

# AN EFFICIENT AUTHENTICATION FRAMEWORK FOR WIRELESS SENSOR NETWORKS

by

REHANA YASMIN

A thesis submitted to  
The University of Birmingham  
for the degree of  
DOCTOR OF PHILOSOPHY

School of Computer Science  
College of Engineering and Physical Sciences  
The University of Birmingham  
November 2012

UNIVERSITY OF  
BIRMINGHAM

**University of Birmingham Research Archive**

**e-theses repository**

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

## **Abstract**

This study investigates the broadcast/multicast authentication problems in wireless sensor networks (WSNs), particularly sensor nodes broadcast authentication and outside user authentication, and proposes efficient and secure solutions for them. The low cost and immunity from cabling have become motivations for many applications of WSNs, for instance, the forest fire alarm, the intelligent traffic system etc. However, the sensitive nature of communication in these applications makes authentication a compulsory security requirement for them. The conventional security solutions are unfeasible for WSNs due to the unique features of sensor networks. Designing a new security mechanism for WSNs, on the other hand, is a challenging task due to the nature of WSNs.

This research proposes a solution to the above mentioned authentication problems in the form of an authentication framework for wireless sensor networks. The proposed framework is comprised of two authentication protocols: one for sensor nodes broadcast authentication and the other for outside user authentication. The latter also facilitates a third type of authentication, i.e., base station to sensor nodes broadcast authentication. These protocols can be applied in WSNs independently tackling individual security problems to achieve different level of security. However, deployed as a unified framework, they ensure a high degree of security with efficiency, providing a single solution to all three authentication problems in WSNs. The performance evaluation results showed that the proposed framework is the most efficient solution when compared to the existing authentication schemes for WSNs, giving a reasonable trade-off between security and efficiency.

*To my beloved parents who have supported me and prayed for my success  
throughout my life*

# Acknowledgements

I wish to thank, first and foremost, my lead supervisor Dr. Eike Ritter and second supervisor Dr. Guilin Wang. I share the credit of my work with them. Without their guidance and persistent help my thesis would not have been possible. The excellent knowledge, support and friendship of them has been invaluable for which I am extremely grateful. I consider it an honor to work with them.

I am truly indebted and thankful to Professor Mark Ryan for his interest in my work and financial support in my experiments for buying equipment and publishing the experimental results. My experiments would not have been possible without his financial support. These experiments gave me an opportunity to interact with the real sensor nodes devices and obtain accurate statistics of my cryptographic protocols. I would also like to acknowledge the efforts of my thesis group members Dr. Tom Chothia and Professor Uday Reddy in continually reviewing my research progress and giving productive feedback throughout this work.

It gives me a great pleasure in acknowledging the support and help of Diego Aranha, the Adjunct Professor of The University of Brasilia, Brazil. Diego Aranha, the lead developer of RELIC library, did not only provide me with the latest version of his library before publishing it but also helped me in modifying it for my own experiments. His expert knowledge in the area of implementing and evaluating cryptographic protocols on sensor nodes helped me in my experiments in collecting my results and evaluating them.

I would like to gratitude my colleagues Sergiu Bursuc and Myrto Arapinis who, other than friends, have always helped me in my work. In addition, I am thankful to Sergiu for reading my thesis and helping me to improve it. I would also like to acknowledge the support of (Brig.) Shiraz Baig, my MS supervisor, who has always guided me and has been an inspiration for me.

In the end, I would like to express my deepest appreciation to my family, friends and departmental colleagues who boosted me morally.



# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Acronym</b>	<b>xv</b>
<b>List of Publications</b>	<b>xvi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Wireless Sensor Networks . . . . .	1
1.2 Applications . . . . .	2
1.3 Security Concerns . . . . .	4
1.4 Authentication . . . . .	4
1.4.1 Authentication in Wireless Sensor Networks . . . . .	5
1.5 Scope and Adopted Approach . . . . .	7
1.6 Research Questions . . . . .	8
1.7 Thesis Contribution . . . . .	9
1.8 Overview and Structure . . . . .	10
<b>2 Security and Cryptography: A Background</b>	<b>13</b>
2.1 Security in Wireless Sensor Networks . . . . .	13
2.1.1 Characteristics of Wireless Sensor Networks . . . . .	14
2.1.2 Security Goals . . . . .	16
2.1.3 Attackers and Security Attacks . . . . .	17
2.1.4 Constraints in Wireless Sensor Networks . . . . .	24
2.2 Cryptography . . . . .	26
2.2.1 Notations and Conventions . . . . .	26
2.2.2 Cryptographic Primitives for Authentication . . . . .	26
2.2.3 ID-based Signature . . . . .	29

2.2.4	ID-based Online/Offline Signature . . . . .	31
2.2.5	Cryptosystems Used . . . . .	33
2.2.6	Examples of IBS and IBOOS Schemes . . . . .	36
2.2.7	Security Proofs . . . . .	40
2.3	Concluding Remarks . . . . .	42
<b>3</b>	<b>Authentication in Wireless Sensor Networks: A Review</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	Broadcast Authentication . . . . .	44
3.2.1	Schemes Based on Symmetric Cryptography . . . . .	44
3.2.2	Schemes Based on Asymmetric Cryptography . . . . .	52
3.2.3	Discussion . . . . .	55
3.3	Outside User Authentication . . . . .	55
3.3.1	User Authentication . . . . .	55
3.3.2	Session Key Establishment . . . . .	59
3.3.3	Discussion . . . . .	61
3.4	Concluding Remarks . . . . .	61
<b>4</b>	<b>Authentication Framework</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Motivations . . . . .	66
4.2.1	Sensor Nodes Broadcast Authentication . . . . .	67
4.2.2	Outside User Authentication . . . . .	69
4.3	Security Goals . . . . .	70
4.4	Security Attacks . . . . .	71
4.5	Threat Model . . . . .	72
4.5.1	Assets to Protect . . . . .	72
4.5.2	Adversary's Goals . . . . .	73
4.5.3	Adversary's Capabilities . . . . .	73
4.6	Assumptions . . . . .	73
4.7	Trust Model . . . . .	74
4.8	Proposed Authentication Framework . . . . .	75
4.9	Framework Instantiation and Evaluation . . . . .	78
4.10	Concluding Remarks . . . . .	79



<b>5</b>	<b>Authenticated Broadcast by Sensor Nodes Protocol</b>	<b>81</b>
5.1	Introduction . . . . .	82
5.2	Options for IBOOS Schemes . . . . .	84
5.2.1	Available IBOOS Schemes . . . . .	84
5.2.2	Adapted IBOOS Scheme . . . . .	85
5.3	Proposed Authenticated Broadcast by Sensor Nodes Protocol . . . . .	87
5.3.1	Is an Online/Offline Signature Scheme Secure for Wireless Sensor Networks? . . . . .	90
5.4	Performance Evaluation . . . . .	90
5.4.1	Performance of the IBOOS Schemes . . . . .	90
5.4.2	Performance of the Proposed Protocol . . . . .	100
5.5	Security Analysis . . . . .	103
5.5.1	Security of the IBOOS Schemes . . . . .	103
5.5.2	Security of the Proposed Protocol . . . . .	104
5.6	Comparison with Existing Protocols . . . . .	106
5.7	Concluding Remarks . . . . .	109
<b>6</b>	<b>Outside User Authentication Protocol</b>	<b>111</b>
6.1	Introduction . . . . .	112
6.2	Options for IBS Schemes . . . . .	114
6.2.1	Available IBS Schemes . . . . .	114
6.3	Options for Session Key Establishment Schemes . . . . .	114
6.3.1	Key Establishment Options . . . . .	115
6.3.2	Available Key Establishment Protocols . . . . .	117
6.3.3	Proposed Key Establishment Protocol . . . . .	117
6.4	Proposed Outside User Authentication Protocol . . . . .	122
6.5	Performance Evaluation . . . . .	125
6.5.1	Performance of IBS Schemes and Session Key Establishment . . . . .	125
6.5.2	Performance of the Proposed Protocol . . . . .	127
6.6	Security Analysis . . . . .	129
6.6.1	Security of the IBS Schemes . . . . .	129
6.6.2	Security of the Session Key Establishment . . . . .	130
6.6.3	Security of the Proposed Protocol . . . . .	140
6.7	Comparison with Existing Protocols . . . . .	142
6.7.1	Comparison with Existing SKE Protocols . . . . .	142
6.7.2	Comparison with Existing User Authentication Protocols . . . . .	148

6.8	The Proposed Authentication Framework: A Single Solution . . . . .	149
6.9	Concluding Remarks . . . . .	150
<b>7</b>	<b>Conclusion and Future Work</b>	<b>153</b>
7.1	Summary of the Thesis Contributions . . . . .	154
7.2	Future Research Directions . . . . .	156
7.2.1	Extension of the Authentication Framework . . . . .	157
7.2.2	Experimental Evaluation of the Proposed Authenticated Broad- cast by Sensor Nodes Protocol . . . . .	158
7.2.3	Pairing-free IBS and IBOOS Schemes . . . . .	159
7.2.4	Other Applications of the Authentication Framework . . . . .	159
	<b>Appendix A: Experimental Details</b>	<b>161</b>
	<b>Appendix B: ID-based One-Pass Session Key Establishment Protocol</b>	<b>165</b>
	<b>List of References</b>	<b>169</b>

# List of Figures

1.1	A Wireless Sensor Network . . . . .	2
2.1	Attacks Against Privacy . . . . .	18
2.2	Attack Against Data Aggregation . . . . .	19
2.3	Impersonation Attack . . . . .	20
2.4	Denial of Service Attack . . . . .	21
2.5	Hello Flood Attack . . . . .	21
2.6	Sinkhole Attack together with Selective Forwarding Attack . . . . .	22
2.7	Wormhole Attack . . . . .	23
2.8	Sybil Attack . . . . .	23
2.9	Replication or Clone Attack . . . . .	24
2.10	Message Authentication Code . . . . .	27
2.11	Digital Signature . . . . .	28
2.12	ID-based Signature . . . . .	30
2.13	ID-based Online/Offline Signature . . . . .	32
4.1	Authentication Framework for Wireless Sensor Networks . . . . .	76
4.2	Authentication Framework: A Countermeasure to Security Attacks . . . . .	77
5.1	MICA2 node without an antenna [Cro] . . . . .	91
6.1	Authenticated ID-based One-Pass Session Key Establishment . . . . .	119
A.1	A MICA2 (MPR4x0) wireless module without an antenna [Cro]. . . . .	162
A.2	Top view of MIB510 Serial Interface Board [Cro]. . . . .	162
B.1	Authenticated One-Pass Session Key Establishment Protocol . . . . .	167



# List of Tables

2.1	Commercially available sensor nodes . . . . .	15
5.1	Time and Energy Consumption of X-IBOOS Scheme . . . . .	93
5.2	Memory Consumption of X-IBOOS Scheme in Bytes . . . . .	94
5.3	Time and Energy Consumption of B-IBOOS Scheme . . . . .	96
5.4	Memory Consumption of B-IBOOS Scheme in Bytes . . . . .	97
5.5	Optimized Time and Energy Consumption of B-IBOOS Scheme . . .	98
5.6	Optimized Memory Consumption of B-IBOOS Scheme in Bytes . . .	98
5.7	Summary of Time and Energy Consumption of X-IBOOS and B- IBOOS Schemes . . . . .	99
5.8	Summary of Memory Consumption of X-IBOOS and B-IBOOS Schemes in Bytes . . . . .	99
5.9	Comparison of proposed broadcast authentication scheme with ex- isting digital signature based broadcast authentication schemes for WSNs . . . . .	107
6.1	Time and Energy Consumption of BNN-IBS Scheme . . . . .	126
6.2	Time and Energy Consumption of ID-1P-SKE Scheme . . . . .	126
6.3	Total Time and Energy Consumption of User Authentication and Session Key Establishment . . . . .	127
6.4	Computation cost comparison of ID-1P-SKE protocol with the existing session key establishment protocols for WSNs . . . . .	144
6.5	Communication cost comparison of ID-1P-SKE protocol with the existing session key establishment protocols for WSNs . . . . .	145
6.6	Comparison of ID-1P-SKE protocol with existing ID-based one-pass key establishment protocols for traditional networks . . . . .	147
6.7	Comparison of proposed user authentication scheme with existing distributed user authentication schemes for WSNs . . . . .	149



# List of Acronyms

<b>AKE</b> authenticated key establishment .....	115
<b>CA</b> Certificate Authority .....	28
<b>CDH</b> Computational Diffie-Hellman .....	35
<b>DH</b> Diffie-Hellman .....	116
<b>DoS</b> Denial-of-Service .....	20
<b>ECC</b> Elliptic Curve Cryptography .....	33
<b>ECDL</b> Elliptic Curve Discrete Logarithm .....	35
<b>IBOOS</b> ID-based Online/Offline Signature .....	31
<b>IBS</b> ID-based Signature .....	30
<b>KE</b> key establishment .....	113
<b>MAC</b> Message Authentication Code .....	27

<b><math>\mu</math>TESLA</b> Micro Timed Efficient Stream Loss-tolerant Authentication .....	45
<b>OOS</b> Online/Offline Signature .....	31
<b>PBC</b> Pairing Based Cryptography .....	34
<b>PKC</b> public key cryptography .....	28
<b>PKG</b> private key generator .....	29
<b>TPM</b> Trusted Platform Modules .....	74
<b>WSN</b> wireless sensor network .....	1



# List of Publications

Some portions of this thesis are also parts of the following publications:

## Conference Papers

- R. Yasmin, E. Ritter, and G. Wang. 2011. *A Pairing-Free ID-based One-Pass Authenticated Key Establishment Protocol for Wireless Sensor Networks*. In Proc. of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM '11). (**Best Paper Award**).
- R. Yasmin, E. Ritter, and G. Wang. 2010. *An Authentication Framework for Wireless Sensor Networks using Identity-Based Signatures*. In Proc. of the 10th IEEE International Conference on Computer and Information Technology (CIT '10). IEEE Computer Society, USA. (**Best Paper Award**).

## Journal Papers

- R. Yasmin, E. Ritter, and G. Wang. *An Authentication Framework for Wireless Sensor Networks using Identity-Based Signatures: Implementation and Evaluation*. Special section on Trust, Security and Privacy in Computing and Communication Systems in IEICE Transactions on Information and Systems (IEICE). 2012.
- R. Yasmin, E. Ritter, and G. Wang. *Provable Security of a Pairing-free One-Pass Authenticated Key Establishment Protocol*. Submitted to International Journal of Information Security (IJIS).



# Part I



# Chapter 1

## Introduction

***Chapter Overview:** This chapter presents the scope of this thesis. It briefly introduces wireless sensor networks, authentication problems in wireless sensor networks and the adopted approach to address the problems. The major contributions made by this thesis are also highlighted in this chapter.*

### 1.1 Wireless Sensor Networks

A wireless sensor network (WSN) is a wireless ad hoc network consisting of a large number of small low cost devices called sensor nodes or motes. A sensor node is a self-contained unit typically consisting of a battery, transceiver, micro-controller and sensors. These sensor nodes are tiny resource constrained devices with the limitations of *low* battery power and communication range and *small* computation and storage capabilities. They are usually deployed in open environments where they collaboratively monitor the physical and environmental data such as temperature, pressure, vibration etc., and report/relay the sensed data to other sensor nodes over a wireless network. The final destination of this data is a base station also called

a sink node which is a powerful device, e.g., a laptop. The base station acts as a gateway and links the WSN to the outer network e.g., the Internet.

The recent advancements in embedded technologies as well as in wireless communications have broadened the prospects for many applications of WSNs. These applications include, but are not limited to, environmental monitoring, ocean reading, forest fire alarm and military applications [ASSC02]. In some WSN applications, the data collected by the sensor nodes is valuable for different types of users such as research organizations, universities, businesses or individuals, called outside users. The outside users of the sensor nodes data are usually equipped with resourceful devices like notebooks, mobile phones etc., in order to query the sensor nodes. Hence, a typical WSN scenario involves three entities: a base station, sensor nodes and outside users as shown in Figure 1.1.

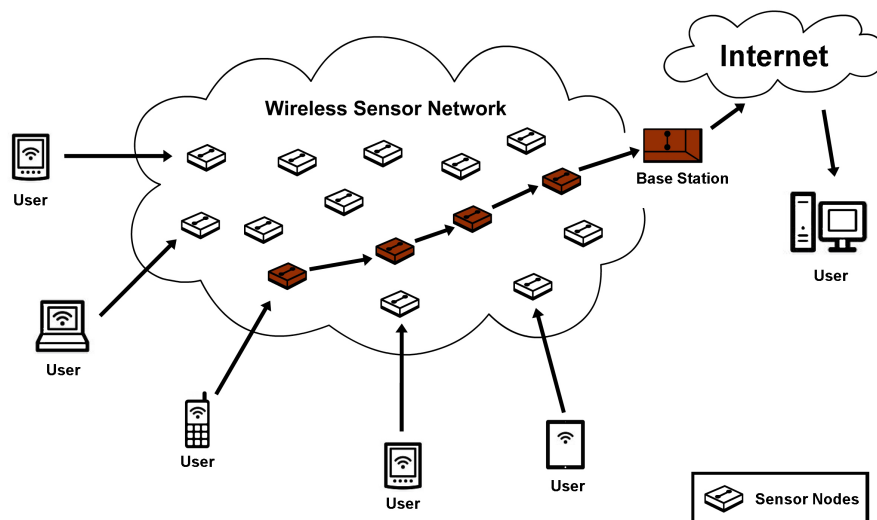


Figure 1.1: A Wireless Sensor Network

## 1.2 Applications

The low cost and the immunity from cabling have become motivations for many applications of WSNs [ASSC02, CK03, CES04, Xu02, Sto05, BHUW08, FHB<sup>+</sup>08, APM05], for instance,

- Disaster handling: (Forest Fire Detection, Flood Detection, Earthquake

Detection and Surveillance etc.)

*Example (Forest Fire Detection):* In a forest fire alarm application, the sensor nodes are deployed in a forest to detect a fire event and inform authorities about event and the exact location of event.

- Road safety: (Intelligent Traffic System)

*Example:* In a traffic application, the sensor nodes are deployed along the roadside to monitor the status of a road and traffic on the road. These sensor nodes sense an accident, a traffic jam or a dangerous road condition, such as ice on the road, and alert other traffic approaching this location. A driver can use the *accident prevention* service of this application to access data which helps him to safely drive on the road. Whereas an insurance company or road patrol can use the *post-accident investigation* service which is helpful for them to judge the causes of an accident, for instance, a driver's driving style at the time of accident.

- Monitoring huge structures: (Structural Health Monitoring)

*Example:* In a structural health monitoring application, the sensor nodes are intended to monitor the structural health of bridges, tunnels, huge buildings, etc. The sensor nodes are assumed to monitor any damages or cracks in structures and report them so that precautionary measures can be taken.

- Environmental monitoring: (Ocean Reading, Habitat Monitoring, Weather Forecast, Monitoring Glacier Behavior, Precision Agriculture etc.)

*Example (Ocean Reading):* WSNs are deployed under the water for oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention and assisted navigation. Underwater sensor networks also help in detecting underwater oilfields or reservoirs, and assist in exploration for valuable minerals.

- Healthcare: (Tracking and Monitoring Doctors and Patients inside a Hospital, At-Home Health, Drug Administration, Elderly Care etc.)

*Example (Elderly Care):* This application aims to improve the life quality of the elderly people through smart environments. It closely monitors changes in a person's vital signs and provides feedback to help maintain an optimal health status. It can also alert medical personnel when life-threatening changes occur.

- Military applications: (Enemy Tracking, Monitoring Enemy Forces, Equipment and Ammunition, Detecting Nuclear, Biological and Chemical Attacks, Battlefield Surveillance etc.)

*Example (Enemy Tracking):* In an enemy tracking application, the sensor nodes are deployed in the battlefield to detect the presence and location of enemy's tanks, vehicles or personnels in the battlefield and track their movements. The soldiers obtain this information from the sensor nodes which helps them to safely position themselves in the battlefield.

### 1.3 Security Concerns

The popularity of WSNs has increased as potential low-cost cable-less solutions to a variety of applications. For instance, a WSN is the best solution for the applications of ocean reading, volcano monitoring, forest fire alarm, intelligent traffic system and battlefield applications, where running wires or cabling is usually impractical or too costly. However, the vulnerability of wireless communication and the ad-hoc nature of deployment open the door for a wide variety of malicious attacks, making security a key concern for these applications. In particular, the wireless medium enables anyone to interrupt the in-channel communication. It compromises the secure communication of the data collected by the sensor nodes which is an important security requirement of the above mentioned applications of WSNs. On the other hand, the resource constrained nature of sensor nodes, i.e., limited *power*, *computing* and *storage* resources, does not allow to use complex security solutions and raises a need for highly efficient security solutions for WSNs. This restriction has significantly impacted the field of application security. For such applications, the efficiency of a security scheme is as important as its security. Any security scheme which is computationally expensive, no matter how secure it is, does not suit resource constrained sensor nodes. Among the broad domain of security problems faced in WSNs, this thesis focuses on authentication problems.

### 1.4 Authentication

Authentication is a process by which one verifies that someone is who he or she claims to be. Authentication enables a receiver of a message to confirm the

- claimed message sender or origin of a message (source authentication),



- contents of a message has not been modified (message integrity/authentication).

Based on types of communication, authentication may be classified as follows:

- **Unicast** or **point-to-point** authentication, where an entity authenticates itself to a single entity.
- **Multicast** authentication, where an entity authenticates itself to a small group of entities.
- **Broadcast** authentication, where an entity authenticates itself to all entities in the network.

#### 1.4.1 Authentication in Wireless Sensor Networks

Beneson [BGK04] distinguished between the insider security and the outsider security in WSNs as follows:

- *Insider security* addresses secure communication between the sensors and between the sensors and the base station(s).
- *Outsider security* addresses secure communication between the WSN (sensors and base station) and the outside user.

Authentication is a crucial security requirement in WSNs which is a part of both insider and outsider security. In the absence of a strong authentication mechanism, an adversary can frequently generate dummy data packets and make the sensor nodes relay them to deplete their energy. Moreover, a fake or modified message can cause the sensor nodes to accept wrong information and may result in serious attacks against the sensor network. For example, it is important for the base station to send some crucial information, like the current time for synchronization, to all sensor nodes in the network. An adversary can modify a time synchronization message or send forged data to desynchronize the network or to disturb the receiver's clock. A countermeasure to this kind of attacks is authentication. Authentication in a typical WSN can be classified into three categories as follows:

- Base station to sensor nodes authentication
- Sensor nodes to other sensor nodes authentication
- Outside users to sensor nodes authentication

The base station to sensor nodes authentication has been widely addressed by the current authentication schemes for WSNs [PST<sup>+</sup>02, LN04, LNZJ05, DG06, CKDZ08]. Therefore, we focus on the other two authentication problems of the sensor nodes to other sensor nodes authentication and the outside user to sensor nodes authentication. This thesis only deals with the broadcast/multicast authentication since the typical point-to-point or pairwise authentication in WSNs have been studied in detail. In the first authentication problem, the sensor nodes need to authenticate their messages to either a set of sensor nodes or all sensor nodes in the network. However, in the second authentication problem, the outside users authenticate themselves to a set of sensor nodes (in their communication range) only and not all sensor nodes in the network. Any solution to broadcast authentication can also be used to address multicast authentication problem in WSNs. Therefore, broadcast and multicast are treated in the same way in first authentication problem and referred as broadcast in rest of the thesis.

#### **1.4.1.1 Sensor Nodes Broadcast Authentication**

Sensor nodes are usually applied over an area for the purpose of collecting some data or monitoring a critical phenomenon. Whenever a sensor node detects a critical event which needs attention, it informs all other sensor nodes (broadcast message) or a group of sensor nodes (multicast message) in the sensor network. These broadcast and multicast messages can be treated in the same way in WSNs. Provision of broadcast authentication in sensor networks is a much harder task than provision of pairwise authentication. The reason is the constrained resources available at sensor nodes. On the other hand, secure broadcast of messages by the sensor nodes is an essential security requirement in WSNs. This feature is compulsory for many attractive applications in civilian and military operations, such as forest fire alarm and enemy tracking. Unfortunately, the problem of sensor nodes broadcast authentication has not gained attention by the existing research work so far. To the best of our knowledge, our work is the first one to identify and address this problem and propose a solution for it.

#### **1.4.1.2 Outside User Authentication**

Sensor nodes usually collect a variety of data. The data collected by the sensor nodes is of interest to different types of users such as research organizations, universities, businesses or individuals. For example, the humidity level in an area might be

a useful piece of information for a farmer. An individual may be interested to know about the weather in his surroundings. A researcher may be interested in environmental data collected by the sensor nodes. An oil company might be keen to obtain ocean reading data. On the other hand, the deployment and maintenance cost of a large scale WSN makes it difficult for everyone to deploy own sensor networks to collect data of their interests. The users of the sensor nodes data, thus, pay the deployment agencies of the large scale WSNs to obtain this data. Therefore, owners and users of the networks are different for some large scale WSNs. NOPP (National Oceanographic Partnership Program) [NOP] is an example of such large scale WSNs to observe earth, ocean and atmosphere. Hence, the sensor nodes data in these large scale WSNs is *valuable* and only the subscribed users, who have paid for the data, are allowed to obtain it. Apart from these commercial applications, there are many army applications which gather *sensitive* and *confidential* data which should be accessible to authorized army officers and soldiers only. These facts raise the issue of authentication of a legitimate user in WSNs. User authentication is a process by which the system verifies the identity of a user who wants to access the sensor nodes data. A user authentication mechanism is necessary to prevent unauthorized users from accessing sensor nodes data. Providing a secure user access to sensor nodes data requires two basic tasks:

1. *Authentication* allows only legitimate users of the data to access it.
2. *Session Key Establishment* enables secure transmission of confidential sensor nodes data to users after authentication.

As a part of this research, we aim to address the problems of user authentication and session key establishment under the heading of outside user authentication.

## 1.5 Scope and Adopted Approach

To address the above mentioned authentication problems, this thesis proposes an authentication framework for WSNs using identity (ID) based signature schemes. The proposed framework is comprised of two authentication protocols; *authenticated broadcast by sensor nodes* protocol to address the problem of sensor nodes broadcast authentication and *outside user authentication* protocol to provide a secure user access to sensor nodes data. The aim of this research work is to design efficient and secure authentication protocols to address the authentication problems in WSNs. The proposed authentication framework is discussed in detail in Chapter 4.

## 1.6 Research Questions

More specifically, this thesis is concerned to answer the following research questions:

- How useful is the proposed authentication framework in aiding secure communication in wireless sensor networks?
  - How useful are the ID-based signatures to handle the authentication problems in wireless sensor networks?
  - To what extent is the proposed authentication framework able to handle the problem of sensor nodes broadcast authentication, a problem neglected by the existing security solutions?
  - What are the advantages of the proposed solution to user authentication over the existing solutions for same the problem.
- Has the proposed authentication framework improved the performance and security over the existing solutions?
  - Is the proposed solution of sensor nodes broadcast authentication problem efficient and secure?
  - Is the proposed solution of user authentication problem more efficient than the existing solutions as well as secure?
- To what extent does the proposed authentication framework support scalability and dynamism, the two required features of security schemes for wireless sensor networks?
  - Does it support large scale sensor networks?
  - Does it allow adding/removing sensor nodes and users to/from sensor networks?

These questions are investigated in detail in this thesis. The security of the different components of the proposed framework is analyzed formally and informally. In addition, the experiments are performed on the actual sensor nodes to measure the efficiency statistics. The experimental results show the performance advantages of the proposed framework over the existing solutions while providing scalability and dynamism.

## 1.7 Thesis Contribution

This thesis makes the following main contributions:

1. Proposes an authentication framework for WSNs using ID-based signature schemes which is comprised of two components: sensor nodes broadcast authentication and outside user authentication.
2. Highlights for the first time the need of authenticated broadcast by sensor nodes and gives a solution to the problem in the form of a sensor nodes broadcast authentication protocol.
3. Proposes for the first time to use ID-based Online/Offline signature schemes in WSNs to provide authentication. Online/Offline signatures, described in next chapter, are used to provide sensor nodes broadcast authentication in this work.
4. Implements for the first time several ID-based Online/Offline signature schemes, a cryptographic primitive new to sensor nodes devices, on real sensor nodes and evaluates their performance.
5. Securely modifies an ID-based signature scheme to obtain an efficient ID-based Online/Offline signature scheme for WSNs and implements two different variations of it for further efficiency improvement.
6. Designs a new secure and efficient cryptographic ID-based one-pass authenticated session key establishment protocol mainly for WSNs.
7. Formally analyzes the security of the newly designed cryptographic ID-based one-pass session key establishment protocol as well as its performance.
8. Proposes a user authentication protocol which, as compared to the existing user authentication protocols, not only authenticates the users but also establishes a session key between the user and the sensor node after successful user authentication.

## 1.8 Overview and Structure

### Part I

The first part of the thesis is an introductory part which discusses the scope of the thesis, some background knowledge and a literature survey of the existing work.

- Chapter 1 has given an overview of the scope of this research thesis after introducing WSNs and the authentication problems faced in WSNs. It has also described our adopted approach to provide a solution, i.e., an authentication framework, and an outline of the major contributions made by this thesis.
- Chapter 2 provides background knowledge about the security problems and requirements in a WSN together with the constraints in WSNs. It also introduces the cryptographic terminologies, primitives and approaches used in this thesis.
- Chapter 3 provides a brief literature survey of the security solutions that have been proposed to handle authentication problems in WSNs and highlights their shortcomings which become motivations for this research work.

### Part II

To address the shortcomings identified by the survey of existing authentication schemes, an authentication framework is proposed for WSNs in this thesis. The second part of the thesis focuses on the proposed authentication framework by introducing it and describing its both authentication protocols in detail and concludes the thesis in the end.

- Chapter 4 introduces the proposed authentication framework after highlighting the motivations behind this work. It also describes the security goals of the proposed framework, threat model and trust model used by the proposed framework together with the assumptions made by it.
- Chapter 5 discusses the first authentication protocol in detail that is the authenticated broadcast by sensor nodes protocol. The details of security and performance evaluations of this protocol are also given in this chapter which include the details of experiments to evaluate the online/offline signatures on sensor nodes, the modification of an ID-based signature scheme to an ID-based online/offline signature scheme and the security of online/offline signature schemes.

- 
- Chapter 6 discusses the second authentication protocol in detail that is the outside user authentication protocol. The details of security and performance evaluations of this protocol are also described in this chapter which include the formal security analysis of the newly designed cryptographic ID-based one-pass session key establishment protocol and its performance evaluation.
  - Chapter 7 concludes the thesis by presenting an overview of the contributions made by this research work. It also suggests the possible directions for future research.





## Chapter 2

# Security and Cryptography: A Background

***Chapter Overview:** This chapter describes several background concepts necessary for the understanding of this thesis. The first part of the chapter presents an overview of wireless sensor networks security including security objectives, types of attackers and security attacks in wireless sensor networks. It also highlights the constraints in wireless sensor networks which are barriers to provide security. The second part of the chapter reviews the cryptographic tools, primitives and notions and describes the computational assumptions used in this thesis.*

### 2.1 Security in Wireless Sensor Networks

The particular characteristics of WSNs offer an advantage to any adversary who intends to compromise security. For instance, the sensor nodes use radio-link as a communication medium which is in fact insecure. The broadcast nature of

communication medium makes WSNs more vulnerable to security attacks than wired networks. On the other hand, provision of security in WSNs is a challenging task since the resources in sensor nodes devices are not sufficient for executing complex security protocols. This section reviews the particular characteristics of a WSN and security concerns in a typical WSN.

### 2.1.1 Characteristics of Wireless Sensor Networks

A WSN can be seen as a special case of ad hoc networks. A wireless ad hoc network is the one which does not rely on a fixed infrastructure, such as routers or access points. Instead, the nodes in an ad hoc network organize themselves on the fly to provide pathways for data to be routed from other nodes. They do not have a fixed topology. The routing decisions in an ad hoc network are made dynamically based on the network connectivity. WSNs share some common features with ad hoc networks, such as they have random network topology and infrastructure-less architecture. Besides, sensor networks possess some characteristics which are different from ad hoc networks and traditional wired and wireless networks. The following are the main characteristics of WSNs.

- *Resource limitation:* Typical sensor nodes are usually tiny resource constrained devices who have very *limited* computational capability, storage capacity, communication bandwidth and on-board energy available. In general, the sensor nodes are significantly more resource constrained devices than typical mobile devices. The battery power is the most scarce resource in sensor nodes among other resources. It is a usual assumption about sensor nodes that once they are deployed, their batteries cannot be replaced or recharged since the sensor nodes are often inaccessible. Thus, the lifetime of a sensor node depends on the lifetime of its battery. Table 2.1 shows the specifications of a few commercially available sensor nodes (MICA2 [MIC], Tmote Sky [Tmo], SmartPoint [Sma]).
- *Nature of deployment:* In order to achieve the highly accurate sensing results, the sensor nodes are usually densely deployed with certain level of redundancy. The number of sensor nodes in a sensor network may be several orders of magnitude higher than the nodes in an ad hoc network. Sensor nodes are usually scattered randomly in inaccessible environments where they self-organize into an infra-structureless network.




			
	<b>Mica2</b> (Crossbow)	<b>T-mote Sky</b> (Moteiv)	<b>SmartPoint</b> (Ambient System)
Processor	8MHz	8MHz	16MHz
RAM	4KB	10KB	10KB
Flash	128KB	48KB	1MB
Data Rate	40Kbps	250Kbps	250Kbps
Range	100m	50-125m	50m
Frequency	2.4GHz	2.4GHz	2.4GHz
Battery	2×AA batteries	2×AA batteries	2×AAA batteries

Table 2.1: Commercially available sensor nodes

- *Unattended after deployment:* The WSNs are usually deployed in an open air environment where they are left unattended without a constant supervision. This fact allows easy physical access for anyone to the sensor nodes. Moreover, most often WSNs operate in harsh and even hostile environments, such as extreme weather and natural disasters, that may affect their performance.
- *Dynamic network topology:* The sensor network topology is unknown prior to the deployment. Moreover, the sensor nodes fail due to depletion of energy or physical damage and new nodes are added to the network. Node addition and node failure make the network topology dynamic.
- *Communication:* A sensor node usually has a limited communication range and every node may not be in direct communication range of the base station. Therefore, the sensor nodes send their collected data through intermediate nodes to the nodes closer to the base station who ultimately forward the data to the base station. Hence, the sensor nodes do not only collect data but also relay data for other nodes that are further away from the base station. The communication paradigm in WSNs is mainly broadcast where a message sent by a sensor node can be received by everyone in the communication range.

### 2.1.2 Security Goals

The goal of security services in WSNs is to protect information (authenticity, verification, integrity, confidentiality, access control, and freshness) and resources (availability) from attacks and misbehavior in the presence of a resourceful adversary.

**Authentication** enables each message sender in the sensor network, including the base station, sensor nodes and outside users, to prove its identity to the receiver, i.e., the legitimacy of the source of a message. It allows the receiver of the message to check that received messages are actually originated from the claimed source.

**Message Integrity** verifies the genuineness of the received message contents. It must be implemented to ensure a receiver that the contents of received message have not been modified in transit by an adversary.

**Verification** empowers each sensor node in the network to attest the legitimacy of the received message. It is important to note that authentication does not imply verification in WSN environment. A legitimate message sender may send an authenticated message to the sensor nodes, however, the sensor nodes may not have access to authentication information of the message sender or may not be capable of performing efficiently the computation that is required to verify authentication information. This capability is ensured by the verification property in WSNs which enables sensor nodes to verify authenticated messages. Verification can be seen as a counterpart of authentication where authentication presents the proof of identity and verification implies the ability to attest the proof of identity. In sensor networks, it is essential for all three entities to have the ability to confirm that the message received was actually sent by a trusted sender and not by an adversary.

**Freshness** means that a received message is new and a recent one. Freshness could mean both data freshness and key freshness. Data freshness implies that the received data is recent and it ensures that no adversary has replayed old messages. Key freshness implies that the session key established between the two parties in each session is fresh and it is unique for each session.

**Confidentiality** prevents unauthorized parties or adversaries from accessing the data being sent to the authorized parties. The confidentiality objective is required in WSNs environment to protect data traveling between the sensor nodes, between the sensor nodes and the base station, and between the sensor nodes and the outside users from disclosure. A confidential message should not reveal its contents to an eavesdropper.

**Access Control** ensures that only the authorized sensor nodes are involved in providing information to network services and only an authorized user obtains a certain type of data according to his access privileges. User access control is required in those applications of WSNs which collect a variety of data. For such applications, the users have different access privileges for different types of data due to the data security and privacy reasons.

**Availability** ensures the survivability of sensor network services to authorized parties when needed despite the presence of internal or external attacks.

### 2.1.3 Attackers and Security Attacks

#### 2.1.3.1 Attackers

- *Mote Class VS Laptop Class.* Depending on the resource capabilities, an attacker in WSNs may be categorized in either *mote* class or a *laptop* class. A *mote* class adversary uses a similar mote (sensor node) to launch attacks against the sensor networks and is less powerful in terms of resources with fewer capabilities. A *Laptop* class adversary, on the other hand, uses a more powerful device (e.g., a laptop) with higher resource and processing capabilities than the network nodes in order to attack a WSN.
- *Insider VS Outsider.* A further classification within the above two classes is based on the access level. An adversary may be *insider* or *outsider*. An *insider* adversary is the one who becomes a part of the sensor network, e.g., by compromising the legitimate sensor nodes or adding his own sensor nodes to the network. The *insider* adversary is a big threat as it has direct access to the network. The *outsider* adversary, on the contrary, does not belong to a WSN and has no or less direct access to the sensor network.

#### 2.1.3.2 Major Security Attacks

An attack can be defined as an action taken to harm a resource of value such as data in case of WSNs. The need of security solutions comes mainly from possible attacks. If there are no attacks, there is no need for security schemes. Usually, the probability of attacks within the WSNs is higher than in any other type of network due to the unique nature of WSNs. Attacks against WSNs may be categorized as passive versus active attacks. Passive attacks include eavesdropping on or monitoring the communication exchanged within a WSN. Active attacks, on

the other hand, involve some modifications of the actual data or insertion of the false data into the communication channel. This section lists the major attacks against WSNs described in [CP03, PSW04, WAR06, DC08].

- *Attacks Against Privacy:* A privacy attack is aimed at obtaining confidential or valuable information.
  - *Eavesdropping.* Since radio links are insecure, an adversary having the appropriate equipment may easily eavesdrop on the communication to obtain sensor nodes data (see Figure 2.1). By eavesdropping, the adversary can also overhear other secret information such as user queries and routing information. The adversary may use this information for malicious purposes, for instance, to know the interests of his business competitors from their queries to the sensor network. Furthermore, by stealing routing information the adversary can launch attacks against routing protocols.

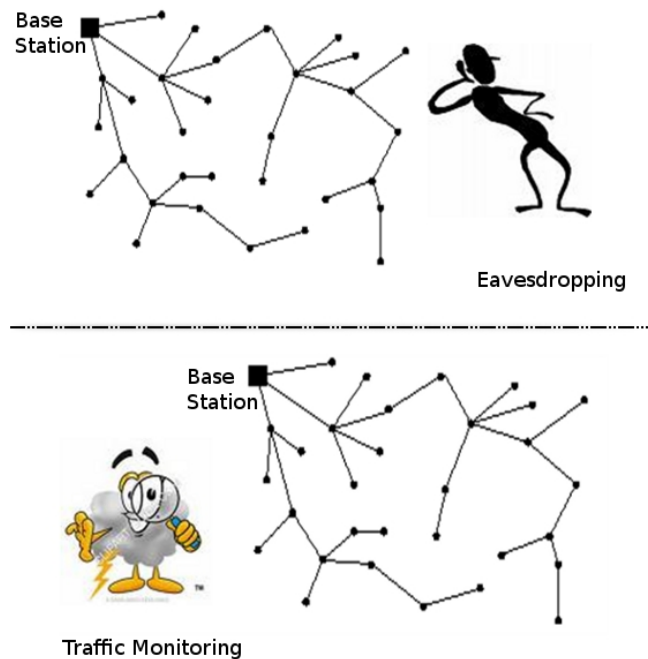


Figure 2.1: Attacks Against Privacy

- *Traffic Monitoring.* If user queries to the sensor network are encrypted, adversary can not know them. However, by monitoring the traffic flow

(Figure 2.1), he can guess the nature of queries. He can also find out the location of the base station by monitoring traffic. The ultimate goal of the adversary is to launch attacks against base station to make the base station unavailable.

Countermeasures to these attacks are data encryption which hides the communication contents, and bogus traffic which deceives traffic monitoring.

- *Attacks Against Data Aggregation:*

- *False Data Injection.* During the data aggregation process, an intruder can add fake sensor readings by injecting data packets or alter original sensor readings by modifying the packets. It can affect the overall data aggregation results as shown in Figure 2.2. A countermeasure to this attack is authentication which prevents from injecting fake data packets or modifying packet contents.

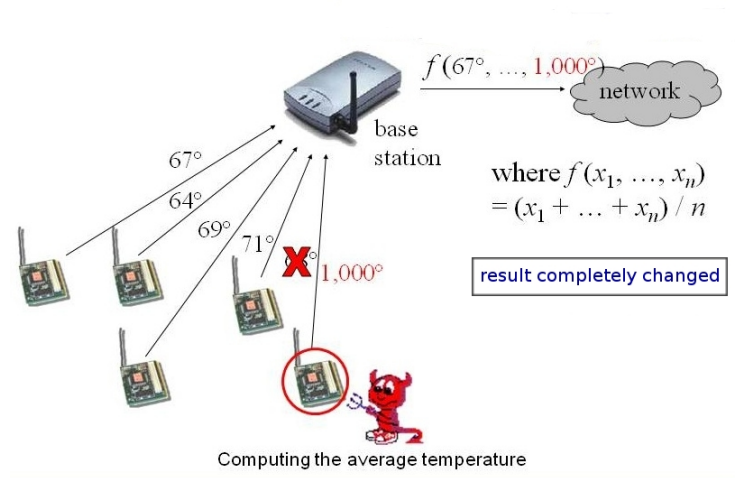


Figure 2.2: Attack Against Data Aggregation

- *Message Replay Attack.* A replay of a data message originally sent by a legitimate sensor node will have the same effect on data aggregation process as the above mentioned attack. A countermeasure to avoid a message replay attack is to ensure the freshness of data, for instance, via attaching a timestamp with each message.

- *Impersonation Attack*: This attack is an attempt by the adversary to deceive sensor nodes by impersonating a legitimate sensor node or an outside user (Figure 2.3). The ultimate goal of this attack is to send fake messages on behalf of a legitimate sensor node or obtain sensor nodes data on behalf of a legitimate user.

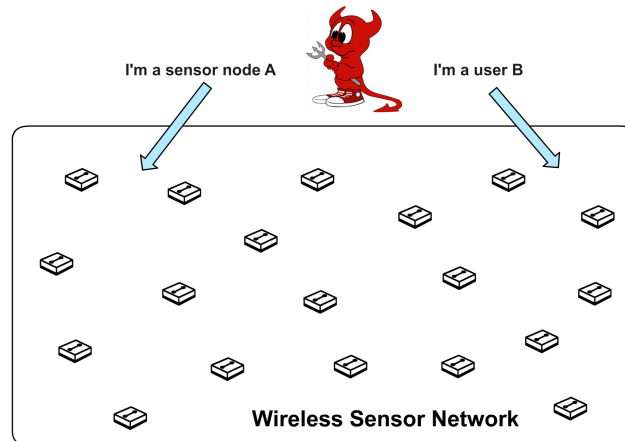


Figure 2.3: Impersonation Attack

- *DoS Attack or Spam Attack*: The Denial-of-Service (DoS) attack or spam attack is an attempt to make a system or a service unavailable. One example of DoS attacks is making the base station unavailable. In this DoS attack, the attacker frequently generates dummy data packets (spams) and makes sensor nodes relay them towards the base station as shown in Figure 2.4. The ultimate purpose of the attacker is to deplete the battery power of the sensor nodes closer to the base station. The nodes closer to the base station fail sooner because they relay more data packets than other nodes. This causes the base station to be disconnected from the sensor network and hence, unavailable. There are also other types of DoS attacks, for example, the one against the sensor nodes storage. In this attack, the attacker forces sensor nodes to store fake packets and run out of storage. The wireless communication medium makes it much easier for an adversary to launch a DoS attack. A countermeasure to most of the DoS attacks is authentication which blocks spams or fake data packets.



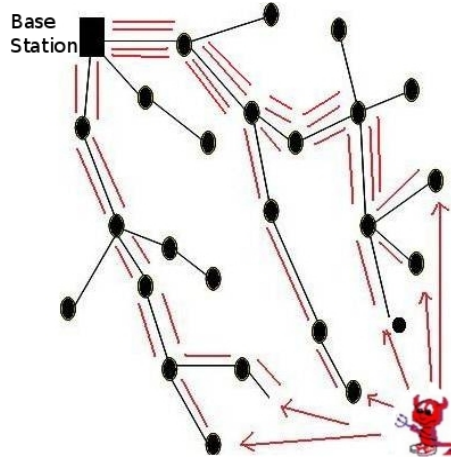


Figure 2.4: Denial of Service Attack

- *Attacks Against Routing Protocols:*
  - *Hello Flood Attack.* After the deployment of a WSN, the topology discovery phase starts. In topology discovery phase, the sensor nodes send HELLO messages to present themselves to neighboring nodes. This helps sensor nodes to find their neighbor nodes and build their routing tables. In hello flood attack, the attacker uses a powerful transmitter and gives fake information about his location. The attacker pretends to be a neighbor node to those sensor nodes who are actually far from the attacker. Because of the signal strength, the sensor nodes accept attacker as their neighbor node (Figure 2.5). The intention behind this attack is to disrupt the topology discovery process of sensor nodes.

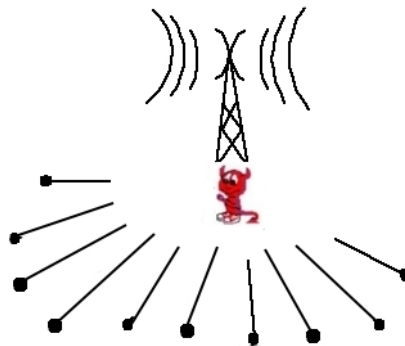


Figure 2.5: Hello Flood Attack

- *Sinkhole Attack*. In sinkhole attack, the attacker gives the wrong routing information to the sensor nodes in order to route all or nearly all traffic via an intruder node as shown in Figure 2.6. In this way, the adversary obtains a control over the whole communication in the network. Sinkhole can result into a variety of other attacks such as selective forwarding.

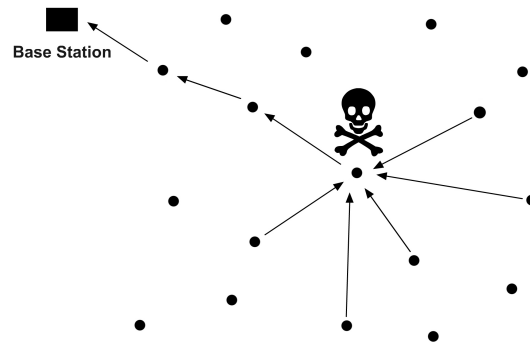


Figure 2.6: Sinkhole Attack together with Selective Forwarding Attack

- *Selective Forwarding Attack, Black Hole Attack*. In selective forwarding attack, an intruder node selectively drops some of the packets routed via it and forwards the rest, as described by Figure 2.6. The intruder node does not drop all packets to lower the risk of being detected otherwise the surrounding nodes may conclude the intruder as a dead node. If all the packets are dropped by the intruder node, the attack is called a black hole attack.
- *Wormhole Attack*. In this attack, the adversary uses an out of band low latency channel between two parts of the sensor network, which are in fact not close to each other, to route traffic. The sensor nodes, that are far from the base station, mark this route in their routing tables as the preferred (shortest) route to reach the base station. Figure 2.7 describes how a wormhole attack works. The wormhole attack can create a sinkhole where all traffic is routed via the intruder nodes considering them as the shortest routes. The aim of this attack is to pretend to be a node closer to the base station presenting a shortest path to the base station.

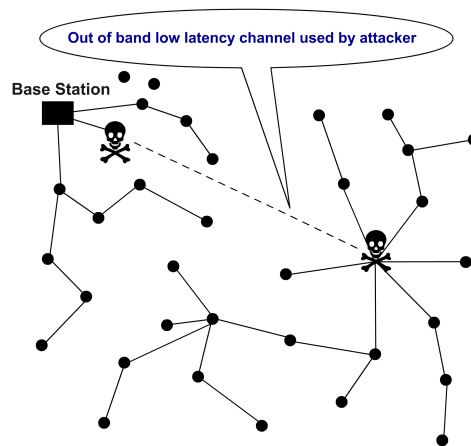


Figure 2.7: Wormhole Attack

- *Sybil Attack*. In this attack, a single intruder node adopts multiple identities as shown in Figure 2.8. Consequently, an intruder node with multiple identities presents multiple paths passing through the single physical node. Sybil attack can also result in different attacks such as a sinkhole attack.

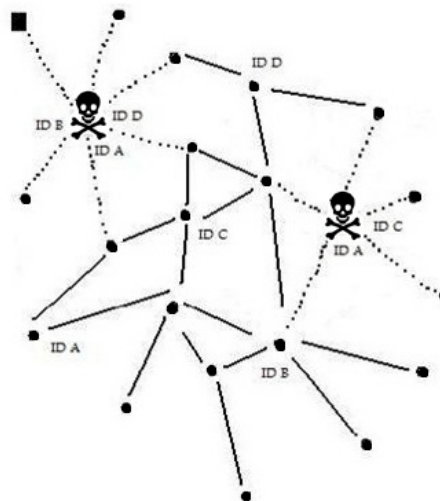


Figure 2.8: Sybil Attack

- *Replication or Clone Attack*. This attack is the one in which one or more intruder nodes copy the identity of an existing legitimate node. As a result, there are more than one sensor node in the network having the

same identity, as shown by Figure 2.9. This attack enables the replicated intruder nodes to impersonate the legitimate sensor node and participate in network communication on behalf of the legitimate node.

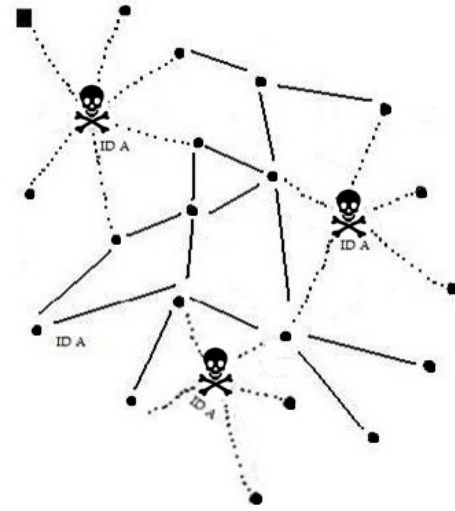


Figure 2.9: Replication or Clone Attack

A countermeasure to attacks against routing protocols is the ability of the sensor nodes to accept routing information only from the legitimate entities in the network, for example, the base station or other legitimate sensor nodes. It ultimately arises the need of authentication. Secure authenticated routing protocols are also required to avoid routing via intruder nodes.

#### 2.1.4 Constraints in Wireless Sensor Networks

A wireless sensor network is a special type of network which presents more constraints than the traditional wired and other wireless networks. These constraints make it impossible to employ the existing strong but complex security solutions to the WSNs. In order to design efficient and useful security mechanisms for WSNs, it is important to understand the constraints in WSNs. These constraints are:

- *Resource Limitations:* The primary challenge of security in WSNs is maximizing security while minimizing resource consumption. The resources in this context include energy (battery power), processing (CPU cycles), storage (memory) and the communication bandwidth.

- *Limited amount of energy.* WSNs operate under very strict energy constraints (available energy usually  $2\times$ AA batteries). Hence, security needs to limit the energy consumption to maximize the life of the individual node as well as the entire network lifetime.
  - *Limited processing capability.* Sensor nodes processors are very slow (up to few MHz) and they do not support some arithmetic and logic operations. Hence, they cannot perform very complex cryptographic operations.
  - *Limited storage capability.* The memory available for security is very low (only a few KBs). This requires that any security scheme designed for sensor networks should consume as less memory as possible.
  - *Limited bandwidth.* Wireless links have low communication bandwidth. The security schemes should consume as little bandwidth as possible.
- *Unattended Operations:* The sensor networks are usually deployed in an environment open to adversary. The unattended operations of sensor networks after deployment provides an adversary with a greater access to the sensor nodes than the typical PCs located in a secure place. The fact that the sensor nodes are not equipped with tamper resistant devices provides a complete freedom to an adversary in compromising a sensor node and obtaining all data and security material stored on it. Therefore, a security scheme should still protect against possible attacks, even if a few sensor nodes are compromised.
  - *Nature of Deployment:* The topology of the sensor network is not known prior to the deployment. Hence, the security schemes cannot benefit from the knowledge of neighboring nodes. Moreover, ad hoc deployment implies no maintenance or battery replacement after deployment which results into node failures. A security scheme should continue to provide services even in the presence of nodes failure.
  - *Unreliable Communication:* Another serious problem to sensor network security is unreliable communication. Due to unreliable communication, the packets in sensor network may drop or get corrupted and latency is high. Broadcast communication paradigm is another difficulty for security in WSNs. The wireless communication medium makes it much easier for an adversary to launch a DoS attack against any security scheme.

## 2.2 Cryptography

One essential aspect of security is that of cryptography which is necessary for secure communications. Cryptography does not only protect data from stealing or alterations but also provides necessary authentication. This section describes in detail the cryptographic concepts, notions and primitives used in this thesis.

### 2.2.1 Notations and Conventions

We let denote  $\{0, 1\}^*$  the set of finite binary strings of arbitrary length and  $\{0, 1\}^k$  the set of binary strings of length  $k$ . Let  $\mathbb{G}$  denotes a group containing a set of points under a certain group operation and  $\mathbb{F}_p$  denotes a finite field of integers modulo a prime number  $p$ .  $\mathbb{Z}_n$  denotes a set of integers modulo  $n$  which forms a finite additive group of  $n$  elements. We define  $\mathbb{Z}_n^*$  to be the subset of  $\mathbb{Z}_n$  containing elements relatively prime to  $n$  which forms a finite multiplicative group. For instance,  $\mathbb{Z}_{10}^* = \{1, 3, 7, 9\}$  [Mao03]. If  $n$  is prime, then  $\mathbb{Z}_n^* = \mathbb{Z}_n \setminus \{0\}$ . If  $\mathbb{Z}$  is a finite set, then  $t \in_R \mathbb{Z}$  denotes that  $t$  was chosen from  $\mathbb{Z}$  uniformly at random.

An algorithm which runs in polynomial time is called a polynomial time algorithm. The output of a deterministic algorithm is always unique whereas that of a probabilistic algorithm is a random variable. The adversary  $\mathcal{A}_d$  is computationally bounded with some limitations. In fact, the adversary is a probabilistic polynomial time (PPT) algorithm. A security parameter  $k$  for a cryptographic algorithm is typically length of the key used by the algorithm. The success probability of an adversary is formalized as too small to matter, i.e., a negligible function of the security parameter  $k$ . A function  $\epsilon(k)$  (usually a probability function) is negligible, if it approaches zero faster than the reciprocal of any polynomial  $p$  expressed in terms of  $k$ , for large enough  $k$ , i.e.,  $\epsilon(k) \leq \frac{1}{p(k)}$ . Informally, it implies that the probability of an event is negligible if the event happens with a probability less than the inverse of any polynomial expressed in  $k$ . A reverse of the negligible function is the non-negligible function.

### 2.2.2 Cryptographic Primitives for Authentication

The two cryptographic primitives usually used for authentication in WSNs are:

- Message Authentication Code
- Digital Signature

### 2.2.2.1 Message Authentication Code

A Message Authentication Code (MAC) [BCK96] is a symmetric key cryptography based primitive to provide message authentication. In order to protect a message  $M$  via a MAC, the sender and the verifier(s) first need to share a secret key, known as MAC key. The message sender then generates the MAC value of  $M$  by passing  $M$  through an algorithm, similar to a hash function, called MAC algorithm. A MAC algorithm takes two inputs, a message  $M$  and a MAC key  $K$ , and outputs a MAC value, i.e.,  $MAC = MAC_K(M)$ . The message  $M$  together with the MAC value is sent to the verifier(s). To authenticate a message, a verifier uses the same MAC algorithm and the shared MAC key and generates a MAC value of the received message. The computed MAC value and the received MAC value are then compared against each other to authenticate the message. Figure 2.10 shows how MACs work.

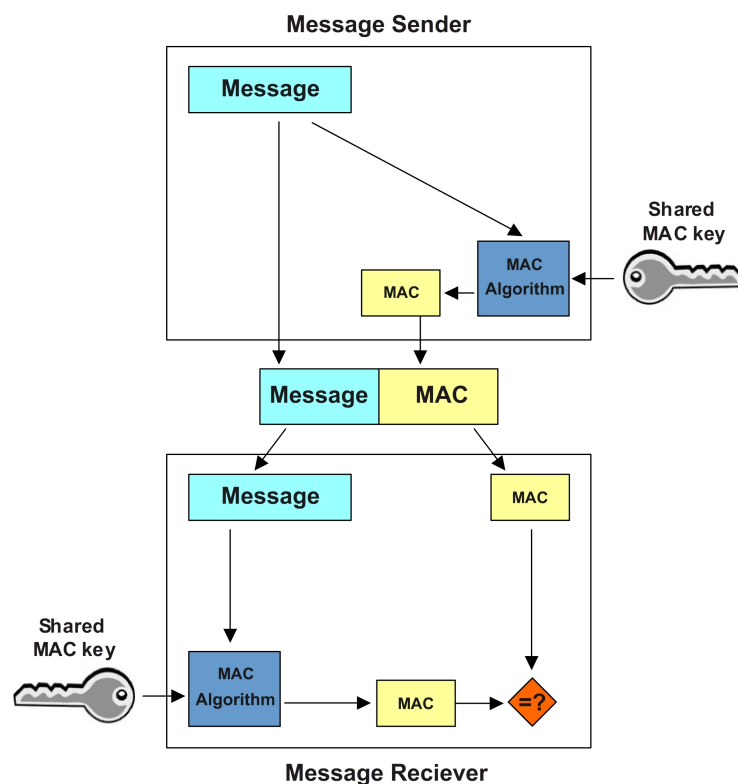


Figure 2.10: Message Authentication Code

The security of a MAC lies in the secret shared MAC key  $K$ . Given  $M$  and  $MAC$ , it is hard to find  $K$ . Without MAC key, one cannot generate a correct MAC, and thus messages that look authentic. A MAC is efficient and fast to compute

on sensor nodes and reduces the memory consumption. However, it cannot provide source authentication in broadcast scenario where the same MAC key is shared among more than one sender and one verifier to generate and verify MAC values.

### 2.2.2.2 Digital Signature

A digital signature [RSA78] is a public key cryptography (PKC) based primitive to ensure authentication. Digital signatures provide a method to assure that a message in fact originates from the person who claims to have generated the message (source authentication) and the contents of the message have not been altered in transit (message authentication or integrity). In contrast to a single shared MAC key, each user now owns a matched pair of private and public keys. The private key is kept secret by the user whereas the public key is made available to everyone. A message signer creates a signature for a message using his private key whereas a message verifier verifies the signature for the message using the signer's public key. Thus, a digital signature scheme provides a way to sign a message so that the signature can be verified publicly by anyone. Other than public and private key pair, every user owns a certificate. A certificate is an electronic document which associates a user's identity with his public key in public key cryptography settings. A certificate is usually issued by a trusted Certificate Authority (CA) and is digitally signed by the CA. It usually contains identity information of the user it identifies, his public key, expiry date and other information. Certificates help to prevent the use of fake public keys for impersonation.

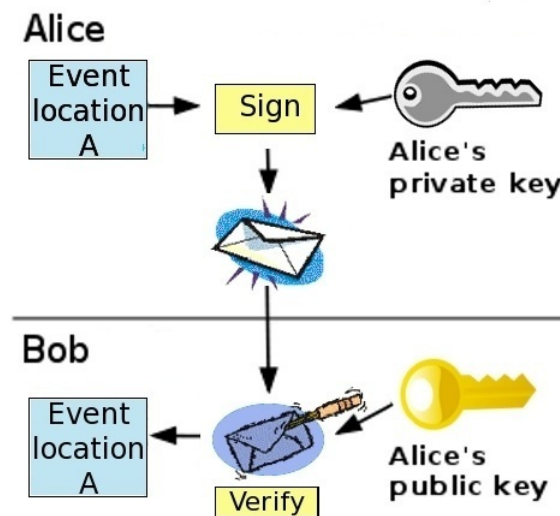


Figure 2.11: Digital Signature



The security of a digital signature lies in the secrecy of the signing key (private key). The intuitive security notion for digital signatures is the impossibility to forge a user's signature without the knowledge of his private key. Figure 2.11 illustrates the use of digital signatures.

### 2.2.3 ID-based Signature

In 1984, Shamir introduced a novel type of public key cryptography named as Identity-based cryptography or ID-based cryptography [Sha85] to replace the traditional certificate based public key cryptography. Implementing the traditional public key cryptography based signature schemes have the following requirements which do not suit WSNs environment:

1. Managing the public key infrastructure (PKI) is cumbersome (particularly in a WSN environment).
2. Public keys should be stored on each receiver to verify signed messages (increased storage overhead).
3. To avoid storage overhead, signed certificates can be sent along with the signed message to obtain public keys (increased transmission overhead).
4. Receivers should validate the signed certificate before using public key (increased computation overhead).

In contrast to certificate based public key cryptography, ID-based cryptography replaces a user's public key with his unique public identifier (ID), such as email address, phone number, physical IP address etc., which uniquely identifies him. The corresponding private key is generated by a private key generator (PKG), a trusted third party. The PKG generates a master secret key  $msk$  and a master public key  $mpk$  (called *Setup* phase). The PKG then computes the private keys for users corresponding to their IDs by using the  $msk$  (called *Key Extract* phase). The users obtain their private keys and other system parameters from the PKG via a secure channel. ID-based cryptography allows anyone to generate others' public key (public information) from a known ID information without a certificate. Consequently, the ID-based cryptography dismisses the need for certificate transmission and verification to obtain the public keys and, hence, reduces the transmission and the processing costs. For these reasons, we choose to use ID-based signature schemes to ensure authentication in our proposed framework.

In an ID-based Signature (IBS) scheme, a message signed using a signer's private key is verified using his ID information, as shown in Figure 2.12.

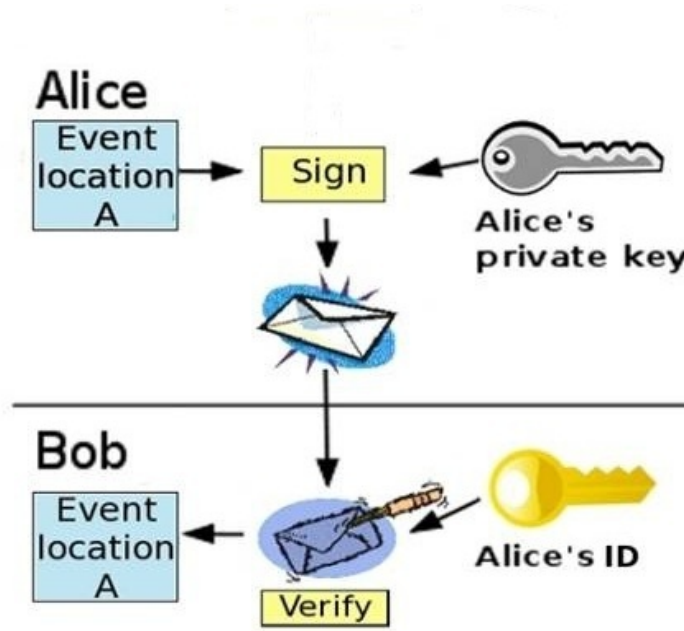


Figure 2.12: ID-based Signature

**Definition 2.1.** An IBS scheme consists of four algorithms as follows:

1. **System Setup (SS):** Given a security parameter  $1^k$ , it outputs a master secret key  $SK_{PKG}$  and system parameters  $SP$ .
2. **Key Extract (KE):** Given a user's identity  $ID_i$  and the master secret key  $SK_{PKG}$ , it outputs a corresponding private key  $D_{ID_i}$ , i.e.,

$$D_{ID_i} \leftarrow KE(ID_i, SK_{PKG}).$$

3. **Signature Generation (Sign):** Given a message  $m$  and a signing key  $D_{ID_i}$ , it outputs a signature  $\sigma$ , i.e.,

$$\sigma \leftarrow Sign(m, D_{ID_i}).$$

4. **Signature Verification (Verify):** Given a message  $m$ , user's identity  $ID_i$ , a signature  $\sigma$  and system parameters  $SP$ , it returns 1 if the signature is valid or 0 if not. Namely,

$$0/1 \leftarrow Verify(m, ID_i, \sigma, SP).$$

### 2.2.4 ID-based Online/Offline Signature

The notion of Online/Offline Signature (OOS) was first introduced by Even, Goldreich, and Micali [EGM90]. An online/offline signature is a special case of a digital signature which divides the process of message signing into two phases, the *offline* phase and the *online* phase. The *offline* phase is performed before the message to be signed becomes available. To achieve performance efficiency, the most complex computations of signature generation process are performed in the *offline* phase. The partial signature obtained as a result of these computations is known as the *offline* signature. Once the message to be signed is known, the *online* phase starts. In this phase, the *offline* signature computed during the *offline* phase is retrieved and some minor quick computations are performed to obtain the final signature of the message. The *online* phase computations are supposed to be very efficient and fast. Since the *offline* phase computations are independent of the message to be signed, they can be performed by any other resourceful device. The online/offline signature schemes are particularly useful for resource constrained application environments, for instance smart card applications and WSNs. In WSNs, an online/offline signature scheme enables a resource constrained sensor node to sign a message as soon as possible, once it has some critical event to report due to the efficient *online* phase.

A categorization of online/offline signature schemes is direct and indirect online/offline signature schemes. The first type of online/offline signature scheme takes a message and directly signs it in two phases without the involvement of any other signature scheme, for instance the online/offline signature scheme described in [XMS05]. The indirect online/offline signature scheme, on the other hand, takes a signature scheme which is not an online/offline signature scheme and provides a way to convert this signature scheme into an online/offline signature scheme, for instance the online/offline signature scheme described in [RMS08]. The online/offline signature scheme of [RMS08] also represents a class of online/offline signature schemes which allows to securely reuse the same partial *offline* signature to sign more than one message.

An ID-based Online/Offline Signature (IBOOS) scheme is an ID-based version of online/offline signature scheme which signs a message in two phases: an *Offline* phase and an *Online* phase, as shown in Figure 2.13.

**Definition 2.2.** An IBOOS scheme consists of five algorithms as follows:

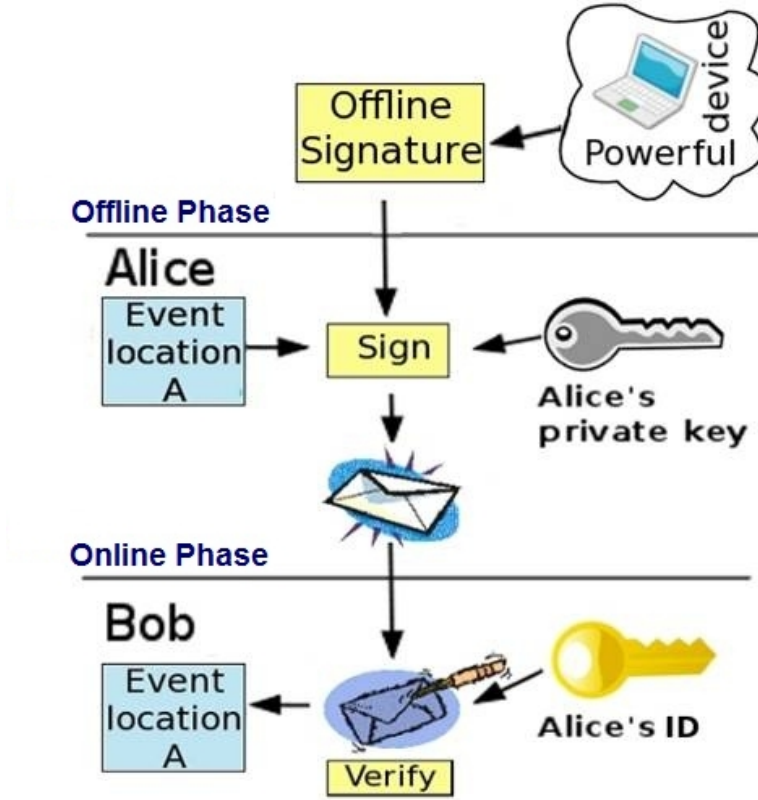


Figure 2.13: ID-based Online/Offline Signature

1. **System Setup (SS):** Given a security parameter  $1^k$ , it outputs a master secret key  $SK_{PKG}$  and system parameters  $SP$ .
2. **Key Extract (KE):** Given a user's identity  $ID_i$  and the master secret key  $SK_{PKG}$ , it outputs a corresponding private key  $D_{ID_i}$ , i.e.,

$$D_{ID_i} \leftarrow KE(ID_i, SK_{PKG}).$$

3. **Offline Signing (OffSign):** Given a signing key  $D_{ID_i}$ <sup>1</sup> and system parameters  $SP$ , it outputs an offline signature  $S$ , i.e.,

$$S \leftarrow OffSign(D_{ID_i}, SP).$$

4. **Online Signing (OnSign):** Given a message  $m$  and an offline signature  $S$ , it outputs an online signature  $\sigma$ , i.e.,

$$\sigma \leftarrow OnSign(m, S).$$

<sup>1</sup>In some IBOOS schemes, the signing key is used in online phase rather than in offline phase

5. **Signature Verification (Verify):** Given a message  $m$ , user's identity  $ID_i$ , a signature  $\sigma$  and system parameters  $SP$ , it returns 1 if the signature is valid and 0 if not. Namely,

$$0/1 \leftarrow \text{Verify}(m, ID_i, \sigma, SP).$$

### 2.2.5 Cryptosystems Used

Before we give examples of some IBS and IBOOS schemes, we describe the cryptosystems used by these schemes as well as by other cryptographic schemes in our thesis in order to understand these schemes.

#### 2.2.5.1 Elliptic Curve Cryptography

The Elliptic Curve Cryptography (ECC) is said to be ideal for implementing public key cryptography on resource-constrained systems. The reason behind is the fact that for the same level of security ECC permits shorter key sizes than other types of public key cryptography. In particular, ECC uses a considerably shorter key size and offers the same level of security as RSA offers using much larger ones. A 160-bit key in ECC is determined to have the same level of security as a 1024-bit key in RSA. Because of its smaller key size, ECC outperforms RSA on 8-bit processors [GPW<sup>+</sup>04] and results in less power consumption and higher speed.

An elliptic curve  $\mathbb{E}$  is defined in a standard, two dimensional  $x, y$  coordinate system by an equation of the form  $y^2 = x^3 + ax + b$ , where  $4a^3 + 27b^2 \neq 0$ . In this equation,  $x$  and  $y$  are variables and  $a$  and  $b$  are constants. These quantities ( $x$ ,  $y$ ,  $a$  and  $b$ ) are not necessarily real numbers, instead they may be values from any *finite field* ( $\mathbb{F}$ ). An example of a finite field is integers modulo a prime number  $p$ , i.e.,  $\mathbb{F}_p$ . All points of the form  $(x, y)$  which satisfy the above equation plus a point at infinity  $O$  lie on the elliptic curve. The set of points on an elliptic curve forms a *group*  $\mathbb{G}$  under a certain addition rule, written as  $+$ . The point  $O$  is the identity element of the group. A group  $\mathbb{G}$  generated by a point  $P$  will be  $\{O, P, P + P, P + P + P, \dots\}$ . The point  $P$  is called the *generator* of  $\mathbb{G}$ . The *order of a group* is the number of elements in the group. The *order of a point*  $P$  is the least positive integer  $n$  such that  $nP = O$ , where  $nP$  denotes  $n$  times addition of  $P$ .

**Point multiplication** is simply the multiplication of a scalar  $k$  with any point  $P$  on the elliptic curve to obtain another point  $Q$  on the same curve such that  $Q = kP$ . It is the main cryptographic operation in ECC based cryptographic

schemes. The *point multiplication* is achieved by two basic elliptic curve operations, point doubling (e.g.,  $2P$ ) and point addition (e.g.,  $2P+P$ ), and is feasible to compute by the sensor nodes [GPW<sup>+</sup>04]. The *point multiplication* operation itself is fairly simple and possible to compute, however, its inverse called the *elliptic curve discrete logarithm* is very difficult to compute and is thought to be intractable. It is defined as: given two points  $P$  and  $Q$  on an elliptic curve, where  $Q = kP$  for some random unknown integer  $k$ , compute the value of  $k$ . The security of ECC based schemes depends on the hardness of elliptic curve discrete logarithm (ECDL) problem. The ID-based signature schemes we will use to evaluate our authentication protocols are based on elliptic curve cryptography.

### 2.2.5.2 Pairing Based Cryptography

The Pairing Based Cryptography (PBC) is a class of elliptic curve cryptosystem, but have very different features than the conventional elliptic curve cryptosystem. PBC revolves around the idea of construction of a mapping between two useful cryptographic groups. A pairing function is a computable map satisfying certain special properties which allows the construction of interesting cryptographic schemes, particularly ID-based schemes.

A **pairing function**  $\hat{e}$  maps a pair of typically elliptic curve points to an element of the multiplicative group of a finite field and is of the form  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ . Here  $\mathbb{G}_1$  and  $\mathbb{G}_2$  are the two cyclic groups of prime order  $q$  written in additive notation (+) with identity element 0, while  $\mathbb{G}_T$  is a cyclic group of the same order  $q$  written in multiplicative notation ( $\times$ ) with identity element 1. The pairing map or bilinear map  $\hat{e}$  satisfies the following basic properties:

1. *Bilinearity*: For all  $P \in \mathbb{G}_1$ ,  $Q \in \mathbb{G}_2$  and  $a, b \in \mathbb{Z}_q^*$ ,

$$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} = \hat{e}(bP, aQ).$$

2. *Non-degeneracy*: There exist  $P \in \mathbb{G}_1$ ,  $Q \in \mathbb{G}_2$  such that  $\hat{e}(P, Q) \neq 1$ .
3. *Computability*: There exists an efficient algorithm to compute  $\hat{e}(P, Q)$  for all  $P \in \mathbb{G}_1$ ,  $Q \in \mathbb{G}_2$ .

Shamir proposed the idea of ID-based cryptography and described an ID-based signature scheme in 1984 [Sha85]. However, the first practical ID-based encryption scheme was found using pairing based cryptography in 2001 [BF03],

making it a popular choice for ID-based cryptography. Although pairing function has many desirable features for securing WSNs, it has been one of the most expensive cryptographic operations for resource constrained sensor nodes in terms of computational and memory requirements. Computation of a single pairing operation on a standard MICA2 [MIC] sensor node takes time almost equal to the computation of six elliptic curve point multiplication operations on the same node [OAG<sup>+</sup>11, ADLO10]. Therefore, pairing based cryptographic schemes are not considered efficient for resource constrained sensor nodes as compared to conventional non-pairing based elliptic curve cryptographic schemes. The ID-based online/offline signature schemes we initially selected to evaluate sensor nodes broadcast authentication protocol are using the pairing based cryptography.

### 2.2.5.3 Computational Assumptions and Definitions

The security of cryptographic schemes is generally reduced to a computational assumption (intractable problem). Following are the computational assumptions and other security definitions which will be used in this thesis.

**Definition 2.3. (ECDL Problem and Assumption).** Let  $\mathbb{G}$  be a group of prime order  $q$  generated by  $P$ . Given an instance  $\langle P, aP \rangle$  where  $a \in_R \mathbb{Z}_q^*$  and  $P \in \mathbb{G}$ , the Elliptic Curve Discrete Logarithm (ECDL) problem in  $\mathbb{G}$  is to compute  $a \in \mathbb{Z}_q^*$ . We say that the ECDL assumption holds in  $\mathbb{G}$  if the ECDL problem in  $\mathbb{G}$  is computationally hard to compute. In other words, the ECDL assumption holds in  $\mathbb{G}$  if there is no algorithm running in polynomial time at most which can solve the ECDL problem in  $\mathbb{G}$  with a non-negligible probability in security parameter  $k$ .

**Definition 2.4. (CDH Problem and Assumption).** Let  $\mathbb{G}$  be a group of prime order  $q$  generated by  $P$ . Given an instance  $\langle P, aP, bP \rangle$  where  $a, b \in_R \mathbb{Z}_q^*$  and  $P \in \mathbb{G}$ , the Computational Diffie-Hellman (CDH) problem in  $\mathbb{G}$  is to compute  $abP \in \mathbb{G}$ . We say that the CDH assumption holds in  $\mathbb{G}$  if the CDH problem in  $\mathbb{G}$  is computationally hard to compute. In other words, the CDH assumption holds in  $\mathbb{G}$  if there is no algorithm running in polynomial time at most which can solve the CDH problem in  $\mathbb{G}$  with a non-negligible probability in security parameter  $k$ .

**Definition 2.5. (CDH Solver Algorithm).** A CDH solver (challenger) algorithm  $\mathcal{C}$  is a probabilistic polynomial time algorithm which can compute the function  $CDH(P, aP, bP) = abP \in \mathbb{G}$  with a non-negligible probability.

**Definition 2.6. (Valid Diffie-Hellman Tuple).** Given  $P \in \mathbb{G}$  and  $a, b, c \in_R \mathbb{Z}_q^*$ , the tuple  $(P, aP, bP, cP)$  is defined as a valid Diffie-Hellman tuple if

- $c = ab$ , in elliptic curve settings of Section 2.2.5.1.
- $\hat{e}(aP, bP) = \hat{e}(P, cP)$ , in pairing based settings of Section 2.2.5.2.

**Definition 2.7. (Existential Unforgeability under Adaptively Chosen Message and ID Attacks (euf-cma-ida)).** An ID-based signature scheme is secure against existential forgery on adaptively chosen message and ID attacks or (euf-cma-ida)-secure, if there is no polynomial time adversary  $\mathcal{A}_d$  allowed to ask the signer for signature on any message of its choice that outputs a new valid ID and message-signature pair with a non-negligible advantage [CC03].

## 2.2.6 Examples of IBS and IBOOS Schemes

We now present an IBS scheme [BNN04] proposed by Bellare et al. called BNN-IBS scheme and an IBOOS scheme [XMS05] proposed by Xu et al. We named this IBOOS scheme [XMS05] as X-IBOOS scheme for convenience. The IBS scheme is based on ECC while the IBOOS scheme relies on PBC.

### 2.2.6.1 BNN-IBS Scheme

The BNN-IBS scheme has four algorithms: *Setup*, *Key Extract*, *Sign* and *Verify*.

**Setup.** This algorithm sets up the system parameters which are  $(\mathbb{E}/\mathbb{F}_p, \mathbb{G}, P, q, p, P_0, H_1, H_2)$ . The *Setup* algorithm performs the following steps:

- Specify the parameters  $\mathbb{E}/\mathbb{F}_p, q, p, P$  and  $\mathbb{G}$ , where
  - $\mathbb{E}/\mathbb{F}_p$  is an elliptic curve  $\mathbb{E}$  over a finite field  $\mathbb{F}_p$ ,
  - $q$  is a large prime number and  $p$  is the field size,
  - $P$  is a point of order  $q$  on the curve  $\mathbb{E}$  and,
  - $\mathbb{G}$  is a cyclic group of order  $q$  under the point addition “+” generated by  $P$ .
- Choose a master secret key  $s \in_R \mathbb{Z}_q^*$  uniformly.
- Compute the master public key as  $P_0 = sP$ .



- Choose one cryptographic hash function  $H_1 = \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$ .
- Choose another cryptographic hash function  $H_2 = \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$ .
- Output the system parameters  $\{\mathbb{E}/\mathbb{F}_p, \mathbb{G}, P, q, p, P_0, H_1, H_2\}$  and keep  $s$  secret.

**Key Extract.** This algorithm computes the private keys corresponding to the  $ID$ s using the well known Schnorr signature [Sch91]. Given an identity  $ID_u$  of a user  $U$ , the corresponding private key  $s_u$  is generated as

- Choose at random  $r_u \in_R \mathbb{Z}_q^*$  and compute
- $R_u = r_u P$
- $c_u = H_1(ID_u, R_u)$
- $s_u = r_u + c_u s$

The user  $U$  obtains  $(R_u, s_u)$  via a secure channel. Here,  $s_u$  is the secret information whereas  $R_u$  is public. Note that  $(R_u, s_u)$  is actually a Schnorr signature of the message (identity)  $ID_u$  signed with the master secret key  $s$  of the PKG. Finding a valid triplet  $(ID_u, R_u, s_u)$  without the knowledge of  $s$  is equivalent to forging the Schnorr signature.

**Sign.** For a user  $U$  with identity  $ID_u$  and private key  $s_u$ , this algorithm signs a message  $m$  as follows:

- Choose at random  $y \in_R \mathbb{Z}_q^*$  and compute
- $Y = yP$
- $h = H_2(ID_u, m, R_u, Y)$
- $z = y + hs_u$

The tuple  $\langle R_u, Y, z \rangle$  is  $U$ 's signature on message  $m$ .

**Verify.** Given the signature tuple  $\langle R_u, Y, z \rangle$ ,  $U$ 's identity  $ID_u$  and the message  $m$ , this algorithm verifies the signature as follows:

- Compute  $c_u = H_1(ID_u, R_u)$

- Compute  $h = H_2(ID_u, m, R_u, Y)$
- Check whether the following equation holds

$$zP \stackrel{?}{=} Y + h(R_u + c_u P_0)$$

The signature is accepted if the answer is *yes* and rejected otherwise.

**Correctness.** The correctness of the scheme follows from

$$\begin{aligned} zP &= Y + h(R_u + c_u P_0) \\ &= yP + h(r_u P + c_u sP) \\ &= yP + h(r_u + c_u s)P \\ &= yP + h s_u P \\ &= (y + h s_u)P \end{aligned}$$

The BNN-IBS scheme was later on improved by Cao et al. [CKDZ08] to reduce the signature size of the BNN-IBS scheme. The signature generation process is the same in both the original BNN-IBS scheme and the Cao's modified scheme, however the signature itself and the signature verification algorithm are different. The signature and the signature verification algorithm of the modified Cao's scheme are as follows:

$U$ 's signature on message  $m$  is now the tuple  $\langle R_u, h, z \rangle$  in modified Cao's scheme.

**Verify.** Given the signature tuple  $\langle R_u, h, z \rangle$ ,  $U$ 's identity  $ID_u$  and the message  $m$ , the *Verify* algorithm verifies the signature as follows:

- Compute  $c_u = H_1(ID_u, R_u)$
- Check whether the following equation holds

$$h = H_2(ID_u, m, R_u, zP - h(R_u + c_u P_0))$$

The signature is accepted if the answer is *yes* and rejected otherwise.

**Correctness.** The correctness of the Cao's modified scheme follows from

$$\begin{aligned} h &= H_2(ID_u, m, R_u, zP - h(R_u + c_u P_0)) \\ &= H_2(ID_u, m, R_u, (y + h s_u)P - h(R_u + c_u P_0)) \\ &= H_2(ID_u, m, R_u, yP + h s_u P - h R_u - h c_u P_0) \\ &= H_2(ID_u, m, R_u, Y + h(s_u - r_u - c_u s)P) \\ &= H_2(ID_u, m, R_u, Y) \end{aligned}$$

### 2.2.6.2 X-IBOOS Scheme

The X-IBOOS is a direct online/offline signature scheme which signs a message in two phases without the help of any other signature scheme. This scheme has five algorithms: *Setup*, *Key Extract*, *OffSign*, *OnSign*, and *Verify*.

**Setup.** This algorithm sets up the system parameters which are  $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, P_0, H_0, H_1)$ . Here  $\mathbb{G}_1$  is a cyclic additive (+) group of a prime order  $q$  generated by  $P$ .  $\mathbb{G}_2$  is a cyclic multiplicative ( $\times$ ) group with the same order  $q$ . Let  $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  be a bilinear mapping with the following properties:

1. *Bilinearity*:  $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$  for all  $Q, R \in \mathbb{G}_1, a, b \in \mathbb{Z}_q^*$ .
2. *Non-degeneracy*: There exist  $Q, R \in \mathbb{G}_1$  such that  $\hat{e}(Q, R) \neq 1$ .
3. *Computability*: There exists an efficient algorithm to compute  $\hat{e}(Q, R)$  for all  $Q, R \in \mathbb{G}_1$ .

The two hash functions chosen are defined as  $H_0: \{0, 1\}^* \rightarrow \mathbb{G}_1$  and  $H_1: \{0, 1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$ . For a random number  $s \in \mathbb{Z}_q^*$ , the master public key is computed as  $P_0 = sP$ . The master secret key is  $s$  which is kept secret.

**Key Extract.** This algorithm computes the private keys corresponding to the *IDs*. Given an identity *ID*, the corresponding private key  $D_{ID}$  is computed as

- $D_{ID} = sH_0(ID)$

**OffSign.** This algorithm computes the partial offline signature as follows:

- Pick two random numbers  $r, x \in \mathbb{Z}_q^*$
- Compute  $S = \frac{1}{r}D_{ID}$  and  $R = xP$
- Output the offline signature pair  $(S, R)$

**OnSign.** Given a message  $m$  and the offline signature pair  $(S, R)$ , the OnSign algorithm computes the online signature as follows:

- Compute  $\sigma = H_1(m, R)x + r$
- Output the resulting signature triplet  $\langle \sigma, S, R \rangle$

**Verify.** Given the signature triplet  $\langle \sigma, S, R \rangle$ , the message  $m$  and the identity  $ID$ , this algorithm checks whether  $(P_0, \sigma P - H_1(m, R)R, S, Q_{ID})$  is a valid Diffie-Hellman tuple by checking if  $\hat{e}(\sigma P - H_1(m, R)R, S) = \hat{e}(P_0, Q_{ID})$ . If true, the signature is accepted otherwise rejected.

**Correctness:**  $(P_0, \sigma P - H_1(m, R)R, S, Q_{ID})$  is a valid Diffie-Hellman tuple since,

$$\begin{aligned}
 \hat{e}(\sigma P - H_1(m, R)R, S) &= \hat{e}((H_1(m, R)x + r)P - H_1(m, R)xP, \frac{1}{r}D_{ID}) \\
 &= \hat{e}(H_1(m, R)xP + rP - H_1(m, R)xP, \frac{1}{r}D_{ID}) \\
 &= \hat{e}(rP, \frac{1}{r}sQ_{ID}) \\
 &= \hat{e}(P, sQ_{ID}) \\
 &= \hat{e}(sP, Q_{ID}) \\
 &= \hat{e}(P_0, Q_{ID})
 \end{aligned}$$

## 2.2.7 Security Proofs

In this thesis, we will use game-based reductionist proof technique to show the provable security of our cryptographic session key establishment protocol in the random oracle model.

### 2.2.7.1 Provable Security

Provable security is a methodology which uses mathematical tools to ensure that a cryptographic algorithm or protocol is secure.

According to Menezes [MvOV96],

*“A cryptographic method is said to be provably secure if the difficulty of defeating it can be shown to be essentially as difficult as solving a well-known and supposedly difficult (typically number-theoretic) problem, such as integer factorization or the computation of discrete logarithms. Thus, “provable” here means provable subject to assumptions.”*

In computational complexity approach of provable security (the one used in this thesis), a proof constitutes the cryptographic protocol, adversarial model and reductionist argument. First the cryptographic protocol is defined. Then the adversarial model (also known as security model) is defined which specifies the goals and the capabilities of the adversary. Finally, a reductionist argument is applied to show that if an efficient attacker is able to break the cryptographic protocol, then one can construct another efficient algorithm to break the underlying well-known

and supposedly intractable problem, for example integer factorization, elliptic curve discrete logarithm etc.

### 2.2.7.2 Reductionist Security Proof

An algorithm which uses the attacker as a sub-routine to break an intractable problem is called a reduction [Poi05]. The proofs in provable security provide reductions from an intractable problem (integer factorization or elliptic curve discrete logarithm etc.) to an attack against a cryptographic protocol. A reduction of an intractable problem  $P$  to an attack against a cryptographic protocol  $CP$  implies that if an attacker  $A$  breaks  $CP$  then  $A$  can be used as a sub-routine to break the problem  $P$ .

### 2.2.7.3 Game-based Security Proof

Game-based method is an approach to construct verifiable security proofs for cryptographic protocols or algorithms. In game-based security proofs, the security model is written as a game or a sequence of games and reductionist security proofs are sequences of game transformations. The security model, in game-based security proofs, is expressed as a game or a sequence of games played between a polynomial time attacker and a challenger. The attacker asks queries to the challenger who will reply the attacker's queries. A security model defines:

- the random oracles (defined in next section) to which the attacker has access,
- the challenger's response to the attacker's queries,
- the winning condition for the attacker achieving to which is the only way to break the protocol.

The attacker can ask as many queries of his choice to the challenger as he wants. At some stage, the challenger gives a challenge to the attacker and the attacker outputs his guess to the challenge. If the attacker's guess is correct, the protocol is regarded as insecure. If the attacker manages to break the security of the protocol with non-negligible probability, the challenger can use this attacker as a sub-routine to break the underlying intractable problem with non-negligible probability. A security model is described in Section 6.6.2.2 which will be used to prove the formal security of the session key establishment protocol proposed in this thesis in Section 6.3.3.1.

#### 2.2.7.4 Random Oracle Model

In cryptography, a random oracle is a data structure (a theoretical black box) used in security proofs. A random oracle is a mathematical function that responds to every query asked. It is a pseudo-random function which maps every possible query to a random response from its output domain. More precisely, it maps elements from a set of bitstrings to uniformly and independently sampled bitstrings [Poi05]. The random oracles are instantiated with cryptographic hash functions in practice.

A random oracle model is a popular model used in the security proofs of the cryptographic protocols. A random oracle model uses a hash function as a random oracle which produces a truly random value for each new query asked by the adversary. It produces the same results only if the same query is asked repeatedly. In random oracle model, the adversary is given a complete access to the cryptographic protocol, but as a black-box and he can ask a query of his choice which is answered correctly by the random oracles in constant time [Poi05].

### 2.3 Concluding Remarks

The particular characteristics of WSNs make them prone to a variety of security attacks. On the other hand, resource constrained nature of the sensor nodes is a hurdle in applying complex cryptographic primitives to secure wireless sensor networks. A MAC is efficient to compute for sensor nodes in terms of resource consumption but it cannot provide source authentication in broadcast scenario. The traditional PKC based digital signatures, on the other hand, require public keys to verify a signed message. ID-based cryptography gives a solution to the public key and certificate management problem. ECC is efficient to implement PKC on sensor nodes when compared to RSA and PBC. Provable security ensures that a cryptographic protocol is secure and it possesses the required security properties.

# Chapter 3

## Authentication in Wireless Sensor Networks: A Review

***Chapter Overview:** This chapter provides a detailed review of the existing research work related to authentication problems in wireless sensor networks including broadcast authentication and outside user authentication. While reviewing the existing authentication schemes, it highlights their limitations in the context of both authentication problems.*

### 3.1 Introduction

Authentication is the act of proving one's identity. The provision of authentication in WSNs is a challenging task due to the nature of the network. This chapter reviews the existing solutions to authentication problems in WSNs in detail with the main focus on two authentication problems, i.e., broadcast authentication and outside user authentication. Furthermore, existing key establishment protocols to establish a session key between a user and a sensor node are also the focus of this survey.

In traditional networks, authentication is achieved via digital signatures, i.e., via PKC. However, PKC had been thought too resource hungry to use in WSNs because of its large processing and storage requirements. So, for a long time, the security solutions for the WSNs were considered only by using symmetric cryptography. In 2004, Gura et al. [GPW<sup>+</sup>04] showed the possibility of both ECC and RSA on 8-bit processors with ECC demonstrating a performance advantage over RSA. Since then PKC has also become a part of WSN security schemes. The following sections discuss the previously proposed schemes of broadcast authentication and outside user authentication and their inadequacy to address certain issues.

## 3.2 Broadcast Authentication

The existing work in the area of broadcast authentication in WSNs can be categorized as symmetric cryptographic schemes (MAC based schemes) and asymmetric cryptographic schemes (digital signature based schemes). All the existing broadcast authentication schemes, except the two of them, provide a solution by assuming the broadcast sender either a base station or any other resourceful device but not a sensor node.

### 3.2.1 Schemes Based on Symmetric Cryptography

In WSNs, authentication was conventionally provided through Message Authentication Code (MAC), a symmetric cryptographic approach. In a typical pairwise MAC based authentication scheme, the MAC generated by a sender is verified by the receiver using the same MAC key shared between the two. This approach provides an efficient solution to the pairwise authentication problems in WSNs. However, the typical pairwise MAC based authentication approach cannot be applied to the broadcast settings in WSNs due to the presence of compromised nodes in the network. In a broadcast scenario, the MAC key used to authenticate the broadcast messages needs to be shared among all potential receivers. This facilitates any compromised broadcast receiver node, with the possession of MAC key, to successfully modify the messages sent by a legitimate broadcast sender. Moreover, the compromised node can send fake messages on behalf of a legitimate sender. Therefore, a typical MAC based authentication scheme can only provide mutual (pairwise) authentication in WSNs and cannot handle the broadcast authentication.



A solution to this problem was given by  $\mu$ TESLA<sup>1</sup> [PST<sup>+</sup>02], a symmetric cryptographic scheme described later, which first addressed the problem of broadcast authentication in WSNs. The major symmetric cryptography based broadcast authentication schemes for WSNs are variations of  $\mu$ TESLA scheme. These schemes can be further categorized depending on the type of broadcast senders. One category is where the broadcast sender is either the base station or any other resourceful device while the second category assumes the resource constrained sensor nodes as the broadcast senders.

### 3.2.1.1 Schemes with Resourceful Broadcast Sender

The broadcast authentication schemes in this section assume the resourceful devices as the broadcast senders and the resource constrained sensor nodes as the receivers.

$\mu$ TESLA [PST<sup>+</sup>02] was mainly proposed for the base station to sensor nodes broadcast authentication.  $\mu$ TESLA is a MAC based symmetric cryptographic protocol which introduces asymmetry through the delayed disclosure of MAC key. In  $\mu$ TESLA, the MAC key used to authenticate broadcast messages is not shared among the broadcast receivers before the actual broadcast. A broadcast message is sent along with a MAC generated by a MAC key. This MAC key is initially unknown to all broadcast receivers. After some delay, when every broadcast receiver has received the broadcast message, the MAC key is sent to all broadcast receivers who use the received MAC key to authenticate broadcast message received earlier. This delayed disclosure of MAC key helps to avoid the attacks launched by the compromised sensor nodes.  $\mu$ TESLA protocol works in multiple phases: Sender Setup, Broadcasting Authenticated Packets, Bootstrapping New Receivers, Authenticating Packets, and Nodes Broadcast. The following is the detailed description of  $\mu$ TESLA protocol.

*Sender Setup.* Each MAC key in  $\mu$ TESLA is a key from a key chain of length  $n$  generated by applying a one-way hash function to the previous key. In sender setup phase, the broadcast sender first generates a sequence of MAC keys (key chain). To generate the key chain, a sender randomly chooses the last key  $K_n$  and calculates the rest of the keys by repeatedly applying the one-way hash function  $F$  to the previous keys as follows:

$$K_{n-1} = F(K_n), K_{n-2} = F(K_{n-1}), \dots, K_0 = F(K_1)$$

---

<sup>1</sup>Micro Timed Efficient Stream Loss-tolerant Authentication ( $\mu$ TESLA)

Due to the one-way hash function  $F$ , it is possible to compute  $K_{i-1}$  from  $K_i$  but the reverse is computationally infeasible. Hence, any key  $K_i$  can be used to authenticate the subsequent keys  $(K_{i+1}, K_{i+2}, \dots, K_n)$  from the same key chain. For instance,  $K_1$  is authentic if  $K_1 = F(K_0)$  and  $K_2$  is authentic if  $K_2 = F(K_1)$  or  $F(F(K_0))$ . The first key from the key chain, i.e.,  $K_0$ , is therefore distributed to every potential receiver in the beginning of the protocol to authenticate the subsequent received key(s). The key chain is then used for authenticated broadcast in the reverse order of generation, i.e.,  $K_1, K_2, \dots, K_n$ . The key used to authenticate the later key(s) is called the key commitment.

*Broadcasting Authenticated Packets.* In this phase, the broadcast time is divided into  $n$  uniform time intervals i.e.,  $1, 2, \dots, n$  and the sender associates each MAC key from the key chain  $(K_1, K_2, \dots, K_n)$ , in the same order, to a corresponding time interval. MACs for all broadcast packets during a time interval  $i$  are computed using the key  $K_i$  associated to that time interval  $i$ , for  $i = 1, 2, \dots, n$ . In a time interval  $i$ , the sender computes the MAC of the packet(s) using  $K_i$  and broadcasts. The sender waits for a certain predefined time period  $\sigma$  to make sure that every receiver in the network has received the broadcast packet(s). He then reveals the MAC key  $K_i$  after the end of the time interval  $i$ . The delay  $\sigma$  depends on the round trip time between the sender and the farthest receiver(s) in the network. This time delay ensures that when the MAC key is being released, every receiver has already received all the broadcast packets of the time interval  $i$  and the compromised sensor nodes cannot exploit the knowledge of that MAC key.

*Bootstrapping New Receivers.* In order to authenticate a received MAC key belonging to a certain one-way key chain, the new receiver needs to have a previous authentic MAC key from the same key chain as a commitment to the entire key chain. To obtain the key commitment, a new receiver in time interval  $i + 1$  sends a request message to the broadcast sender. The broadcast sender replies the new receiver with a key  $K_i$  of the one-way key chain used in the past time interval  $i$  and the other  $\mu$ TESLA parameters, for instance, the duration of a time interval, the delay in disclosure of MAC key etc.

*Authenticating Packets.* After receiving a packet along with the MAC, the receiver *first* checks whether the MAC key for this packet has not already been disclosed. If the packet passes this security check, the receiver stores it and waits for the corresponding MAC key otherwise discards the packet. In the *second* step, the receiver verifies the newly received MAC key. When a new MAC key (say  $K_i$ ) of

a previous time interval, not disclosed before, is received by the receiver, the receiver verifies  $K_i$  using the previously disclosed authentic MAC key  $K_{i-1}$  by performing a test  $K_{i-1} = F(K_i)$ . The successful verification implies that the received MAC key is a valid key from the sender's key chain. In the *third* step, the receiver authenticates all broadcast packets that were received during the last time interval  $i - 1$  using  $K_i$ . The receiver also replaces the previously stored key  $K_{i-1}$  with  $K_i$ .

*Sensor Nodes Broadcast.*  $\mu$ TESLA is mainly proposed to facilitate authenticated broadcast by the base station which is a resourceful device. A sensor node, with a limited storage capability, cannot store the keys of a long key chain and the  $\mu$ TESLA parameters of all other sensor nodes in the network including their key chain commitments. Moreover, exchanging the authenticated key chain commitments securely and broadcasting the disclosed MAC keys to all receivers in the network are other problems in case of sensor nodes broadcast. All these facts do not allow sensor nodes to become broadcast senders in  $\mu$ TESLA. Therefore,  $\mu$ TESLA suggests sensor nodes to broadcast messages via the base station, i.e., a sensor node sends a broadcast message to the base station which then broadcasts this message on behalf of that sensor node.

*Limitations:* In  $\mu$ TESLA, a broadcast sender is required to store long key chains consisting of  $n$  MAC keys while a receiver is required to store key chain commitment of every broadcast sender.  $\mu$ TESLA suffers from the following major problems:

- Does not provide quick authentication since the receiver waits for the MAC key to authenticate a broadcast message.
- Can cause a DoS attack against the storage of sensor nodes by forcing them to store packets until they receive authentication keys to authenticate them.
- Very slow for large scale sensor networks since a sender waits until the message reaches the other end of a large network before disclosing MAC key.
- Suffers from the problem of distribution of key chain commitment of a new key chain to all broadcast receivers, once all the keys from a previous key chain have been used.
- Requires time synchronization between the sender and the receiver.
- Assumes the base station as the only broadcast sender in sensor network and allows a sensor node to broadcast a message only via base station.

- Not scalable in terms of number of broadcast senders since the sensor nodes (receivers) cannot store the initial  $\mu$ TESLA parameters (i.e., key chain commitment, duration of time interval etc.) of a large number of broadcast senders.

Different variations [LN04, LNZJ05, DG06, GD07] of  $\mu$ TESLA scheme have been proposed later on to address the problems of distribution of new key chain commitment, scalability in terms of number of broadcast senders, and speed issue for large scale sensor networks, which are discussed below.

**Multi-level  $\mu$ TESLA** [LN04] attempts to handle the problem of distribution of new key chain commitments to all broadcast receivers. Once all the keys from a key chain have been used, distributing the new key chain commitment to all broadcast receivers in the network is a problem faced in  $\mu$ TESLA. The new key chain commitment can be sent to all broadcast receivers in the same way as other broadcast messages. However, any receiver that does not receive a key chain commitment due to the packet loss in the network would not be able to authenticate future messages. Moreover, an attacker can target the packet containing the key chain commitment and interrupt its distribution, for example, by launching a jamming attack. The motivation behind *Multi-level  $\mu$ TESLA* is to prolong the life time of  $\mu$ TESLA broadcast without storing the long key chains on broadcast senders and authentically distributing the new key chain commitments to the broadcast receivers. This scheme introduces different levels of key chains. The lower level key chains are used to authenticate broadcast messages (authenticated message broadcast) from the base station. The higher level key chains are used to authenticate the new key chain commitments of lower level key chains (authenticated distribution of new key chain commitment).

*Limitations:* Although this variation of  $\mu$ TESLA prolongs the life time of  $\mu$ TESLA broadcast by handling the problem of distribution of key chain commitment, it still suffers from the delayed authentication, DoS attack and scalability problems of  $\mu$ TESLA.

**Multi-Sender  $\mu$ TESLA** [LNZJ05], another variation of  $\mu$ TESLA, provides scalability in terms of number of broadcast senders (a few resourceful broadcast senders only). The idea behind this scheme is that the time period is divided among multiple senders (say  $m$ ) instead of one broadcast sender. The broadcast senders broadcast messages one by one in their allocated time intervals. The life

time (time interval) of each broadcast sender is further divided into  $n$  intervals. The  $\mu$ TESLA parameters (key chain commitment, starting time, duration of time intervals etc.) and the certificates (containing  $\mu$ TESLA parameters) are generated for all broadcast senders. To start an authenticated broadcast in a time interval  $i$ , a sender  $S$  first broadcasts its certificate containing its key chain commitment at the beginning of  $i$ . The receiving sensor nodes verify the certificate using some stored information. After that, the sender broadcasts messages during the time interval  $i$  (which is further divided into  $n$  time intervals) in the same way as in the original  $\mu$ TESLA scheme. A sender is allowed to broadcast messages during its predefined time intervals only.

*Limitations:* This variation of  $\mu$ TESLA addresses the scalability problem to some extent, however, it does not allow multiple senders to broadcast at the same time. Due to the limited storage, the sensor nodes cannot store packets from multiple senders at the same time before receiving authentication keys. Hence, different senders are allowed to broadcast messages turn by turn in predefined fixed time intervals only which does not suit real time applications. Besides, this scheme also suffers from the delayed authentication and above mentioned DoS attack.

**L-TESLA** [DG06, GD07] aims to speed up the authentication process of  $\mu$ TESLA for large scale sensor networks. It assumes the presence of trusted sensor nodes in the network who are secure with more computing and communication capabilities than the ordinary resource constrained sensor nodes. The basic idea behind this scheme is to divide the sensor nodes in the network into equal subsets and provide the parallel broadcast and authentication of a single message in these subsets. In this scheme, the whole sensor network is divided into small virtual networks each containing a subset of the sensor nodes. For each small network, there is one trusted node acting as a broadcast sender for that small network. The trusted node plays the same role as is played by the base station in the original  $\mu$ TESLA scheme. It maintains its own key chain and other  $\mu$ TESLA parameters for that small network. The  $\mu$ TESLA parameters of the trusted node are stored on each sensor node in its small network. Whenever the trusted node receives a broadcast message (either directly from the base station or from another trusted node), it verifies the message and computes a new MAC for this message using a MAC key from its own key chain. It then broadcasts this message along with the new MAC to the sensor nodes in its own small network (subset of sensor nodes). Once all sensor nodes in its small network have received this message, the trusted node releases the

corresponding MAC key to authenticate that message. Since the size of a trusted node's corresponding network is small, every sensor node receives this message in a short order. The trusted node also forwards this message to other nearby trusted nodes which after verification repeat the process. The communication among the trusted nodes is also carried out through  $\mu$ TESLA using a separate key chain. The motivation behind this scheme is to decrease the delay in message authentication for real time data in large scale WSNs.

*Limitations:* Although *L-TESLA* decreases the authentication delay of  $\mu$ TESLA for large scale sensor networks, the presence of trusted nodes in sensor networks is not usual. Moreover, the experimental results of the scheme show that this scheme does not work well with unevenly distributed networks. For such networks, it rather increases the overall delay as compared to the delay incurred in original  $\mu$ TESLA scheme. Evenly distributed sensor networks are rare since the topology of a WSN is unpredictable.

### 3.2.1.2 Schemes with Resource Constrained Broadcast Sender

The broadcast authentication schemes in this section assume the resource constrained sensor nodes as the broadcast senders as well as the receivers.

The first attempt in this regard is [CC05] which enables sensor nodes to broadcast messages to nearby sensor nodes only. [CC05] proposes to use *Multi-level*  $\mu$ TESLA protocol, a variation of  $\mu$ TESLA, for sensor nodes broadcast authentication. To use *Multi-level*  $\mu$ TESLA for sensor nodes broadcast, a sensor node needs to distribute its key chain commitments to all nearby sensor nodes. However, distributing the key chain commitments authentically to all nearby sensor nodes is a problem of  $\mu$ TESLA. One possibility is to store the key chain commitments on sensor nodes before the deployment of sensor network. However, before deployment a sensor node does not know who will be its neighboring nodes after deployment. Due to the limited storage, it is not possible for a sensor node to store the key chain commitments of all  $N - 1$  sensor nodes (for a large scale sensor network of  $N$  sensor nodes, the number  $N$  is in thousands). Another possibility is to distribute (broadcast) the key chain commitments to the nearby sensor nodes after the deployment which is also not an easy task. To address this problem, a scheme to bootstrap key chain commitments for *Multi-level*  $\mu$ TESLA after the deployment of sensor network has been proposed in [CC05].

In [CC05], the base station calculates a key for each sensor node  $s$  before deployment, known as Identification Key ( $IDK$ ), as follows,

$$IDK_s = Hash(GMK \parallel s)$$

and stores it on  $s$ , where  $GMK$  is a group master key known to the base station only. During the *bootstrapping* phase, each sensor node  $s$  broadcasts its key chain commitment to all of its neighboring nodes, encrypted by its  $IDK_s$ . After a fixed time interval, the base station broadcasts  $GMK$  to all sensor nodes in the network. Each sensor node then computes the Identification Key,  $IDK_s$ , of each nearby sensor node  $s$  using  $GMK$  and decrypts the received key chain commitments.

*Limitations:* This scheme uses a  $\mu$ TESLA based scheme for the broadcast message authentication, and thus limits the number of broadcast senders and receivers. Moreover, it allows broadcast to the neighboring nodes only. Furthermore, the key chain commitments are distributed in bootstrapping phase only. The sensor nodes added after bootstrap phase are not able to obtain/distribute the key chain commitments from/to nearby sensor nodes and therefore cannot participate in broadcast communication.

To address the problem of addition of new sensor nodes in [CC05], the scheme in [KKLL07] proposes to maintain a key chain of group master keys  $GMK$ s, called  $GMK$  hash chain. After *bootstrapping* phase in [CC05], if new sensor nodes are to be added to the system, their Identification Keys ( $IDK$ s) are calculated using the next key from  $GMK$  hash chain. The new sensor nodes broadcast their key chain commitments, encrypted with their  $IDK$ s, to the nearby sensor nodes. After some time, the base station releases the corresponding  $GMK$ . The whole process of [CC05] then repeats in the same way to obtain the key chain commitments of the newly added sensor nodes.

*Limitations:* This approach enables newly added sensor nodes to distribute their key chain commitments to nearby sensor nodes. However, the new sensor nodes are not able to obtain the key chain commitments of the old sensor nodes who are close to them and hence, not able to authenticate broadcast messages from the old sensor nodes. The number of broadcast senders is also limited in this scheme.

*Comments.* Since all the above mentioned symmetric cryptography based broadcast authentication schemes are based on the basic mechanism of  $\mu$ TESLA, they all inherit the flaws of  $\mu$ TESLA scheme. A summary of major issues with  $\mu$ TESLA based schemes is as follows:

- Delayed authentication
- DoS attack due to delayed authentication
- Time consuming for large scale sensor networks
- Lack of scalability in terms of number of broadcast senders
- Multiple senders cannot broadcast simultaneously
- Multiple senders can broadcast only in their predefined fixed time slots
- Distribution of key chain commitments for multiple broadcast senders
- Broadcast by sensor nodes not supported

### 3.2.2 Schemes Based on Asymmetric Cryptography

Multiple digital signature based broadcast authentication schemes for WSNs have been proposed in [RLZ07, RLZM07, CKDZ08] addressing several of the limitations of  $\mu$ TESLA based schemes. All these schemes assume resourceful devices as the broadcast senders and not typical sensor nodes. The first five schemes [**CAS**, **DAS**, **MAS**, **BAS**, **HAS**] use the traditional PKC based signature schemes where a message signer signs a message using his private key and the receiver verifies the signed message using the signer's public key. In fact, they provide a solution to the public keys and certificates management problem faced in WSNs. The last two schemes [**IDS**, **IMBAS**] use ID-based signature schemes.

**CAS** [RLZ07, RLZM07] discusses a certificate based authentication scheme for WSNs. A certificate contains the public key of broadcast sender signed by the private key of a fixed sink or the base station. A broadcast sender signs a message using its private key and broadcasts the signed message along with its certificate. On receiving a message, the receiver first verifies the signed certificate using the public key of the base station. If the verification succeeds, the receiver obtains the signer's public key from the certificate. The receiver then verifies the signed message using the received public key. To avoid the certificate transmission and verification for every message, **DAS** [RLZ07] suggests every receiver (sensor node) to store the IDs and the corresponding public keys of all legitimate broadcast senders. A broadcast sender now signs a message using its private key and broadcasts the signed message along with its ID and public key pair. The receiver checks whether the received



ID and public key pair is present in its local memory or not in order to verify the legitimacy of received public key. If this check passes, the receiver verifies the signed message using this public key.

*Limitations:* **CAS** and **DAS** both increase the communication overhead where the certificates or public keys are transmitted for every signed broadcast message. Furthermore, the receiving sensor nodes in **CAS** need to verify two signatures which increases computation overhead; first to verify the signed certificate to obtain the sender's public key, and second to verify the signed message. On the other hand, storing public keys of all broadcast senders in **DAS** increases storage overhead on sensor nodes and hence, limits the number of broadcast senders for a large scale sensor network due to the limited storage capabilities of typical sensor nodes.

Merkle hash tree [Mer80] has been suggested to use in another scheme **MAS** in [RLZM07] to manage the public keys of senders. Merkle hash tree helps to avoid the storage overhead of **DAS**. In this scheme, each leaf node of the Merkle hash tree is a hash value of the public key of a broadcast sender. The value of each internal node is derived as the hash of two of its children nodes' values. Each receiver stores the value of the root node of Merkle hash tree in order to authenticate the public keys of broadcast senders. A sender, along with the signed message, broadcasts the hash values of all sibling nodes of the nodes on the path from its corresponding leaf node to the root node named as *AAI* and its own public key. The receiver constructs a partial hash tree using the sender's public key and *AAI*. If public key is authentic, the calculated root value is the same as the one already stored on the receiver.

*Limitations:* For  $N$  broadcast senders, *AAI* for each sender contains  $\log_2 N$  hash values. This approach increases message size and therefore the transmission cost especially for large scale sensor networks where  $N$  corresponds to thousands of sensor nodes. Moreover, this approach is not dynamic. The reason is that once the broadcast senders are decided, the Merkle hash tree of senders' public keys is calculated and the root node value is stored on every receiver, adding a new broadcast sender is an issue.

**BAS**, another scheme proposed in [RLZ07], uses bloom filter [Mit02] to store the public keys. A bloom filter is a space-efficient probabilistic data structure to represent a set and to test whether an element is a member of that set. In a bloom filter, false positives are possible but false negatives are not. It implies that a test result returns either an element is inside the set (may be wrong) or definitely not in the set. Unlike DAS, BAS does not preload each sensor node with the ID and

public key pairs of the broadcast senders. Instead, each sensor node only stores the hash mappings of the ID and public key pairs employing the bloom filter. When a sensor node receives a broadcast message together with the sender's ID and public key pair, it verifies the authenticity of the received public key by checking if the corresponding hash mapping of the public key is stored in its local memory or not. **HAS** [RLZ07], on the other hand, describes a bloom filter and Merkle hash tree based hybrid authentication scheme to enhance the number of broadcast senders in BAS. This approach combines the previously described BAS and MAS schemes and trades the message length for the storage space to obtain scalability in terms of number of broadcast senders.

*Limitations:* **BAS** suffers from the probability of false positives while authenticating public keys. This probability increases with the increase in number of broadcast senders and therefore limits the number of broadcast senders. Considering a reasonable probability of a false positive and the capabilities of a typical sensor node, the maximum number of broadcast senders supported by BAS is about 434 [RLZ07]. **HAS** supports more broadcast senders than **BAS** but at the cost of increased message size. The addition of new broadcast senders is a problem in both schemes requiring an update of a bloom filter or a bloom filter and Merkle hash tree stored on every sensor node. Both schemes require the transmission of a signer's public key with every message incurring transmission overhead.

*Comments.* All the above mentioned asymmetric cryptography based authentication schemes discussed in [RLZ07, RLZM07] mainly focus on the management of public keys and/or certificates in WSNs. All these schemes assume either ECDSA (Elliptic Curve Digital Signature Algorithm) [JMV01] or RSA [RSA83] as underlying signature scheme to sign a message.

**IDS** [RLZM07] uses a pairing cryptography based digital signature for broadcast authentication in WSNs. **IMBAS** [CKDZ08] uses another digital signature scheme based on ECC for broadcast authentication in WSNs. Since both these schemes are based on ID-based cryptography, they solve the problem of public key and certificate management.

*Limitations:* The message signers in both of these schemes **IDS** and **IMBAS** are assumed to be powerful devices. Although it is possible for resource constrained sensor nodes devices to compute a pairing operation, it is the most expensive cryptographic operation for sensor nodes in terms of resource consumption. One can conclude that ECC based signatures are more efficient than pairing based

signatures for sensor nodes in terms of computation time and energy consumption. However, they both still take a considerable time in signing a message on resource constrained sensor nodes. Consequently, these both authentication schemes consume considerable time on sensor nodes to broadcast a signed message.

### 3.2.3 Discussion

All the symmetric and asymmetric cryptography based authentication schemes that we have presented suffer from significant limitations. In case of symmetric schemes ( $\mu$ TESLA based schemes), these limitations are the broadcast at regular and predefined intervals, the storage of  $\mu$ TESLA parameters, the distribution of key chain commitments and the delayed authentication.  $\mu$ TESLA based schemes fail to provide a solution to the real-time applications of WSNs. In case of asymmetric schemes (digital signature based schemes), these limitations are the management of public keys and certificates and the cost of applying PKC on sensor nodes particularly the time cost. The former raises the scalability problem and the latter is critical for real-time applications. Furthermore, all of the above mentioned schemes except [CC05, KKLL07] assume broadcast senders to be powerful devices and not ordinary sensor nodes.

## 3.3 Outside User Authentication

As mentioned earlier, the outside user access to the sensor nodes data requires to handle two basic tasks of ‘user authentication’ and ‘session key establishment’.

### 3.3.1 User Authentication

User authentication in WSNs may be implemented using some user credentials for instance user’s ID and a password known only to the user. It requires sensor nodes to store the ID and password pair of each user. However, a single compromised node will reveal the passwords of all the users. Alternatively, user authentication may be enforced with a public key cryptosystem with public and private keys. A simple approach to handle user authentication is a *centralized* mechanism. In a *centralized* approach, the user sends his login request to a central entity, say a base station. The base station, after successful user authentication, forwards the user query to the sensor nodes to obtain the requested data from them. The base station then replies the user with data obtained from the sensor nodes. The user can also send the login

request directly to the sensor nodes. The sensor nodes forward the user information to a central entity e.g., the base station. The base station verifies the legitimacy of the user request and decides whether the access should be granted or not. The base station then replies back to the sensor node with the verification outcomes. Based on the outcomes, the sensor nodes either provide the requested data to the user or refuse to process the user request. Both centralized approaches are simple and easy to deploy because of the fact that the base station is a powerful device which can perform complex computations to authenticate a user. An alternative approach to handle user authentication is a *distributed* mechanism. In *distributed* approach, the sensor nodes who receive the user request locally verify it and process the user query. There is no involvement of a third party in this approach.

### 3.3.1.1 Centralized Schemes

The centralized user authentication schemes described in [WZCW06, TJY07, Lee08, Das09] divide the user authentication process into three phases: registration, login, and authentication. The registration phase is carried out via a secure channel in which each user registers himself to a registration node, for instance, the base station or any dedicated node. After registration, whenever the user wants to access data from the sensor nodes, he sends a login request to the login node with his credentials. The login node forwards user's credentials to the registration node who verifies the authenticity of the user and gives feedback to the login node. Depending on the feedback from the registration node, the login node either accepts or rejects the user authentication request.

*Limitations:* Although, the centralized user authentication schemes are easy to deploy and efficient for sensor nodes in terms of processing, they all suffer from certain problems. Firstly, they carry the limitation of a single point of failure (registration node or the base station). If the third party responsible to authenticate users fails, the whole scheme will fail. Secondly, they require one round trip communication between the registration node and the sensor node (login node) for every user request and hence, result in increased communication overhead. They also cause traffic congestion in the network in case of multiple simultaneous user requests. Thirdly, they are vulnerable to a severe DoS attack against sensor network. In this attack, an adversary sends fake user requests to the login nodes forcing them to forward fake user requests towards the registration node for verification. The result is in-network traffic congestion by increased communication and depletion of sensor

nodes battery power while relaying fake requests. Furthermore, they do not deal with the session key establishment between the user and the sensor nodes for secure query and data transfer.

### 3.3.1.2 Distributed Schemes

A distributed user authentication scheme was proposed by [BGK04] which first addressed the outside user authentication problem in WSNs. This scheme realized the presence of compromised sensor nodes in the network affecting the user authentication process and introduced the notion of a  $(t, n)$ -threshold authentication. A  $(t, n)$ -threshold authentication means the authentication succeeds only if the user successfully authenticates to at least  $(n - t)$  out of  $n$  sensor nodes ( $t$  is the number of potentially compromised nodes in the network). In this scheme, a user  $U$  separately authenticates himself to each of  $n$  sensor nodes in his communication range. If  $U$  successfully authenticates himself to a node  $n_i$ , the node  $n_i$  broadcasts to other  $n - 1$  nodes its vote ‘yes’ otherwise sends nothing. If within a timeout time,  $n - t$  or more ‘yes’ votes are collected, the user is successfully authenticated otherwise not.

*Limitations:* The functionality of this protocol is compromised by the fact that a legitimate user will not be authenticated if  $n - t$  votes are not collected. Furthermore, the communication cost is increased when each sensor node broadcasts its authentication results to every other sensor node within its communication range.

**RRUASN** [BGR05], a PKC based distributed user authentication scheme, was proposed by the same author later on. In this scheme, every user obtains from the base station a private key and a certificate containing the user’s corresponding public key. This certificate is signed by the base station using the private key of the base station. In first step of the protocol, a user sends his signed certificate along with his identity  $U$  to the sensor nodes in his communication range. The sensor nodes store this certificate and send a challenge nonce back to the user. In second step, the user signs this nonce together with his identity  $U$  using his private key and sends them to the sensor nodes. The sensor nodes first verify the user certificate to obtain user’s public key and then verify the signed nonce. The successful verification proves the legitimacy of the user. In order to revoke the access rights of a user after user’s access time period expires, the base station periodically updates its public and private keys and broadcasts its public key to all sensor nodes in the network. Thus, only a user who possesses a certificate signed by the current private key of the base station can access data.

*Limitations:* In this scheme, the receiving sensor nodes verify two signatures for each authentication request; one to verify the signed user's certificate and second to verify the signed nonce. Thus, verifying a user is expensive for a sensor node in terms of computation cost. Moreover, the messages from a user to the sensor nodes are sent in two steps: first the user's certificate which is stored by the sensor nodes and then the signed value of nonce (originally received from the sensor nodes). An adversary may exploit this fact by replaying the certificates of legitimate users and forcing sensor nodes to store them. This may result into DoS attack against the sensor nodes storage. Furthermore, the periodic broadcast of the public key by the base station to all sensor nodes increases the communication overhead.

**DP<sup>2</sup>AC** [ZZR09] describes a user authentication scheme which uses a distributed approach to query sensor network after successfully authenticating a user. In DP<sup>2</sup>AC, a user is authenticated with the help of a token. A token is a  $\lambda$ -bit random integer signed by the network owner using RSA signature. A user can purchase a token from anywhere like a mobile phone voucher. In order to access data from the sensor nodes, the user sends his data query along with his token to the sensor node in his communication range. The sensor node verifies the signed token using network owner's public key. The interesting part of this protocol is the re-usability check of a used token. For this purpose, the whole sensor network is divided into virtual horizontal and vertical lines. Every used token is stored in sensor network on all sensor nodes who are on any *one* (same) vertical line, from one end of the network to the other end. When a user sends a request with a token, after signature verification, his token is checked on all sensor nodes who are on any *one* (same) horizontal line, from one end of the network to the other end. The idea behind this scheme is that a reused token will be stored on at least one sensor node at the intersection of those vertical and horizontal lines.

*Limitations:* This approach only works well where the sensor nodes are deployed in a grid form, forming horizontal and vertical lines. Moreover, it results in increased storage overhead because each used token is stored on more than one sensor node in the network. Due to the increased storage overhead, it restricts the number of outside users of WSNs. This scheme also results in increased communication overhead because for each user request more than one sensor node are consulted to detect token re-usability.

*Comments.* None of the centralized and distributed user authentication schemes handle the session key establishment between a user and the sensor nodes.

### 3.3.2 Session Key Establishment

To establish a pair-wise key between the user and the sensor nodes, [JLX07] proposes a key establishment scheme based on the self-certified-key cryptosystem [PH97]. In this scheme, the user sends a request along with his ID to the sensor nodes in his range. In response to the user request, each sensor node computes a key using its private key and other public parameters, encrypts a nonce using the computed key and sends it to the user. The user, if he is the legitimate one, computes the same key using his own private key and other public parameters and decrypts the nonce. He then sends the decrypted nonce back to the sensor node who verifies the correct decryption. This scheme is efficient in terms of storage and communication overhead and supports a large number of users as compared to the other above mentioned user authentication schemes.

*Limitations:* This scheme only handles the key establishment between a user and the sensor nodes which implicitly provides user authentication. The sensor nodes compute a key for every valid or invalid user request. An adversary may exploit the situation and launch DoS attack by sending bogus user requests and forcing sensor nodes to perform the key computations, nonce encryption and broadcast. The result will be the wastage of sensor nodes resources. Another issue with this scheme is that it always establishes the same key between a user and a particular sensor node since there is no involvement of the ephemeral keys. Hence, if a key established between a user and a particular sensor node has been compromised once, it will enable the adversary not only to hijack all future communication between the two participants but also decrypt any previous communication between the same participants, eavesdropped by the adversary.

A PKC based hybrid key establishment protocol between a sensor node and a security manager (user in our case) is proposed by Huang et al. [HCK<sup>+</sup>03]. This protocol exploits the differences in resource capabilities between the sensor nodes and the security manager and puts the cryptographic burden on less resource constrained security manager. Like an outside user, a security manager is a powerful device (compared to a sensor node) which establishes a session key with a sensor node for the subsequent use. In this protocol, each party obtains a certificate containing its public key signed by a certification authority. In the beginning of the protocol, both parties exchange their signed certificates to obtain the public keys of each other. The contents of the certificates are verified on both sides. The protocol then proceeds by exchanging some messages and establishing a key between both parties. The

key confirmation messages at the end of the protocol assure that both parties have computed the key. The successful key computation authenticates both parties to each other by proving the fact that both parties have knowledge of the private keys corresponding to the public keys extracted from their certificates.

*Limitations:* In this protocol, the knowledge of the corresponding private keys is only proved after the complete run of the protocol on both sides via key confirmation messages. An adversary can exploit this fact and repeat this protocol with the sensor node by replaying a valid certificate. This will force sensor nodes to perform unnecessary computations and communications and hence, result into DoS attack. Before a sensor node detects the replayed certificate, it would have performed expensive computations and communications wasting its resources, particularly battery power. Later on Tian et al. [TWZ05] detected another serious security attack against this protocol. They showed in [TWZ05] that a security manager (user in our case) can easily learn the long-term private key of a sensor node after having one normal run of the protocol with the sensor node.

Kim et al. [KLP<sup>+</sup>07] propose an ID-based key establishment protocol based on pairing based cryptography which aims to reduce the communication cost of [HCK<sup>+</sup>03]. Being an ID-based protocol, it replaces the public keys of both parties with their IDs. It eliminates the need of exchanging the certificates of both parties to obtain public keys which ultimately reduces the communication cost of [HCK<sup>+</sup>03].

*Limitations:* This protocol reduces the communication cost but increases the overall computation cost of the protocol due to the expensive pairing computations. Like [HCK<sup>+</sup>03], this protocol also experiences a delayed user authentication (again by the proof of knowledge of private key) on the sensor node's side which causes a DoS attack.

An attempt to reduce the computation cost of [KLP<sup>+</sup>07] is made by Zhang et al. in [ZW09]. They propose another version of the protocol relying on the pairing based cryptography. Compared with Kim et al.'s protocol, their contribution is to scale down the number of point multiplication operations (the most expensive cryptographic operation of ECC) on a sensor node under the same communication complexity as in [KLP<sup>+</sup>07].

*Limitations:* Unfortunately, Zhang et al.'s protocol does not authenticate the security manager at all which enables any one to establish a session key with the sensor nodes.



### 3.3.3 Discussion

The centralized user authentication approaches relieve sensor nodes from verifying the user requests. However, they suffer from the problems of in-network traffic congestion, communication overhead, single point of failure and a DoS attack. The distributed approaches to user authentication, on the other hand, overcome the problems of centralized approaches but make the resource constrained sensor nodes responsible for the verification of a user's authenticity. Therefore, they require a lightweight user authentication mechanism on sensor nodes. However, the existing distributed user authentication schemes result in high processing cost ([BGK04, BGR05] and storage overhead ([ZZR09])). Moreover, none of them facilitate a session key establishment after successful user authentication. The scalability in terms of number of outside users is another issue in these schemes. The existing session key establishment schemes for WSNs, on the other hand, are either expensive in terms of computation and communication costs or they suffer from serious security problems.

## 3.4 Concluding Remarks

The existing authentication schemes for WSNs failed to handle certain problems:

- The existing broadcast authentication schemes do not handle the problem of sensor nodes broadcast authentication.
- The existing outside user authentication schemes, on the other hand, are expensive and lack session key establishment.
- The existing session key establishment schemes are also expensive and lack security feature.



## Part II



# Chapter 4

## Authentication Framework

***Chapter Overview:** Based on the literature survey of previous chapter, this chapter describes the problem definition and the motivations to find a solution. It also describes the security goals to achieve, the potential attacks to face and comes up with a threat model and a trust model. In the light of the motivations for solution, it briefly introduces the proposed authentication framework consisting of two authentication protocols, leaving the details of individual protocols for successive chapters.*

### 4.1 Introduction

This chapter presents our proposed authentication framework for WSNs using the ID-based signature schemes. The proposed framework aims to tackle the shortcomings of the existing approaches that emerged during the literature review in Chapter 3. The main shortcomings are:

- The broadcast authentication schemes for WSNs only focus on two types of authentication problems, i.e., base station broadcast authentication and

outside user authentication. No research has been done to address sensor nodes broadcast authentication whereas having this feature is essential to build many useful applications of WSNs as described in next section.

- The MAC based broadcast authentication schemes fail to provide a solution to the above mentioned problem. The digital signature based broadcast authentication schemes do not meet the needs of real-time applications and also suffer from the problem of public key and certificate management.
- Centralized user authentication schemes cause traffic congestion and DoS attacks. Moreover, the base station is a single point of failure in these schemes.
- Distributed user authentication schemes are not efficient with increased communication and storage overhead. Moreover, they do not provide session key establishment for the secure exchange of sensor nodes data after user authentication. Scalability is another problem faced by these schemes.
- Session key establishment protocols for wireless sensor networks are expensive posing considerable computation and communication burden on sensor nodes. In addition, they are not secure and suffer from serious security attacks.

The main focus of our proposed framework is the two authentication problems, i.e., sensor nodes broadcast authentication and outside user authentication. However, it can handle all three authentication problems of WSNs including base station to sensor nodes broadcast authentication.

**Contribution.** The major contribution of this chapter towards the thesis is the authentication framework proposed for wireless sensor networks.

## 4.2 Motivations

Although considerable advancements have been made to address authentication problems in sensor networks, they are inadequate. Most of the existing solutions target a specific authentication problem but ignore others. There are certain issues related to authentication in WSNs which still need to be explored, for instance, authenticated broadcast of real time data by the sensor nodes.

### 4.2.1 Sensor Nodes Broadcast Authentication

There are many critical situations where a sensor node is obliged to send a quick message. For example:

- Consider a forest fire alarm application of WSNs discussed in [Sto05]. In case a fire starts in the forest, the sensor nodes deployed there immediately inform other sensor nodes and/or authorities about the event and the exact location of the event before the fire begins to spread. This timely detection gives firefighters an advantage to arrive at the scene before the fire spreads uncontrollably.
- In a traffic application [BHUV08], whenever a sensor node senses an accident, a traffic jam or a dangerous road condition, it sends an immediate message in all directions to alert other traffic approaching this location. This prevents traffic jams to build up on the roads and helps drivers in safe driving.
- In a structural health monitoring application [SCL08], the sensor nodes are intended to monitor the condition of large structures such as bridges, subway tunnels and water pipes etc., so that structural damages or cracks may be noticed and reported immediately. This way precautionary measures can be taken before the structure weakens to the point of failure.
- Wireless sensor networks are also used in many military applications. For instance, consider the military application scenario discussed in [Sto05], where a troop of soldiers needs to move through a battlefield. The sensor nodes deployed there detect the presence of an enemy's tank, vehicle or personnel and broadcast this information immediately throughout the network. The soldiers obtain this information from their nearby sensor nodes and use it to strategically position themselves in the battlefield.

All these scenarios represent a *real-time event* and require a message to be sent as quickly as possible to report this event. However, the transmission and reception of a message consume a considerable time due to the wireless media. Moreover, in most of the cases a message propagates through several hops to reach the desired destinations. Therefore, the message sending time should be as short as possible. A delayed message may cause undesirable effects. For example, it may leave a fire to spread uncontrollably, a traffic jam to become worse and a structure to collapse.

A delayed message about the presence of an enemy in the battlefield may cause the deaths of soldiers.

In addition, the *message authentication* is equally important in all the above situations otherwise a malicious entity may exploit the situation and cause unnecessary actions and even serious damages in some cases. For example, an adversary can cause a fake fire alarm or can give the wrong location of the fire event. The former case forces the authorities to take unnecessary actions whereas the latter case misguides the firefighters and ultimately causes delay in rescue efforts. The adversary may send fake messages to block traffic towards a specific region or to turn traffic towards a specific direction in a traffic application. Similarly, an attack on the sensor network deployed for structural monitoring may cause the structures to collapse. The attacker may send fake data to make believe that there is no fault in a structure when there is one. Thus, he can delay the maintenance necessary to fix the problem, especially when the fault is created by the attacker himself. In a battlefield scenario, the malicious sensor nodes added by the enemy can disseminate wrong information about the enemy's movement, thus deceiving the soldiers. All these situations point towards the fact that the message authentication is a compulsory requirement in all these applications.

Moreover, in all the above mentioned scenarios, sensor nodes on the path from the sender node to the receiver(s) relay the messages towards destination. The wireless communication allows an adversary to inject false messages during multi-hop forwarding and causes sensor nodes to relay false data and deplete their energy [LPW06]. It implies that the sensor nodes on the path should be able to authenticate and filter out false messages as early as possible to save relaying energy [ZSW08, ZSJM07]. Therefore, they are also potential receivers of these messages, arising the need of *authenticated multicast* by the sensor nodes. In a battlefield application, all sensor nodes in the network are potential receivers of the critical information, arising the need of *authenticated broadcast* by the sensor nodes.

To summarize, there is a need of a secure mechanism which

- enables sensor nodes to broadcast a message without the involvement of the base station, unlike  $\mu$ TESLA.
- empowers all sensor nodes in the sensor network to broadcast an authenticated message efficiently in terms of resource consumption.
- enables a sensor node to send a message as quickly as possible to report a



critical event in real time.

- allows addition of new sensor nodes as broadcast senders as well as broadcast receivers.
- enables all potential receivers to verify a broadcast message sent by any other sensor node in the network.
- enables all sensor nodes on path from the sender node to the receivers to verify a message to detect injected false data earlier to save network resources.
- makes the typical sensor nodes the broadcast senders rather than other resourceful devices.

### 4.2.2 Outside User Authentication

As mentioned previously, a distributed approach to authenticate an outside user is desirable to avoid the security attacks of centralized approaches. The distributed approach puts the burden of user verification on sensor nodes. Since the sensor nodes are resource constrained devices, a lightweight user authentication mechanism is needed for them to verify the authenticity of an outside user. However, the existing distributed user authentication schemes are expensive for sensor nodes increasing processing, storage and communication overheads. The session key establishment between a sensor node and a user is another requirement to provide user access to the sensor nodes data. Nevertheless, none of the existing user authentication schemes for WSNs handle the session key establishment. Similar problems arise in existing session key establishment protocols for WSNs. They are either very expensive for sensor nodes or prone to security attacks.

To summarize, there is a need of a lightweight user authentication mechanism which

- enables all sensor nodes in the network to authenticate any user locally without the involvement of a third party efficiently in terms of processing, storage and communication overheads.
- provides scalability in terms of the number of outside users.
- allows addition of new sensor nodes (as user verifiers) as well as new users to the system.

- establishes a session key between a user and a sensor node after successful user authentication.
- makes the typical sensor nodes the user verifiers rather than other resourceful devices.

### 4.3 Security Goals

The primary security goals of authentication framework are to satisfy the security properties of *authentication*, *integrity*, *verification*, *freshness*, *confidentiality* and *availability*. The security goals discussed in Section 2.1.2 are restated here specifically with reference to broadcast authentication and user authentication in WSNs. The security goals for user authentication overlap with those of broadcast authentication with the addition of *confidentiality* property.

**Authentication** enables the broadcast sender nodes and the outside users to prove their identities to other sensor nodes in the network. Authentication is required to prove that a broadcast message or a user authentication request message received by a sensor node is actually sent by a legitimate broadcast sender or an outside user respectively. In addition, a session key is established with a legitimate user. Authentication distinguishes the legitimate broadcast senders and outside users from intruders.

**Message Integrity** guarantees that the contents of a received broadcast message or user authentication request message have not been modified en-route and any modified message can be detected. Lack of message integrity can result in serious issues, especially in authenticated broadcast, since the consequences of using false or altered information could be disastrous.

**Verification** empowers a sensor node to attest the legitimacy of any broadcast sender node or any outside user. Verification property implies the ability of the sensor nodes to perform necessary tests to verify the authenticity, for instance, sensor node's access to sender's authentication information like public key and capability to perform necessary computation to verify authentication information.

**Freshness** ensures that a broadcast message or a user request message received by a sensor node is fresh and not the replay of an old message from a legitimate sender by the adversary to take some deceitful advantage or to waste the resources of the sensor nodes.

**Confidentiality** prevents unauthorized users or intruders from accessing the sensor nodes data being sent to authorized users after successful user authentication. Since an adversary may eavesdrop on the in-transit data, this data should be resistant to disclose its meaning.

**Availability** ensures that the services of authenticated broadcast and user authentication are available even in the presence of a DoS attack.

## 4.4 Security Attacks

We now consider some potential security attacks against authentication in WSNs which are to be handled by the proposed framework.

1. *Impersonation Attack.* In this attack, an adversary impersonates a legitimate sensor node or a user. He uses the identity of a legitimate sensor node to broadcast messages on its behalf and the identity of a user to login to the system on user's behalf. The motivations behind this attack are to deceive broadcast receivers by giving them false information on behalf of the targeted legitimate sensor node and to access sensor nodes data for which the adversary is not a legitimate user.
2. *False Data Injection Attack.* In this attack, an adversary injects random false data in the sensor network. There are two motivations behind this attack. The first one is to give some wrong information about an event. For instance, reporting a fire in a forest whereas there is no fire or giving the wrong location of the fire whereas the fire has set up at a different location. The first case causes a fake fire alarm whereas the second case wastes the time of the rescue team and makes the fire uncontrollable. The second motivation behind the false data injection is to cause sensor nodes relaying false data and depleting their battery power.
3. *DoS Attack.* The DoS attacks disrupt the functionality of the sensor network in one or another way. For instance, the above mentioned false data injection attack can lead to a DoS attack where the relaying nodes deplete their battery power and become nonfunctional. This results in the disruption of network functionality.  $\mu$ TESLA based broadcast authentication schemes suffer from the DoS attack against the sensor node's storage due to the delayed authentication. Moreover, successfully targeting only the base station

collapses the whole protocol of  $\mu$ TESLA. The centralized user authentication schemes suffer from the DoS attack against the sensor network resources where an adversary sends fake user requests forcing sensor nodes to forward them towards the base station. It does not only result into network traffic congestion (wasting bandwidth) but also wastes the resources of the relaying sensor nodes. As a result, the relaying sensor nodes deplete their battery power and fail to function. Targeting the sensor nodes closer to the base station, makes the base station disconnected from the sensor network.

4. *Message Replay Attack.* In this attack, the adversary captures the previous legitimate messages exchanged between nodes and between users and nodes, and replays them later. The motivations behind this attack are to cause confusion, impersonate a legitimate user or make sensor nodes to waste their resources in unnecessary processing.
5. *Node Compromise Attacks.* A compromised node does not only reveal the cryptographic material stored on it but can also be used to launch any of the above mentioned attacks. The compromised nodes attacks are very successful in symmetric MAC based authentication schemes. In such schemes, a single compromised node revealing a MAC key enables an intruder to impersonate all sensor nodes sharing the same MAC key for broadcast authentication. Multiple compromised nodes can also collude and launch any of the above mentioned attacks to achieve devastating results.

## 4.5 Threat Model

A threat model for the proposed authentication framework describes the sensor network assets which are to be protected, adversary's goals and his capabilities to launch attacks against WSNs assets.

### 4.5.1 Assets to Protect

The valuable assets of a WSN that must be protected are:

- Confidential sensor nodes data
- Sensor nodes resources, such as battery power, processing and storage
- System availability (in presence of a DoS attack)

### 4.5.2 Adversary's Goals

The adversary's goals are to:

- impersonate a legitimate broadcast sender to send messages on its behalf and a user to access sensor nodes data,
- modify the contents of the broadcast messages and the user request messages,
- send bogus messages to waste the resources of the sensor nodes,
- replay old broadcast messages and user requests to fool the sensor nodes,
- obtain the session key established between a user and the sensor node.

### 4.5.3 Adversary's Capabilities

To achieve his goals, the adversary may use an ordinary sensor node or a resourceful device like a laptop. The adversary can eavesdrop on all communication because of the insecure radio link. The wireless nature of communication helps adversary in interrupting communication. Moreover, he may compromise a few sensor nodes in the network. The adversary does not compromise the majority of or all the sensor nodes in the network since this reveals his presence. Moreover, if he compromises majority of the sensor nodes, it breaks down all the security mechanisms. He is able to extract all cryptographic material stored on the compromised nodes including their private keys and the session keys established with users. He may use these compromised nodes to attack the security of the sensor network. In addition, the adversary may add a few sensor nodes of his own in the network. He may launch jamming attacks on the data link layer and the physical layer. However, we do not take into account these attacks as almost all authentication schemes are vulnerable to such attacks.

## 4.6 Assumptions

The proposed authentication framework makes the following assumptions:

- The wireless sensor network is a large scale network consisting of several thousands of sensor nodes.
- The sensor nodes are stationary and not mobile sensor nodes.

- The sensor nodes are densely deployed in the area of interest.
- The topology of the sensor network is not known prior to the deployment of the network.
- The network topology is prone to frequent changes due to the addition of new nodes, revocation of malicious nodes, depleted batteries and nodes failure.
- The wireless sensor network employs a broadcast communication paradigm rather than a point-to-point communication paradigm.
- The medium of communication is a radio link.
- The sensor nodes are deployed in a hostile environment where they are directly accessible by any one.
- All the sensor nodes in the sensor network are similar and equally resource constrained in terms of battery power, memory, computation, bandwidth and the transmission range.
- To keep the cost low, the sensor nodes are not equipped with the tamper resistant devices (e.g., Trusted Platform Modules (TPM) technology).
- The base station is comparatively a resourceful device, e.g. a laptop, with large battery power, memory, computation and bandwidth, which connects the sensor nodes to the outer networks.
- The outside users of the sensor nodes data are equipped with the resourceful devices, for instance PDA, notebook or mobile phone, to query sensor nodes.

## 4.7 Trust Model

- Since the sensor nodes are usually placed unattended in open places and they are not equipped with TPM chips, any adversary can compromise any sensor node in the network and obtain the cryptographic material stored on it. The adversary can further use the compromised sensor node for his malicious intentions to launch attacks against the sensor network. Therefore, no trust requirements are placed on the sensor nodes.
- The base station, on the other hand, is a powerful device which can protect itself and is considered as the trustworthy entity in the wireless sensor network.

## 4.8 Proposed Authentication Framework

The proposed authentication framework is a detailed solution which covers all three previously mentioned broadcast authentication problems in WSNs counteracting the security attacks against sensor networks. The proposed authentication framework utilizes ID-based cryptography and online/offline signature (OOS) schemes and is comprised of two authentication schemes; one for sensor nodes broadcast authentication and second for user authentication. The user authentication scheme can also handle the base station to sensor nodes broadcast authentication. It makes the proposed authentication framework a single solution to all three authentication problems in WSNs.

- The name of the first scheme is ***authenticated broadcast by sensor nodes***. This scheme aims to achieve two primary goals:
  1. It enables each sensor node in the sensor network to *broadcast or multicast authenticated messages as soon as possible* without the involvement of the base station. In other words, it enables every sensor node in the sensor network to become a broadcast sender.
  2. It also enables *every potential receiver to verify a signed message* sent by any other broadcast sender node in the network. It also allows sensor nodes on the path from the sender node to the receivers to verify a valid message and drop the false injected data.

Authenticated broadcast by sensor nodes scheme together with its security and performance evaluation is discussed in detail in Chapter 5.

- The name of the second scheme is ***outside user authentication***. This scheme also achieves two primary goals:
  1. It enables each sensor node in the sensor network to *verify the legitimacy of any outside user* without storing the user specific information.
  2. It also enables a sensor node to *establish a session key* with the user after the successful user authentication to securely exchange the confidential sensor nodes data.

Outside user authentication scheme and its security and performance evaluations are discussed in Chapter 6 together with the details of its use for the base station broadcast authentication.

The proposed authentication framework uses digital signatures to grant authentication in both schemes while maintaining the efficiency requirement of the resource constrained sensor nodes. It uses ID-based Online/Offline Signature (IBOOS) scheme for the first authentication scheme and ID-based Signature (IBS) scheme for the second authentication scheme. ID-based cryptography replaces a public key with the ID, and thus eliminates the need of a signed certificate to extract the public key. An online/offline signature scheme performs most of the computations of signature generation before the message to be signed is known in *offline* phase. The *online* phase performs only minor quick computations to obtain the final signature when there is a message to send. The *online* phase is assumed to be very efficient while the *offline* phase can be performed by other resourceful device, for instance, the base station in case of WSNs. An IBOOS scheme thus enables a resource constrained sensor node to sign and broadcast a message immediately, once it has some critical event to report. Authenticity and efficient signature generation are the two main features of IBOOS schemes.

Figure 4.1 illustrates the proposed authentication framework which achieves the aimed security goals, described in Section 4.3. Due to the use of ID-based cryptography and secure digital signature schemes (IBS and IBOOS), *verification*

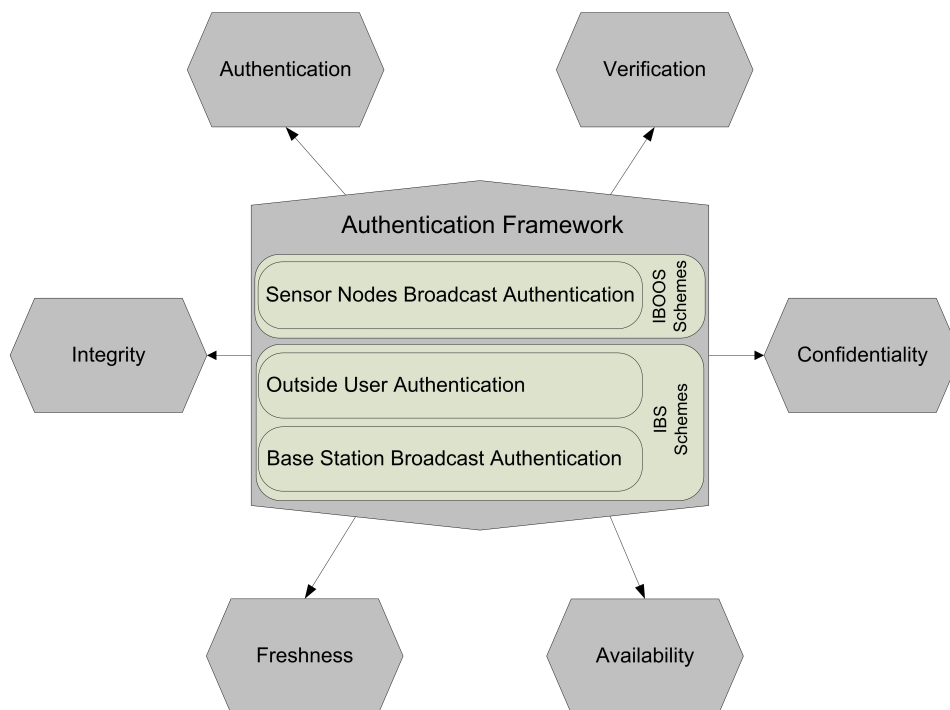


Figure 4.1: Authentication Framework for Wireless Sensor Networks



and *integrity* are ensured on top of *authentication*. The session key establishment in user authentication protocol meets the necessary *confidentiality* requirement. *Freshness* is achieved via time stamps and *availability* is ensured by adopting distributed approaches of authentication in both schemes. A detail discussion about how these security properties are satisfied in both types of authentication problems is given in subsequent chapters with individual authentication schemes.

By ensuring these security properties, the proposed framework safeguards the WSNs from the potential attacks (given in Section 4.4) against it saving its assets. Figure 4.2 illustrates the attacks against WSNs and how they are defeated by the

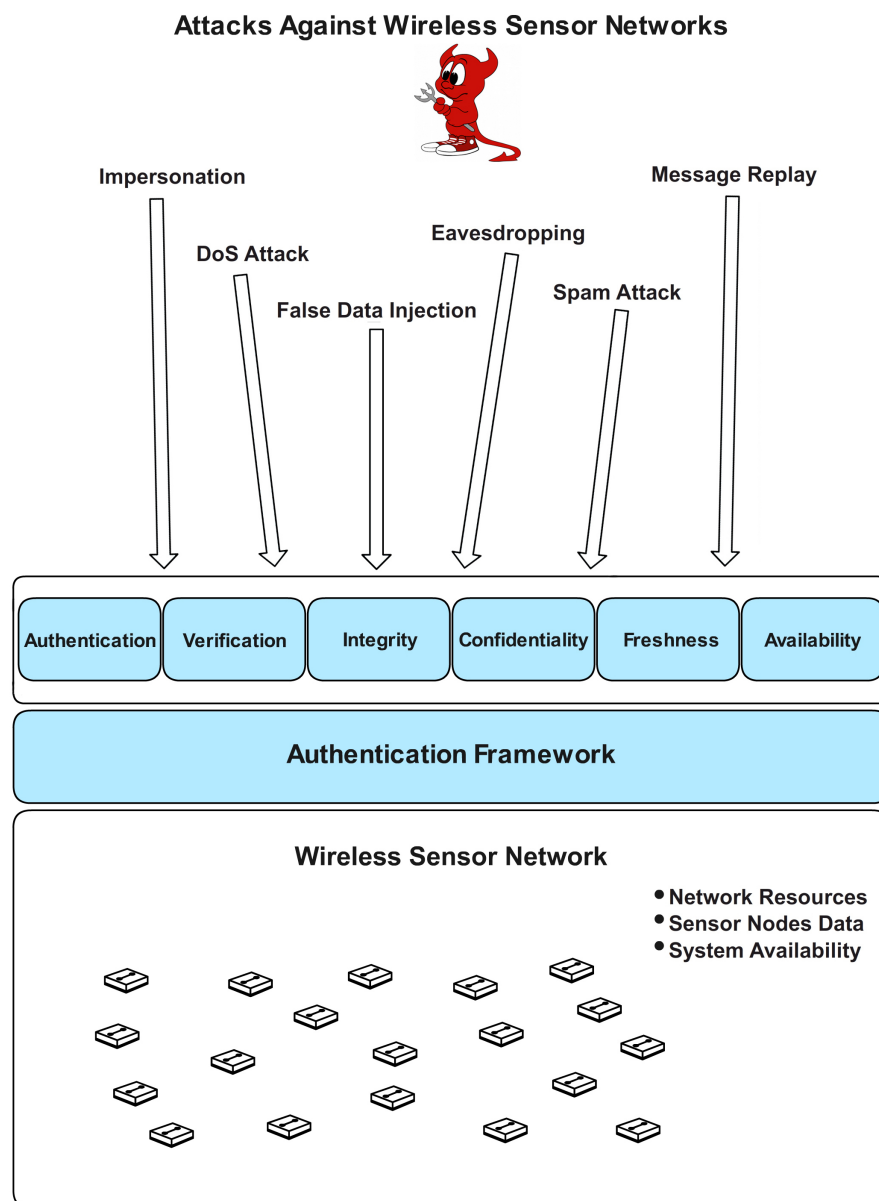


Figure 4.2: Authentication Framework: A Countermeasure to Security Attacks

proposed framework. The attacker in these attacks could be a compromised node or an adversary himself. The proposed framework introduces a logical layer between a WSN and the attacker which shields the WSN from the attacks. Again, the detail about how these attacks are defeated by our proposed framework is given in coming chapters. Besides these attacks, the proposed framework can also help in defeating other attacks against WSNs such as attacks against secure data aggregation, e.g. *False Data Injection Attack* and attacks against routing protocols, e.g. *Hello Flood Attack* etc., by providing required authentication.

The primary objective of the proposed framework is to design an authentication mechanism which solves the above mentioned authentication problems efficiently in terms of power consumption, processing time and storage overhead on sensor nodes. The primary advantage of this framework is its re-usability which means it does not restrict the solution to existing IBS and IBOOS schemes, rather it can be reused with any IBS and IBOOS schemes. Once new IBS and IBOOS schemes are available, which are more secure and efficient than the existing IBS and IBOOS schemes, they can replace the existing ones to achieve better security and performance results. Security and efficiency are the two central design features of the proposed authentication framework.

## 4.9 Framework Instantiation and Evaluation

To instantiate the proposed authentication framework, the most efficient IBS and IBOOS schemes were selected while keeping the security with maximum efficiency objective of this research in mind. There are many IBS and IBOOS schemes available, for instance, based on RSA signatures. The verification of RSA signature is efficient for sensor nodes since one can set small verification exponents. This fact can be utilized in a user authentication scheme, where the sensor nodes only verify a signed user request. However, RSA based signatures result in lengthy messages due to the large signature sizes. The ECC based signatures, on the other hand, are equally efficient for signing and verification of messages and also enjoy the short signature sizes. For this reason, the ECC based signatures are considered more efficient for WSNs than RSA signatures [GPW<sup>+</sup>04]. Therefore, the proposed authentication framework was instantiated using the efficient ECC based IBS and IBOOS schemes.

Initially, the authentication framework was theoretically evaluated at an early stage of the performance evaluation using Cao et al.'s IBS scheme [CKDZ08], Ren et

al.'s IBOOS scheme [RMS08], and Xu et al.'s IBOOS scheme [XMS05]. The details of these signature schemes will be discussed later on with the individual authentication schemes. The two selected IBOOS schemes represent the two different classes of ID-based Online/Offline Signature schemes, discussed in Section 2.2.4. The actual implementation of these schemes on the real sensor nodes was done at a later stage. To the best of our knowledge, these IBS and IBOOS schemes were the most secure and efficient schemes of that time for the resource constrained sensor nodes among the available IBS and IBOOS schemes. However, later on some changes were made based on implementation results which will be discussed in detail in the chapters of individual authentication schemes.

## 4.10 Concluding Remarks

In order to address the shortcomings of existing authentication schemes for WSNs, an authentication framework has been proposed. The proposed authentication framework provides a detailed solution to address the shortcomings of the existing schemes. Other than sensor nodes broadcast authentication, outside user authentication and session key establishment, it can also deal with the base station to sensor nodes broadcast authentication. It therefore provides a single solution to all authentication problems in WSNs. Security, efficiency and re-usability are the main features of the proposed authentication framework.



## Chapter 5

# Authenticated Broadcast by Sensor Nodes Protocol

***Chapter Overview:** This chapter presents the proposed authenticated broadcast by sensor nodes protocol using the ID-based online/offline signature schemes together with its performance and security evaluations. The first half of this chapter highlights the challenges faced in the design of a broadcast authentication protocol and discusses the available ID-based online/offline signature schemes. It also describes our adapted ID-based online/offline signature scheme. It then presents the proposed authenticated broadcast by sensor nodes protocol in detail. The second half of this chapter evaluates the performance as well as security of the proposed protocol. At the end of this chapter, the proposed protocol has been compared with the existing protocols for authenticated broadcast in WSNs.*

## 5.1 Introduction

The realization of many WSN applications (for instance, forest fire alarm application, enemy tracking application etc.) depends on the existence of a secure and efficient protocol for broadcast authentication. On the other hand, designing a secure and efficient broadcast authentication protocol for WSNs is a challenging task due to the resource constraints and nature of deployment of the network. Some major challenges faced in designing a secure as well as efficient broadcast authentication protocol for WSNs, adapted from [LPW06, Per01], are:

- *Efficient generation and verification.* Since the sensor nodes are resource constrained devices with limited computation and storage capabilities, the generation and verification overheads of the authentication information should be small.
- *Low communication.* The battery power is the most scarce resource on sensor nodes and communication consumes most of it. Thus, a protocol with low communication overhead is highly desirable for low power sensor nodes.
- *Instant/Individual message authentication.* Some applications send messages at irregular and unpredictable times and require instant message authentication, for instance, the fire alarm application. For such applications, the authentication protocol should enable a receiver to verify a broadcast message individually and instantly once the message has been received.
- *Scalability.* The broadcast applications have a potentially large number of receivers and, in some applications, a large number of senders as well. The protocol should be independent of the number of broadcast senders and receivers. It should also be dynamic allowing the addition of new broadcast senders and receivers.
- *Robustness to packet loss.* Due to radio communications, WSN applications face a high level of packet loss. The lost packets are not retransmitted in many broadcast applications. Hence, the broadcast authentication protocol should be tolerant to packet loss.

Unfortunately, all the existing MAC based efficient schemes for broadcast authentication in WSNs, discussed earlier in Chapter 3, cannot support instant/individual message authentication, scalability and robustness to packet loss.

On the other hand, the digital signature based broadcast authentication schemes provide instant/individual message authentication and robustness to packet loss. However, signing a message consumes more time and battery power than a MAC computation. The public key and certificate management in WSNs is another issue causing the scalability problem. For a large sensor network, it is not possible for a sensor node to store public keys of all other sensor nodes in the network, restricting the number of broadcast senders. Moreover, all these previously proposed schemes assume powerful devices as broadcast senders rather than the resource constrained sensor nodes.

This chapter presents the proposed *authenticated broadcast by sensor nodes* protocol for WSNs using the ID-based Online/Offline Signature (IBOOS) schemes. The IBOOS schemes enable a sensor node to quickly sign and broadcast a message as soon as it has some time critical event to report since the computation of online signature of an IBOOS scheme is very fast. The IBOOS schemes allow the offline phase to be performed by some other resourceful device. Hence, it is possible for the base station to perform the complex computations of the offline phase and distribute the partial offline signature to the sensor nodes. The sensor nodes then only perform the small, energy efficient computations of the online phase. In addition, some IBOOS schemes, like [RMS08], facilitate a signer to reuse the partial signature computed in the offline phase to sign more than one message. This feature of an IBOOS scheme can further reduce the computation burden on the sensor nodes. The ID-based public key cryptosystem does not require the public keys and certificates and, as a result, solves the scalability problem of the digital signatures.

Due to the resource constrained nature of sensor nodes, the security and the efficiency are the two most important aspects of a security protocol to decide the suitability of that protocol for WSNs. This chapter also presents the *performance* evaluation and the *security* analysis of the proposed protocol. Since our proposed protocol is the first proposal to use IBOOS schemes in WSNs, a cryptographic primitive previously untested on the sensor nodes devices, we have implemented a few IBOOS schemes on actual sensor nodes as part of this research work. Although the theoretical evaluation before the actual implementation strengthened the idea of applying IBOOS schemes to sensor nodes devices, the actual implementation confirmed the suitability of the idea. The implementation results helped to evaluate the performance of the proposed protocol. Besides performance, the security of the proposed authenticated broadcast by sensor nodes protocol using IBOOS schemes together with our adapted IBOOS scheme is assessed.

**Contribution.** The major contribution made by this chapter is addressing the problem of broadcast by sensor nodes authentication in WSNs for the first time and proposing to use the online/offline signature schemes in WSNs for the first time. Another major contribution is the practical implementation of several online/offline signature schemes on real sensor nodes devices for the first time and the transformation of an IBS scheme to an efficient and secure IBOOS scheme.

## 5.2 Options for IBOOS Schemes

We now discuss the IBOOS schemes options for our proposed authenticated broadcast by sensor nodes protocol.

### 5.2.1 Available IBOOS Schemes

There are many IBOOS schemes available, for example, based on ECC or RSA. Since ECC based signature schemes are more efficient to process for sensor nodes than RSA ones [GPW<sup>+</sup>04], we only consider ECC based signature schemes in this work. We selected two ECC based IBOOS schemes [XMS05] and [RMS08] to evaluate our proposed protocol. This selection was made keeping the security and efficiency requirements for WSNs in mind. We name the first scheme [XMS05], proposed by Xu et al., as X-IBOOS scheme and the second scheme [RMS08], proposed by Ren et al., as R-IBOOS scheme for convenience. These schemes represent two different categories of direct (X-IBOOS scheme) and indirect (R-IBOOS scheme) online/offline signature schemes, mentioned in Section 2.2.4. Both X-IBOOS and R-IBOOS schemes have been proved to be existentially unforgeable under the adaptive chosen message attacks in [XMS05] and [RMS08] respectively. The offline signature in R-IBOOS scheme can be securely reused to sign more than one message. We roughly estimated the cost of both schemes on sensor nodes before implementation. To see how efficient these IBOOS schemes would be on sensor nodes, we chose them to implement on actual sensor nodes. However, we only implemented and evaluated the X-IBOOS scheme and based on the expensive implementation results of this scheme, we decided to skip the implementation of the R-IBOOS scheme. Instead, we implemented and evaluated the adapted B-IBOOS scheme given in next section. The implementation results are discussed in detail in later sections.



### 5.2.2 Adapted IBOOS Scheme

To achieve our efficiency aim, we also adapted an ID-based signature scheme to an ID-based online/offline signature scheme and evaluated it. The reason behind this adaption is the expensive results of X-IBOOS scheme discussed later in Section 5.4.1.5. To obtain a more efficient IBOOS scheme than the X-IBOOS and R-IBOOS schemes, we noticed that the ID-based signature scheme (BNN-IBS) proposed by Bellare et al. [BNN04] and improved by Cao et al. [CKDZ08] could be securely transformed to an IBOOS scheme. The BNN-IBS scheme and its improved versions are given in Section 2.2.6.1. The BNN-IBS is an ECC based pairing-free ID-based signature scheme having only one point multiplication as an expensive operation in the signature generation process. This point multiplication computation results in a partial signature and is independent of the message to be signed. This can be computed as an offline signature before the message to be signed is known. Thus, this point multiplication operation forms the offline phase of the online/offline version of BNN-IBS that we propose. The rest of the signature generation process uses this offline signature and the message and only performs integer arithmetics to get the final signature of the message. Integer arithmetics is very efficient for sensor nodes in terms of time and power consumption. Integer arithmetics operations performed when the message to be signed is known form the online phase of the adapted online/offline version of BNN-IBS scheme. We named the adapted IBOOS scheme as B-IBOOS scheme after the name of BNN-IBS scheme which was proposed by Bellare et al. This transformation can be applied to both BNN-IBS scheme and Cao's variant of BNN-IBS scheme since both have the same *Setup*, *Key Extract* and *Sign* algorithms and only the *Verify* algorithm is different.

#### 5.2.2.1 B-IBOOS Scheme

Like the X-IBOOS scheme, the B-IBOOS scheme is a direct online/offline signature scheme. The B-IBOOS scheme has five algorithms instead four algorithms of BNN-IBS scheme. The additional algorithm in B-IBOOS scheme is introduced due the fact that the message is signed in two phases in B-IBOOS scheme unlike BNN-IBS scheme. The five algorithms of the B-IBOOS scheme are *Setup*, *Key Extract*, *OffSign*, *OnSign*, and *Verify*. The *Setup*, *Key Extract* and *Verify* algorithms of the B-IBOOS scheme are the same as in Cao's variant of BNN-IBS scheme whereas the *Sign* algorithm of the BNN-IBS version is split into *OffSign* and *OnSign* algorithms in our adapted B-IBOOS scheme.

**Setup.** This algorithm sets up the system parameters which are  $(\mathbb{E}/\mathbb{F}_p, \mathbb{G}, P, q, p, P_0, H_1, H_2)$ . The *Setup* algorithm performs the following steps:

- Specify the parameters  $\mathbb{E}/\mathbb{F}_p, q, p, P$  and  $\mathbb{G}$ , where
  - $\mathbb{E}/\mathbb{F}_p$  is an elliptic curve  $\mathbb{E}$  over a finite field  $\mathbb{F}_p$ ,
  - $q$  is a large prime number and  $p$  is the field size,
  - $P$  is a point of order  $q$  on the curve  $\mathbb{E}$  and,
  - $\mathbb{G}$  is a cyclic group of order  $q$  under the point addition “+” generated by  $P$ .
- Chose a master secret key  $s \in_R \mathbb{Z}_q^*$  uniformly.
- Compute the master public key as  $P_0 = sP$ .
- Choose one cryptographic hash function  $H_1 = \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$ .
- Choose another cryptographic hash function  $H_2 = \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$ .
- Output the system parameters  $\{\mathbb{E}/\mathbb{F}_p, \mathbb{G}, P, q, p, P_0, H_1, H_2\}$  and keep  $s$  secret.

**Key Extract.** Given an identity  $ID_i$  of a user  $I$ , this algorithm computes the corresponding private key as follows:

- Choose a random  $r_i \in_R \mathbb{Z}_q^*$  and compute
- $R_i = r_i P$
- $c_i = H_1(ID_i, R_i)$
- $s_i = r_i + c_i s$

Here the private key  $s_i$  is the Schnorr signature of the identity  $ID_i$  of  $I$  signed with the master secret key of the *PKG*. The user  $I$  obtains  $(R_i, s_i)$  via a secure channel. Here  $s_i$  is secret information whereas  $R_i$  is public.

**OffSign.** The offline phase is performed before the message to be signed is known. The *OffSign* algorithm proceeds as follows:

- Choose  $y \in_R \mathbb{Z}_q^*$

- Compute  $Y = yP$

The offline signature is  $(y, Y)$ .

**OnSign.** The online phase is performed after the message becomes available. The *OnSign* algorithm computes

- $h = H_2(ID_i, m, R_i, Y)$
- $z = y + hs_i$

The tuple  $\langle R_i, h, z \rangle$  is  $I$ 's signature on message  $m$ .

**Verify.** Given the signature tuple  $\langle R_i, h, z \rangle$ ,  $I$ 's identity  $ID_i$  and the message  $m$ , the *Verify* algorithm verifies the signature as follows:

- Compute  $c_i = H_1(ID_i, R_i)$
- Check whether the following equation holds

$$h = H_2(ID_i, m, R_i, zP - h(R_i + c_iP_0))$$

The signature is accepted if the answer is *yes* and rejected otherwise.

### 5.3 Proposed Authenticated Broadcast by Sensor Nodes Protocol

In the proposed broadcast authentication protocol, the sensor nodes sign a broadcast message in two phases. In first phase, the offline signature is computed before any time critical information to report is available. The offline phase can be performed by the base station or by the sensor node itself depending on the nature of the application. The offline signature is stored on the sensor node. In the second phase, when a critical event happens, the sensor node uses the offline signature to compute the final signature of the message and broadcasts the message. The proposed scheme for broadcast authentication using an IBOOS scheme consists of four phases, *System Initialization*, *Key Generation*, *Message Broadcast and Authentication*, and *Sender Revocation*. The first two phases are performed only once before the deployment of the sensor network. As mentioned earlier, in ID-based signature schemes a private key generator (PKG), which is a trustworthy entity, initializes the system

and computes the private keys corresponding to the users' IDs. In the proposed protocol, the base station, which is a trustworthy and a resourceful device, plays the role of PKG to initialize the system and compute the private keys. The details of the protocol are explained as follows:

**System Initialization:** In this phase, the Setup algorithm of an IBOOS scheme runs on the base station and computes the public system parameters and the master secret key. Let  $SK_{BS}$  be the secret key of the base station, which will be called the master secret key. The base station calculates the corresponding public key  $PK_{BS}$ , which will be called the master public key. The master secret key  $SK_{BS}$  is only kept by the base station while the master public key  $PK_{BS}$  is made public. The base station also sets up public system parameters ( $SP$ ) in this phase which include  $PK_{BS}$ .

**Key Generation:** In this phase, the base station calculates the private keys of all sensor nodes corresponding to their IDs using the master secret key  $SK_{BS}$  and other system parameters. For a sensor node  $I$  with identity  $ID_i$ , the private key  $D_{ID_i}$  is computed using the Key Extract algorithm as

$$D_{ID_i} \leftarrow KE(ID_i, SK_{BS})$$

The ID, private key, system parameters and other related information (if any) are stored on individual sensor nodes before the deployment of sensor network. Hence, every sensor node  $I$  stores  $\{ID_i, D_{ID_i}, SP\}$ .

**Message Broadcast and Authentication:** The process of a signature generation for a broadcast message is divided in two phases: the offline phase and the online phase.

*Offline phase:* The offline phase is preformed before the message to broadcast is available. The offline signature generation algorithm (OffSign) runs in this phase and performs most of the signature computations in order to compute the partial signature  $S$  as

$$S \leftarrow OffSign(D_{ID_i}, SP)$$

The resulting partial signature  $S$  is stored on sensor node  $I$ .

*Online phase:* As soon as a sensor node  $I$  senses an event which requires quick reporting, the online phase starts. In this phase, the sensor node  $I$  retrieves the

partial signature  $S$  calculated in the offline phase and performs very minor and fast computations to get final signature  $\sigma$  over message  $m$  as

$$\sigma \leftarrow OnSign(\langle m, TS \rangle, S)$$

Here  $TS$  is the current time stamp. The final broadcast message then contains the message  $m$ , the time stamp  $TS$ , identity of the sensor node  $ID_i$  and the signature  $\sigma$ , i.e.,

$$\{m, TS, ID_i, \sigma\}.$$

To sign a broadcast message, our adapted ID-based online/offline signature scheme i.e., the B-IBOOS scheme, or any other efficient as well as secure IBOOS scheme can be used here.

*Authentication:* On receiving a broadcast message, the receiver node first checks the time stamp  $TS$  to avoid the verification of a replayed message. If it is a fresh one, the receiver node further proceeds with signature verification, else it discards the message. The receiver node verifies the signature  $\sigma$  using the sender node's identity  $ID_i$  and other system parameters as

$$\text{yes/no} \leftarrow Verify(\langle m, TS \rangle, ID_i, \sigma, SP)$$

If the verification holds, the receiver node accepts the message otherwise discards it. If necessary it rebroadcasts (relays) the legitimate message to all sensor nodes belonging to the next hop.

**Sender Revocation:** To revoke a compromised sensor node  $I$ , the base station broadcasts its identity  $ID_i$  to all other sensor nodes in the network, who store  $ID_i$ . If in the future a sensor node receives a message containing  $ID_i$ , it simply rejects the message without going through authentication process. An adversary is assumed to compromise only a few sensor nodes in the network. Storing the  $ID$ s of a few compromised nodes incurs a reasonable storage overhead for sensor nodes. Moreover, the base station can periodically update system parameters and secret keys of all legitimate sensor nodes excluding the malicious nodes. However, this update might be costly. Another possible solution is to manually detach these compromised sensor nodes from the sensor network.

### 5.3.1 Is an Online/Offline Signature Scheme Secure for Wireless Sensor Networks?

An interesting and important question to ask here is whether it is secure to use an online/offline signature scheme in WSNs in the presence of the node compromise attack? An online/offline signature scheme signs a message in two phases. The partial offline signature is computed before the message is known and is stored on the sensor node. Once the message is known, the sensor node uses this offline signature to compute the final signature on the message. What if an adversary compromises the sensor node before the online phase starts and obtains the offline signature stored on it? Will it give an extra advantage to an adversary in comparison to the situation where an ordinary digital signature is used? The answer to this question is ‘NO’. The reason is that in both cases, once the adversary compromises a sensor node, he has a full control over the compromised sensor node and all cryptographic material stored on it including the private key of the node. With the private key of the compromised sensor node, the adversary can sign the messages on behalf of that node in both cases. Then, the presence of an extra phase in message signing process of an online/offline signature does not give any extra benefit to the adversary. Hence, using an online/offline signature scheme in WSNs is as secure as using any other digital signature scheme.

## 5.4 Performance Evaluation

The performance of the proposed protocol is evaluated in two steps. In first step, we discuss the efficiency of several IBOOS schemes on sensor nodes. In second step, we analyze the efficiency of the proposed authenticated broadcast by sensor nodes protocol using IBOOS schemes.

### 5.4.1 Performance of the IBOOS Schemes

The goal of implementing the IBOOS schemes on sensor nodes was to validate the idea of applying IBOOS to WSNs experimentally. Specifically, we aimed to find the answers to the following questions:

1. Is it possible for a typical resource constrained sensor node processor to perform IBOOS operations?

2. How efficient it is to compute an IBOOS scheme on sensor nodes in terms of resource consumption with respect to the computation, communication and memory costs?

#### 5.4.1.1 Hardware and Software Used

**MICA2.** For implementation purposes, the hardware platform selected was the standard MICA2 [MIC] sensor node. MICA2 has an integrated ATMEGA 128L micro-controller from the AVR family having 8-bit processor, 4KB of SRAM, 128KB of flash memory (ROM) with a clock speed of 7.3828MHz. MICA2 radio operates on 868/916 MHz ISM band. 868 MHz is a license-free frequency band for Europe. MICA2 is a popular choice among research community. Several research groups [GPW<sup>+</sup>04, PLP06, ADLO10, OAG<sup>+</sup>11] all over the world have used MICA2 nodes for the evaluation of security protocols and cryptographic operations. This fact helped us to compare our experimental results with others. Figure 5.1 shows a MICA2 (MPR4x0) node without an antenna.

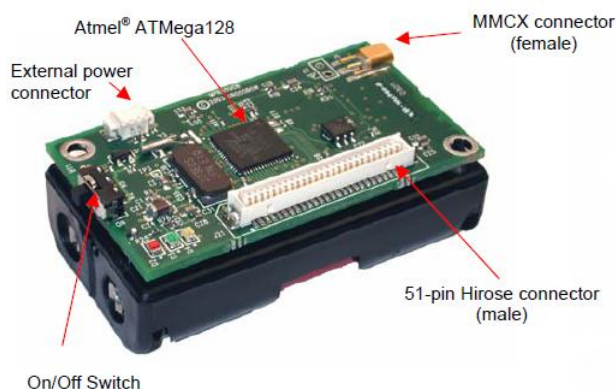


Figure 5.1: MICA2 node without an antenna [Cro]

**TinyOS.** The MICA2 motes use a special operating system called TinyOS [Tin]. TinyOS is an open source operating system designed for wireless embedded sensor networks, released under the BSD license. Its component-based architecture enables rapid innovation and implementation while minimizing the code size to meet the severe memory constraints inherent in sensor networks. The component library of TinyOS includes network protocols, distributed services, sensor drivers, and data acquisition tools all of which can be used as they are or be further refined for a custom application. Since sensor nodes are application specific, only one application runs on a sensor node. Therefore, TinyOS does not support memory management

or process management. TinyOS was originally developed as a research project at the University of California Berkeley, but has since grown to have an international community of developers and users.

**NesC.** The TinyOS operating system has been implemented in a language called nesC [Nes]. This language is an extension of C language. It has been designed to embody the structuring concepts and execution model of TinyOS. For further details of TinyOS and nesC, see tutorials online available at [Tin] and [Nes], respectively.

The further details about how hardware and software are setup to build a test sensor network and how sensor nodes are programmed are given in Appendix A.

#### 5.4.1.2 Performance Metrics

The primary goal of these experiments was to gather the actual statistics about the resource consumption of an IBOOS scheme on real sensor nodes and study its performance. In the case of a signature scheme, the primary factors that affect a sensor node's resources are the signature generation and verification costs (computation and memory costs) and signature size (transmission cost). Therefore, the IBOOS schemes are evaluated for the following performance metrics: computation cost (time and battery power consumption), memory consumption (ROM/RAM usage), and signature size. The transmission cost is proportional to the signature size, thus we only count the signature size.

#### 5.4.1.3 Experimental Details

The implementation involved a base station (a laptop with TinyOS installed on it) and two MICA2 sensor nodes; one acting as a signer while the other acting as a verifier. However, both nodes had the ability to sign as well as verify the messages. To perform the cryptographic operations, we used RELIC [AG]. RELIC is a publicly available highly efficient library to implement cryptographic operations on sensor nodes particularly the most efficient implementation of pairing computation operation. Since both IBOOS schemes chosen for experiments required pairing computations, we decided to use this library. The security level of  $\sim 80$ -bit (RSA-1024 equivalent), considered adequate for resource constrained sensor nodes by NIST, was adopted.  $\eta_T$  pairing [BGHS07] was chosen to compute pairing operation.  $\eta_T$  pairing is the fastest one to compute on resource constrained sensor nodes and a best choice at this security level [OAG<sup>+</sup>11]. The *Setup* and the *Key Extract* phases of ID-based settings were performed at the base station. The system parameters and



other relevant information for the sensor nodes were stored on them via the base station. All the software programs (including the codes for online/offline signature schemes) running on the sensor nodes for evaluation had been implemented in the nesC language installed on the TinyOS operating system. We developed and tested our programs first on TOSSIM [Tos] and Avrora [Avr], two popular simulation and analysis tools for MICA micro-controllers. These tools allow both the code development and the debugging. The programs were then installed on MICA2 sensor nodes. The reported implementation results of the computation cost and memory consumption are the average of running the code 50 times.

#### 5.4.1.4 Results of X-IBOOS Scheme

We implemented the X-IBOOS scheme on sensor nodes first. This section presents the implementation results of the X-IBOOS scheme.

##### a) Computation Cost

The X-IBOOS scheme involves two pairing computations in signature verification as the most expensive cryptographic operations. Table 5.1 shows the time and energy consumption of this scheme. It took about 1.697s to compute the offline signature while only 0.018s to compute the online part. Thus, this scheme enables a sensor node to generate a final signature of a real time message in 0.018s only, which is quite fast considering the resource constraints of a sensor node. However, the signature verification is very expensive which consumes considerable time of 5.099s and hence the battery power<sup>1</sup> due to the two pairing computations. The computation of a single pairing operation using RELIC takes about 1.9s [OAG<sup>+</sup>11].

	<b>Time (s)</b>	<b>Energy (mWs)</b>
<b>Offline Sign</b>	1.697	50.92
<b>Online Sign</b>	0.018	0.54
<b>Verify</b>	5.099	177.01

Table 5.1: Time and Energy Consumption of X-IBOOS Scheme

<sup>1</sup>Power consumption is computed using the MICA2 data sheet [MIC] and the computed number of clock cycles for each stage. The power consumption is calculated at 3V power supply and 7.3728MHZ clock frequency.

### b) Signature Size

A signature in the X-IBOOS scheme is comprised of two group elements of the form  $(x, y)$  and one number. Based on our selection of  $\eta_T$  pairing and  $\sim 80$ -bit security level, a random number takes about 271 bits and a group element is about  $2 \times 271$  bits. Therefore, the resulting signature size is 1355 bits or 170 bytes. This signature size can be reduced to 102 bytes by applying compression and including only one co-ordinate ( $x$ ) of the group element. Given  $x$  and a single bit of  $y$ , the receiver can regenerate  $y$ , the second co-ordinate of the group element. However, this is a trade-off between the transmission cost of sending both co-ordinates  $(x, y)$  and the computation cost of deriving  $y$  on the receiver side.

### c) Memory Consumption

Table 5.2 summarizes the memory requirement of the X-IBOOS scheme including the size of both signature generation and verification codes. It also includes the code size of RELIC, TinyOS code, node's ID (16 bits) and private key ( $2 \times 271$  bits), master public key ( $2 \times 271$  bits) and other system parameters. The memory consumption of ROM, Global RAM and Stack RAM is 63,972, 1,933 and 1,911 bytes respectively. The stack memory is consumed only during the execution of the program, i.e., during the signature generation and verification. Once the program stops execution, this memory is available for other operations. Note that this is the total storage consumption on a sensor node when a sensor node acts as both a signer and a verifier. In our proposed authenticated broadcast by sensor nodes scheme, a sensor node acts as a sender as well as a receiver of broadcast messages. This memory consumption can be reduced by storing only one co-ordinate  $x$  of the group elements on sensor node. Given  $x$  and a single bit of  $y$ , the node can derive  $y$  when it needs. This will reduce the storage consumption per one group element stored on the sensor node by 270 bits.

ROM	Global RAM	Stack RAM
63,972	1,933	1,911

Table 5.2: Memory Consumption of X-IBOOS Scheme in Bytes

#### 5.4.1.5 Optimization: Our Adapted B-IBOOS Scheme

We proposed to evaluate the two IBOOS schemes, X-IBOOS and R-IBOOS, as mentioned earlier. However, the evaluation results of X-IBOOS scheme depict the fact that the X-IBOOS scheme is costly for sensor nodes in terms of resource consumption. In fact, the reason behind this cost is the pairing computations which consume considerable resources on sensor nodes including processing time, battery power and memory. The signature verification in X-IBOOS scheme consumed 5.099s which is not very ideal for real-time applications. A single pairing computation using RELIC takes about 1.9s [OAG<sup>+</sup>11]. To the best of our knowledge, this is the most efficient implementation result of pairing operation for MICA2 sensor nodes. The resulting signature size is 170 bytes which implies considerable communication cost. Since the R-IBOOS scheme also requires pairing computations, similarly expensive results are to be expected from the implementation of R-IBOOS scheme. Hence, we decided to skip its implementation as we were looking for IBOOS schemes efficient for sensor nodes. Our next step was to find a pairing-free IBOOS scheme to implement and evaluate for sensor nodes.

To the best of our knowledge, there are a few ECC based IBS schemes without pairing [BNN04, CKDZ08, GG09] but no ECC based IBOOS scheme without pairing. Therefore, to obtain a pairing-free IBOOS scheme, we securely transformed a pairing-free IBS scheme to a pairing-free IBOOS scheme, i.e., B-IBOOS scheme. The details of this modification together with the B-IBOOS scheme have already been discussed in Section 5.2.2. The B-IBOOS scheme has only one point multiplication as the expensive operation in signature generation which is computed during the offline phase. The online phase only performs integer arithmetics to obtain the final signature of the message, which is very efficient for sensor nodes. The signature verification in B-IBOOS scheme requires three point multiplication operations which, although expensive, are far less expensive than a single pairing computation for sensor nodes. Thus, we implemented and evaluated the adapted B-IBOOS scheme in next step.

#### 5.4.1.6 Results of B-IBOOS Scheme

In our implementation of B-IBOOS scheme, that is the IBOOS version of the IBS scheme presented in [CKDZ08], the sensor nodes are both signers and verifiers. The IBS scheme in [CKDZ08] is actually an improvement over the BNN-IBS [BNN04] scheme to reduce the signature size. This improved version of the BNN-IBS

scheme has already been proposed to use in WSNs in [CKDZ08] without the actual implementation on sensor nodes. In [CKDZ08], the improved signature scheme is used to provide outside user authentication where sensor nodes are the verifiers only.

### a) Computation Cost

The B-IBOOS scheme computes three point multiplication operations in signature verification as expensive cryptographic operations for sensor nodes. The offline phase is comprised of only one point multiplication. A point multiplication took 0.295s in our implementation which confirmed the point multiplication cost obtained by [ADLO10] using the same RELIC library. Table 5.3 shows the time and energy consumption of this scheme. It took about 0.295s to compute the offline signature whereas only 0.025s to compute the online part. The computation cost of the online phase is almost the same for both B-IBOOS (Table 5.3) and X-IBOOS (Table 5.1) schemes. Nevertheless, in the offline phase X-IBOOS scheme took more time, and thus consumed more battery power than B-IBOOS scheme. The same is the case with the verification phase. The B-IBOOS scheme, being a pairing-free signature scheme, verifies the signature in 1.044s only as compared to the verification time of 5.099s of the X-IBOOS scheme. A comparison of Table 5.1 and Table 5.3 revealed the fact that the B-IBOOS scheme is very efficient for resource constrained sensor nodes in terms of computation cost when compared with the X-IBOOS scheme.

	Time (s)	Energy (mWs)
<b>Offline Sign</b>	0.295	8.85
<b>Online Sign</b>	0.025	0.74
<b>Verify</b>	1.044	31.33

Table 5.3: Time and Energy Consumption of B-IBOOS Scheme

### b) Signature Size

The signature in B-IBOOS scheme is comprised of one elliptic curve point of the form  $(x, y)$  and two numbers. We used 163-bit field for ECC to meet the ~80-bit security level. For these settings, a number takes 160 bits while one elliptic curve point takes  $2 \times 163$  bits. Therefore, the resulting signature size is 646 bits (80 bytes) without compression while 484 bits (60 bytes) with compression. This signature size is much smaller than the one in X-IBOOS scheme, and thus results in a reduced transmission cost. The standard IEEE Std. 802.15.4 [ICS03] for the low-power

sensor networks allows a variable payload of up to 102 bytes. With this packet size, a sensor node still has 22 bytes available to include its *ID* and the message *m*, other than uncompressed 80 bytes of the signature, to send them all together in a single packet. The messages exchanged to report any critical event, for instance the location of an enemy, are usually short in size up to a few bytes. Therefore, 22 bytes provide enough space for both the *ID* and the message *m*.

### c) Memory Consumption

Table 5.4 shows the memory requirement of the B-IBOOS scheme. Like X-IBOOS scheme (Table 5.2), it includes the memory consumed by the signature generation and verification code, RELIC code, TinyOS code, node's ID (16 bits) and private key (160 bits), master public key ( $2 \times 163$  bits) and other system parameters. In case of B-IBOOS scheme, ROM, Global RAM and Stack RAM consume 47,798, 1,902 and 1,821 bytes respectively. The memory consumed by the stack is returned once the program completes its execution. Like in the case of X-IBOOS scheme, this is the total storage consumption on a sensor node when a sensor node acts as both a signer and a verifier. This memory consumption can also be reduced by storing only one co-ordinate (*x*) of the elliptic curve points of the form (*x*, *y*). It reduces the memory consumption per one elliptic curve point stored on the sensor node by 162 bits. Compared with the memory consumption in X-IBOOS scheme (Table 5.2), the ROM consumption is lower in B-IBOOS scheme than in X-IBOOS scheme while the Global RAM consumption is almost the same in both schemes. The stack usage is slightly lower in B-IBOOS scheme than in X-IBOOS scheme. However, the overall RAM consumption of B-IBOOS scheme is slightly smaller than the RAM consumption in X-IBOOS scheme.

ROM	Global RAM	Stack RAM
47,798	1,902	1,821

Table 5.4: Memory Consumption of B-IBOOS Scheme in Bytes

#### 5.4.1.7 Optimization

In the light of the results we obtained for computation cost, signature size and memory consumption of both IBOOS schemes, X-IBOOS and B-IBOOS, it is clear that the B-IBOOS scheme outperforms the X-IBOOS scheme in terms of

computation cost and signature size. The memory usage, however, did not make a big difference in both schemes. One factor, which was contributing towards the memory usage of B-IBOOS scheme, was the fact that RELIC used a precomputed table to speed up the computation of point multiplication for ECC based schemes. This precomputed table was also stored and consumed some memory space on the sensor node. To optimize the B-IBOOS scheme for memory consumption, we decided to evaluate this scheme without the precomputed table of RELIC. It restrained us from using one efficient function of RELIC used to compute the point addition of the two point multiplications, i.e.,  $(aP + bQ)$ . Table 5.5 and Table 5.6 show the implementation results of the B-IBOOS scheme obtained after removing the precomputed table of RELIC.

	<b>Time (s)</b>	<b>Energy (mWs)</b>
<b>Offline Sign</b>	0.317	9.52
<b>Online Sign</b>	0.025	0.74
<b>Verify</b>	1.118	33.54

Table 5.5: Optimized Time and Energy Consumption of B-IBOOS Scheme

<b>ROM</b>	<b>Global RAM</b>	<b>Stack RAM</b>
45,612	1,634	1,381

Table 5.6: Optimized Memory Consumption of B-IBOOS Scheme in Bytes

Compared with Table 5.3 and Table 5.4, avoiding the precomputed table reduced the memory consumption of B-IBOOS scheme, particularly the RAM consumption. Although, it slightly increases the time and power consumptions of the offline phase and the signature verification phase, this increment is not a drastic change. It is an acceptable trade-off between the computation cost and the memory usage giving 10% of free memory. However, it depends on the nature of the application whether it can compromise on speed or memory.

Table 5.7 and Table 5.8 summarize the performance results of X-IBOOS and B-IBOOS schemes for comparison purposes.

#### 5.4.1.8 Application Possibilities for Different IBOOS Schemes

The two implementations of B-IBOOS scheme offer a trade-off between the computation cost and the memory usage. Memory can be saved by removing the

Schemes		Time (s)	Energy (mWs)
X-IBOOS	Offline Sign	1.697	50.92
	Online Sign	0.018	0.54
	Verify	5.099	177.01
B-IBOOS	Offline Sign	0.295	8.85
	Online Sign	0.025	0.74
	Verify	1.044	31.33
B-IBOOS - Optimized	Offline Sign	0.317	9.52
	Online Sign	0.025	0.74
	Verify	1.118	33.54

Table 5.7: Summary of Time and Energy Consumption of X-IBOOS and B-IBOOS Schemes

Schemes	ROM	Global RAM	Stack RAM
X-IBOOS	63,972	1,933	1,911
B-IBOOS	47,798	1,902	1,821
B-IBOOS - Optimized	45,612	1,634	1,381

Table 5.8: Summary of Memory Consumption of X-IBOOS and B-IBOOS Schemes in Bytes

precomputed table and slightly increasing the computation time. However, this can be decided depending on the type of application. The offline signature is computed before the message to be signed is available and the online phase takes the same time in both implementations, i.e., 0.025s. Therefore, the time to compute the final signature, once the message is known, is the same in both cases. For the time critical applications, it is reasonable to use the first implementation of B-IBOOS scheme if the receiver is a sensor node, and the second implementation of B-IBOOS scheme if the receiver is a powerful device. Moreover, if the offline phase is performed on the base station and the resulting offline signature is stored on the sensor node, the X-IBOOS scheme can also be useful for such applications of WSNs where the signature verifier is a powerful device. For such applications, the X-IBOOS scheme enables a sensor node to broadcast a signed message only in 0.018s which is quicker than the B-IBOOS scheme.

#### 5.4.1.9 Impact of Applying IBOOS Schemes on Sensor Nodes

In previous sections, we presented the experimental results of implementing and executing IBOOS schemes on sensor nodes. In the light of those results, we

demonstrate the impact of IBOOS schemes on sensor nodes lives in this section. The public key cryptography based digital signatures are expensive to compute on sensor nodes as compared to the MAC operation. However, the application of public key cryptography operations on sensor nodes does not affect a node's life time drastically, if the number of operations is smaller or spread over time [PLP06]. In time critical applications of WSNs, the broadcast of a message by a sensor node is not a very frequent event. An example of such applications is a forest fire alarm application. In a forest fire alarm application, a message is sent by a sensor node only when a fire is set up somewhere in the forest which is not very frequent. Signing and verifying a message occasionally only in critical situations is not very expensive for the sensor nodes. With 2AA batteries in ordinary MICA sensor nodes, the available battery power is 6750,000mW [PLP06]. If only 2% of this battery power i.e., 135,000mW, is available for signing broadcast messages, a sensor node can sign 14,077 messages applying the first B-IBOOS implementation and 13,158 messages applying the second B-IBOOS implementation during the life time of the batteries. For the same available battery power, a sensor node can verify 4,308 messages in the first B-IBOOS implementation and 4,025 messages in the second B-IBOOS implementation. This number of broadcast messages is big enough for the considered applications, for instance, the forest fire alarm application. These figures are calculated by assuming the fact that both the offline and the online phases are performed by the sensor nodes themselves.

### 5.4.2 Performance of the Proposed Protocol

We now evaluate the performance of the proposed authenticated broadcast by sensor nodes protocol using IBOOS schemes. To the best of our knowledge, the proposed protocol is the first attempt to highlight and provide a solution to the problem of sensor nodes broadcast authentication. The proposed protocol introduces the online/offline signature schemes to the WSNs for the first time. It therefore handles the problem of real-time broadcast applications providing security with efficiency. Due to the online/offline signature, a resource constrained sensor node can sign a message quickly and efficiently. Moreover, the ID-based cryptography copes with the problem of public keys and certificates management. The proposed protocol using the IBOOS schemes does not only meet the design challenges mentioned in the beginning of this chapter but also provides some extra features as follows:



#### 5.4.2.1 Broadcast by Sensor Nodes

In the proposed protocol, a sensor node can broadcast a message by itself without the involvement of the base station or any other third party unlike  $\mu$ TESLA based authentication schemes where a sensor node can broadcast only via base station. To the best of our knowledge, this is the first broadcast authentication protocol for large scale sensor networks which empowers all sensor nodes in the sensor network to become broadcast senders.

#### 5.4.2.2 Quick Broadcast

An online/offline signature scheme computes the most time consuming offline phase of the signature generation beforehand and the sensor nodes only need to compute the quick and efficient computations of the online phase when the message is known. It enables sensor nodes to sign a message quickly once they have some event to report. Consequently, the proposed protocol enables a quick broadcast of signed messages by the sensor nodes to respond to the time critical situations.

#### 5.4.2.3 Storage Efficiency

The broadcast receiver nodes store the  $\mu$ TESLA parameters, in  $\mu$ TESLA based schemes, and the ID and public key pairs, in digital signature based schemes, of all broadcast senders. In the proposed protocol, the receiving nodes do not need to store any broadcast sender specific information for the verification of broadcast messages. They are only required to store their own public and private information and the system parameters. Thus, the proposed protocol achieves the storage efficiency.

#### 5.4.2.4 Computation Efficiency

In the proposed protocol, as discussed previously, the offline signature generation phase can be performed by any resourceful device, for instance, the base station. By performing the most complex computations of offline phase on the base station, the sensor nodes are only left with the online phase computations. The online phase computations involve only integer arithmetic and are very efficient to compute for sensor nodes in terms of time and power consumption. As a result, the proposed protocol reduces the computation burden of a signature generation on sensor nodes. However, it depends on the nature of the application and the trade off between the computation and communication costs. For example, in a fire alarm application,

the broadcast of a message is not a very frequent event. The base station can compute and distribute a few offline signatures to the sensor nodes that can store them for later use. The distribution and storage of a few offline signatures incur only a reasonable communication and storage overheads on sensor nodes, reducing the computation overhead on the other hand. Moreover, reusing an offline signature to sign more than one message results into further computation efficiency.

#### 5.4.2.5 Communication Efficiency

The ID-based signature schemes do not require a broadcast sender to send a public key and/or a certificate with all messages, as is done in some existing signature based authentication schemes for WSNs in order to avoid storage overhead. Thus, the proposed protocol using ID-based online/offline signatures reduces the communication overhead of the signature based authentication schemes without increasing storage overhead.

#### 5.4.2.6 Multiple Senders

Unlike  $\mu$ TESLA based broadcast authentication schemes, which allow only one broadcast sender (i.e., the base station), the proposed protocol enables more than one sensor node to send authenticated broadcast messages. Preloaded with their IDs, private keys and other system parameters, the broadcast sender nodes can broadcast authenticated messages which can be verified by any other sensor node in the network using the ID information of the broadcast sender sent along with the messages. Moreover, the multiple broadcast sender nodes can send broadcast messages simultaneously at any time which are verified by the receivers as soon as they have been received. Therefore, the proposed protocol using the ID-based online/offline signatures supports multiple broadcast senders.

#### 5.4.2.7 Scalability

Scalability in the context of designing a broadcast authentication scheme for WSNs refers to building an application that can scale well and easily support large number of broadcast senders and receivers. In most of the existing broadcast authentication schemes for WSNs, the high storage requirement restricts the number of broadcast senders. For example, the  $\mu$ TESLA based schemes require every sensor node to store the long hash key chains and  $\mu$ TESLA parameters of every other broadcast sender node in the network whereas the digital signature based schemes require to

store the ID and public key pairs of every broadcast sender. Due to the limited storage capability, a sensor node can store the sender related information, for instance the  $\mu$ TESLA parameters or the ID and public key pairs, for only a limited number of broadcast senders, restricting the number of broadcast senders. ID-based cryptography resolves the public key and certificate management problem faced in WSN environment and therefore, the sensor nodes do not need to store the ID and public key pair of any sender node. As a result, the proposed protocol using the ID-based signature schemes supports a large number of broadcast senders. Moreover, new sensor nodes can be added to the WSN easily at any time. Preloaded with ID, private key and system parameters, a new sensor node can broadcast authenticated messages as well as verify signed messages sent by any other broadcast sender node. Hence, the proposed protocol meets the scalability challenge of designing a broadcast authentication protocol described at the beginning of this chapter.

#### 5.4.2.8 Robustness to Packet Loss

In  $\mu$ TESLA based authentication schemes, first broadcast packets are sent and then the MAC keys to authenticate those packets are sent. If a packet carrying a MAC key is lost due to the radio communication, the broadcast receivers will not be able to authenticate all those broadcast packets received in a previous time interval. Hence,  $\mu$ TESLA based broadcast authentication protocols do not provide robustness to packet loss. In the proposed protocol, as soon as a signed message is received it can be verified individually and if a packet is lost, it only affects the single message carried by that packet, providing robustness to packet loss.

## 5.5 Security Analysis

This section first reviews the security of the IBOOS schemes i.e., the B-IBOOS scheme, and then the security of the proposed authenticated broadcast by sensor nodes protocol using IBOOS schemes.

### 5.5.1 Security of the IBOOS Schemes

The performance evaluation of the B-IBOOS scheme proved this signature scheme the most efficient IBOOS scheme among the existing ones. We therefore only discuss the security of the B-IBOOS scheme. The BNN-IBS scheme has been shown to be *existentially unforgeable against the chosen message and ID attacks*

(i.e., `euf-cma-ida` secure) in [BNN04]. Using BNN-IBS as an IBOOS does not affect the security of this signature scheme. It is secure to compute the offline part before the message is known and store it. If an attacker compromises a sensor node and obtains both the offline signature  $Y$  and the random number  $y$  used to generate  $Y$ , he still would not be able to get any extra benefits other than the ones obtained by compromising a sensor node. The attacker will not be able to reuse that offline signature to impersonate the sensor node since the offline signature does not use the private key of the sensor node or any other node specific information. After computing the final signature and broadcasting it, the sensor node deletes both  $y$  and  $Y$ . In addition, when an attacker compromises a sensor node he has access to the private key of the sensor node. In that scenario, none of the signature schemes are secure anymore.

## 5.5.2 Security of the Proposed Protocol

### 5.5.2.1 Security Properties Achieved

We now explain how the security properties listed in Section 4.3 are achieved by the proposed authenticated broadcast by sensor nodes protocol.

**Authentication.** The proposed sensor nodes broadcast protocol provides the required authentication, i.e., a proof of the identity of the claimed message sender. The required authentication is achieved via the signed messages that are signed by a broadcast sender using his private key and a secure IBOOS scheme, giving a proof of the sender's identity. This proof of identity is affirmed by verifying the signed message. The successful signature verification implies that the claimed message sender is the actual source of the message.

**Message Integrity.** The proposed protocol using IBOOS schemes ensures message integrity since any changes made in the contents of a broadcast message during the transmission are detected through the signature verification. Any message modified in transit will not pass the signature verification process.

**Verification.** The proposed protocol empowers every sensor node in the sensor network to verify a broadcast message sent by any sender in the network. The verifiers only need the signer's ID information sent along with the message to verify the signed message. The receivers do not need any other sender specific information, like  $\mu$ TESLA parameters, in order to be able to verify a signed message. Moreover, our experimental results proved the fact that it is possible for the resource

constrained sensor nodes to perform signature verification computations required in an efficient IBOOS scheme.

**Freshness.** In the proposed protocol, every broadcast message includes a timestamp and only the messages with the fresh timestamps are accepted. Thus, the proposed protocol provides freshness.

**Availability.** In  $\mu$ TESLA based authentication schemes, where the base station makes a single point of failure, if the base station fails the whole protocols collapse. On the other hand, in the proposed broadcast authentication protocol even if the base station becomes unavailable for some time, the message senders can still broadcast the signed messages and the receivers can verify them.

### 5.5.2.2 Countermeasures to Security Attacks

This section discusses how the security attacks listed in Section 4.4 are countered by the proposed broadcast authentication protocol.

1. *Node Impersonation Attack.* The proposed broadcast authentication protocol employs secure IBOOS schemes providing strong authentication and message integrity. The secure IBOOS schemes make it impossible for an intruder to sign a message on behalf of or modify a valid message sent by a legitimate broadcast sender node. Only the legitimate broadcast sender node with a valid secret key can sign a message. For an attacker to impersonate a broadcast sender node, he must be able to compute a valid message signed by the target legitimate sender node (i.e., a valid message signature pair). However, it is hard to compute such a valid pair without the knowledge of private key of the target legitimate sender. At the same time, no one can forge a signature in the presence of an existentially unforgeable signature scheme.
2. *False Data Injection Attack.* The verification of the signed messages in the proposed broadcast authentication protocol distinguishes the actual data from the false data injected by the intruder to waste the resources of the relaying sensor nodes. The proposed protocol enables all sensor nodes on the message path, during multi-hop forwarding, to verify any data messages in the network and filter out false injected data as soon as it has been injected.
3. *DoS Attack.* The proposed broadcast authentication protocol yields authentication of a broadcast message without any delay. A signed broadcast message

is verified as soon as it has been received. This instant and individual authentication prevents the DoS attack faced in  $\mu$ TESLA based schemes. Moreover, the early detection of false data injected into the sensor network avoids the DoS attack against the resources of the relaying sensor nodes.

4. *Node Compromise Attacks.* The proposed protocol avoids the node compromise attack typically faced in MAC based authentication schemes. In the proposed protocol, an intruder can impersonate only the compromised node. The collusion of multiple compromised sensor nodes will not give any extra benefits to the adversary than those obtained by the individual compromised nodes. Furthermore, after revocation process the compromised nodes will not be able to successfully broadcast messages in the network.
5. *Message Replay Attack.* The proposed protocol uses timestamps in order to provide freshness which ultimately resists message replay attacks. A receiver node first checks the timestamp before actual signature verification to avoid the verification of a replayed message. Depending on the transmission delay imposed by the communication channel between the sender and the receiver, the sensor node sets a time threshold leaving a potential attacker little time to mount a replay attack.

## 5.6 Comparison with Existing Protocols

Now we compare the proposed authenticated broadcast by sensor nodes protocol using IBOOS schemes with the existing digital signature based authentication protocols for WSNs: CAS [RLZ07], DAS [RLZ07], IDS [RLZM07], and IMBAS [CKDZ08]. The details of these schemes are given in Section 3.2.2. We do not include the  $\mu$ TESLA based broadcast authentication schemes here due to the fact that they cannot provide a solution to the broadcast authentication problem in real-time applications of WSNs. For the comparison purposes, we assume the pairing-free B-IBOOS scheme for our proposed protocol. The actual implementation results of the existing digital signature based schemes on real sensor nodes are not available. These authentication schemes assume broadcast senders as powerful devices, however for comparison purposes, we estimate the cost of applying these schemes to ordinary sensor nodes. Using the most efficient implementation results of the computation costs of pairing operation, point multiplication and elliptic curve

Schemes	Signature	Sign Time		Verify Time	Storage <sup>α</sup>	Msg Size
		Offline s	Online s	s	KB	Bytes
Existing Broadcast Authentication Schemes						
CAS [RLZ07]	ECDSA	0	0.36	2×0.63	0	60 <sup>β</sup>
DAS [RLZ07]	ECDSA	0	0.36	0.63	1172 <sup>γ</sup>	40
IDS [RLZM07]	Pairing based	0	2.2	4.98	0	84 <sup>ρ</sup>
IMBAS [CKDZ08]	BNN-IBS	0	0.32	1.044	0	80
Proposed Broadcast Authentication Scheme						
Proposed	B-IBOOS	0.295	0.025	1.044	0	80

<sup>α</sup> storage only considers the additional authentication information stored

<sup>β</sup> Msg size ignores the signed certificate sent to extract ECDSA public key

<sup>γ</sup> computed for a large WSN of about 60,000 sensor nodes and 20 bytes public key

<sup>ρ</sup> signature size given in [RLZ07]

Table 5.9: Comparison of proposed broadcast authentication scheme with existing digital signature based broadcast authentication schemes for WSNs

digital signature algorithm (ECDSA) on MICA2 sensor nodes and ignoring all other costs including hash computations, we roughly estimated the time cost of these schemes for comparison purposes. A point multiplication operation on MICA2 takes about 0.295s (our result), pairing operation takes 1.9s [OAG<sup>+</sup>11] (and our result), signature generation and verification for ECDSA take 0.36s and 0.63s respectively [ADLO10]. Using these results, Table 5.9 gives the time cost of all schemes. The message size includes both the signature size as well as the public key (20 bytes for ECDSA) ignoring the certificate size for the first scheme while only the signature size for all other schemes.

The first two schemes **CAS** and **DAS** propose to use ECDSA signature scheme to sign a message. CAS requires the signer's public key and certificate to be sent with every signed message, increasing message size and hence the transmission cost. The receiver verifies two ECDSA signatures for every received message; one to verify the signed certificate and another to verify the signed message. DAS, on the other hand, requires all sensor nodes to store the public keys of every broadcast sender in the network, increasing the storage overhead. For a large scale sensor network, it is not possible for a sensor node having a limited storage capability to store the

public key of every other sensor node in the network. For instance, considering a large scale sensor network of about 60,000 sensor nodes and ECDSA public key of size 20 bytes, every sensor node is required to store 1172KB which is beyond the storage capabilities of a sensor node.

The signature generation in **IDS** comprises one pairing and one point multiplication computations while signature verification involves two pairing computations and one exponentiation in target group  $\mathbb{G}_T$  (say  $E_{TG}$ ). Note that if the basic operation in  $\mathbb{G}$  is denoted multiplicatively ( $\times$ ) instead of additively ( $+$ ), the point multiplication in  $\mathbb{G}$  is then called exponentiation (say  $E_G$ ) correspondingly, and hence takes 0.295s. However, the exponentiation in the target group  $\mathbb{G}_T$  (in the settings of pairing [CMS08]) takes more time than exponentiation (or point multiplication) in  $\mathbb{G}$  because of the fact that it computes arithmetic in  $\mathbb{G}_T$  which is operated in a field much bigger than the field in which  $\mathbb{G}$  is defined. In usual implementations of pairing, one exponentiation in  $\mathbb{G}_T$  costs about equal to four exponentiations in multiplicative group [CMS08] and hence four point multiplications in an additive group. Thus, the signature verification cost for IDS is two pairing operations and four point multiplications (for  $E_{TG}$ ). Like exponentiation operation, a point multiplication in pairing-based settings also takes longer than in pairing-free ECC based settings. However, we use the cost of point multiplication in pairing-free ECC based settings here to evaluate IDS scheme. **IMBAS** proposes BNN-IBS signature scheme for sensor networks where sensor nodes are only receivers (verifiers). The signature verification in BNN-IBS requires three point multiplications as expensive operations. The signature verification costs of BNN-IBS and B-IBOOS schemes are the same since B-IBOOS is the adapted IBOOS version of BNN-IBS scheme.

The figures in Table 5.9 show that the proposed sensor nodes broadcast authentication scheme using B-IBOOS scheme enables a sensor node to sign and broadcast a message in 0.025s only whenever it has some time-critical event to report as compared to the existing schemes which take significantly longer. In signature verification, only DAS consumes less time than the proposed scheme. However, it increases the storage overhead of storing senders public keys beyond the storage capabilities of the sensor nodes. The comparison results given in Table 5.9 suggest that the proposed authenticated broadcast by sensor nodes protocol using IBOOS schemes is the most efficient and suitable scheme for the time critical applications of WSNs when compared with the existing signature based authentication schemes. It also allows the base station to compute the offline signature on behalf of a sensor node, reducing computation overhead on them.



Note that for MICA2, the active power consumption is 30mW [PLP06]. Therefore, the energy consumption  $Y$  can be computed using the time consumed  $X$  as  $Y = X \times 30$  (mWs). Moreover, the transmission cost is proportional to the message size.

## 5.7 Concluding Remarks

In this chapter, the proposed broadcast by sensor nodes authentication protocol and its performance and security evaluations have been presented. The existing  $\mu$ TESLA based authentication schemes failed to handle the broadcast scenario of multiple senders. The existing signature based authentication schemes, on the other hand, failed to meet the requirements of real-time applications. Compared to the existing digital signature based authentication schemes, the proposed protocol introduces the online/offline signatures to the WSNs. An IBOOS scheme does not only address the problem of real time broadcast applications of WSNs but also brings efficiency on resource constrained sensor nodes side. The implementation and evaluation of several IBOOS schemes on real sensor nodes prove the fact that the IBOOS schemes are suitable for the sensor nodes. The X-IBOOS scheme proved expensive for the sensor nodes, consuming considerable resources on them. However, the reason behind this cost was not the online/offline signature itself but the expensive pairing based cryptography. The implementation results of our adapted B-IBOOS scheme proved this argument. It implies that if we use pairing-free efficient ECC based IBOOS schemes for WSNs, we can obtain efficient results. Moreover, if in future a more efficient implementation of pairing computation for sensor nodes processors is available than the one in hand now, the pairing-based IBOOS scheme may also become eligible to use in WSNs. The implementation results of the IBOOS schemes confirmed the suitability of the proposed authenticated broadcast by sensor nodes protocol, demonstrating particularly low computation overhead of the proposed protocol as compared to the existing digital signature based broadcast authentication protocols for WSNs.



## Chapter 6

# Outside User Authentication Protocol

***Chapter Overview:** This chapter presents the proposed outside user authentication protocol using the ID-based signature schemes together with its performance and security evaluations. The first half of the chapter highlights the challenges faced in the design of a user authentication protocol and discusses the available ID-based signature schemes. It also describes the available session key establishment options and protocols and our proposed ID-based one-pass session key establishment protocol. It then presents the proposed outside user authentication protocol in detail. The second half of this chapter evaluates the performance and security of the proposed protocol. At the end of the chapter, the proposed protocol has been compared with the existing user authentication protocols for WSNs.*

## 6.1 Introduction

The favorable outcomes of many WSN applications rely upon the presence of a secure and efficient outside user authentication protocol. An example of such applications is a large scale sensor network set up for business purposes. This sensor network collects the data of interest to multiple outside users, like research organizations, other businesses and individuals, and sells this data to these users in return of money. The major concern of this application is to securely deliver the valuable data only to authorized users who have paid for the data. Like for broadcast authentication protocol, designing a secure and efficient user authentication protocol for WSNs is challenging. The major efficiency concern in the case of a user authentication protocol is the user verification task of the protocol performed on the resource constrained sensor node's side. Indeed, the outside users are equipped with the resourceful devices to query sensor nodes data which can perform expensive computations. The major challenges faced in designing a secure as well as efficient authentication protocol for WSN, as discussed in [LPW06, Per01], are revised here in the context of user authentication as follows:

- *Efficient verification.* Since the sensor nodes are resource constrained devices with limited computation, the verification overhead of the user authentication information should be small.
- *Low communication.* Due to the scarce battery power, a protocol with low communication overhead is needed for low power sensor nodes.
- *Instant and distributed user authentication.* In order to avoid the high communication overhead and in-network traffic congestion, the user authentication protocol should enable every receiver node to verify a user authentication request message locally without the involvement of any third party as soon as the message has been received.
- *Scalability.* The above mentioned WSN business applications have a potentially large number of outside users to compensate the deployment expenses of the large scale sensor networks. The user authentication protocol should be independent of the number of outside users as well as the verifier nodes. It should also allow to add new users and delete the ones whose access time period has expired. At the same time, it should empower every newly added sensor node to act as a verifier.

- *Session key establishment.* Due to the radio communication, anyone can overhear the valuable or confidential data in transit. Hence, the user authentication protocol should also facilitate the establishment of a session key between the user and the sensor node for the safe transfer of valuable data. The session key establishment protocol should also meet the challenges of efficiency and scalability.

Unfortunately, most of the existing user authentication schemes for WSNs are centralized, and hence do not provide instant, distributed user authentication. The distributed schemes, on the other hand, are not efficient in terms of resource consumption, also lacking scalability and key establishment features.

This chapter presents the proposed *outside user authentication* protocol for WSNs using the ID-based Signature (IBS) schemes. The proposed protocol adopts a distributed user authentication approach to provide instant authentication and avoid the problems of centralized approaches. The IBS schemes overcome the resource consumption of digital signature based distributed approaches. An IBS scheme enables a sensor node to verify a signed user request message using the user's ID instead of user's public key and, as a result, solves the scalability problem by reducing the storage overhead. After successful authentication of a user, a session key establishment between a user and a sensor node is another feature of the proposed protocol. A *key establishment (KE)* protocol provides the communicating parties with a secret and shared session key which can be used to encrypt and decrypt the data exchanged between the parties. A session key establishment protocol is required here for the secure communication of the sensor nodes data to the user. This chapter also describes our ID-based one-pass session key establishment (ID-1P-SKE) protocol which is mainly designed for WSNs.

As mentioned in previous chapter, both security and efficiency are important aspects in making a decision to adopt a protocol for WSNs. The *performance* and the *security* of the proposed protocol have also been analyzed in this chapter. A cryptographic protocol must come with a security proof to attest that it satisfies the required security properties. Since the proposed ID-1P-SKE protocol is a new cryptographic protocol, it requires a formal security analysis. Therefore, we formally analyze the security of ID-1P-SKE protocol using the reductionist proof technique. The chosen IBS scheme had already been proved to be existentially unforgeable against the chosen message and ID attacks (i.e., *euf-cma-ida* secure) in [BNN04] and did not require another security analysis.

**Contribution.** The major contribution made by this chapter is a complete user authentication protocol using IBS schemes which other than user authentication facilitates the establishment of a session key between a user and a sensor node. Another major contribution is a new ID-based one-pass session key establishment protocol (ID-1P-SKE) together with formal security analysis and performance evaluation.

## 6.2 Options for IBS Schemes

We now discuss the IBS schemes options for our proposed outside user authentication protocol.

### 6.2.1 Available IBS Schemes

Like IBOOS schemes, there are many IBS schemes available which are mainly based on ECC or RSA signatures. However, ECC based IBS schemes are given preference in this research work because of their efficiency advantages over RSA based IBS schemes. We selected the most efficient IBS scheme called BNN-IBS [BNN04] to evaluate our proposed protocol. There are some variants of BNN-IBS scheme, for instance SLL-IBS [GG09] and the Cao's modified version of BNN-IBS, i.e., vBNN-IBS [CKDZ08]. The vBNN-IBS scheme has been adapted as a paring-free IBOOS (B-IBOOS) scheme and discussed in previous chapter (Section 5.2.2) in detail. The BNN-IBS and vBNN-IBS have the same computation complexities with the only difference of signature sizes. The vBNN-IBS scheme modifies the BNN-IBS scheme to reduce the signature size of the latter. The SLL-IBS scheme is not discussed here because of its close similarities to BNN-IBS and vBNN-IBS schemes. Any of these IBS schemes can be used for the proposed outside user authentication protocol.

## 6.3 Options for Session Key Establishment Schemes

As mentioned earlier, a *key establishment* protocol provides a secret and shared session key which can be used for the encryption and decryption of data. In a secure key establishment protocol, no third party, other than the communicating parties, is able to impersonate any of the legitimate parties participating in the protocol. This

is called the *authenticated key establishment (AKE)* problem, which is harder than the *key establishment (KE)* problem [BWJM97]. Designing a secure and efficient AKE protocol for resource constrained devices like sensor nodes is a challenging task. Before presenting the proposed ID-based one-pass session key establishment (ID-1P-SKE) protocol, the two party key establishment options and the available protocols are outlined in order to understand the inspirations behind designing a new session key establishment protocol for WSNs.

### 6.3.1 Key Establishment Options

#### 6.3.1.1 Two Pass Key Establishment

To establish a key between the two parties, one option is a two pass key establishment protocol. In a two-pass key establishment protocol, two messages are exchanged and processed in order to compute a common key since both parties exchange their ephemeral public keys. The two-pass key establishment protocols have been a focus of research since the introduction of a pioneering solution, Diffie-Hellman (DH) key establishment protocol [DH76]. However, the typical DH protocol suffers from the man-in-the middle attack (MIMA) due to the lack of authentication (of both parties). The efforts to counter this attack have given rise to various two-pass key establishment protocols, for instance, the secure Station-to-Station (STS) [DVOW92] protocol. The STS protocol demands every message exchanged between the two parties to be digitally signed. This approach reduces the chances of MIMA, however, increases the computational requirements of a key establishment protocol, making it costly. The high computational and communication costs of the secure two-pass protocols make them unsuitable for use in several applications that require low-cost one-way communication, for instance, email, SMS and store-and-forward applications (where the receiver cannot reply immediately or do not reply at all), and low-power mobile environments, for instance, wireless sensor networks (where low communication cost is critical).

#### 6.3.1.2 One Pass Key Establishment

To satisfy the resource constraints of sensor nodes, a session key establishment protocol with *high* security and a *minimum* amount of computation and number of passes is required. A secure one-pass key establishment protocol is an attractive alternative for the sensor nodes. A one-pass key establishment protocol between two

parties is the one in which only one message is exchanged for the establishment of a session key, i.e., only the sender (protocol initiator) generates an ephemeral private key and computes its public part, i.e., the ephemeral public key. The sender sends the ephemeral public key to the receiver (protocol responder). Both parties then compute a shared session key using their own private keys, ephemeral keys and other public information. A one-pass key establishment protocol reduces the transmission and processing costs due to the fact that only a single message is transmitted and processed.

This fact can be utilized in the proposed user authentication protocol to achieve transmission cost efficiency. Since in a one-pass key establishment protocol only one party computes and sends its ephemeral key to the other party, that single message can be combined with the signed user authentication request message (in user authentication phase). This helps to achieve an efficient transmission cost in the proposed outside user authentication protocol. Rather than two separate messages, one for user authentication and another for key establishment, a single message will serve both jobs. It will also counter the man-in-the-middle attacks faced in a typical two-pass Diffie-Hellman (DH) protocol. The only message exchanged between the user and the sensor node for the key establishment will be signed by the user and verified by the sensor node. It prevents an intruder from impersonating a user and at the same time sending a fake ephemeral public key on behalf of the user, avoiding the man-in-the-middle attack.

Besides the reduced transmission and processing costs, another advantage of a one-pass key establishment protocol is its use in off-line communications. In an off-line communication, one party (sender) is on-line whereas the other party (receiver) may or may not be on-line. For instance, the sender of an email is on-line when he sends an email whereas the email receiver does not need to be necessarily on-line at that time. He can receive the email later on when he comes on-line. A one-pass key establishment protocol is used to secure off-line communications as follows: The sender computes its ephemeral private and public keys and the shared session key, encrypts the message  $m$  (any confidential message) using the computed session key and sends both the ephemeral public key and the ciphertext of  $m$  to the off-line receiver. The receiver, when he comes on-line, computes the same shared key using the received sender's ephemeral public key and decrypts the received message.

This feature is particularly useful for applications where only one entity is on-line. However, it can also benefit those applications of WSNs where the privacy of a user query is a requirement, as described in [CYS<sup>+</sup>07]. An example of such applications



is ocean readings data collected by a WSN. In this application, different oil business companies who are users of the sensor nodes data may be curious to know the data interests of each other and eavesdrop on each other's data queries. Therefore, the queries should also be hidden in such applications. In order to provide query privacy, the only message sent by the user for key establishment can be combined with the encrypted user query. If the user authentication succeeds, the sensor node computes the session key and decrypts user query otherwise it discards the message. Here only a single message is exchanged to authenticate a user, establish a key and send an encrypted user query. This feature helps to achieve highly efficient transmission cost for those applications of WSNs where the query privacy is mandatory.

### 6.3.2 Available Key Establishment Protocols

The existing protocols [HCK<sup>+</sup>03, JLX07, KLP<sup>+</sup>07, ZW09] for WSNs to establish a session key between a user and a sensor node, briefly discussed in Section 3.3.2, have several limitations. They are all either expensive for sensor nodes in terms of computation and communication costs or lack some security features. In addition, not one of these protocols is a one-pass key establishment protocol. In recent years, a few ID-based one-pass key establishment protocols [BJT04, OTO05, Wan05, GBGN08] have been designed for traditional networks. However, none of these protocols are computationally efficient as all of these require pairing computations. The extensive use of pairing computations makes the existing ID-based one-pass key establishment protocols quite slow and computationally expensive, particularly for resource constrained devices like sensor nodes.

### 6.3.3 Proposed Key Establishment Protocol

Due to the inadequacy of the existing key establishment protocols to provide an efficient and secure solution to the low power WSN environment, we have designed a new pairing-free ID-based one-pass session key establishment (ID-1P-SKE) protocol for WSNs. The proposed ID-1P-SKE protocol does not require both parties (the initiator and the responder) to compute any pairing operation. The lack of pairing computations makes ID-1P-SKE protocol highly efficient than the existing ID-based one-pass schemes and therefore suitable for sensor nodes devices by providing security with efficiency. Besides efficient computation and communication costs, another advantage of ID-1P-SKE protocol is its same ID-based setup as used by the BNN-IBS scheme. Consequently, using the same ID-based parameters, the sensor

nodes can not only authenticate users via BNN-IBS scheme but also establish a session key with them. The ID-1P-SKE protocol is described in next section.

### 6.3.3.1 ID-1P-SKE Protocol

The proposed ID-1P-SKE protocol has three algorithms: *Setup*, *Key Extract* and *Key Establishment*.

**Setup.** This algorithm generates the system parameters including master public key ( $mpk$ ), and the corresponding master secret key ( $msk$ ). This algorithm performs the following steps:

- (a) Specify the parameters  $\mathbb{E}/\mathbb{F}_p$ ,  $q$ ,  $p$ ,  $P$  and  $\mathbb{G}$ , where
  - $\mathbb{E}/\mathbb{F}_p$  is an elliptic curve  $\mathbb{E}$  over a finite field  $\mathbb{F}_p$ ,
  - $q$  is a large prime number and  $p$  is the field size,
  - $P$  is a point of order  $q$  on the curve  $\mathbb{E}$  and,
  - $\mathbb{G}$  is a cyclic group of order  $q$  under the point addition “+” generated by  $P$ .
- (b) Chose  $s \in_R \mathbb{Z}_q^*$  uniformly as  $msk$ .
- (c) Compute  $mpk$  as  $P_{PKG} = sP$ .
- (d) Choose one hash function  $H: \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$ .
- (e) Choose one function  $\chi: \mathbb{G} \rightarrow \{0, 1\}^k$  to derive the session key where  $k$  is the security parameter.
- (f) Output the system parameters  $\{\mathbb{E}/\mathbb{F}_p, q, p, P, \mathbb{G}, P_{PKG}, H, \chi\}$  and keep  $s$  secret.

**Key Extract.** This algorithm takes  $msk$  and an  $ID$  as input and generates a private key corresponding to that  $ID$  using the Schnorr signature. For an identity  $ID_i$  of  $I$ , this algorithm performs the following steps:

- (a) Choose  $r_i \in_R \mathbb{Z}_q^*$ .
- (b) Compute  $R_i = r_i P$ .
- (c) Compute  $c_i = H(ID_i, R_i)$ .

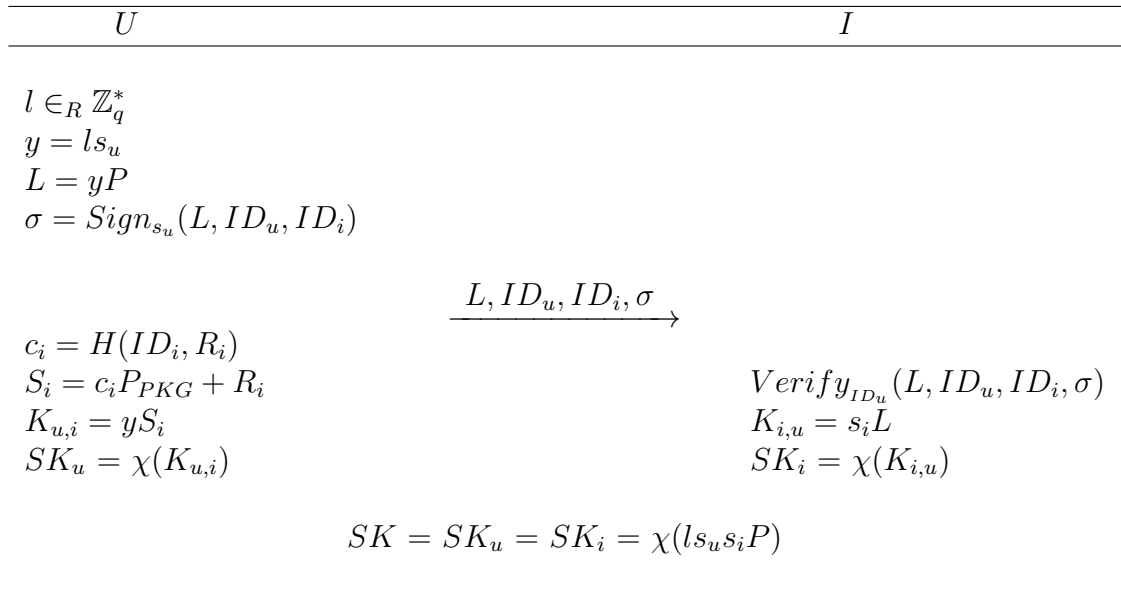


Figure 6.1: Authenticated ID-based One-Pass Session Key Establishment

- (d) Compute  $s_i = c_i s + r_i$ .
- (e) Output  $(s_i, R_i)$ .

$I$  obtains  $(s_i, R_i)$  via a secure channel. Here,  $s_i$  is the secret information whereas  $R_i$  is public.

**Key Establishment:** This algorithm establishes a key between a sender (initiator)  $U$  and a receiver (responder)  $I$ . Figure 6.1 describes the steps of the key establishment.

- (a)  $U$  chooses at random  $l \in_R \mathbb{Z}_q^*$  as ephemeral private key and
- computes  $y = ls_u$ ,
  - computes the ephemeral public key as  $L = yP$ ,
  - signs  $L$  together with  $ID_u$  and  $ID_i$  as  $\sigma = \text{Sign}_{s_u}(L, ID_u, ID_i)$ .

Here  $Sign_{s_u}(L, ID_u, ID_i)$  is a signature signed by  $U$  with its private key  $s_u$  and an ID-based signature (IBS) using the same ID-based setup as used by ID-1P-SKE, for instance, the secure BNN-IBS scheme. Computing  $y$  from  $L$  is the Elliptic Curve Discrete Logarithm (ECDL) problem, which is intractable.

- $U \longrightarrow I: \{L, ID_u, ID_i, \sigma\}.$

(c)  $I$  first verifies  $\sigma$  using  $ID_u$ .

- $Verify_{ID_u}(L, ID_u, ID_i, \sigma)$ .

The successful verification implies that the ephemeral public key is actually sent by  $U$ , and hence  $I$  accepts it. Otherwise the protocol is terminated at this stage. Next,  $I$  computes the shared secret  $K_{i,u}$  as

- $K_{i,u} = s_i L$   

$$\left\{ \begin{array}{l} = s_i y P \\ = s_i l s_u P \end{array} \right\}$$

and deletes  $L$ .

(d)  $U$  computes the same shared secret  $K_{u,i}$  as

- $c_i = H(ID_i, R_i)$
- $S_i = c_i P_{PKG} + R_i$
- $K_{u,i} = y S_i$   

$$\left\{ \begin{array}{l} = y s_i P \\ = l s_u s_i P \end{array} \right\}$$

$U$  then deletes  $L$ ,  $l$  and  $y$ .

(e) Both parties then compute the shared session key as

$$SK = \chi(K_{u,i}) = \chi(K_{i,u}) = \chi(l s_u s_i P),$$

where  $\chi$  is the key derivation function defined in Setup.

The session key at this stage is ready to be used for exchanging encrypted data. However, there are chances that at the end of a secure run of a key establishment protocol, the receiver of the last message may not compute the key. Indeed, in any key establishment protocol, the sender of the last message cannot make sure whether or not his last message is received by the other party. He may successfully finish the protocol with a key output. Although the adversary is not able to learn the computed key, the receiver might not receive the sender's last message, and hence might not be able to compute the key. The assurance against this scenario is achieved via an *authenticated key establishment protocol with key confirmation (AKC)*. This

is usually achieved by adding a key confirmation message to an authenticated key establishment protocol after the key has been established. To add this feature to ID-1P-SKE protocol, the *Key Establishment* algorithm proceeds as follows:

(f) After key computation,  $I$  performs the following steps:

- Computes the *XOR* of its computed key  $SK_i$  with  $ID_u$  and  $ID_i$  as follows:  

$$E = (SK_i \oplus ID_u \oplus ID_i).$$
- Encrypts  $E$  with  $SK_i$  using a secure symmetric encryption algorithm, i.e.,  

$$E' = Enc_{SK_i}(E)$$
 and sends  $E'$  to  $U$ .

(g) After  $U$  receives  $E'$ , he performs the following steps:

- Decrypts  $E'$  using his computed key  $SK_u$  to obtain  $E$ , i.e.,  $E = Dec_{SK_u}(E')$ .
- Checks whether  $E \stackrel{?}{=} (SK_u \oplus ID_u \oplus ID_i)$ .

The successful verification implies that both parties have computed a shared session key. Since  $U$  does not expect to receive any message from  $I$  to compute the key, he does not need to send a key confirmation message to  $I$ .

*Comments.* In WSNs, the outside user is the initiator and the sensor node is the responder of the protocol. A sensor node can send the key confirmation message to the user together with the encrypted query results. However, the key confirmation part is optional in WSN environment and can be skipped since after receiving last message from user, the sensor node sends the encrypted query results to the user which provides implicit key confirmation. In addition, key confirmation can be replaced with the time out option. In time out option, a time-out value is set up for the user. The user waits for the feedback from the sensor node until the time-out value expires. If he does not receive any feedback after the time-out value expires, he resends his last message to the sensor node or restarts the protocol. A detailed application of the proposed ID-1P-SKE protocol in WSNs environment together with the four phases of *System Initialization*, *Key Generation*, *User Registration* and *Key Establishment* has been discussed in [YRW11] and given in Appendix B. Other than WSNs, the proposed ID-1P-SKE protocol can be used in any other low power application environment. For instance, RFID tags can use it to authenticate a RFID tag reader and establish a key between the tag and the reader.

## 6.4 Proposed Outside User Authentication Protocol

In the proposed outside user authentication protocol, a user first registers himself to the base station and obtains his private key and other system parameters. After that, whenever he wants to access data from sensor nodes, he sends a signed request to the sensor nodes in his communication range who verify his signed request locally using his ID. If the verification succeeds, the sensor nodes and the user both compute a session key for further communication. This session key enables the user to send encrypted queries, if query privacy is required, to the sensor nodes and obtain confidential data from them. Whether the user query is processed by a single sensor node or a set of sensor nodes is related to the topic of *query processing in wireless sensor networks*, which is not a part of this research. We now present the proposed scheme for user authentication using IBS scheme, which consists of six phases, i.e., *System Initialization*, *Key Generation*, *User Registration*, *User Authentication*, *Session Key Establishment* and *User Revocation*. Like authenticated broadcast by sensor nodes protocol, the first two phases of this scheme are also performed only once before the deployment of the sensor network. The details of the protocol are explained as follows:

**System Initialization:** Again the base station plays the role of a private key generator (PKG) and initializes the system in the proposed user authentication protocol. Let  $SK_{BS}$  be the master secret key of the base station, only known to the base station. The base station calculates the corresponding master public key  $PK_{BS}$  known to everyone. The base station also sets up other public system parameters ( $SP$ ) which include  $PK_{BS}$ .

**Key Generation:** In this phase, the base station calculates the private keys of all sensor nodes corresponding to their IDs using the master secret key  $SK_{BS}$  and other system parameters in the same way as in first authentication scheme. For a sensor node  $I$  with identity  $ID_i$ , the corresponding private key  $D_{ID_i}$  is computed as

$$D_{ID_i} \leftarrow KE(ID_i, SK_{BS})$$

The ID, private key, system parameters and other related information (if any) are stored on individual sensor nodes before the deployment of sensor network. Hence, every sensor node  $I$  stores  $\{ID_i, D_{ID_i}, SP\}$ .

**User Registration:** This phase is performed every time when a new user is added to the system. In this phase, a user  $U$  with identity  $ID_u$  registers with the system. The base station computes his private key  $D_{ID_u}$  as

$$D_{ID_u} \leftarrow KE(ID_u, SK_{BS})$$

The user obtains his private key and other system parameters from the base station through a secure channel. Hence, every user receives  $\{ID_u, D_{ID_u}, SP\}$ . The user also obtains the  $ID$ s and other public information, if any, of the sensor nodes in his communication range.

**User Authentication:** In order to query sensor nodes data, the user sends his signed request to the sensor nodes in his range who verify the legitimacy of the user.

*User Request:* Let  $U$  be the user with identity  $ID_u$  and  $N$  be the number of sensor nodes in his range.  $U$  signs his authentication request message  $RM$  together with the current time stamp  $TS$  using an IBS scheme and his private key  $D_{ID_u}$  as

$$\sigma = \text{Sign}(\langle RM, TS \rangle, D_{ID_u})$$

$U$  sends his authentication request message  $RM$  along with his authentication information which is the time stamp  $TS$ , his  $ID_u$  and the signature  $\sigma$  to  $N$  sensor nodes in his communication range i.e.,

$$\{RM, TS, ID_u, \sigma\}.$$

To sign a user request message, the BNN-IBS scheme or any other secure and efficient IBS scheme can be used here.

*Authentication:* On receiving a user request, a sensor node  $I$  first checks the time stamp  $TS$  to filter out a replayed request message. If it is a fresh one, the sensor node verifies the user  $U$ 's signature using his identity  $ID_u$  and other system parameters stored on it as

$$0/1 \leftarrow \text{Verify}(\langle RM, TS \rangle, ID_u, \sigma, SP)$$

If the verification succeeds, the sensor node proceeds with the session key establishment, otherwise it terminates the protocol and stops further computation and communication at this stage.

**Session Key Establishment:** To provide a secure transmission of data from sensor nodes to a user, a session key needs to be established after successful user

authentication. However, because of a one-pass session key establishment protocol it is possible to carry out the key establishment phase in parallel to the user authentication phase in the proposed outside user authentication protocol. In the proposed user authentication protocol, the only message that needs to be exchanged for the one-pass key establishment is combined with the user request message in user authentication phase.

In user authentication phase,  $U$  also computes his ephemeral key  $EK$  and signs it together with his authentication request message  $RM$  as

$$\sigma = \text{Sign}(\langle RM, TS, EK \rangle, D_{ID_u})$$

$U$  now sends  $\{RM, TS, EK, ID_u, \sigma\}$  to the sensor node  $I$ . If  $U$ 's signature is valid and user authentication succeeds, both  $I$  and  $U$  compute the session key  $SK$  using the key derivation function  $\chi$  as  $SK = \chi(ID_i || ID_u || S_{iu})$ . Here  $S_{iu}$  is a common secret computed by both parties using  $EK$  and their private keys. At this point, the session key  $SK$  is ready for encrypting data. To establish a session key, the proposed ID-based one-pass key establishment protocol i.e., the ID-1P-SKE protocol or any other efficient and secure ID-based one-pass key establishment protocol can be used.

**User Revocation:** User revocation is divided in two cases: revoking a user with expired access time period and revoking a malicious user. These two cases are treated differently. To handle the first case, at the time when the base station calculates the private key for a user  $U$ , the access expiry time  $ET$  of the user is used as a parameter to calculate the private key as follows:

$$D_{ID_u} \leftarrow KE(\langle ID_u, ET \rangle, SK_{BS})$$

In user authentication phase, the user  $U$  sends his access expiry time  $ET$  together with his other authentication information  $\{RM, TS, ID_u, \sigma\}$  to the sensor node. On sensor node's side,  $ID_u$  and  $ET$  are both used to generate the public information corresponding to user's private key in order to verify his signature. It binds the correctness of user's corresponding public information to the correctness of both his  $ID_u$  and  $ET$ . After user's access time period expires, if he sends a signed request together with a fake  $ET$ , it will not pass signature verification since the public information generated from his  $ID_u$  and fake  $ET$  will not correspond to his private key and will be invalid.

In the second case, the base station issues an authenticated revocation list containing the malicious users'  $ID$ s. The sensor nodes store it until the malicious



user's expiry time is passed. Thus, if next time a malicious user attempts to access data from sensor nodes, the sensor nodes reject his request without going through the authentication process. After his access time period  $ET$  expires, he is not able to successfully authenticate himself to the system. In WSNs, the case of malicious users is not very common. Therefore, storing the  $IDs$  of a few malicious users until their access time period expires will impose only a reasonable storage overhead on sensor nodes. To efficiently handle the storage, user's access time period can be kept short so that the sensor nodes do not store malicious users'  $IDs$  for a long time. After that time period only the private keys of the legitimate users can be updated for the next time period. However, the duration of this period can be decided depending on how frequently the event of malicious users occur. Moreover, the base station, as suggested in the first authentication scheme, can periodically update system parameters and secret keys of all legitimate users excluding the malicious ones. However, this option might be costly.

## 6.5 Performance Evaluation

The performance of the proposed protocol is evaluated in two steps. In first step, we evaluate the efficiency of an IBS scheme and the proposed ID-1P-SKE protocol on sensor nodes. In second step, we analyze the performance of the proposed outside user authentication protocol using IBS schemes.

### 6.5.1 Performance of IBS Schemes and Session Key Establishment

In order to evaluate the performance of the proposed outside user authentication protocol, we considered the most efficient pairing-free BNN-IBS scheme and the ID-1P-SKE scheme among the available schemes. Since the sensor nodes are more resource-constrained than the users (more specifically, the devices held by the users), we pay more attention to the efficiency of the protocol on the sensor node side than on the user side. In the proposed protocol, a sensor node only plays the role of a signature verifier to verify the signed user requests. Thus, the main factor to evaluate the user authentication cost is the signature verification cost of the BNN-IBS scheme. We have already adapted a variation of the BNN-IBS scheme (vBNN-IBS) as an IBOOS scheme (B-IBOOS) for the first authentication protocol. The modified B-IBOOS scheme was implemented on MICA2 sensor nodes to obtain the

signature generation and verification costs where the signature generation consisted of the offline phase and the online phase, as explained in Section 5.4.1.6. Combining the computation costs of the offline and the online phases of B-IBOOS scheme gives the total computation cost of the signature generation for BNN-IBS scheme. The signature verification cost of BNN-IBS scheme is the same as in B-IBOOS scheme. The memory cost of BNN-IBS scheme will, however, decrease in the case of user authentication protocol because the sensor nodes do not store the code of the signature generation algorithm of BNN-IBS scheme now. They only need to store the code of the signature verification algorithm to verify the signed user requests.

The performance of two different implementations of B-IBOOS scheme has already been discussed in detail in Section 5.4.1.6. Since the sensor nodes only need to store the signature verification code, the memory consumption is not an issue in this case. Therefore, the first implementation of B-IBOOS scheme is considered here to evaluate the proposed protocol.

### 6.5.1.1 Computation Cost

Using the results from Table 5.3, the Table 6.1 gives the computation cost of the BNN-IBS scheme.

	<b>Time (s)</b>	<b>Energy (mWs)</b>
<b>Sign</b>	0.32	9.59
<b>Verify</b>	1.044	31.33

Table 6.1: Time and Energy Consumption of BNN-IBS Scheme

The establishment of the session key on sensor nodes side using the ID-1P-SKE protocol adds only one point multiplication in computation cost. As discussed in preceding chapter, one point multiplication computation takes only 0.295s on MICA2 sensor node. Table 6.2 gives the computation cost of the key establishment on sensor nodes side using the ID-1P-SKE protocol.

	<b>Time (s)</b>	<b>Energy (mWs)</b>
<b>ID-1P-SKE</b>	0.295	8.85

Table 6.2: Time and Energy Consumption of ID-1P-SKE Scheme

Hence, the total computation cost of the user authentication (the BNN-IBS signature verification cost) and the key establishment (ID-1P-SKE cost) on sensor

nodes is given by the Table 6.3. To the best of our knowledge, the BNN-IBS scheme and our proposed ID-1P-SKE scheme are the most efficient available IBS and the one-pass session key establishment schemes which enable a resource constrained sensor node to authenticate an outside user and establish a session key with him only in 1.339s without incurring any storage overhead.

	Time (s)	Energy (mWs)
<b>User Authentication</b>	1.044	31.33
<b>Key Establishment</b>	0.295	8.85
<b>Total</b>	1.339	40.18

Table 6.3: Total Time and Energy Consumption of User Authentication and Session Key Establishment

#### 6.5.1.2 Communication Cost

In the proposed user authentication and session key establishment protocol, only one message is exchanged to authenticate a user and establish a key with him. The only message exchanged contains the signature and the ephemeral public key of the user. Using the same security level and curve parameters as used in the evaluation of B-IBOOS scheme, the size of BNN-IBS signature will be  $(2 \times (2 \times 163) + 160)$  bits) 812 bits whereas the size of the ephemeral public key will be  $(2 \times 163)$  bits) 326 bits, a group element of the form  $(x, y)$ . Hence, the total message size will be about 1154 bits or 144 bytes including IDs (16 bits). However, it reduces to 124 bytes if vBNN-IBS scheme is used here instead of BNN-IBS scheme.

### 6.5.2 Performance of the Proposed Protocol

We now evaluate the performance of the proposed outside user authentication protocol using IBS schemes. To the best of our knowledge, the proposed protocol is the first solution to the problem which does not only address the user authentication but also provides the session key establishment feature. The proposed protocol uses the ID-based signature schemes to authenticate users, ensuring security with efficiency. The proposed outside user authentication protocol using the IBS schemes does not only meet the design challenges highlighted in the beginning of this chapter but also provides some extra features as follows:

### 6.5.2.1 Distributed User Authentication

The proposed protocol adopts a distributed approach to authenticate outside users. It means the legitimacy of a user is verified locally by the sensor node who receives the user request rather than a third party. Hence, the proposed protocol avoids the in-network traffic congestion, the communication overhead and authentication delay of the centralized user authentication schemes.

### 6.5.2.2 Storage Efficiency

The sensor nodes store the users' passwords in password based authentication schemes and the users' IDs and public key pairs in digital signature based authentication schemes, of all users. In the proposed protocol, the sensor nodes do not need to store any user specific authentication information for the verification of user authentication request. The sensor nodes preloaded with their own public and private information and the system parameters can verify the legitimacy of any outside user of the sensor nodes data. Therefore, the proposed protocol achieves the goal of storage efficiency.

### 6.5.2.3 Computation Efficiency

In the proposed protocol, the ephemeral public key of the user is sent together with the signed user authentication request. Only one signature verification is required to authenticate a user and the ephemeral public key of the user. There is no need to authenticate the ephemeral public key separately in order to avoid accepting a fake ephemeral public key. Hence, the user authentication and session key establishment together attains computation efficiency.

### 6.5.2.4 Communication Efficiency

Only a single message is exchanged to authenticate a user and establish a key with him. Thus, the proposed protocol reduces the communication overhead. Moreover, the ID-based signature schemes do not require an outside user to send a public key or a certificate with a user request message which further reduces the communication overhead without increasing storage overhead.

#### 6.5.2.5 Multiple Users

Like in the case of proposed authenticated broadcast by sensor nodes protocol, the ID-based signature schemes handle the problem of public keys and certificates management in the proposed user authentication protocol. Therefore, the proposed protocol using the ID-based signatures facilitates the maximum number of outside users of the sensor nodes data.

#### 6.5.2.6 Scalability

New users as well as new sensor nodes can be added to the WSN easily at any time. New users simply need to register themselves to the base station and obtain the required information whereas new sensor nodes are preloaded with the required information. Having the required information (ID, private key and system parameters), a new user can send a signed user authentication request to any sensor node in the network whereas a new sensor node can verify the signed user request message sent by any outside user of the sensor nodes data. The sensor nodes do not need to store any user specific information of every outside user. It removes the restriction on the number of outside users. Therefore, the proposed protocol meets the scalability challenge.

#### 6.5.2.7 Session Key Establishment

The proposed protocol also features an efficient session key establishment where only one message is being transmitted and processed to authenticate a user and establish a key, achieving both the communication and the computation efficiency.

### 6.6 Security Analysis

This section first reviews the security of the IBS schemes i.e., the BNN-IBS scheme, and then analyzes the security of the proposed session key establishment protocol i.e., ID-1P-SKE protocol. In the end, the security of our outside user authentication protocol using the IBS scheme is discussed.

#### 6.6.1 Security of the IBS Schemes

We consider the security of the BNN-IBS scheme, the most efficient IBS scheme. The BNN-IBS scheme has been shown to be *existentially unforgeable against the*

*chosen message and ID attacks* (i.e., **euf-cma-ida** secure) in [BNN04] and does not need another security proof. Therefore, we do not analyze here the formal security of the BNN-IBS scheme.

## 6.6.2 Security of the Session Key Establishment

In session key establishment, we analyze the security of the proposed ID-based one-pass authenticated key establishment (ID-1P-SKE) protocol which has been used to evaluate the outside user authentication protocol. The security of the ID-1P-SKE protocol has been analyzed using the reductionist proof techniques.

### 6.6.2.1 Desirable Security Properties

Informally, a one-pass key establishment protocol is desired to achieve the following security properties identified by [BWJM97] and improved by [OTO05].

**Unknown Key-Share.** An entity  $A$  cannot be coerced into sharing a key with an entity  $B$  when  $A$  believes that the key is shared with some other entity  $C$ .

**Known Session Keys.** The knowledge of the previous session keys does not enable an adversary to compromise other session keys.

**Key Control.** Neither of the participating parties should be able to force the session key to be a pre-selected value.

**Key Compromise Impersonation.** The compromise of an entity  $A$ 's long-term private key allows an adversary to impersonate  $A$  but it should not allow adversary to impersonate other entities in the presence of  $A$ .

**Forward Secrecy.** If the long-term private keys of one or more entities are compromised, it should not affect the secrecy of previously established session keys.

According to [OTO05], one-pass key establishment protocols cannot achieve the two security properties identified by [BWJM97], namely *Key Compromise Impersonation* and *Forward Secrecy*. The reason is that an adversary  $\mathcal{A}_d$ , who has compromised a receiver  $B$ 's long-term private key and eavesdropped one message from  $A$ , will be able to compute the session key.  $\mathcal{A}_d$  will then be able to impersonate (only)  $A$  to  $B$ , launching *Key Compromise Impersonation* attack. The adversary can also reveal the previously established session keys in the same way, compromising *Forward Secrecy*. Therefore, [OTO05] redefined these two security properties for

one-pass key establishment protocols as *Sender's Key Compromise Impersonation* and *Sender's Forward Secrecy*.

***Sender's Key Compromise Impersonation.*** The compromise of a sender  $A$ 's long-term private key allows an adversary to impersonate  $A$  but not other entities in the presence of  $A$ .

***Sender's Forward Secrecy.*** The compromise of a sender's long-term private key does not affect the secrecy of previously established session keys.

### 6.6.2.2 Security Model

We now describe the security model, introduced by [GBGN08], for ID-based one-pass authenticated key establishment protocols. This model, say (ID-eCK) model, is in fact the extension of extended Canetti-Krawczyk (*eCK*) [LLM07] model for two-pass AKE protocols to the ID-based one-pass AKE settings. The formal definition of the (ID-eCK) model is based on the following game involving a challenger  $\mathcal{C}$  and an adversary  $\mathcal{A}_d$ :

A protocol  $\pi$  is modeled as a collection of parties (probabilistic polynomial time (PPT) Turing machines) running the proposed AKE protocol, representing a real time scenario. All the communication among the parties is controlled by a PPT adversary  $\mathcal{A}_d$ . For any arbitrary identities,  $\mathcal{A}_d$  can also obtain corresponding private keys from the PKG to add fictitious parties. All of these parties, including both the honest parties and the fictitious parties, are activated by  $\mathcal{A}_d$  which can run multiple instances of the protocol at each party. A particular instance of the AKE protocol executed by a party is called a *session*. A legitimate instance of an AKE protocol between two parties consists of two sessions; a *session* ( $sid$ ) initiated at initiator and a *matching session* ( $sid^*$ ) at responder, where ( $sid$ ) and ( $sid^*$ ) are the session ids. A *matching session*<sup>1</sup> may not exist, if the communication sent by the responder is corrupted by the attacker  $\mathcal{A}_d$ . The attacker  $\mathcal{A}_d$  is allowed to make the following queries during the game which are answered by the challenger  $\mathcal{C}$ :

- **Send( $ID_i, ID_j, msg$ ):** Sends a message  $msg$  to  $ID_i$  on behalf of  $ID_j$ . The response from  $ID_i$  is returned to  $\mathcal{A}_d$ . An empty message  $\lambda$  activates  $ID_i$  as an *initiator*. A non-empty message makes the role of  $ID_i$  to be that of a *responder*.

<sup>1</sup>In case of one-pass protocol, there is only one message transmitted during the protocol, i.e., from the initiator. The initiator takes the empty string  $\lambda$  as input and transmits the message  $msg$  and the responder takes  $msg$  as input and transmits  $\lambda$ .

For one-pass AKE the session identifier is the tuple  $(ID_i, ID_j, msg, role)$ , where  $role$  is either *initiator* or *responder*.

- **Long-Term Key Reveal** $(ID_i)$ : Reveals the long-term private key of an honest party  $ID_i$ .
- **Ephemeral Key Reveal** $(sid)$ : Reveals the ephemeral key of possibly incomplete session  $sid$ .
- **Reveal** $(sid)$ : Reveals the session key of the session  $sid$  which is a completed session.
- **Extract** $(ID_i)$ : Returns a long-term private key corresponding to any arbitrary identity  $ID_i$  selected by adversary to add a dishonest party.

The **Long-Term Key Reveal** query and the **Extract** query both reveal the long-term private keys. However, the former returns the long-term private key of an honest party  $ID_i$  whereas the latter returns the the long-term private key of a dishonest party  $ID_i$  which allows adversary to add fictitious party corresponding to its chosen  $ID$  i.e.,  $ID_i$ . In addition to the above queries,  $\mathcal{A}_d$  is allowed, at any later stage, to make a **Test** $(sid)$  query to a completed session. In response to this query, a challenge value  $\alpha$  is given to  $\mathcal{A}_d$ .

- **Test** $(sid)$ : The challenger flips a fair coin  $b \in_R \{0, 1\}$ . If  $b = 1$ , it returns the session key held in the session  $sid$  otherwise it picks a random value sampled from the session key distribution space  $\{0, 1\}^k$  (where  $k$  is the security parameter) and returns it to  $\mathcal{A}_d$ .

The **Test** query may be made only once. The  $\mathcal{A}_d$ 's job is now to guess  $b$ .  $\mathcal{A}_d$  continues the experiment after the **Test** query. It outputs its guess  $b'$  in an attempt to distinguish the session key from the random string. The experiment terminates as soon as  $\mathcal{A}_d$  outputs its guess by calling the **Guess** $(b')$  query.

- **Guess** $(b')$ : If  $b' = b$  returns 1, otherwise 0.

$\mathcal{A}_d$  wins the game, if the selected test session is *clean* and he guesses the challenge value correctly, i.e.,  $b' = b$ .

**Clean Session.** Let  $sid$  be a session activated at  $ID_i$  and  $sid^*$  be the matching session activated at  $ID_j$  ( $sid^*$  may not exist). Let  $sk_i$  and  $sk_j$  denote the long-term



private keys of  $ID_i$  and  $ID_j$  respectively. Let  $esk_i$  and  $esk_j$  be the ephemeral private keys of  $ID_i$  and  $ID_j$  in sessions  $sid$  and  $sid^*$  respectively. A session  $sid$  is not clean, if any of the following conditions holds:

- $\mathcal{A}_d$  reveals the master secret key of the  $PKG$ .
- $ID_i$  or  $ID_j$  is a dishonest party whose private key is obtained by the adversary via **Extract** query.
- $\mathcal{A}_d$  reveals the session key of  $sid$  or  $sid^*$  (if  $sid^*$  exists).
- A matching session  $sid^*$  exists and
  - If  $sid$  is an initiator session,  $\mathcal{A}_d$  reveals both  $sk_i$  and  $esk_i$  or it reveals  $sk_j$ .
  - If  $sid$  is a responder session,  $\mathcal{A}_d$  reveals  $sk_i$  or both  $sk_j$  and  $esk_j$ .
- A matching session  $sid^*$  does not exist and
  - $\mathcal{A}_d$  reveals  $sk_i$  or  $sk_j$ .

In all other cases, the session is considered as *clean*. In order to win the game, the adversary  $\mathcal{A}_d$  must keep the session clean until the end of the experiment. The advantage of adversary  $\mathcal{A}_d$  in distinguishing the real session key from a random value is defined as

$$Adv_{\pi}^{\mathcal{A}_d} = \Pr[Suc_{\mathcal{A}_d}] - \frac{1}{2},$$

where  $\Pr[Suc_{\mathcal{A}_d}]$  is the success probability of  $\mathcal{A}_d$ . Informally, we say that an ID-based one-pass AKE protocol is secure if no adversary can learn anything about a session key held by two uncorrupted entities  $ID_i$  and  $ID_j$ . The security of an ID-based one-pass AKE protocol, in the above model, is formally defined as follows:

**Definition 6.1. (ID-eCK Security).** *An ID-based one-pass AKE protocol is (ID-eCK)-secure if the following two conditions hold.*

- *If two honest parties complete matching sessions, they compute the same session key.*
- *For any PPT adversary  $\mathcal{A}_d$ ,  $Adv_{\pi}^{\mathcal{A}_d}$  is negligible.*

### 6.6.2.3 Security Analysis of ID-1P-SKE Protocol

The security analysis of the proposed ID-1P-SKE protocol includes the formal security proof using the ID-eCK security model and a discussion about how the security properties listed in Section 6.6.2.1 are achieved by the ID-1P-SKE protocol.

#### a) Security Proof

The security of the proposed ID-1P-SKE protocol is reduced to the hardness of Computational Diffie-Hellman (CDH) problem. We proceed to show that the proposed ID-1P-SKE protocol is secure in the random oracle model assuming that the CDH problem is hard in  $\mathbb{G}$ .

**Theorem 1.** *Suppose that CDH assumption holds in  $\mathbb{G}$ . Then the proposed ID-1P-SKE protocol is (ID-eCK)-secure (in the sense of Definition 6.1) with  $\chi$  and  $H$  modeled as random oracles.*

**Proof.** If two honest and uncorrupted parties  $ID_i$  and  $ID_j$  complete matching sessions successfully, then  $ID_j$  must have received  $ID_i$ 's ephemeral public key. So both parties establish the same session key as shown in Figure 6.1. Thus, the first condition of Definition 6.1 holds. To prove the second condition, let us assume by contradiction that there exists a PPT adversary  $\mathcal{A}_d$  with running time  $t(k)$ , who has a non-negligible advantage  $\epsilon$  against the ID-1P-SKE protocol. Then we need to construct an algorithm  $\mathcal{C}$  to solve the CDH problem which uses  $\mathcal{A}_d$  as a subroutine. Since  $\chi$  is modeled as a random oracle, after the **Test** query,  $\mathcal{A}_d$  can distinguish a session key from a random string only in three ways:

1. **Guessing Attack.**  $\mathcal{A}_d$  correctly guesses the session key.
2. **Key Replication Attack.**  $\mathcal{A}_d$  creates a session, not matching to the test session, but that has the same session key as the test session.  $\mathcal{A}_d$  then learns the test session key by querying the session that has the same key.
3. **Forging Attack.**  $\mathcal{A}_d$  computes  $(ls_i s_j P)$  used in the test session and queries  $\chi$  to derive the session key.

Since  $\chi$  is a random oracle, the probability of guessing the output of  $\chi$  is  $\mathcal{O}(1/2^k)$ , which is negligible. Since two non-matching sessions have different communicating parties and ephemeral keys information, the key replication attack is equivalent to finding a collision for  $\chi$  which has a negligible probability. Therefore, the probability

of key replication attack is also negligible. Thus, the first two ways can be ruled out. The rest of the section will analyze the forging attack.

Let  $\langle P, aP, bP \rangle$  be the CDH instance given to the CDH challenger  $\mathcal{C}$ .  $\mathcal{C}$ 's goal is to compute  $abP$ .  $\mathcal{C}$  executes the ID-1P-SKE protocol with the adversary  $\mathcal{A}_d$  and modifies the data returned by the honest parties in a way that if  $\mathcal{A}_d$  breaks the security of the proposed ID-1P-SKE protocol,  $\mathcal{C}$  can solve the CDH problem. To execute the simulated ID-1P-SKE protocol,  $\mathcal{C}$  picks  $s \in_R \mathbb{Z}_q^*$  uniformly as master secret key ( $msk$ ), computes the master public key ( $mpk$ )  $P_{PKG} = sP$  and gives  $P_{PKG}$  to  $\mathcal{A}_d$ .  $\mathcal{C}$  randomly picks a set of identities  $\{ID_1, ID_2, \dots, ID_n\}$  for  $n$  users. Assume  $\mathcal{A}_d$  activates each of these users  $m$  times at most ( $n$  and  $m$  both are polynomials in security parameter  $k$ ).  $\mathcal{C}$  randomly selects  $t_s$  as the test session which  $\mathcal{A}_d$  will chose to perform the **Test** query. Similarly,  $\mathcal{C}$  picks the parties  $ID_A$  and  $ID_B$  randomly from the set  $\{ID_1, ID_2, \dots, ID_n\}$ .  $\mathcal{C}$  will be correct in its guess, if  $\mathcal{A}_d$  chooses  $t_s$  session activated at  $ID_A$  with  $ID_B$  as the test session.  $\mathcal{C}$  replies to queries of  $\mathcal{A}_d$  as follows:

**S Query.**  $\mathcal{C}$  maintains a list  $L_S$ . On input  $ID_i$ , it first checks the list to make sure if there is a value corresponding to  $ID_i$ . If there is a value in the list, it returns that value, i.e.,  $S_i$  and  $R_i$ , to  $\mathcal{A}_d$ . Otherwise it picks two numbers  $s_i, c_i \in_R \mathbb{Z}_q^*$ , computes  $S_i = s_iP$  and  $R_i = S_i - c_iP_{PKG}$  and returns them.  $\mathcal{C}$  stores the tuple  $(ID_i, R_i, S_i, c_i, s_i)$  in  $L_S$ . For  $ID_B$ , it sets  $S_B = bP$  and  $R_B = S_B - c_BP_{PKG}$ , where  $c_B \in_R \mathbb{Z}_q^*$ , and stores the tuple  $(ID_B, R_B, S_B, c_B, \perp)$  in  $L_S$ .

**H Query.**  $\mathcal{C}$  checks the list  $L_S$  and returns the value of  $c_i$  corresponding to the input of  $ID_i$  and  $R_i$ .

**$\chi$  Query.**  $\mathcal{C}$  similarly maintains another list  $L_\chi$ . On a query, it randomly picks a value from the session key distribution  $\{0, 1\}^k$  and returns it to  $\mathcal{A}_d$ . The input and the output both are stored in the list  $L_\chi$ .

**Send**( $ID_i, ID_j, msg$ ). This query is handled as follows:

1. The *role* of the  $ID_i$  is initiator, i.e.,  $msg = \lambda$ .
  - If  $i = A, j = B$  and the session is  $t_s$ ,  $\mathcal{C}$  selects  $l_a \in_R \mathbb{Z}_q^*$ , retrieves  $s_a$  from  $L_S$ , computes  $L_a = l_a s_a(aP)$ <sup>1</sup> and returns  $L_a$  to  $\mathcal{A}_d$ .
  - If  $i = B$ ,  $\mathcal{C}$  randomly selects  $\beta \in \mathbb{Z}_q^*$ , computes  $L_b = \beta P$  and returns  $L_b$  to  $\mathcal{A}_d$ .  $\mathcal{C}$  stores  $\beta$  in the list maintained for session  $sid$ .

---

<sup>1</sup>Note that  $(l_a a)$  produces another random number.

- For all other cases,  $\mathcal{C}$  chooses  $l_i \in_R \mathbb{Z}_q^*$  and returns  $L_i = l_i s_i P$ .
2. The *role* of the  $ID_i$  is responder, i.e.,  $msg \neq \lambda$ .
- $\mathcal{C}$  accepts the session marking it as completed.

**Extract/Long-Term Key Reveal**( $ID_i$ ).  $\mathcal{C}$  first checks the list  $L_S$  to see if there is an entry corresponding to  $ID_i$ . If there is, it retrieves  $s_i$  from the list and returns it to  $\mathcal{A}_d$ . Otherwise, it makes  $S$  query, retrieves  $s_i$  from  $L_S$  and returns it to  $\mathcal{A}_d$ . In case of *Extract* query, it also returns the corresponding  $R_i$ . On *Extract* or *Long-Term Key Reveal* query with input  $ID_B$ ,  $\mathcal{C}$  aborts its execution.

**Ephemeral Key Reveal**( $sid$ ). If this is the  $t_s$  session between  $ID_A$  and  $ID_B$ ,  $\mathcal{C}$  outputs “fail”. The *Ephemeral Key Reveal* query is not handled at  $ID_B$  for all sessions. For all other cases,  $\mathcal{C}$  returns the ephemeral private key selected while answering the *Send* query.

**Reveal**( $sid$ ). This query is handled as follows:

- If this is the  $t_s$  session between  $ID_A$  and  $ID_B$ ,  $\mathcal{C}$  aborts its simulation.
- If this session is at  $ID_B$ ,  $\mathcal{C}$  retrieves  $\beta$  and returns the value of  $\chi(\beta S_j)$ .
- For all other cases,  $\mathcal{C}$  returns the session key with its knowledge of private and ephemeral private keys.

**Test**( $sid$ ). The *Test* query is replied as follows:

- If this is not the  $t_s$  session between  $ID_A$  and  $ID_B$ ,  $\mathcal{C}$  outputs “fail”.
- If  $t_s$  is not a clean session,  $\mathcal{C}$  aborts its simulation.
- Otherwise  $\mathcal{C}$  has to return the session key for session  $sid$  or a random value from the session key distribution after tossing a coin. The session key between  $ID_A$  and  $ID_B$  would be of the following form:

$$SK_{AB} = \chi(l_a s_a a S_B) = \chi(l_a s_a a b P),$$

which requires solving the CDH instance  $\langle P, aP, bP \rangle$ . Hence,  $\mathcal{C}$  returns a random value from the key distribution.

**Solving the CDH Problem.** If  $\mathcal{A}_d$  can distinguish a real session key from the given random value with a non-negligible probability, then it must have issued a  $\chi$  query to compute session key with the input  $(l_a s_a abP)$ . Now  $\mathcal{C}$  can answer the CDH challenge with  $(l_a s_a abP)/(l_a s_a) = abp$ . If  $\mathcal{A}_d$ 's guess is right, then  $\mathcal{C}$ 's answer is correct.

**Time Analysis.** If  $\mathcal{A}_d$  with running time  $t$ , asking  $q_\chi, q_H, q_S, q_{send}, q_{ex}, q_s, q_{ep}$  and  $q_r$  queries (all polynomials in  $k$ ) to  $\chi, H, S, Send, Extract, Long-Term Key Reveal, Ephemeral Key Reveal$  and  $Reveal$  oracles respectively, has non-negligible advantage in breaking the security of ID-1P-SKE scheme, then the CDH problem can be solved in running time  $t' \leq (t + (2q_S + q_{send} + q_r)t_m)$ , where  $t_m$  is the time to compute a scalar multiplication in  $\mathbb{G}$  and is a constant. A  $q_S$  query computes two scalar multiplications whereas  $q_{send}$  and  $q_r$  both queries compute one scalar multiplication each.

**Probability Analysis.** Similar to the probability analysis given in [GBGN08],  $\mathcal{C}$ 's probability of success involves three parts:

*Firstly*,  $\mathcal{C}$  outputs “fail”, if  $\mathcal{A}_d$  issues an *Ephemeral Key Reveal* query at  $ID_B$ . As there are  $m(n-1)$  maximum possible sessions at  $ID_B$  (including the test session), the probability of  $\mathcal{A}_d$  of not asking this query is  $Pr[I] = 1 - \frac{m(n-1)}{mn(n-1)} = \frac{n-1}{n}$ .

*Secondly*,  $\mathcal{C}$  outputs “fail”, if  $\mathcal{A}_d$  does not choose  $t_s$  as test session, foreseen by  $\mathcal{C}$ . The probability of not occurring of this event is  $Pr[II] = \frac{1}{mn(n-1)}$ .

*Thirdly*, the success probability of  $\mathcal{A}_d$  making the right guess without asking the  $\chi$  query is  $Pr[III] \leq \frac{1}{2}$ .

Let  $Pr[\chi]$  be the probability that  $\mathcal{A}_d$  asks the  $\chi$  query. The success probability of  $\mathcal{A}_d$  is then

$$\begin{aligned} Pr[Suc_{\mathcal{A}_d}] &= Pr[Suc_{\mathcal{A}_d}|\chi]Pr[\chi] + Pr[III] \\ &\leq Pr[Suc_{\mathcal{A}_d}|\chi]Pr[\chi] + \frac{1}{2} \\ &\leq Pr[\chi] + \frac{1}{2} \end{aligned}$$

hence,

$$Pr[\chi] \geq Pr[Suc_{\mathcal{A}_d}] - \frac{1}{2}$$

Let  $Suc_{\mathcal{C}}$  be the probability that  $\mathcal{C}$  is able to solve the CDH problem which is only possible if  $Pr[\chi] \neq 0$ . If  $Pr[\chi] \neq 0$  and  $q_\chi$  is the number of  $\chi$  queries made by  $\mathcal{A}_d$ , then the probability that  $\mathcal{C}$  picks the correct  $\kappa$  from the list  $L_\chi$  to solve CDH problem is given as  $Pr[IV] = \frac{1}{q_\chi}$ .

The success probability of  $\mathcal{C}$  when  $\mathcal{A}_d$  asks  $\chi$  query is then given as

$$\begin{aligned} \Pr[Suc_{\mathcal{C}}|\chi] &= \Pr[I \cap II \cap IV] \\ &= \frac{n-1}{n} \frac{1}{mn(n-1)} \frac{1}{q_{\chi}} \\ &= \frac{1}{mn^2 q_{\chi}} \end{aligned}$$

The success probability of  $\mathcal{C}$  in solving the CDH problem is then given as

$$\begin{aligned} \Pr[Suc_{\mathcal{C}}] &= \Pr[Suc_{\mathcal{C}}|\chi] \Pr[\chi] \\ &\geq \frac{1}{mn^2 q_{\chi}} \left( \Pr[Suc_{\mathcal{A}_d}] - \frac{1}{2} \right) \end{aligned}$$

As described earlier, the advantage  $Adv_{\pi}^{\mathcal{A}_d}$  of  $\mathcal{A}_d$  in distinguishing the real session key from a random string is  $\epsilon$  which is  $(\Pr[Suc_{\mathcal{A}_d}] - \frac{1}{2})$  (by the ID-eCK model). Therefore,

$$\Pr[Suc_{\mathcal{C}}] \geq \frac{\epsilon}{mn^2 q_{\chi}}$$

Since  $\epsilon$  is non-negligible, it means  $\mathcal{C}$  can solve CDH problem with non-negligible probability; a contradiction to the CDH assumption. Hence, our assumption that there exists a PPT adversary  $\mathcal{A}_d$  with a non-negligible advantage  $\epsilon$  against the ID-1P-SKE protocol is wrong and we can conclude by contradiction.

## b) Security Properties of ID-1P-SKE Protocol

We now discuss how the informal security properties listed in Section 6.6.2.1 are satisfied by the ID-1P-SKE protocol. The ID-eCK model ensures the security properties of *Unknown Key Share*, *Sender's Forward Secrecy* and *Known Session Key* while it does not handle the two security attributes of *Key Control* and *Sender's Key Compromise Impersonation*.

**Unknown Key Share.** If a protocol is (ID-eCK)-secure, then it is resilient against the *Unknown Key-Share* attack. For instance, if  $A$  establishes a session key with  $D$  while he believes that he has established a key with  $B$  then there is a session  $t_s^*$ , between  $ID_A$  and  $ID_D$ , other than the test session  $t_s$  between  $ID_A$  and  $ID_B$ , which also holds  $SK_{AB}$ . It means that there are two different sessions which hold the same session key  $SK_{AB}$ . Since  $t_s$  and  $t_s^*$  are not matching sessions,  $\mathcal{A}_d$  can make the **Reveal**( $t_s^*$ ) query to learn  $SK_{AB}$  before the **Test**( $t_s$ ) query and can break the security of the protocol with non-negligible probability. Moreover, if  $\mathcal{A}_d$  learns  $SK_{AB}$  through unknown key share attack with non-negligible probability, then the

key replication attack is also possible with non-negligible probability because there are two different sessions with the same session key. Informally, the ephemeral public key signed with  $A$ 's private key assures  $B$  that the key is established with  $A$ . The value of  $S_B$ , computed from  $B$ 's public information, assures  $A$  that the key is established with  $B$ .

**Known Session Key.** In ID-eCK model, the *Known Session Key* attack is captured by allowing adversary to reveal the session key of any session other than the test session. Each session has a different ephemeral key which is uniformly chosen from  $\mathbb{Z}_q^*$  at random which makes the session key of each session computationally independent. If  $\mathcal{A}_d$  reveals any previous session key via **Reveal** query that key would be of the form  $\chi(l_i s_i s_j P)$  which is fully computationally independent of the key  $\chi(l_a s_a abP)$  for the test session. Due to the computational independence of session keys for each session, from the knowledge of one session key nothing can be implied about the value of the other session keys.

**Remark 1.** *If the attacker learns a previous session key between  $A$  and  $B$  (ID-eCK model does not allow it) and replays the corresponding message from  $A$  (including  $A$ 's signature) to  $B$  for that session, he would be able to impersonate  $A$  to  $B$  as  $B$  cannot differentiate a replayed message. This attack (faced in all one-pass protocols) can be handled by including time-stamp in the message [CBHVS09].*

**Sender's Forward Secrecy.** This security attribute is reflected in the ID-eCK model by allowing the adversary to reveal the long-term private key of the sender in test session. Adversary is also allowed to reveal the ephemeral key of the sender in ID-eCK model. However, if the adversary reveals both the long-term private key as well as the ephemeral key of the sender in a session, he can reconstruct the shared secret and can compute the session key. Hence, ID-eCK model excludes this combination for the test session. Informally, if  $\mathcal{A}_d$  reveals the sender  $A$ 's long-term private key  $s_a$  through **Long-Term Key Reveal** query, eavesdrops the ephemeral public key  $L_a$  and obtains  $S_B$  from  $\mathcal{C}$ ,  $\mathcal{A}_d$  still faces the CDH problem of computing  $(l_a s_a abP)$  from  $s_a$ ,  $L_a$  and  $S_B$ , where  $L_a = l_a s_a (aP)$  and  $S_B = bP$ .

**Key Control.** This property is not captured by the ID-eCK model. However, informally we see that the receiver  $B$  cannot control the session key  $SK_{AB}$  since the ephemeral public key  $L_a$  is computed by the sender  $A$ .  $A$ , too, cannot control the session key since it is computationally impossible to find a  $l_a \in \mathbb{Z}_q^*$  for a preselected value of shared secret  $K_{AB}$ . However, it is possible for  $A$  to influence the value of the session key which is unavoidable in all one-pass key establishment protocols.

**Sender's Key Compromise Impersonation.** The sender's key compromise impersonation (S-KCI) resilience (not implied by (ID-eCK)-security) is related to authentication. The S-KCI attack succeeds for majority of the one-pass AKE schemes due to the fact that they do not include a sender verification mechanism [CBHVS09]. In ID-1P-SKE protocol, the ephemeral public key is signed by the sender  $A$  providing sender verification. If  $\mathcal{A}_d$  has compromised the sender  $A$ 's long-term private key, he can sign ephemeral public key on behalf of  $A$  and impersonate  $A$  to other entities. However, he cannot sign the ephemeral public key on behalf of any other entity (other than  $A$ ), when it has compromised  $A$ 's long-term private key. Hence, it makes impossible for the adversary to impersonate other entities to  $A$ . Consequently, the ID-1P-SKE protocol achieves S-KCI resilience. Including the responder's  $ID$  in the signed message further avoids the possibility of an attacker re-using  $A$ 's signature from a protocol run between  $A$  and a different entity.

### 6.6.3 Security of the Proposed Protocol

#### 6.6.3.1 Security Properties Achieved

We now explain how the security properties listed in Section 4.3 are achieved by the proposed outside user authentication protocol.

**Authentication.** The proposed protocol provides the required authentication, i.e., the proof a user's legitimacy. The required authentication is achieved via the signed user request which is signed by a user using his private key and a secure IBS scheme, giving a proof of the user's identity. This proof of user's identity is validated by verifying the signed user's request message by the sensor nodes. The successful signature verification implies that the request sender is the actual legitimate user of the sensor nodes data. The successful signature verification also yields the fact that the ephemeral public key is actually sent by the claimed user and the session key is actually established with a legitimate user. It helps to avoid the man-in-the-middle attack faced in typical Diffie-Hellman style key establishment protocols.

**Message Integrity.** The proposed protocol using IBS schemes ensures message integrity since any changes made in the contents of a user request message can be detected through the signature verification. Any modified user request message will not pass the signature verification process.

**Verification.** The proposed distributed user authentication protocol enables every sensor node in the sensor network to verify the legitimacy of any outside



user of the sensor nodes data efficiently. The sensor nodes only need the user's ID information sent along with the user request message to verify the signed request message. The sensor nodes do not need any other user specific information, for instance passwords or public keys, in order to be able to verify a signed user request. Moreover, the experimental results of BNN-IBS scheme proved the fact that it is possible for the resource constrained sensor nodes to verify a message signed using an efficient IBS scheme.

**Freshness.** Every user request contains a timestamp and only the request messages with the fresh timestamps are verified and accepted by the sensor nodes. Thus, the proposed protocol provides freshness.

**Confidentiality.** The proposed protocol does not only provide user authentication but also preserves the confidentiality of the sensor nodes data in transit. The session key established between a sensor node and a user after successful user authentication, establishes a secure communication channel between the user and the sensor node. This helps to securely transfer the sensor nodes data only to legitimate users by hiding it from the eavesdroppers.

**Availability.** In centralized user authentication schemes, a third party providing authentication represents a single point of failure. If the third party fails, the whole protocol collapses. In the proposed user authentication protocol, even if a few sensor nodes fail, the user requests are still processed by the other sensor nodes.

### 6.6.3.2 Countermeasures to Security Attacks

This section discusses how the security attacks listed in Section 4.4 are countered by the proposed user authentication and session key establishment protocol.

1. *Impersonation Attack.* The proposed user authentication protocol uses the secure IBS schemes providing strong authentication and message integrity. The secure IBS schemes make it impossible for an illegitimate user to sign a user request message on behalf of a legitimate user or modify a valid user request message sent by a legitimate user. The reason is that it is hard to compute a valid message signature pair without the knowledge of the legitimate user's private key in the presence of an existentially unforgeable signature scheme. Hence, it is hard for the attacker to impersonate a legitimate user.
2. *DoS Attack.* The proposed user authentication protocol adopts a distributed approach to authenticate outside users where a user's legitimacy is verified

locally by the sensor nodes. This local verification prevents the traffic congestion and the DoS attack usually faced in centralized user authentication schemes.

3. *Message Replay Attack.* The proposed protocol uses timestamps which provide freshness and ultimately resist message replay attacks. The sensor node first checks the time stamp before signature verification to avoid the verification of a replayed user request message from a previous legitimate session. A time threshold is setup depending on the transmission delay imposed by the communication channel between the user and the sensor node, leaving a potential attacker little time to mount a replay attack.

## 6.7 Comparison with Existing Protocols

None of the existing schemes provide both features of user authentication and session key establishment. Therefore, the proposed outside user authentication protocol is compared separately with the existing user authentication schemes and the session key establishment schemes. We first compare the proposed ID-1P-SKE protocol with the existing session key establishment (SKE) protocols and then the proposed outside user authentication protocol using BNN-IBS scheme with the existing user authentication protocols for WSNs.

### 6.7.1 Comparison with Existing SKE Protocols

The performance of the proposed session key establishment (ID-1P-SKE) protocol is discussed in two ways: firstly, by comparing it with the existing session key establishment protocols for WSNs in Table 6.4 and Table 6.5 and secondly, by comparing it with other ID-based one-pass session key establishment protocols in Table 6.6. The factors used to evaluate the performance are the number of complex cryptographic operations including pairing computation, point multiplication and exponentiation operations (computation overhead), total number of messages exchanged in each protocol run (communication overhead) and the memory requirements (storage overhead). Since the sensor nodes are more resource-constrained than users (more specifically, the devices held by users), we consider the efficiency of the protocols on the sensor node side rather than on the user side.

For  $\sim 80$ -bit security, in an efficient and optimized implementation on a standard MICA2 sensor node, one pairing computation takes 1.90s [OAG<sup>+</sup>11] and one point multiplication takes 0.295s (our results). In usual implementations of pairing, one exponentiation in  $\mathbb{G}_T$  costs about the same as four point multiplications in an additive group [CMS08]. Since the overheads of hash operation and arithmetic operations in  $\mathbb{Z}_q^*$  are very small compared to the above mentioned expensive cryptographic operations, we only consider the expensive cryptographic operations for performance comparison. In all tables,  $P$  denotes one pairing computation,  $H$  denotes one hash function evaluation,  $M$  denotes one point multiplication or exponentiation in  $\mathbb{G}$  and  $E$  denotes one exponentiation in target group  $\mathbb{G}_T$ .

#### 6.7.1.1 SKE Protocols for Wireless Sensor Networks

This section compares the proposed ID-1P-SKE protocol with the existing session key establishment protocols for WSNs proposed by Huang et al. [HCK<sup>+</sup>03], Kim et al. [KKLL07], and Zhang et al. [ZZR09]. These protocols have already been discussed in Section 3.3.2. Table 6.4 and Table 6.5 show the comparison results.

##### a) Computation Overhead

In the scenario of WSNs, it is highly desirable for a security protocol to have low computational overhead on resource constrained sensor nodes. In Huang et al.'s key establishment protocol [HCK<sup>+</sup>03], the computation overhead on a sensor node is the verification of a signed certificate to extract user's public key and the computations of three point multiplications to compute the session key. The user authentication, however, is achieved via key confirmation messages. For comparison purpose, we assume that the certificate verification requires the verification of an *ECDSA* signature. The *ECDSA* signature is considered more efficient to compute for sensor nodes than RSA signature because of shorter key and signature sizes. *ECDSA* requires two point multiplications as expensive operations to verify a signature. Hence, the total computation overhead of Huang et al.'s protocol will be five point multiplications. Kim et al.'s protocol [KKLL07] requires sensor nodes to compute three point multiplications and one exponentiation in  $\mathbb{G}_T$ . Zhang et al.'s protocol [ZZR09] brings down the computation cost of [KKLL07] by one point multiplication without providing user authentication. Our proposed protocol ID-1P-SKE requires a sensor node to compute only one point multiplication to compute the session key and one signature verification to authenticate the user. For comparison with

	Key Establishment Cost		User Authentication Cost		Time (s)
	<i>User</i>	<i>Sensor Node</i>	<i>Sensor Node</i>		<i>Sensor Node</i>
Huang et al. [HCK <sup>+</sup> 03]	$4M + 3H$	$3M + 3H$	<i>ECDSA</i>	$2M$	1.48
Kim et al. [KKLL07]	$2P + 1M + 1E + 2H$	$3M + 1E + 2H$	Implicit verification	NA	2.07
Zhang et al. [ZZR09]	$2P + 1M + 4H$	$2M + 1E + 3H$	Not supported	NA	1.77
Our scheme	$3M + 1H$	$1M$	<i>BNN-IBS</i>	$3M$	1.18

Table 6.4: Computation cost comparison of ID-1P-SKE protocol with the existing session key establishment protocols for WSNs

[HCK<sup>+</sup>03] and [KKLL07], we assume that the secure and efficient ID-based signature scheme *BNN-IBS* [CKDZ08] is used for user authentication in ID-1P-SKE protocol. *BNN-IBS* requires three point multiplications for signature verification. It is clear from Table 6.4 that the overall computational load of the proposed protocol is still lower than the computational loads of both [HCK<sup>+</sup>03] and [KKLL07]. Moreover, the key computation cost is lower than the key computation cost of [ZZR09]. At the same time, the proposed protocol has stronger security properties as compared to the other protocols, as we shall discuss later on in this section.

## b) Time Consumption

We now compare the estimated total computation time taken by a sensor node to authenticate a user and establish a session key. The results of this time analysis are also given in Table 6.4. Huang et al.'s protocol [HCK<sup>+</sup>03] requires a sensor node to compute five point multiplications, and hence takes about 1.48s on it. Kim et al.'s protocol [KKLL07], on the other hand, computes three point multiplications and one exponentiation in  $\mathbb{G}_T$ . Considering the fact that the exponentiation in  $\mathbb{G}_T$  costs equal to four times a point multiplication costs, the estimated computation time cost is about 2.07s for their protocol. Zhang et al.'s protocol [ZZR09] requires a sensor node to compute two point multiplications and one exponentiation in  $\mathbb{G}_T$  for key computation (this protocol does not provide user authentication) and consumes about 1.77s on a sensor node. Considering the ID-based signature scheme *BNN-IBS* [CKDZ08], the total estimated computation time for the proposed ID-1P-SKE protocol is about 1.18s for four point multiplications. This implies that compared with the protocols proposed by Huang et al., Kim et al., and Zhang et al.,

	Messages Exchanged	
	<i>Key Establishment</i>	<i>Key Confirmation</i>
Huang et al. [HCK <sup>+</sup> 03]	4	2
Kim et al. [KKLL07]	3	1
Zhang et al. [ZZR09]	3	NA (Does not support)
Our scheme	1	1

Table 6.5: Communication cost comparison of ID-1P-SKE protocol with the existing session key establishment protocols for WSNs

the ID-1P-SKE protocol reduces the total computation time for the authenticated key establishment on a sensor node by 20%, 33%, and 43%, respectively, without mentioning that Huang et al.'s and Zhang et al.'s protocols are quite weak in security as shown in the end of this section. In addition, note that the proposed protocol also improves the performance of a user by 25%, 82%, and 75% over Huang et al.'s, Kim et al.'s and Zhang et al.'s solutions, respectively. As improving the efficiency of the user side is not our focus in this research, we do not discuss this issue in detail.

### c) Communication Overhead

To achieve network resource efficiency and minimum latency, the number of messages exchanged between the sensor node and the user should be as small as possible. Huang et al.'s protocol [HCK<sup>+</sup>03] and Kim et al.'s protocol [KKLL07] exchange six and four messages, respectively, for key establishment and user authentication. The key confirmation messages are compulsory to provide user authentication in their protocols. Zhang et al.'s protocol [ZZR09] exchanges three messages for the key establishment. Our proposed protocol ID-1P-SKE exchanges only one message for both the key establishment and the user authentication. Hence, the proposed protocol causes very low communication overhead than the other three protocols for WSNs as shown in Table 6.5.

### d) Storage Overhead

The storage overhead of the proposed ID-1P-SKE protocol is similar to the other considered protocols and is not very high. The proposed protocol does not require sensor nodes to store any user credentials (IDs, public keys, certificates etc.), and hence provides storage efficiency. The only storage requirement is the sensor node's ID, its corresponding ID-based key and the system parameters which are required in all ID-based schemes.

### e) Performance Versus Security Comparison

As discussed earlier in Section 3.3.2, Huang et al.'s protocol [HCK<sup>+</sup>03] is not secure since a user can easily learn a sensor node's private key after one run of the protocol with the sensor node. This is a severe security attack against a key establishment protocol which cannot be tolerated, no matter how efficient a protocol is. Another drawback is the DoS attack caused by the delayed user authentication. On the other hand, Zhang et al.'s protocol [ZZR09] does not support user authentication at all allowing any adversary to establish a session key and obtain sensor nodes data. Hence, these two protocols lack the required security. Kim et al.'s protocol [KKLL07] also suffers from the DoS attack caused by the delayed user authentication wasting sensor node's resources. Our protocol authenticates a user at the first step by verifying the signed user's ephemeral public key and the time stamp. Furthermore, it is not possible for any participant or any adversary to learn any participant's private key.

However, an adversary can cause a sensor node to verify a fake signature in our protocol wasting its resources. To see how serious this attack is when compared to the DoS attack in Kim et al.'s protocol, we assume the secure and efficient ID-based signature scheme *BNN-IBS* [CKDZ08] used for signing the user's ephemeral public key in our protocol. To detect a fake ephemeral public key sent by an adversary, a sensor node will perform three point multiplications in our protocol and three point multiplications and one exponentiation in  $\mathbb{G}_T$  in Kim et al.'s protocol. Before a user is authenticated, four messages will be exchanged in Kim et al.'s protocol while only one message will be exchanged in our protocol since after receiving the first message from the user the sensor node can find out the fake ephemeral public key and terminate the protocol. To sum up, the DoS attack in our protocol is far less harmful than in Kim et al.'s protocol saving both the communication and the computation costs. Therefore, the proposed ID-1P-SKE protocol provides better performance versus security than the existing session key establishment protocols for the WSNs.

#### 6.7.1.2 ID-based One-Pass SKE Protocols

In this section, by comparing the proposed ID-1P-SKE protocol with the existing ID-based one-pass session key establishment protocols proposed by Benit et al. [BJT04], Okamoto et al. [OTO05], Wang [Wan05], and Gorantla et al. [GBGN08], we show that a significant efficiency improvement achieved by our protocol is its

	Key Establishment Cost		Time (s)
	<i>User</i>	<i>Sensor Node</i>	<i>Sensor Node</i>
Benit et al. [BJT04]	$1P + 2M + 1H$	$1P + 1H$	1.90
Okamoto et al. (II) [OTO05]	$1P + 3M + 2H$	$1P + 1M + 2H$	2.195
Wang [Wan05]	$1P + 3E + 2H$	$1P + 2E + 2H$	4.260
Gorantla et al. [GBGN08]	$1P + 2M + 1H$	$1P + 1M + 1H$	2.195
Our protocol	$3M + 1H$	$1M$	0.295

Table 6.6: Comparison of ID-1P-SKE protocol with existing ID-based one-pass key establishment protocols for traditional networks

very low computation and time costs. Note that the existing ID-based one-pass key establishment protocols are not in fact designed for WSNs. What we are showing here is that these protocols do not suit WSNs due to their low performances and high costs. Table 6.6 compares our protocol with the existing protocols by listing the key establishment costs for both sides of each protocol and the time cost for the sensor node side. The proposed protocol is computationally efficient on the sensor node's (responder's) side requiring only one point multiplication but no pairing computation, contrary to existing protocols. One pairing computation on a standard MICA2 sensor node takes 1.9s versus 0.295s for a point multiplication, and hence consumes resources equal to about six point multiplications. Due to the absence of pairing computations on both sides, our protocol provides much better performance than the existing protocols. Table 6.6 also shows the estimated time that a sensor node consumes if the existing protocols are applied in WSNs. This time is computed using the time costs for the expensive operations of point multiplication in ECC based schemes (0.295s), exponentiation in  $\mathbb{G}_T$  (1.18s) and pairing computation (1.9s) and ignoring the hash function computations. Although the point multiplication operation in pairing based schemes takes longer to compute than in pairing-free ECC based schemes, we assume the same time cost of the point multiplication for both the existing pairing based schemes and our pairing-free ID-1P-SKE scheme. It is clear from Table 6.6 that our proposed protocol is almost six times faster than Benit et al.'s protocol [BJT04], which is the best existing ID-based one-pass key establishment protocol in terms of efficiency on the sensor node's side. Moreover, if we also count the user authentication cost (BNN-IBS signature verification cost of three point multiplications), our protocol still outperforms all the existing protocols with the total time of 1.18s. Note that not all the existing protocols provide user authentication, for instance, not Benit et al.'s protocol. To sum up, our proposed ID-1P-SKE protocol is the most suitable ID-based one-pass session key establishment

protocol for WSNs. Other than WSNs, it can also be used for other application environments where the responder is a resource constrained device.

### 6.7.2 Comparison with Existing User Authentication Protocols

This section compares the proposed outside user authentication protocol using IBS schemes with the existing distributed digital signature based user authentication schemes for WSNs. Only two of the existing user authentication schemes use the distributed approach to authenticate outside users, RRUASN [BGR05] and DP<sup>2</sup>AC [ZZR09]. None of them provide session key establishment. The details of these schemes are given in Section 3.3.1. The RRUASN scheme uses ECDSA signature to verify outside users whereas DP<sup>2</sup>AC scheme uses RSA signature. For comparison purposes, we assume the pairing-free BNN-IBS scheme for our proposed outside user authentication protocol.

In **RRUASN**, user authentication involves verification of two ECDSA signatures by the sensor nodes as expensive operations; one to verify the signed certificate and second to verify the signed user request. The signed certificate to extract public key is sent along with every user request increasing communication overhead. **DP<sup>2</sup>AC** involves one RSA signature verification and verification of token re-usability. A major issue with this scheme is the storage overhead. In this scheme, every used token is stored on more than one sensor node in the network. Consider a large scale sensor network of say 60,000 sensor nodes arranged in the form of a grid. Suppose every used token is stored on at least 245 sensor nodes forming one vertical line in the grid. Assuming a token size = 10 bytes and number of used tokens  $T = 10,000$ , the overall storage overhead on the sensor network is about  $(10 \times 10000 \times 245 =) 24500,000$  bytes (roughly 23925 KB) which is considerable for resource constrained sensor nodes. Another issue with this scheme is the communication overhead per user request to verify a used token. Every new token is sent to at least 245 sensor nodes forming one horizontal line in the sensor network for re-usability checking. Therefore, the total verification time cost of this scheme is the RSA signature verification time plus transmission time ( $TT$ ) to send a token to a set of sensor nodes for re-usability check and get a feedback from them. Moreover, this scheme can work well only if the sensor nodes are arranged in a grid forming horizontal and vertical lines. The proposed outside user authentication scheme using an IBS scheme, i.e., BNN-IBS, involves only one signature verification, consisting of three



Schemes	Signature	Verify Time (s)	Storage	Session Key
		(s)	Bytes	
<b>Existing Distributed User Authentication Schemes</b>				
RRUASN [BGR05]	ECDSA	1.26	0	No
DP <sup>2</sup> AC [ZZR09]	RSA	$0.47 + TT^\alpha$	$10 \times T^\beta$	No
<b>Proposed Distributed User Authentication Scheme</b>				
Proposed	BNN-IBS	1.044	0	Yes

<sup>$\alpha$</sup>   $TT$  is transmission time to send a token to a set of sensor nodes for re-usability checking

<sup>$\beta$</sup>   $T$  is the number of used tokens

Table 6.7: Comparison of proposed user authentication scheme with existing distributed user authentication schemes for WSNs

point multiplications, by the sensor nodes to authenticate a user.

The actual implementation results of the existing schemes on real sensor nodes are not available. However, for comparison purposes, we estimate the cost of these schemes by considering the costs of signatures used in these schemes on MICA2 sensor nodes. The signature verification on MICA2 for BNN-IBS scheme takes about 1.044s (our result) and for ECDSA takes 0.63s [ADLO10]. One RSA signature verification with 1024 bit key size takes about 0.47s on MICA2 sensor nodes [PLP06]. Using these facts, Table 6.7 gives the cost of all schemes. It is clear from Table 6.7 that the proposed user authentication scheme using an IBS scheme consumes less time as compared to RRUASN without transmitting any certificate and eliminates the storage and communication overhead of  $DP^2AC$ . It also provides the session key establishment between the user and the sensor node whereas none of the existing user authentication schemes deal with the key establishment problem.

## 6.8 The Proposed Authentication Framework: A Single Solution

As mentioned previously, there are three types of broadcast/multicast authentication problems in WSNs.

- Base station to sensor nodes authentication
- Outside user to sensor nodes authentication
- Sensor node to other sensor nodes authentication

The authentication framework proposed in this thesis focuses on sensor node to other sensor nodes broadcast authentication and outside user to sensor nodes authentication. However, using the proposed authentication framework and the same ID-based setup, the base station to sensor nodes broadcast authentication can also be handled by the proposed framework. The base station can be authenticated in the same way as an outside user is authenticated in the proposed user authentication protocol. However, after successful authentication there is no need to establish a session key between a sensor node and the base station since each sensor node already shares a unique individual key with the base station. Therefore, the protocol is terminated after authentication. Moreover, there is no need for the base station to send an ephemeral public key. It is easy to differentiate the base station from other outside users since every sensor node stores the *ID* of the base station. The sensor nodes do not need to store any additional parameters to authenticate broadcast messages from the base station. With this addition, the proposed authentication framework can address all three types of broadcast authentication problems faced in WSNs.

The most efficient available pairing-free BNN-IBS scheme and our adapted B-IBOOS scheme both use the same system parameters. Moreover, the signature verification process is the same in both signature schemes. Hence, using the same private keys and ID-based setup the sensor nodes can sign broadcast messages as well as authenticate broadcast messages from the base station, outside users and other sensor nodes in the sensor network. The sensor nodes do not need to store different parameters for different authentication types. In addition, the proposed ID-1P-SKE scheme also relies on the same system parameters as used by the BNN-IBS and B-IBOOS schemes. Consequently, the sensor nodes can also establish a session key with outside users using the same parameters. This makes the proposed framework a single efficient and detailed solution to handle authentication problems as well as the session key establishment problem. Furthermore, our proposed authentication framework is the most efficient and secure as compared to the existing individual authentication solutions for WSNs.

## 6.9 Concluding Remarks

In this chapter, the proposed outside user authentication and session key establishment protocol and its performance and security evaluations have been presented. The existing user authentication schemes for WSNs are either prone to security

attacks or expensive for sensor nodes. Moreover, they lack one or the other security feature of user authentication and key establishment, and hence cannot provide a complete solution to the problem requiring separate protocols for user authentication and session key establishment. Compared to the existing user authentication schemes for WSNs, the proposed protocol provides both the user authentication and the session key establishment after successful user authentication. The proposed protocol uses the ID-based signatures to authenticate a user which handles the problem of public keys and certificate management faced in WSNs environment. The performance analysis confirmed the suitability of the proposed protocol using IBS schemes for resource constrained sensor nodes. Moreover, the performance comparison of the proposed protocol with the existing distributed user authentication protocols for WSNs showed the proposed protocol a better option for WSNs, demonstrating the particularly low computation overhead of the proposed protocol as compared to the existing protocols. Scalability is another prominent feature of the proposed protocol. The performance vs security comparison of the proposed ID-1P-SKE protocol with the existing session key establishment protocols for WSNs proved the proposed ID-1P-SKE protocol a best solution for the problem, demonstrating a very low computation and communication overheads while achieving the high level of security. The proposed authentication framework provides a single efficient solution to all three broadcast authentication problems faced in WSNs.



# Chapter 7

## Conclusion and Future Work

***Chapter Overview:** This chapter summarizes the major contributions of the thesis. It also highlights the avenues for future research which could be conducted to extend the proposed authentication framework.*

Wireless sensor networks are a unique class of ad hoc networks typically consisting of tiny low-cost resource constrained devices. These resource constrained devices are usually deployed in an open environment to sense and communicate data to a destination over a wireless medium. The wireless communication and the deployment nature of the WSNs open the door to a variety of security attacks in addition to the security attacks faced by traditional networks. On the other hand, the limited resources of the sensor nodes pose a hurdle in applying complex security solutions which are tailored to traditional networks.

This thesis contributes to the active research area of broadcast authentication in WSNs by considering both sensor nodes broadcast authentication and outside user authentication which also facilitates the base station to sensor nodes broadcast authentication. A traditional mean to provide authentication, i.e., a digital signature, consumes considerable time and energy on resource constrained sensor

nodes. Moreover, public key and certificate management is another issue faced while applying digital signatures in a WSN environment. Public keys and certificates are either sent with every signed message (increasing transmission and processing overheads) or stored on each sensor node (increasing storage overhead and reducing scalability). The MAC based authentication schemes are efficient on sensor nodes, however they fail to provide a solution to the broadcast authentication problems. To tackle these problems, a novel authentication framework for WSNs has been proposed in this thesis giving a solution to the above mentioned authentication problems.

## 7.1 Summary of the Thesis Contributions

To recapitulate, the specific contributions of this thesis are:

1. **Authentication framework.** An authentication framework for WSNs using ID-based signature schemes has been proposed. The proposed authentication framework is comprised of two authentication protocols; one for broadcast by sensor nodes authentication and the other for outside user authentication. The outside user authentication protocol also facilitates the base station to sensor nodes broadcast authentication. These authentication protocols can be applied in WSNs independently tackling individual security problems to achieve different level of security. However, deployed as a unified framework, they ensure a high degree of security with efficiency, providing a single solution to all three authentication problems in WSNs using the same ID-based setup. An important feature of the framework is its re-usability. Once new IBS and IBOOS schemes are available, which are more efficient than the currently available schemes, the framework can be re-used with the new schemes for efficiency improvement.
2. **First time handling of sensor nodes broadcast authentication.** Earlier work in authentication in WSNs focused on two types of broadcast authentication problems only: the base station to sensor nodes and the outsider user to sensor nodes. It completely ignored the third type of broadcast authentication, i.e., the sensor nodes to other sensor nodes broadcast authentication. Our research work highlights for the first time the problem of authenticated broadcast by sensor nodes, particularly for time critical applications of WSNs. We describe real application scenarios where the problem is important. We

also highlight the need for a solution and propose a solution to the problem. Our proposed solution enables sensor nodes to broadcast authenticated real-time messages without the involvement of the base station.

3. **First time application of OOS in WSNs.** MAC based authentication schemes are efficient to compute on sensor nodes, however, they fail to provide a solution to the problem. Digital signatures, on the other hand, take longer to compute on resource constrained sensor nodes which does not suit the time-critical applications of WSNs. The proposed broadcast by sensor nodes authentication protocol proposes to use online/offline signature schemes for the first time in WSNs. The IBOOS schemes enable a sensor node to sign and broadcast a message as soon as possible to respond to a time-critical event by performing the most time consuming computations of message signing before the time-critical event happens. Moreover, performing the expensive offline phase by some other powerful device and re-using the offline signature to sign more than one message can reduce the cost of a signature scheme for sensor nodes bringing significant efficiency achievements.
4. **First time implementation of OOS on sensor nodes devices.** To the best of our knowledge, it was the first proposal of applying online/offline signatures to the WSNs environment, a cryptographic primitive new to sensor nodes devices. In order to evaluate the performance of IBOOS schemes on real sensor nodes devices, a few IBOOS schemes were implemented and their performance was evaluated. This was the first implementation of IBOOS schemes on real sensor nodes. The experimental results proved the IBOOS schemes efficient for resource constrained sensor nodes.
5. **Adaption of an IBS scheme to a pairing-free IBOOS scheme.** The implementation results of the first IBOOS scheme were too expensive for sensor nodes. However, the reason was the expensive pairing based cryptography and not the IBOOS scheme itself. In order to obtain a pairing-free IBOOS scheme, we have securely transformed an IBS scheme to an IBOOS scheme. The implementation results of the transformed B-IBOOS scheme were very efficient, enabling a sensor node to sign a message only in 0.025s. This time cost is quick enough considering the resource constrained nature of sensor nodes devices. The two different implementations of the adapted IBOOS scheme provide a trade-off between computation cost and the memory consumption.

Compared with the existing digital signature based authentication schemes for WSNs, the proposed broadcast authentication scheme using pairing-free B-IBOOS scheme was the most efficient and secure solution to the problem.

6. **A session key establishment protocol (SKP) for WSNs.** The existing session key establishment protocols for WSNs were either expensive or prone to security attacks. The existing ID-based one-pass key establishment protocols for traditional networks were all pairing-based, and hence expensive for sensor nodes. Thus, a new secure and efficient ID-based one-pass authenticated session key establishment protocol (ID-1P-SKE) was designed mainly for WSNs to establish a session key between a resourceful user and a resource constrained sensor node. However, it can be used for any other similar application environment where the key is established between a powerful initiator and a resource constrained responder.
7. **Formal security analysis of SKP.** The security of the proposed cryptographic ID-1P-SKE protocol was formally analyzed using the reductionist proof technique. The security analysis of the protocol proved the proposed protocol secure not only for WSNs environment but also for any other similar application environment.
8. **User authentication together with session key establishment.** The proposed user authentication protocol, compared to the existing user authentication protocols, not only authenticates the outside users but also establishes a session key between the user and the sensor node after successful user authentication. This key establishment is mandatory for the secure exchange of valuable sensor nodes data and in some cases for the exchange of encrypted user queries to provide query privacy. The ID-based signatures handle the problem of public key and certificate management, hence providing scalability. Compared with the existing distributed user authentication schemes for WSNs, the proposed user authentication scheme using the pairing-free BNN-IBS scheme was the most efficient, secure and a complete solution to the problem.

## 7.2 Future Research Directions

The research that has been undertaken for this thesis has successfully met the research aims and the time span proposed. However, the investigated research



area of outside user authentication has been spotted as an appealing aspect of WSNs along which additional research could be conducted. The proposed outside user authentication protocol deals with the user authentication and session key establishment. For future research, we suggest to extend the proposed authentication framework by including the user access control according to user's access privileges or attributes as a part of outside user authentication. Experimental evaluation of the proposed authenticated broadcast by sensor nodes protocol is also in our list of future work. Besides this, we have an interest in further exploring the efficient pairing-free ID-based signature schemes for WSNs and application of the proposed framework to other areas in WSNs as well as to other networks. This section briefly outlines these areas of future work.

### 7.2.1 Extension of the Authentication Framework

Sensor nodes in a WSN usually collect a variety of data. Different users of a WSN may be interested in different types of data and may have different access privileges due to the data security and privacy. For example, in an army application a major have access to more sensitive data than a soldier. Therefore, there is a need of a mechanism to restrict a user's access to sensor nodes data according to user's access privileges or attributes, known as *attribute based user access control*. Like user authentication and session key establishment, access control is also a part of user's access to sensor nodes data task. The attributes based user access control in WSN is enforced to grant a user the right to access only the data for which he is an authorized user. To control a user's access according to his attributes, a sensor node should be able to differentiate between users, which ultimately requires some user specific information on the sensor node's side. Storing user specific information for every user on sensor nodes would be impractical for a large scale sensor network due to the limited storage capability of a sensor node. On the other hand, sending user credentials with a user request would require some kind of verification.

A common approach to handle an attribute based user access control is to combine confidentiality (encryption) with user access control. Attribute-based encryption (ABE) [SW04], a generalization of ID-based encryption (IBE) [BF03], helps in this regard to link encryption to some user attributes. In attribute-based encryption, data intended for a user is encrypted based on attributes or privileges assigned to that user by an authority. Only users that have the required attributes are able to decrypt the data, ensuring confidentiality with access control. However,

attribute-based encryption is not computationally efficient for resource constrained sensor nodes due to the fact that the computational cost of encryption grows linearly with the number of attributes assigned to users.

To cope with this problem, we propose to modify the ID-based encryption to achieve an inexpensive alternative to attribute-based encryption. As mentioned before, in ID-based schemes the private keys corresponding to IDs are computed by a PKG. Our idea is to associate the set of attributes assigned to a user to his private key. The corresponding computed public information is then automatically linked to the same set of attributes. Data is encrypted using a user's public information and decrypted using his corresponding private key. Consequently, only a user with the corresponding private key (computed using his assigned set of attributes by the PKG) is able to decrypt the data.

To develop this idea, we started looking for a pairing-free IBE scheme. Unfortunately, the existing secure and efficient IBE schemes [BF03, Lyn02] rely on pairing computations. In the next step, we tried to design a pairing-free IBE scheme from a pairing-based IBE scheme which could be used for WSNs. Our attempt to design a pairing-free IBE scheme was inspired by a pairing-based authenticated encryption scheme with CCA (Chosen Cipher Attack) security and reduced ciphertext size presented in [Lyn02]. This is an in process work lacking the formal security and performance evaluations of the designed IBE scheme.

## 7.2.2 Experimental Evaluation of the Proposed Authenticated Broadcast by Sensor Nodes Protocol

During this research, we performed some experiments where we implemented a few IBOOS schemes on sensor nodes for our proposed authenticated broadcast by sensor nodes protocol. An IBOOS scheme is a cryptographic primitive which has not been used for sensor nodes. The purpose of our experiments was to see whether it was possible for the sensor nodes to compute IBOOS schemes and how efficient they were to compute on real sensor nodes. Our experimental results (discussed in chapter 5) showed their suitability and performance on sensor nodes answering both questions. In future, we are interested in evaluation of the proposed authenticated broadcast by sensor nodes protocol either by implementing on real sensor network or through simulation. These experiments will be helpful in order to estimate the overheads introduced by the protocol when deployed in a large scale wireless sensor network. The storage requirement of the protocol on a sensor node remains constant and

does not increase with the increase in number of broadcast senders or receivers. The experimental results will help to evaluate the communication and computation overheads introduced by the protocol.

### 7.2.3 Pairing-free IBS and IBOOS Schemes

Pairing-free IBS and IBOOS schemes are another interesting areas which could be further explored. Pairing based cryptography is resource hungry and does not suit the resource constrained application environments such as wireless sensor networks. There is a need to design new efficient pairing-free cryptographic schemes for such application environments. The existing pairing-based cryptographic schemes can also be investigated for possible transformations to obtain efficient pairing-based schemes for WSNs.

### 7.2.4 Other Applications of the Authentication Framework

The proposed authentication framework can be used in WSNs to handle other problems. For instance, data aggregation in WSNs can also benefit from the proposed framework. Data aggregation is an in-network processing of the data collected by the sensor nodes while forwarding it to the base station. The main goal of data aggregation is to gather and aggregate data in an energy efficient manner to increase network lifetime. An aggregator node computes the simple operations of sum, average, minimum or maximum of the data gathered from other sensor nodes before forwarding it to the other aggregator node on the way to the base station. It helps to reduce the size of raw data and save network resources like bandwidth and battery power. Secure data aggregation protocols must satisfy the security requirements of *source authentication* as well as data *confidentiality*, *integrity* and *freshness* [OX09]. The IBOOS schemes and the proposed IBE scheme can be further studied to provide a secure data aggregation protocol accommodating all needed security requirements.

Other than WSNs, the proposed framework can also be studied to apply in other networks. For instance, the vehicular ad-hoc network is a candidate network which can benefit from the proposed framework. In vehicular ad-hoc networks described in [BH08], the proposed authentication protocols can provide a secure communication between vehicles and between vehicles and the roadside infrastructures which include sensor nodes. Mobile ad-hoc networks can also be explored as a potential candidate.



# Appendix A

## Experimental Details

### A.1 MICA2 Hardware Details

A MICA2 [Cro] sensor node consists of a wireless module, sensor boards and a PC interface board. The compulsory parts of a MICA2 sensor node we used in our experiments are a wireless module and a PC interface board. We did not use any sensor board since our experiments did not require any sensing.

#### A.1.1 Wireless Module

A MICA2 wireless module is a combination of a micro-controller, a radio transceiver and a battery pack (usually 2xAA batteries). Sensor board is attached to the wireless module which adds up the sensing ability to MICA2 wireless module. The MICA2 motes come in three models according to their radio frequency bands: MPR400 (915 MHz), MPR410 (433 MHz), and MPR420 (315 MHz). All models utilize a powerful ATmega128L micro-controller and a Chipcon CC1000 frequency tunable radio transceiver with extended range. ATmega128L is a low power micro-controller which runs TinyOS (operating system for sensor nodes) from its internal flash memory. A MICA2 mote has 128KB in-system programmable flash and 4KB SRAM with a clock speed of 7.3828 MHz. The MICA2 wireless module we selected for our experiments was MPR400CB. The reason is that MPR400CB operates on 868 MHz frequency band which is a license-free band for Europe. Figure A.1 shows a MICA2 (MPR4x0) sensor node without an antenna. A sensor board is connected to the mote through the 51 pin connector shown in the figure. In order to program MICA2, it is connected to the PC or laptop through a PC Interface Board.

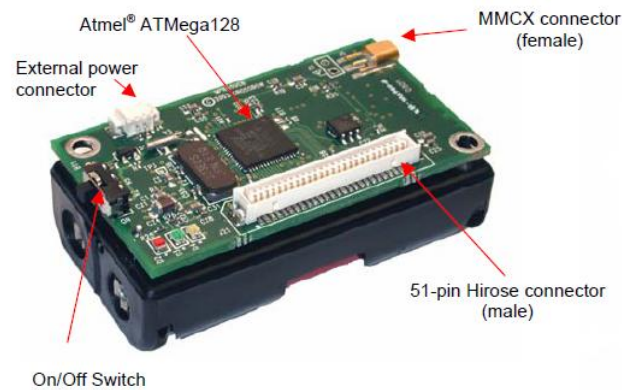


Figure A.1: A MICA2 (MPR4x0) wireless module without an antenna [Cro].

### A.1.2 PC Interface Board

We selected a serial PC interface board to program sensor nodes, i.e., MIB510CB serial interface board. The other options are USB or Ethernet interface boards. The MIB510 interface board, shown in Figure A.2, is a multi-purpose interface board. It supplies power to the wireless module through an external power adapter option while wireless module is attached to it. In addition, it serves two main purposes: it allows the user to (re)program any sensor node by plugging the node directly into the base and it operates as a part of the root node interface giving the PC a data passage onto the radio based sensor network.

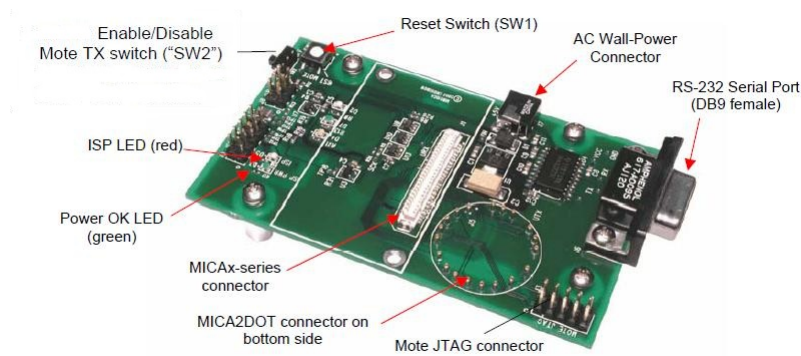


Figure A.2: Top view of MIB510 Serial Interface Board [Cro].

### A.1.3 Hardware Configuration

We developed and debugged our software programs on a laptop. These programs were then installed on MICA2 nodes. In order to install a program on a mote, the wireless module (MPR400CB) shown in Figure A.1 is attached to the PC interface board (MIB510CB) shown in Figure A.2 through the 51 pin connector. The MIB510CB board is then connected to the PC through a serial port (RS-232). The details of this whole procedure are given in [AAKC06].

### A.1.4 Programming a Node

Programming a mote requires the operating system TinyOS to be installed on laptop. We installed TinyOS on our laptop on top of Fedora (Linux). To compile a program for MICA2 in TinyOS, the command is as follows:

```
make mica2
```

This command generates an executable file in the same directory and helps in debugging the program. There is a special command used to install a program on MICA2 nodes which compiles the program and then uploads it to the node.

```
make mica2 reinstall mib510,/dev/USBx
```

Here mib510 is the PC interface board mounted on /dev/USBx. We followed the step by step procedure of installing TinyOS and programming a MICA2 sensor node given in [AAKC06]. The results of our experiments have already been discussed in detail.





# Appendix B

## ID-based One-Pass Session Key Establishment Protocol

### B.1 Proposed ID-1P-SKE Protocol for WSNs

This section presents detailed application of the proposed ID-1P-SKE protocol in WSNs. The proposed ID-based one-pass authenticated key establishment protocol has four phases: *System Initialization*, *Key Generation*, *User Registration* and *Key Establishment*. The first two phases are performed once, before the deployment of the sensor network. In an ID-based cryptosystem, a private key generator (PKG) computes the private keys corresponding to *IDs*. In our scheme, the base station plays the role of *PKG* and computes the private keys for sensor nodes and users.

**System Initialization:** In this phase, the *Setup* algorithm of ID-1P-SKE runs on the base station (before deployment) and generates the system parameters, including master public key (*mpk*), and the corresponding master secret key (*msk*) using a security parameter *k*. This algorithm performs the following steps:

(a) Specify the parameters  $\mathbb{E}/\mathbb{F}_p$ ,  $q$ ,  $p$ ,  $P$  and  $\mathbb{G}$ , where

- $\mathbb{E}/\mathbb{F}_p$  is an elliptic curve  $\mathbb{E}$  over a finite field  $\mathbb{F}_p$ ,
- $q$  is a large prime number and  $p$  is the field size,
- $P$  is a point of order  $q$  on the curve  $\mathbb{E}$  and,
- $\mathbb{G}$  is a cyclic group of order  $q$  under the point addition “+” generated by  $P$ .

- (b) For  $msk$   $s \in_R \mathbb{Z}_q^*$ , compute  $mpk$  as  $P_{PKG} = sP$ .
- (c) Choose one hash function  $H: \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$ .
- (d) Choose one function  $\chi: \mathbb{G} \rightarrow \{0, 1\}^k$  to derive the session key where  $k$  is the security parameter.
- (e) Output the system parameters  $\{\mathbb{E}/\mathbb{F}_p, q, p, P, \mathbb{G}, P_{PKG}, H, \chi\}$  and keep  $s$  secret.

**Key Generation:** In this phase, the *Key Extract* algorithm of ID-1P-SKE runs on the base station (before deployment) and computes the private keys of all sensor nodes. This algorithm takes  $msk$  and a sensor node's  $ID$  as input and generates a private key corresponding to that  $ID$  using the Schnorr signature. For a sensor node  $I$  with identity  $ID_i$ , this algorithm performs the following steps:

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

The  $ID$ s, corresponding private keys and system parameters are stored on sensor nodes before the deployment. Hence, every sensor node  $I$  stores  $\{ID_i, s_i, R_i\}$  and system parameters.

**User Registration:** This phase is repeated every time when a new user is registered with the system. In this phase, the *Key Extract* algorithm runs on the base station and computes the private key  $s_u$  for the new user  $U$  corresponding to his identity  $ID_u$  in the same way as computed for sensor nodes in the *Key Generation* phase. The base station, who runs this algorithm, sends the private key to the user via a secure channel. Hence, every user  $U$  obtains  $\{ID_u, s_u, R_u\}$  and system parameters. To establish a session key with a sensor node, the user also needs the  $ID$  and corresponding  $R$  information of the sensor node. Therefore, he also obtains the  $\{ID_i, R_i\}$  pairs of the sensor nodes in his communication range.

**Key Establishment:** In this phase, a session key is established between a user and a sensor node for the secure transmission of sensor nodes data to the user. Here,

the *Key Establishment* algorithm of ID-1P-SKE is described between a user  $U$  and a sensor node  $I$ . Figure B.1 describes the steps of the protocol.

User $U$	Sensor Node $I$
$l \in_R \mathbb{Z}_q^*$ $y = ls_u$ $L = yP$ $\sigma = \text{Sign}_{s_u}(L, ID_u, ID_i, TS)$	
$\xrightarrow{L, ID_u, ID_i, TS, \sigma}$	
$c_i = H(ID_i, R_i)$ $S_i = c_i P_{PKG} + R_i$ $K_{u,i} = yS_i = (ls_u s_i P)$ $SK_u = \chi(K_{u,i})$	$\text{Verify}_{ID_u}(L, ID_u, ID_i, TS, \sigma)$ $K_{i,u} = s_i L = (s_i ls_u P)$ $SK_i = \chi(K_{i,u})$
$SK = SK_u = SK_i = \chi(ls_u s_i P)$	

Figure B.1: Authenticated One-Pass Session Key Establishment Protocol

(a) The user  $U$  chooses at random  $l \in_R \mathbb{Z}_q^*$  and computes  $y = ls_u$  and  $L = yP$ .  $U$  signs the ephemeral public key  $L$  together with  $ID_u$ ,  $ID_i$  and  $TS$  as follows:  $\sigma = \text{Sign}_{s_u}(L, ID_u, ID_i, TS)$ .  $U$  then sends  $[L, ID_u, ID_i, TS, \sigma]$  to the sensor node  $I$  in his communication range. Here  $TS$  is the current time stamp to avoid a replay attack and  $\sigma$  is a signature signed by  $U$  using his private key  $s_u$  and an ID-based signature (IBS) scheme with the same ID-based setup as used by the ID-1P-SKE protocol, for instance the secure BNN-IBS scheme.

(b) The sensor node  $I$  first checks the time stamp  $TS$  to avoid the verification of a replayed message. If this is a fresh message,  $I$  verifies the signature  $\sigma$  as  $\text{Verify}_{ID_u}(L, ID_u, ID_i, TS, \sigma)$ . The successful signature verification implies that the message is actually sent by the user  $U$ , and hence  $I$  accepts it. Otherwise the protocol is terminated at this stage. Next, the sensor node  $I$  computes the shared secret  $K_{i,u}$  as

$$K_{i,u} = s_i L (= s_i ls_u P)$$

and deletes  $L$ .

(c) The user  $U$  computes the same shared secret  $K_{u,i}$  as

$$c_i = H(ID_i, R_i)$$

$$S_i = c_i P_{PKG} + R_i$$

$$K_{u,i} = y S_i (= l s_u s_i P)$$

$U$  then deletes  $L$ ,  $l$  and  $y$ .

(d) Both parties then compute the shared session key as

$$SK = \chi(K_{u,i}) = \chi(K_{i,u}) = \chi(l s_u s_i P),$$

where  $\chi$  is the key derivation function. The session key at this stage is ready to be used to exchange encrypted data.

The security of the ID-1P-SKE protocol and its performance in WSN environment have already been discussed with outside user authentication protocol.

# List of References

- [AAKC06] Hani Alzaid, Suhail Abanmi, Salil Kanhere, and Chun Tung Chou. Detecting wormhole attacks in wireless sensor networks. <http://eprints.qut.edu.au/9866/>, 2006.
- [ADLO10] Diego F. Aranha, Ricardo Dahab, Julio López, and Leonardo B. Oliveira. Efficient Implementation of Elliptic Curve Cryptography in Wireless Sensors. *Advances in Mathematics of Communications*, 4(2):169–187, 2010.
- [AG] Diego F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient Library for Cryptography. <http://code.google.com/p/relic-toolkit/>.
- [APM05] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3(3):257–279, 2005.
- [ASSC02] Ian F. Akyildiz, W. Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.
- [Avr] Avrora: The AVR Simulation and Analysis Framework. <http://compilers.cs.ucla.edu/avrora/>.
- [BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology - CRYPTO '96*, pages 1–15, London, UK, 1996. Springer-Verlag.
- [BF03] Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

- [BGhS07] Paulo S. L. M. Barreto, Steven Galbraith, Colm O hEigeartaigh, and Michael Scott. Efficient pairing computation on supersingular Abelian varieties. *Designs, Codes and Cryptography*, 42(3):239–271, 2007.
- [BGK04] Zinaida Benenson, Felix Gartner, and Dogan Kesdogan. User authentication in sensor networks (Extended Abstract). In *Proceedings of Informatik 2004, Workshop on Sensor Networks*, 2004.
- [BGR05] Zinaida Benenson, Nils Gedicke, and Ossi Raivio. Realizing robust user authentication in sensor networks. In *Proceedings of Workshop on Real-World Wireless Sensor Networks - REALWSN'05*, 2005.
- [BHUW08] Jens-Matthias Bohli, Alban Hessler, Osman Ugus, and Dirk Westhoff. A secure and resilient WSN roadside architecture for intelligent transport systems. In *Proceedings of WiSec '08*, pages 161–171, NY, USA, 2008. ACM.
- [BJT04] Waldyr Benits Jr and Routo Terada. An IBE Scheme to Exchange Authenticated Secret Keys. Cryptology ePrint Archive, Report 2004/071, 2004. <http://eprint.iacr.org/>.
- [BNN04] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security Proofs for Identity-Based Identification and Signature Schemes. In *Advances in Cryptology - EUROCRYPT '04*, volume 3027 of *LNCS*, pages 268–286. Springer Berlin/Heidelberg, 2004.
- [BWJM97] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key agreement protocols and their security analysis. In *Proceedings of the Cryptography and Coding*, pages 30–45, London, UK, 1997. Springer-Verlag.
- [CBHVS09] Konstantinos Chalkias, Foteini Baldimtsi, Dimitrios Hristu-Varsakelis, and George Stephanides. Two Types of Key-Compromise Impersonation Attacks against One-Pass Key Establishment Protocols. In *Proceedings of e-Business and Telecommunications*, pages 227–238, 2009.
- [CC03] Jae Choon Cha and Jung Hee Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In *Proceedings of Public Key Cryptography - PKC '03*, pages 18–30, 2003.

- [CC05] Wen-Huei Chen and Yu-Jen Chen. A bootstrapping scheme for inter-sensor authentication within sensor networks. *Communications Letters, IEEE*, 9(10):945–947, 2005.
- [CES04] David Culler, Deborah Estrin, and Mani Srivastava. Guest editors’ introduction: Overview of sensor networks. *Computer*, 37(8):41 – 49, 2004.
- [CK03] Chee-Yee Chong and Srikanta P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, 91(8):1247 – 1256, 2003.
- [CKDZ08] Xuefei Cao, Weidong Kou, Lanjun Dang, and Bin Zhao. IMBAS: Identity-based Multi-user Broadcast Authentication in Wireless Sensor Networks. *Computer Communications*, 31(4):659–667, 2008.
- [CMS08] Liquan Chen, Paul Morrissey, and Nigel P. Smart. Pairings in Trusted Computing. In *Proceedings of Pairing’08*, volume 5209 of *LNCS*, pages 1–17. Springer Berlin/Heidelberg, 2008.
- [CP03] Haowen Chan and Adrian Perrig. Security and privacy in sensor networks. *IEEE Computer*, 36(10):103–105, 2003.
- [Cro] Crossbow Technology, Inc. MPR/MIB Mote Hardware Users Manual. Online. [http://bullseye.xbow.com:81/Support/Support\\_pdf\\_files/MPR-MIB\\_Series\\_Users\\_Manual.pdf](http://bullseye.xbow.com:81/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf).
- [CYS<sup>+</sup>07] Bogdan Carbunar, Yang Yu, Weidong Shi, Michael Pearce, and Venu Vasudevan. Query privacy in wireless sensor networks. In *Proceedings of Sensor, Mesh and Ad Hoc Communications and Networks - SECON ’07*, pages 203–212, 2007.
- [Das09] Manik Lal Das. Two-factor user authentication in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 8(3):1086–1090, 2009.
- [DC08] Xiaojiang Du and Hsiao-Hwa Chen. Security in wireless sensor networks. *Wireless Communications, IEEE*, 15(4):60–66, Aug. 2008.

- [DG06] Jawad Drissi and Qijun Gu. Localized broadcast authentication in large sensor networks. In *Proceedings of International Conference on Networking and Services - ICNS '06*, page 25. IEEE, 2006.
- [DH76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
- [DVOW92] Whitfield Diffie, Paul C. Van Oorschot, and Michael J. Wiener. Authentication and Authenticated Key Exchanges. *Design Codes and Cryptography*, 2(2):107–125, 1992.
- [EGM90] Shimon Even, Oded Goldreich, and Silvio Micali. On-Line/Off-Line digital signatures. In *Advances in Cryptology - CRYPTO '89*, volume 435 of *LNCS*, pages 263–275. Springer Berlin, 1990.
- [FHB<sup>+</sup>08] Andreas Festag, Alban Hessler, Roberto Baldessari, Long Le, Wenhui Zhang, and Dirk Westhoff. Vehicle-to-Vehicle and Road-Side Sensor Communication for Enhanced Road Safety. In *ITS World Congress*, 2008.
- [GBGN08] M. Choudary Gorantla, Colin Boyd, and Juan Manuel González Nieto. ID-based One-pass Authenticated Key Establishment. In *Proceedings of Australasian Information Security Conference - AISC' 08*, pages 39–46. Australian Computer Society, Inc., 2008.
- [GD07] Qijun Gu and Jawad Drissi. Dominating Set based Overhead Reduction for Broadcast Authentication in Large Sensor Networks. In *Proceedings of International Conference on Networking and Services - ICNS '07*, page 81, 2007.
- [GG09] David Galindo and Flavio D. Garcia. A Schnorr-Like Lightweight Identity-Based Signature Scheme. In *Proceedings of AFRICACRYPT' 09*, volume 5580, pages 135–148. Springer-Verlag, 2009.
- [GPW<sup>+</sup>04] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Proceedings of Cryptographic Hardware and Embedded Systems - CHES '04*, pages 119–132, 2004.
- [HCK<sup>+</sup>03] Qiang Huang, Johnas Cukier, Hisashi Kobayashi, Bede Liu, and Jinyun Zhang. Fast Authenticated Key Establishment Protocols for



- Self-Organizing Sensor Networks. In *Proceedings of Wireless Sensor Networks and Applications - WSNA '03*, pages 141–150. ACM, 2003.
- [ICS03] IEEE-Computer-Society. 802.15.4 - wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans), 2003.
- [JLX07] Canming Jiang, Bao Li, and Haixia Xu. An efficient scheme for user authentication in wireless sensor networks. In *Proceedings of Advanced Information Networking and Applications - AINA '07*, pages 438–442, 2007.
- [JMV01] Don Johnson, Alfred Menezes, and Scott A. Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security*, 1(1):36–63, 2001.
- [KKLL07] JoonWan Kim, YongHo Kim, Hwaseong Lee, and DongHoon Lee. A practical inter-sensor broadcast authentication scheme. In Constantine Stephanidis, editor, *Proceedings of the Universal Access in Human Computer Interaction - UAHCI '07*, volume 4554 of *Lecture Notes in Computer Science*, pages 399–405. Springer Berlin Heidelberg, 2007.
- [KLP<sup>+</sup>07] Yong Kim, Hwaseong Lee, Jong Park, Laurence Yang, and Dong Lee. Key establishment scheme for sensor networks with low communication cost. In *Proceedings of Autonomic and Trusted Computing - ATC '07*, volume 4610 of *LNCS*, pages 441–448. Springer Berlin / Heidelberg, 2007.
- [Lee08] Tsern-Huei Lee. Simple dynamic user authentication protocols for wireless sensor networks. In *Proceedings of SENSORCOMM '08*, pages 657–660, 2008.
- [LLM07] Brian A. LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger Security of Authenticated Key Exchange. In *Proceedings of ProvSec '07*, pages 1–16, 2007.
- [LN04] Donggang Liu and Peng Ning. Multilevel  $\mu$ TESLA: Broadcast authentication for distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(4):800–836, 2004.

- [LNZJ05] Donggang Liu, Peng Ning, Sencun Zhu, and Sushil Jajodia. Practical broadcast authentication in sensor networks. In *Proceedings of MobiQ-uitous '05: Networking and Services*, pages 118–132. IEEE Computer Society, 2005.
- [LPW06] Mark Luk, Adrian Perrig, and Bram Whillock. Seven cardinal properties of sensor network broadcast authentication. In *Proceedings of Security of Ad Hoc and Sensor Networks - SASN '06*, pages 147–156. ACM, 2006.
- [Lyn02] Ben Lynn. Authenticated Identity-Based Encryption. Cryptology ePrint Archive, Report 2002/072, 2002. <http://eprint.iacr.org/>.
- [Mao03] Wenbo Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall Professional Technical Reference, July 2003.
- [Mer80] Ralph C. Merkle. Protocols for public key cryptosystems. In *IEEE Symposium on Security and Privacy*, pages 122–134, 1980.
- [MIC] MICA2. [http://bullseye.xbow.com:81/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2\\_Datasheet.pdf](http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf).
- [Mit02] Michael Mitzenmacher. Compressed bloom filters. *IEEE/ACM Transactions on Networks*, 10(5):604–612, 2002.
- [MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
- [Nes] NesC. <http://nesc.sourceforge.net/>.
- [NOP] NOPP. <http://www.nopp.org/>.
- [OAG<sup>+</sup>11] Leonardo B. Oliveira, Diego F. Aranha, Conrado P. L. Gouvêa, Michael Scott, Danilo F. Címara, Julio López, and Ricardo Dahab. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. *Computer Communications*, 34(3):485–493, 2011.

- [OTO05] Takeshi Okamoto, Raylin Tso, and Eiji Okamoto. One-Way and Two-Party Authenticated ID-Based Key Agreement Protocols Using Pairing. In *Proceedings of Modeling Decisions for Artificial Intelligence - MDAI'05*, volume 3558 of *LNCS*, pages 122–133. Springer-Verlag, 2005.
- [OX09] Suat Ozdemir and Yang Xiao. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, 53(12):2022 – 2037, 2009.
- [Per01] Adrian Perrig. The BiBa One-Time Signature and Broadcast Authentication Protocol. In *Proceedings of Computer and Communications Security - CCS '01*, pages 28–37, Philadelphia PA, USA, 2001.
- [PH97] Holger Petersen and Delta P. Horster. Self-certified keys - Concepts and Applications. In *Proceedings of Communications and Multimedia Security*, 1997.
- [PLP06] Krzysztof Piotrowski, Peter Langendoerfer, and Steffen Peter. How public key cryptography influences wireless sensor node lifetime. In *Proceedings of Security of Ad Hoc and Sensor Networks - SASN '06*, pages 169–176, NY, USA, 2006. ACM.
- [Poi05] David Pointcheval. *Contemporary Cryptology*, chapter Provable Security for Public Key Schemes, pages 133–189. Advanced Courses in Mathematics - CRM Barcelona. Birkhäuser Publishers, Basel, 2005.
- [PST<sup>+</sup>02] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. SPINS: Security Protocols for Sensor Networks. *Wireless Networks*, 8(5):521–534, 2002.
- [PSW04] Adrian Perrig, John A. Stankovic, and David Wagner. Security in wireless sensor networks. *Communications, ACM*, 47(6):53–57, 2004.
- [RLZ07] Kui Ren, Wenjing Lou, and Yanchao Zhang. Multi-user broadcast authentication in wireless sensor networks. In *Proceedings of Sensor, Mesh and Ad Hoc Communications and Networks - SECON '07*, pages 223–232, 2007.
- [RLZM07] Kui Ren, Wenjing Lou, Kai Zeng, and P.J. Moran. On broadcast authentication in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 6(11):4136–4144, 2007.

- [RMS08] Qiong Ren, Yi Mu, and Willy Susilo. Mitigating phishing with ID-based online/offline authentication. In *Proceedings of Australasian Information Security Conference - AISC '08*, volume 81, pages 59–64, 2008.
- [RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications, ACM*, 21(2):120–126, 1978.
- [RSA83] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems (reprint). *Communications, ACM*, 26(1):96–99, 1983.
- [Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.
- [SCL08] Frank Stajano, Daniel Cvrcek, and Matt Lewis. Steel, Cast Iron and Concrete: Security Engineering for Real World Wireless Sensor Networks. In *Proceedings of Applied Cryptography and Network Security - ACNS '08*, number 5037 in LNCS, pages 460–478. Springer Verlag, 2008.
- [Sha85] Adi Shamir. Identity-based Cryptosystems and Signature Schemes. In *Advances in Cryptology - CRYPTO 1984*, pages 47–53. Springer-Verlag, 1985.
- [Sma] SmartPoint. <http://www.ambient-systems.net/en/products/downloads.html>.
- [Sto05] Ivan Stojmenovi, editor. *Handbook of Sensor Networks - Algorithms and Architectures*. WileyBlackwell, November 2005.
- [SW04] Amit Sahai and Brent Waters. Fuzzy identity based encryption. Cryptology ePrint Archive, Report 2004/086, 2004. <http://eprint.iacr.org/>.
- [Tin] TinyOS. <http://www.tinyos.net/>.
- [TJY07] Huei-Ru Tseng, Rong-Hong Jan, and Wu Yang. An improved dynamic user authentication scheme for wireless sensor networks. In *Proceedings of GLOBECOM '07*, pages 986–990. IEEE, 2007.

- [Tmo] Tmote Sky. <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>.
- [Tos] TOSSIM, Simulating TinyOS Networks. <http://www.cs.berkeley.edu/~pal/research/tossim.html>.
- [TWZ05] Xiaojian Tian, Duncan S. Wong, and Robert W. Zhu. Analysis and improvement of an authenticated key exchange protocol for sensor networks. *Communications Letters*, 9(11):970 – 972, 2005.
- [Wan05] Yongge Wang. Efficient Identity-Based and Authenticated Key Agreement Protocol. Cryptology ePrint Archive, Report 2005/108, 2005. <http://eprint.iacr.org/>.
- [WAR06] Yong Wang, Garhan Attebury, and Byrav Ramamurthy. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 8(1-4):2–23, 2006.
- [WZCW06] Kirk H. M. Wong, Yuan Zheng, Jiannong Cao, and Shengwei Wang. A dynamic user authentication scheme for wireless sensor networks. In *Proceedings of Sensor Networks, Ubiquitous, and Trustworthy Computing - SUTC '06*, pages 244–251. IEEE Computer Society, 2006.
- [XMS05] Shidi Xu, Yi Mu, and Willy Susilo. Efficient authentication scheme for routing in mobile ad hoc networks. In *Proceedings of Embedded and Ubiquitous Computing - EUC '05*, volume 3823 of *LNCS*, pages 854–863. Springer, 2005.
- [Xu02] Ning Xu. A survey of sensor network applications. Survey Paper for CS694a, Computer Science Department, University of Southern California, 2002.
- [YRW11] Rehana Yasmin, Eike Ritter, and Guilin Wang. A Pairing-Free ID-based One-Pass Authenticated Key Establishment Protocol for Wireless Sensor Networks. In *Proceedings of Sensor Technologies and Applications - SENSORCOMM '11*, 2011.
- [ZSJN07] Sencun Zhu, Sanjeev Setia, Sushil Jajodia, and Peng Ning. Interleaved hop-by-hop authentication against false data injection attacks in sensor networks. *ACM Transactions on Sensor Networks*, 3(3):14, 2007.

- [ZSW08] Wensheng Zhang, Nalin Subramanian, and Guiling Wang. Lightweight and compromise-resilient message authentication in sensor networks. In *Proceedings of IEEE INFOCOM '08*, pages 1418–1426. IEEE, 2008.
- [ZW09] Li-Ping Zhang and Yi Wang. An ID-Based Key Agreement Protocol for Wireless Sensor Networks. In *Proceedings of Information Science and Engineering*, pages 2542–2545. IEEE Computer Society, 2009.
- [ZZR09] Rui Zhang, Yanchao Zhang, and Kui Ren.  $DP^2AC$ : Distributed privacy-preserving access control in sensor networks. In *Proceedings of IEEE INFOCOM '09*, pages 1251–1259, 2009.