an elliptic

TABLE 6.2: 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
IBAKAS 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.

curve, we used a lightweight asymmetric cryptographic library suitable for WSN and IoT devices, known as RELIC toolkit [5], 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.

6.5.1 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 [68], MAKA [37], AKAIoTs [85], LSTR [35], and Kumar et al. [61]. It is clear that IBAKAS, AKAIoTs, and Kumar et al. are ECC-based schemes, while the others utilize a pairing technique.

Computation cost

Given that the BS is a powerful device, in this chapter 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 6.3 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 6.4. As seen in this Table, the MTP function and pairing-related operations are computationally expensive.

Figure 6.6(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 6.3, each sensor node executes only 4 point multiplications and 3 one-way hash functions to achieve an authentication and establish a single session key.

TABLE 6.3: 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 6.4: 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

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 6.5 shows the total computation time for IBAKAS and the cluster-based schemes, including ICMDS, MAKA, and LSTR. In this comparison, the number of clusters varies from 2 to 10. Each cluster contains 9 CMs. Based on Table 6.5, we demonstrate that the IBAKAS scheme is lightweight and offers better computation efficiency compared to ICMDS, MAKA and LSTR schemes.

TABLE 6.5: 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

Communication cost

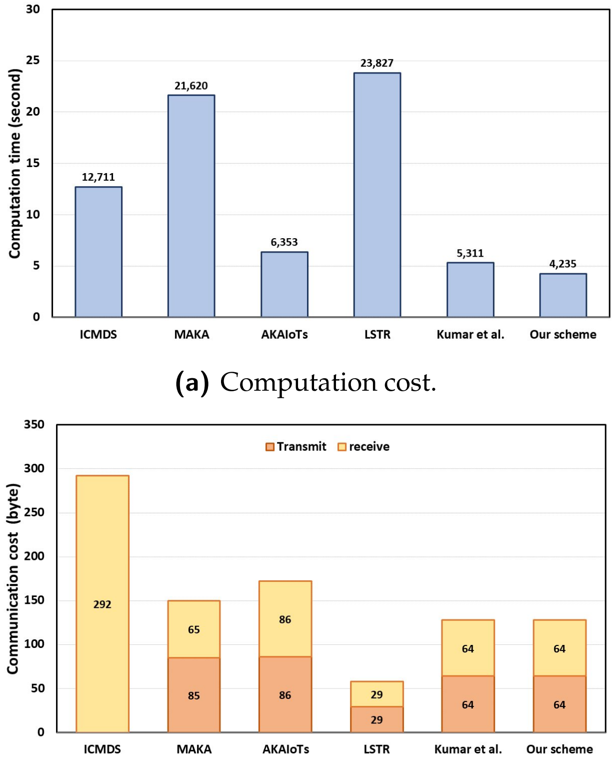
We assume that $|ID|$ and $|nonce|$ are each 2 bytes in size. In the schemes ICMDS [68], MAKA [37] and LSTR [35], we use the pairing-friendly curve BN-P158 over a 158-bit primary field. According to this curve, the size of an element in the groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ is 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.[95]. The size of messages transmitted and received by the schemes are as follows:

- The ICMDS scheme requires a sensor node to receive $(P_{pub}, R, C, C_0, C_1, \dots, C_m)$, 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 (PU, EM), where $EM \in \{|\mathbb{G}_1| + |\mathbb{G}_1| + |nonce|\}$ and $PU \in |\mathbb{G}_2|$. Additionally, it requires a sensor to receive (P, EM), where $EM \in \{|\mathbb{G}_1| + |\mathbb{G}_1| + |nonce|\}$ 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 $PDU \in \{|ID| + |ID| + |ID| + |\mathbb{G}_1| + |nonce|\}$. In addition, LSTR requires a sensor node to receive the same size message as it transmitted. Therefore, the size of a transmitted message is $3|ID| + |\mathbb{G}_1| + |nonce| = 3 \times 2 + 21 + 2 = 29$ bytes. The size of a received message is 29 bytes.

According to AKAIoTs [85], Kumar et al. [61], 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.



(a) Computation cost.

(b) Communication cost.

FIGURE 6.6: Computation and communication costs required by a sensor node to establish one session key.

- 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 6.6(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 6.2, the LSTR scheme suffers from a lack of security features, such as mutual authentication, Unknown key share, and key-compromise impersonation resilience.

Energy consumption

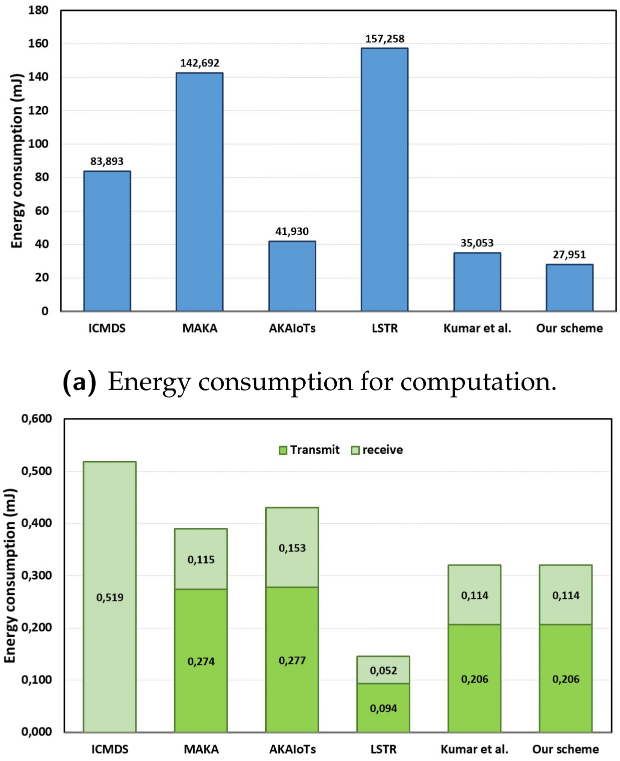
To evaluate the energy consumption associated with computation and communication, we use the equations [95] $W_{comp} = V \times I_c \times t$ and $W_{tx/rx} = V \times I_{tx/rx} \times U \times \frac{8}{dr}$, respectively. Where $W_{comp/tx/rx}$ represents the energy in millijoules (mJ), V is the voltage, 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 [97, 98].

Figure 6.7 illustrates the energy consumed by a sensor node for (a) the computation process and (b) the transmission/reception of messages. From Figure 6.7(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 6.7(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 6.8 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 6.5. Thus, the total energy consumption is also lower.

Key storage cost

Because sensor nodes are resource-constrained, key storage overhead is an important factor to consider. Figure 6.9 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 $76 + 100 = 176$ bytes,



(a) Energy consumption for computation.

(b) Energy consumption for communication.

FIGURE 6.7: Energy consumed by a sensor node to establish one session key.

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.

6.6 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.

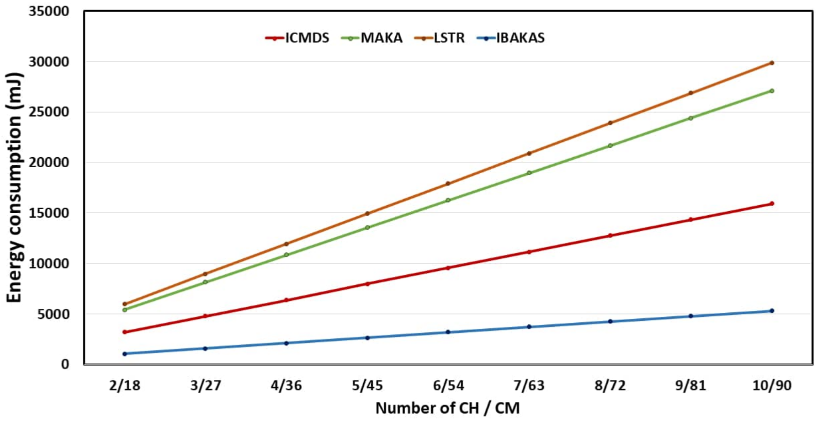


FIGURE 6.8: Total energy consumption according to a number of clusters.

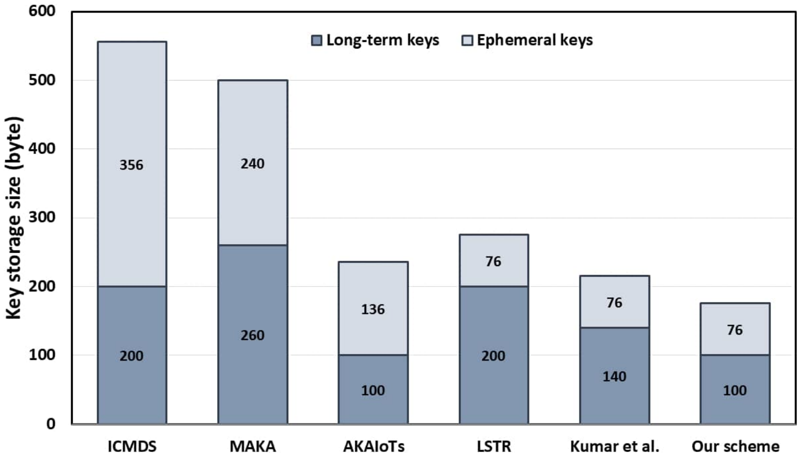


FIGURE 6.9: Key storage size required by a sensor node.

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 6.10 illustrates the proposed architecture.

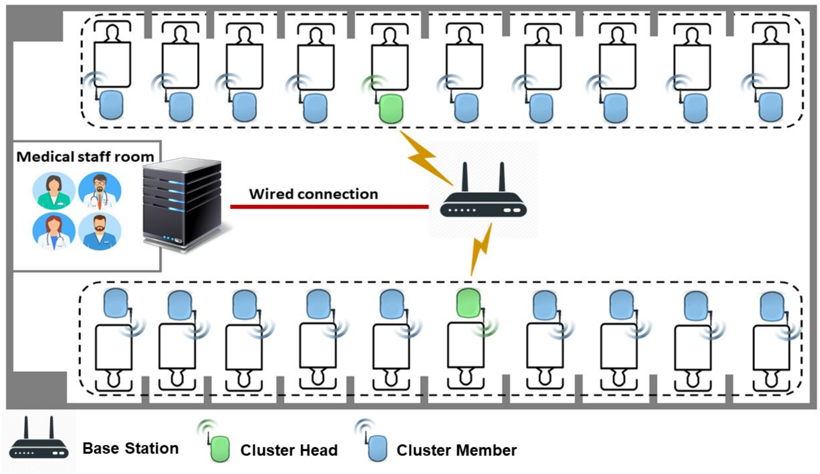


FIGURE 6.10: Patient's monitoring in the field hospital.

6.7 Conclusion

This chapter proposed an Identity-Based Authentication and Key Agreement Scheme (IBAKAS) for CWSN. We aimed to achieve the best possible balance between security and lightness with our design. 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.

General conclusion

CWSN security is a challenging task facing this type of network. Since CWSNs are dynamic by nature and deployed in open areas, they are susceptible to different types of cyber-attacks that can adversely affect their functioning. Additionally, the limited resources of sensor devices make it impossible to implement conventional security schemes in CWSN, which require a high overhead of computation, communications, and increased energy consumption. Cryptography is considered a security technique is protecting communication in open networks such as CWSN. In addition, the use of such technique techniques guarantees the basic requirements of security in CWSN.

SKC-based schemes are highly efficient in terms of computation overhead and energy consumption. Nevertheless, the distribution of keys presents a significant challenge. Additionally, such schemes do not provide a good balance between resilience and the storage of cryptographic keys. The first attempts at implementing PKC in sensor networks using RSA have proven to be infeasible due to its large key size. Additionally, its cryptographic primitives are computationally expensive.

Further research in this area has demonstrated that ECC is a more suitable method of PKC for resource-constrained devices due to its small key sizes and faster execution time. However, to use PKC schemes in CWSNs, a public key must be authenticated. The use of a PKI is not feasible in sensor networks due to the overhead and complexity associated with the operation of public-key certificates, including their distribution, storage, and verification. A PKC extension called IBC was introduced by Shamir. In such a cryptosystem, an entity's public key is derived from its identity. A trusted third party, known as PKG, is responsible for issuing the corresponding private key. Compared to PKI, IBC offers several advantages. It can provide easy public keys management, and there is no need to generate and maintain public key certificates. Consequently, IBC requires low computational and communication overheads.

In this thesis, we studied the security issues and proposed lightweight and robust solutions to ensure CWSN security without compromising performance requirements. Initially, we discussed WSN security, including security constraints, security requirements, and potential cyber-attacks that may target wireless sensor networks. Then, we reviewed existing PKC-based schemes in WSNs, including IBC schemes based on pairings, IBC schemes based on ECC, and schemes based on ECC. We also compared these schemes according to several criteria, highlighting their shortcomings. Subsequently, we proposed three security solutions to enhance the security in CWSN. All of these proposals have as their common objective the trade-off between efficiency and security. They provide a high level of security and require lightweight operations.

Firstly, we proposed a secure version of LEACH referred to as HCBS (Hybrid Cryptography-Based Scheme) for secure data communication in cluster-based WSNs. The protocol was built upon symmetric and asymmetric cryptographic primitives. HCBS is able to satisfy the most important security requirements and resists attacks that pose a threat to LEACH. According to tests conducted on the TOSSIM simulator, our proposal achieved favorable performance in terms of energy consumption, loss rate, and end-to-end delay.

Secondly, we proposed a lightweight and secure version of the LEACH protocol called IDSP. Data routing from the CMs to the BS is secured based on an ID-based authenticated key agreement with bilinear pairing. The IDSP protocol reduces the number of required keys to a signal-ephemeral global key and one long-term private key preloaded in the sensor nodes. The keys are then used to compute pairwise shared keys between nodes without the need for digital certificates. Therefore, IDSP can provide both node authenticity and confidentiality during data transmission at a low cost without the need for BS. The IDSP protocol has been tested in the Cooja network simulator with the WiSMote platform, and the results showed that IDSP is lightweight and secure.

Thirdly, we proposed an identity-based authentication and key agreement scheme for CWSNs (IBAKAS), which combines IBC and ECC 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 CWSNs. Moreover, the AVISPA tool is used to verify the formal security of the proposed scheme. Compared to existing relevant schemes, the proposed scheme reduces computation and communication overheads, saves storage space for keys, and prolongs the network lifetime by reducing the energy consumption of the sensor node.

As a future work, our proposals will be extended with more research. Our future perspectives are summarized as follows:

1. We aim to extend our proposals to support blockchain-based IoT [88] 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 proposals on real resource-constrained sensor devices.

3. We will validate our proposals using the Random Oracle Model (ROM) [80].

Bibliography

[1] IEEE 802.15.4. *IEEE 802.15 Working Group for Wireless Specialty Networks (WSN)*. [Online]. Available: `https://www.ieee802.org/15/`. Accessed Nov 2021.

[2] Ian F Akyildiz et al. "A survey on sensor networks". In: *IEEE Communications magazine* 40.8 (2002), pp. 102–114.

[3] Connectivity Standards Alliance. *ZigBee Alliance*. [Online]. Available: `https://zigbeealliance.org/about/`. Accessed Nov 2021.

[4] Hisham N Almajed and Ahmad S Almogren. "SE-ENC: A secure and efficient encoding scheme using elliptic curve cryptography". In: *IEEE Access* 7 (2019), pp. 175865–175878.

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

[6] Alessandro Armando et al. "The AVISPA tool for the automated validation of internet security protocols and applications". In: *International conference on computer aided verification*. Springer. 2005, pp. 281–285.

[7] International Society of Automation. *ISA100.11a*. [Online]. Available: `http://www.isa.org`. Accessed Nov 2021.

[8] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. "Security proofs for identity-based identification and signature schemes". In: *Journal of Cryptology* 22.1 (2009), pp. 1–61.

[9] Ayoub Benayache et al. "MsM: A microservice middleware for smart WSN-based IoT application". In: *Journal of Network and Computer Applications* 144 (2019), pp. 138–154.

[10] UC Berkeley. *Tmote Sky platform DataSheet*. [Online]. Available: `https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf`. Accessed Nov 2021.

[11] Bharat Bhushan and Gadadhar Sahoo. "Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks". In: *Wireless Personal Communications* 98.2 (2018), pp. 2037–2077.

[12] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. "Key agreement protocols and their security analysis". In: *IMA international conference on cryptography and coding*. Springer. 1997, pp. 30–45.

[13] Dan Boneh and Matt Franklin. "Identity-based encryption from the Weil pairing". In: *Annual international cryptology conference*. Springer. 2001, pp. 213–229.

[14] Djallel Eddine Boubiche et al. "Cybersecurity Issues in Wireless Sensor Networks: Current Challenges and Solutions." In: *Wireless Personal Communications* 117.1 (2021).

[15] Omar Rafik Merad Boudia, Sidi Mohammed Senouci, and Mohammed Feham. "A novel secure aggregation scheme for wireless sensor networks using stateful public key cryptography". In: *Ad Hoc Networks* 32 (2015), pp. 98–113.

[16] D Brown. "Standards for efficient cryptography, SEC 1: elliptic curve cryptography". In: *Released Standard Version* 1 (2009).

[17] Claude Castelluccia, Einar Mykletun, and Gene Tsudik. "Efficient aggregation of encrypted data in wireless sensor networks". In: *The second annual international conference on mobile and ubiquitous systems: networking and services*. IEEE. 2005, pp. 109–117.

[18] Sanjit Chatterjee and Alfred Menezes. "On cryptographic protocols employing asymmetric pairings — The role of Ψ revisited". In: *Discrete Applied Mathematics* 159.13 (2011), pp. 1311–1322.

[19] Liqun Chen and Caroline Kudla. "Identity Based Authenticated Key Agreement Protocols from Pairings". In: *16th IEEE Computer Security Foundations Workshop, 2003. Proceedings.* 2003, pp. 219–233.

[20] Sarika Choudhary and Nishtha Kesswani. "Detection and prevention of routing attacks in internet of things". In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2018, pp. 1537–1540.

[21] Roberto Di Pietro et al. "Security in wireless ad-hoc networks–a survey". In: *Computer Communications* 51 (2014), pp. 1–20.

[22] Whitfield Diffie and Martin Hellman. "New directions in cryptography". In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[23] Danny Dolev and Andrew Yao. "On the security of public key protocols". In: *IEEE Transactions on information theory* 29.2 (1983), pp. 198–208.

[24] Hongzhen Du et al. "A new provably secure certificateless signature scheme for Internet of Things". In: *Ad Hoc Networks* 100 (2020), p. 102074.

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

[26] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors". In: *29th annual IEEE international conference on local computer networks*. IEEE. 2004, pp. 455–462.

[27] Adam Dunkels et al. *Powertrace: Network-level power profiling for low-power wireless networks*. 2011.

[28] Mohamed Elhoseny et al. "A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption". In: *Journal of King Saud University-Computer and Information Sciences* 28.3 (2016), pp. 262–275.

[29] Joakim Eriksson et al. "COOJA/MSPSim: interoperability testing for wireless sensor networks". In: *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. 2009, pp. 1–7.

[30] Fakhrosadat Fanian and Marjan Kuchaki Rafsanjani. “Cluster-based routing protocols in wireless sensor networks: A survey based on methodology”. In: *Journal of Network and Computer Applications* 142 (2019), pp. 111–142.

[31] Amar Abdelmalek Ghehioueche, Noureddine Chikouche, and Fares Mezrag. “Performance Evaluation and Analysis of Encryption Schemes for Wireless Sensor Networks”. In: *2019 International Conference on Digitization (ICD)*. IEEE. 2019, pp. 187–191.

[32] Venkata C Giruka et al. “Security in wireless sensor networks”. In: *Wireless communications and mobile computing* 8.1 (2008), pp. 1–24.

[33] Carles Gomez and Josep Paradells. “Wireless home automation networks: A survey of architectures and technologies”. In: *IEEE Communications Magazine* 48.6 (2010), pp. 92–101.

[34] Zygmunt J Haas et al. “Current challenges and approaches in securing communications for sensors and actuators”. In: *The Art of Wireless Sensor Networks*. Springer, 2014, pp. 569–608.

[35] Khaled Hamouid, Salwa Othmen, and Amine Barkat. “LSTR: Lightweight and Secure Tree-Based Routing for Wireless Sensor Networks”. In: *Wireless Personal Communications* (2020), pp. 1–23.

[36] Darrel Hankerson, Scott Vanstone, and Alfred J Menezes. *Guide to Elliptic Curve Cryptography*. New York, NY: Springer, 2004.

[37] Yasmine Harbi et al. “Enhanced authentication and key management scheme for securing data transmission in the internet of things”. In: *Ad Hoc Networks* 94 (2019), p. 101948.

[38] Yasmine Harbi et al. “Secure data transmission scheme based on elliptic curve cryptography for internet of things”. In: *International Symposium on Modelling and Implementation of Complex Systems*. Springer. 2018, pp. 34–46.

[39] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. “Energy-efficient communication protocol for wireless microsensor networks”. In: *Proceedings of the 33rd annual Hawaii international conference on system sciences*. IEEE. 2000, 10–pp.

[40] Simon Heron. “Advanced encryption standard (AES)”. In: *Network Security* 2009.12 (2009), pp. 8–12.

[41] Florian Hess. “Efficient identity based signature schemes based on pairings”. In: *International Workshop on Selected Areas in Cryptography*. Springer. 2002, pp. 310–324.

[42] Jason Hill et al. “System architecture directions for networked sensors”. In: *ACM Sigplan notices* 35.11 (2000), pp. 93–104.

[43] Shuaiqi Hu. “A hierarchical key management scheme for wireless sensor networks based on identity-based encryption”. In: *2015 IEEE International Conference on Computer and Communications (ICCC)*. IEEE. 2015, pp. 384–389.

[44] Y-C Hu, Adrian Perrig, and David B Johnson. “Packet leashes: a defense against wormhole attacks in wireless networks”. In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*. Vol. 3. IEEE. 2003, pp. 1976–1986.

[45] Usha Jain and Muzzammil Hussain. “Securing wireless sensors in military applications through resilient authentication mechanism”. In: *Procedia Computer Science* 171 (2020), pp. 719–728.

[46] Wassim Jerbi, Abderrahmen Guermazi, and Hafedh Trabelsi. “O-LEACH of routing protocol for wireless sensor networks”. In: *2016 13th international conference on computer graphics, imaging and visualization (CGiV)*. IEEE. 2016, pp. 399–404.

[47] Jinfang Jiang et al. “A survey on location privacy protection in wireless sensor networks”. In: *Journal of Network and Computer Applications* 125 (2019), pp. 93–114.

[48] Sayamuddin Ahmed Jilani, Chandan Koner, and Shovon Nandi. “Security in Wireless Sensor Networks: Attacks and Evasion”. In: *2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA)*. IEEE. 2020, pp. 1–5.

[49] Antoine Joux. “A one round protocol for tripartite Diffie–Hellman”. In: *International algorithmic number theory symposium-Springer*. 2000, pp. 385–393.

[50] Yasin Kabalci. "IEEE 802.15. 4 technologies for smart grids". In: *Smart grids and their communication systems*. Springer, 2019, pp. 531–550.

[51] Jayaprakash Kar, Kshirasagar Naik, and Tamer Abdelkader. "A Secure and Lightweight Protocol for Message Authentication in Wireless Sensor Networks". In: *IEEE Systems Journal* (2020).

[52] Jamal N Al-Karaki and Ahmed E Kamal. "Routing techniques in wireless sensor networks: a survey". In: *IEEE wireless communications* 11.6 (2004), pp. 6–28.

[53] Eirini Karapistoli, Ioanna Mampentzidou, and Anastasios A Economides. "Environmental monitoring based on the wireless sensor networking technology: A survey of real-world applications". In: *International Journal of Agricultural and Environmental Information Systems (IJAEIS)* 5.4 (2014), pp. 1–39.

[54] Chris Karlof and David Wagner. "Secure routing in wireless sensor networks: Attacks and countermeasures". In: *Ad hoc networks* 1.2-3 (2003), pp. 293–315.

[55] M Keerthika and D Shanmugapriya. "Wireless Sensor Networks: Active and Passive attacks-Vulnerabilities and Countermeasures". In: *Global Transitions Proceedings* 2.2 (2021), pp. 362–367.

[56] Shafiullah Khan, Al-Sakib Khan Pathan, and Nabil Ali Alrajeh. *Wireless sensor networks: Current status and future trends*. CRC press, 2016.

[57] Anna N Kim et al. "When HART goes wireless: Understanding and implementing the WirelessHART standard". In: *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE. 2008, pp. 899–907.

[58] Jun Young Kim et al. "Long-term secure management of large scale Internet of Things applications". In: *Journal of Network and Computer Applications* 138 (2019), pp. 15–26.

[59] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[60] Neal Koblitz, Alfred Menezes, and Scott Vanstone. "The state of elliptic curve cryptography". In: *Designs, codes and cryptography* 19.2 (2000), pp. 173–193.

[61] Vivek Kumar and Sangram Ray. "Pairing-Free Identity-Based Digital Signature Algorithm for Broadcast Authentication Based on Modified ECC Using Battle Royal Optimization Algorithm". In: *Wireless Personal Communications* (2021), pp. 1–25.

[62] Vivek Kumar et al. "A Pairing Free Identity Based Two Party Authenticated Key Agreement Protocol Using Hexadecimal Extended ASCII Elliptic Curve Cryptography". In: *Wireless Personal Communications* (2021), pp. 1–17.

[63] HyungJune Lee, Alberto Cerpa, and Philip Levis. "Improving wireless simulation through noise modeling". In: *Proceedings of the 6th international conference on Information processing in sensor networks*. 2007, pp. 21–30.

[64] Philip Levis et al. "TOSSIM: Accurate and scalable simulation of entire TinyOS applications". In: *Proceedings of the 1st international conference on Embedded networked sensor systems*. 2003, pp. 126–137.

[65] An Liu and Peng Ning. "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks". In: *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*. IEEE. 2008, pp. 245–256.

[66] Huang Lu, Jie Li, and Mohsen Guizani. "Secure and efficient data transmission for cluster-based wireless sensor networks". In: *IEEE transactions on parallel and distributed systems* 25.3 (2013), pp. 750–761.

[67] Ming Lu. "Study on secret key management project of WSN based on ECC". In: *Journal of Networks* 7.4 (2012), p. 652.

[68] Amjad Mehmood, Muhammad Muneer Umar, and Houbing Song. "ICMDS: Secure Inter-Cluster Multiple-key Distribution Scheme for wireless sensor networks". In: *Ad Hoc Networks* 55 (2017), pp. 97–106.

[69] Mohamed Nidhal Mejri, Jalel Ben-Othman, and Mohamed Hamdi. "Survey on VANET security challenges and possible cryptographic solutions". In: *Vehicular Communications* 1.2 (2014), pp. 53–66.

[70] MEMSIC. *MicaZ platform DataSheet*. [Online]. Available: `http://courses.ece.ubc.ca/494/files/MICAz_Datasheet.pdf`. Accessed Nov 2021.

[71] Omar Rafik Merad Boudia, Sidi Mohammed Senouci, and Mohammed Feham. "Secure and efficient verification for data aggregation in wireless sensor networks". In: *International Journal of Network Management* 28.1 (2018), e2000.

[72] Fares Mezrag, Salim Bitam, and Abdelhamid Mellouk. "Secure routing in cluster-based wireless sensor networks". In: *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE. 2017, pp. 1–6.

[73] Konstantin Mikhaylov et al. "Wireless sensor networks in industrial environment: Real-life evaluation results". In: *2012 2nd Baltic Congress on Future Internet Communications*. IEEE. 2012, pp. 1–7.

[74] Victor S Miller. "Use of elliptic curves in cryptography". In: *Conference on the theory and application of cryptographic techniques*. Springer. 1985, pp. 417–426.

[75] Shivendu Mishra et al. "ESS-IBAA: Efficient, short, and secure ID-based authentication algorithm for wireless sensor network". In: *International Journal of Communication Systems* 34.8 (2021), e4764.

[76] Leonardo B Oliveira et al. "SecLEACH—On the security of clustered sensor networks". In: *Signal Processing* 87.12 (2007), pp. 2882–2895.

[77] Leonardo B Oliveira et al. "TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks". In: *Computer communications* 34.3 (2011), pp. 485–493.

[78] Luis ML Oliveira and Joel JPC Rodrigues. "Wireless Sensor Networks: A Survey on Environmental Monitoring." In: *J. Commun.* 6.2 (2011), pp. 143–151.

[79] Harsh Kupwade Patil and Stephen A Szygenda. *Security for Wireless Sensor Networks using Identity-Based Cryptography*. Auerbach Publications, 2012.

[80] David Pointcheval. "Provable security for public key schemes". In: *Contemporary cryptology*. Springer, 2005, pp. 133–190.

[81] Zhongyuan Qin et al. "A novel identity-based security scheme for wireless sensor networks". In: *2014 Tenth International Conference on Computational Intelligence and Security*. IEEE. 2014, pp. 662–666.

[82] Priyanka Rawat et al. “Wireless sensor networks: a survey on recent developments and potential synergies”. In: *The Journal of supercomputing* 68.1 (2014), pp. 1–48.

[83] Ronald L Rivest, Adi Shamir, and Leonard Adleman. “A method for obtaining digital signatures and public-key cryptosystems”. In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[84] Sankardas Roy et al. “Secure data aggregation in wireless sensor networks”. In: *IEEE Transactions on Information Forensics and Security* 7.3 (2012), pp. 1040–1052.

[85] Mutaz Elradi S Saeed et al. “AKAIoTs: authenticated key agreement for Internet of Things”. In: *Wireless Networks* 25.6 (2019), pp. 3081–3101.

[86] Ahmed Saidi, Khelifa Benahmed, and Nouredine Seddiki. “Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks”. In: *Ad Hoc Networks* 106 (2020), p. 102215.

[87] R Sakai, K Ohgishi, and M Kasahara. “Cryptosystems based on pairing, 2000. SCIS 2000”. In: *the 2000 symposium on cryptography and information security, Okinawa, japan, January*, pp. 26–28.

[88] Khaled Salah et al. “Blockchain for AI: Review and open research challenges”. In: *IEEE Access* 7 (2019), pp. 10127–10149.

[89] Adi Shamir. “Identity-based cryptosystems and signature schemes”. In: *Workshop on the theory and application of cryptographic techniques*. Springer. 1984, pp. 47–53.

[90] Gaurav Sharma, Suman Bala, and Anil K Verma. “PF-IBS: pairing-free identity based digital signature algorithm for wireless sensor networks”. In: *Wireless personal communications* 97.1 (2017), pp. 1185–1196.

[91] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*. Vol. 43. John Wiley & Sons, 2011.

[92] Limin Shen et al. "A secure and efficient id-based aggregate signature scheme for wireless sensor networks". In: *IEEE Internet of Things Journal* 4.2 (2016), pp. 546–554.

[93] Lei Shi et al. "Distributed localization in wireless sensor networks under denial-of-service attacks". In: *IEEE Control Systems Letters* 5.2 (2020), pp. 493–498.

[94] Kyung-Ah Shim. "A survey of public-key cryptographic primitives in wireless sensor networks". In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 577–601.

[95] Kyung-Ah Shim. "S2DRP: Secure implementations of distributed reprogramming protocol for wireless sensor networks". In: *Ad Hoc Networks* 19 (2014), pp. 1–8.

[96] Ayush Sogani and Aman Jain. "Energy aware and fast authentication scheme using identity based encryption in wireless sensor networks". In: *Cluster Computing* 22.5 (2019), pp. 10637–10648.

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

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

[99] Texas Instruments. *Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee(TM) Ready RF Transceiver*. [Online]. Available: `http://www.ti.com/product/CC2420`. Accessed March 2021.

[100] TinyOS-2.x. *Multilinear-Modular-Hashing function*. https://github.com/tyll/tinyos-2.x-contrib/.

[101] Ivana Tomić and Julie A McCann. "A survey of potential security issues in existing wireless sensor network protocols". In: *IEEE Internet of Things Journal* 4.6 (2017), pp. 1910–1923.

[102] Yuh-Min Tseng, Jian-Lun Chen, and Sen-Shan Huang. "A Lightweight Leakage-Resilient Identity-Based Mutual Authentication and Key Exchange Protocol for Resource-limited Devices". In: *Computer Networks* (2021), p. 108246.

[103] UC Berkeley. *nesC : A Programming Language for Deeply Networked Systems*. http://nescc.sourceforge.net/.

[104] Luca Vigano. "Automated security protocol analysis with the AVISPA tool". In: *Electronic Notes in Theoretical Computer Science* 155 (2006), pp. 61–86.

[105] D Vinodha and EA Mary Anita. "Secure data aggregation techniques for wireless sensor networks: a review". In: *Archives of Computational Methods in Engineering* 26.4 (2019), pp. 1007–1027.

[106] John Paul Walters et al. "Wireless sensor network security: A survey". In: *Security in distributed, grid, mobile, and pervasive computing*. Auerbach Publications, 2007, pp. 367–409.

[107] Tsu-Yang Wu and Yuh-Min Tseng. "An efficient user authentication and key exchange protocol for mobile client–server environment". In: *Computer Networks* 54.9 (2010), pp. 1520–1530.

[108] Mohammad Sadegh Yousefpoor and Hamid Barati. "Dynamic key management algorithms in wireless sensor networks: A survey". In: *Computer Communications* 134 (2019), pp. 52–69.

[109] Mohammad Sadegh Yousefpoor and Hamid Barati. "Dynamic key management algorithms in wireless sensor networks: A survey". In: *Computer Communications* 134 (2019), pp. 52–69. ISSN: 0140-3664.

[110] Mohammad Sadegh Yousefpoor et al. "Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review". In: *Journal of Network and Computer Applications* (2021), p. 103118.

[111] Erdong Yuan et al. "A key management scheme based on pairing-free identity based digital signature algorithm for heterogeneous wireless sensor networks". In: *Sensors* 20.6 (2020), p. 1543.

[112] Ke Zhang et al. "The application of a wireless sensor network design based on zigbee in petrochemical industry field". In: *2008 First International Conference on Intelligent Networks and Intelligent Systems*. IEEE. 2008, pp. 284–287.

[113] Kun Zhang, Cong Wang, and Cuirong Wang. "A secure routing protocol for cluster-based wireless sensor networks using group key management". In: *2008 4th international conference on wireless communications, networking and mobile computing*. IEEE. 2008, pp. 1–5.

[114] Shushan Zhao et al. "A survey of applications of identity-based cryptography in mobile ad-hoc networks". In: *IEEE Communications surveys & tutorials* 14.2 (2011), pp. 380–400.

[115] Hong Zhong et al. "An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks". In: *Journal of Parallel and Distributed Computing* 111 (2018), pp. 1–12.

[116] Zolertia. *RE-Mote platform-Online Resources and documentation*. [Online]. Available: `https://github.com/Zolertia/Resources/wiki`. Accessed Nov 2021.

[117] Zolertia. *Z1 platform DataSheet*. [Online]. Available: `http://wiki.zolertia.com/wiki/images/e/e8/Z1_RevC_Datasheet.pdf`. Accessed March 2021.