Design and Analysis of Anonymous User Authentication Scheme for Wireless Sensor Networks

Bachelor Thesis

By

Olive Chakraborty

A thesis submitted to

BITS, Pilani

for the partial fulfillment of the degree of

Bachelors of Engineering in Computer Science

in

Department of Computer Science and Information Systems

May, 2015

Certificate

This is to certify that the thesis entitled "**Design and Analysis of Anonymous User Authentication Scheme for Wireless Sensor Networks**" being submitted by **Mr. Olive Chakraborty**, an undergraduate student (ID 2010B4A7698P) in the Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani Campus, Rajasthan 333031, India, for the award of **Bachelors of Engineering** in **Computer Science**, is an original research work carried by him under my supervision and guidance. The thesis has fulfilled all the requirements as par the regulation of **BITS Pilani** and in my opinion, has reached the standards needed for submission. The works, techniques and the results presented have not been submitted to any other university or Institute for the award of any other degree or diploma.

(**SK Hafizul Islam, Ph.D**)
Assistant Professor
Department of Computer Science and Information Systems
Birla Institute of Technology and Science
Pilani Campus, Rajasthan 333031, India. May, 2015

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

Acknowledgments

First of all, I would like to take this opportunity to thank my supervisor Dr. SK Hafizul Islam without whose effort this thesis would not have been possible. I am so grateful to him for working tirelessly after me, answering my doubts whenever and wherever possible. I am most grateful to Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani Campus, Rajasthan 333031, India, for providing me this wonderful opportunity to complete my bachelor thesis. I would like to thank my friends in France, in Team CARAMEL, INRIA for providing me with help as and when required. I would like to thank Maike Massierer for being a great motivator and a great friend.

And last but the biggest of all, I want to thank my parents, for always believing in me and letting me do what I wanted, but keeping a continuous check that I never wandered off the track from my goal.

Olive Chakraborty

Adm. No.: 2010B4A7698P
May, 2015

Abstract

This thesis investigates the authentication problems in wireless sensor networks (WSNs), particularly user authentication, and proposes an 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 infeasible for WSNs due to the unique features of sensor networks. Designing a new authentication scheme for WSNs, on the other hand, is a challenging task. This thesis proposes a secure and robust user authentication scheme for WSNs using hash function and elliptic curve. This protocol can be applied in WSNs independently tackling individual security problems to achieve different level of security. The security of the proposed scheme is based on the difficulty of Computational Diffie-Hellman problem and one-wayness of the hash function in the random oracle model. The performance evaluation results showed that the proposed protocol is efficient compared to the existing authentication schemes for WSNs, giving a reasonable trade-off between security and efficiency.

Keywords: Wireless Sensor Networks, Authentication, Elliptic Curve Cryptography, User Anonymity, Random Oracle Model, Password, Biometric, Hash function.

Contents

List of Figures

List of Tables

List of Acronyms

WSN	Wireless Sensor Networks
GW Node	Gateway Node
DoS	Denial of Service
DHP	Diffie Hellman Problem
ECC	Eliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDHP	Elliptic Curve Diffie-Hellman Problem
PKC	Public key Cryptosystem

Chapter 1

Introduction

This chapter presents the introduction of the thesis that includes the brief description of WSNs, authentication problems in WSNs and the adopted approach to address the problems. This chapter also presents the scope of this thesis and the contributions of the thesis.

1.1 Wireless Sensor Networks

A Wireless sensor Network can be broadly described as a network of nodes that makes a collaborative effort in sensing certain specified data around its periphery and thereby controls the surrounding environment [9]. In WSNs, each node consists of processing capability ,it may contain multiple types of memory like program, data and flash memories, having a Radio Frequency (RF) transceiver , a power source, and accommodating various sensors and actuators. The nodes communicate wireless and self-organize after being deployed in an ad-hoc fashion. It is usually a collection of a data acquisition network and a data dissemination network. The data acquisition network consists of the actual sensor nodes along with the mobile or stationary Gateway Node (GW), the data dissemination network is a collection of wired and wireless networks that is involved in post-processing of the acquired data. However, the dissemination network is more equipped with computing, storage and power level capacity as compared to acquisition network.

1.2 Applications of WSNs

Wireless sensor network provides one major advantage over wired networks: immunity from cabling costs, which gives rise to numerous applications. Some of them include

- Disaster handling: Forest Fire Detection, Flood Detection, Earthquake, Detection and surveillance.

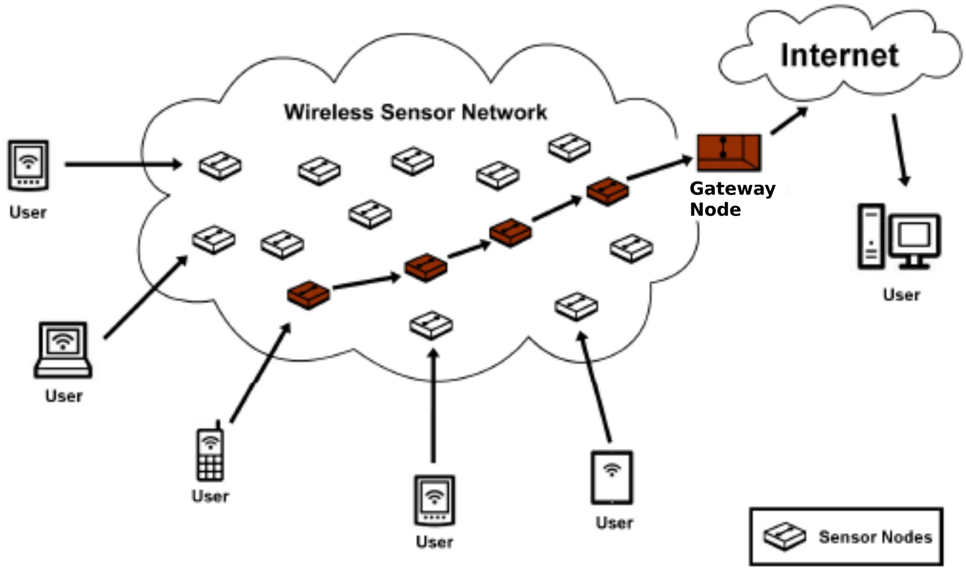

Figure 1.1: Overview of Wireless Sensor Networks (WSNs).

- Military applications: Enemy Tracking, Monitoring Enemy Forces, Biological and Chemical Attacks, Ammunition detection, Nuclear detection, Battlefield Surveillance, etc.
- Wildlife and Ocean monitoring
- Manufacturing machinery performance monitoring
- Structural health Monitoring

1.3 Security Concerns in WSNs

As the wireless industry explodes, it faces the growing need for security. Applications in the sector of economy such as health care, financial services, and government depend on the underlying security available for wireless computing environment. Both for authenticated and private web transactions and signed and encrypted messaging, an efficient public key infrastructure (PKI) is needed. Privacy assures that if an eavesdropper is intercepting communication messages are not. On the other hand, authentication assures that any unauthorized user cannot fraudulently obtain his/her required services from home domains [15]. 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. Also, 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 and robust security solutions for WSNs. This restriction has significantly impacted the field of application security. Thus, efficient and secure mechanisms for WSNs are required in order to provide the authenticity and privacy online. Remote user mutual authentication with key agreement schemes can be used to achieve some of these security goals and is the major focus of this thesis.

1.4 Scope and Adopted Approach

To address the above privacy issues, this thesis proposes an authentication protocol for WSNs using Elliptic Curve Cryptography (ECC)-based schemes. The proposed framework is comprised of an 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 method is discussed in detail in Chapter 4.

1.5 Thesis Contribution

The main contributions of the thesis includes

1. Proposes an user anonymity-preserving authentication scheme with key agreement for WSNs using ECC.

2. Formally analyzes the security of the newly designed protocol as well as its performance.

3. The scheme, as compared to the existing schemes, not only authenticates the users but, also establishes a session key between the user and the sensor node after successful mutual authentication.

4. The scheme provides many security and robustness features of user authentication scheme for WSNs.

1.6 Roadmap of the Thesis

The structure of the thesis is as follows:

1. The Chapter 1 is an introductory part which discusses the scope of the thesis, about the contribution of this thesis and the motivation for writing it.

2. The Chapter 2 provides the background of WSNs, applications of WSNs, security aspects of WSNs, different attacks existing in WSNs and the user authentication systems in WSNs.

3. The Chapter 3 discusses in details the basis of ECC, some definitions for security and some mathematically intractable problems.

4. The Chapter 4 introduces the proposed authentication framework after highlighting the motivations behind this work. It also describes the the various phases involved in the protocol. the protocols have also been pictorially represented.

5. The security analysis comprises of Chapter 5, where, a formal security model of the proposed protocol has been discussed. Also the threat model and trust model used by the proposed framework together with the assumptions made by it.

6. The Chapter 6 comprises of the conclusion and further work of the thesis.

Chapter 2

Security in Wireless Sensor Networks

This chapter describes different concepts required for the understanding of this thesis. This chapter presents an overview of WSNs security including security objectives, types of attackers and security attacks in WSNs. It also highlights the constraints in WSNs which are barriers to provide security.

2.1 Characteristics of WSNs

A WSN can be considered as a special case of ad-hoc networks. A wireless ad-hoc network does not rely on a fixed infrastructure. Instead, the nodes in an ad-hoc network organize themselves on the go 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. This section reviews the particular characteristics of a WSN and security concerns in a typical WSN.

The following summarizes some of properties 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.

- *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.

- *Dynamic network topology*: The sensor network topology is unknown prior to the deployment.
- *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 Gateway (GW) node.

These characteristics of WSNs offer an advantage to any adversary ($\mathscr{A}$) who intends to compromise the 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.

2.2 Security Goals

Security is sometimes seen as a standalone component of a system's architecture, where security is provided by a separate module. However, this is usually a flawed approach to the network security. To achieve a secure system, security must be integrated into every component, since the components are designed without security can become a point of attack.

Authentication enables each message sender in the sensor network, including the GW nodes, sensor nodes and the 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 WSNs 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 WSNs, 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.

Key establishment and Trust setup means when setting up a sensor network, one of the first requirements is to establish cryptographic keys for later use. Public key cryptography is another option beyond the capabilities of today's sensor networks. Its main advantage is that a node can set up a secure key with any other node in the network.

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 of the message is required in WSNs to protect the 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.

Privacy in WSNs is of great concern. Sensor networks have also thrust privacy concerns to the forefront. The most obvious risk is that ubiquitous sensor technology might allow ill-intention individuals to deploy secret surveillance networks for spying on unaware victims.

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.

2.2.1 Attacks on WSN

The WSNs are in general a subclass of Wireless networks. So all the attacks possible on Wireless networks can be directed at WSNs. However, due to the additional constraints talked about previously, WSNs gave rise to whole new set of attacks that can be distinguished into two sets, *"mote-class attacks"* and *"laptop class attacks"*. The Figure 2.1 includes different attacks in WSNs.

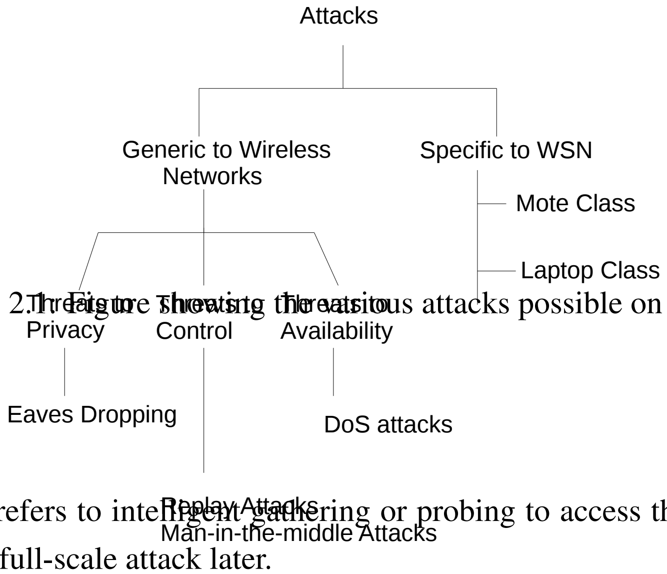

Figure 2.1: Figure showing the various attacks possible on WSNs

Threats to Privacy

Reconnaissance: It refers to intelligent gathering or probing to access the vulnerabilities in a network, to launch a full-scale attack later.

Eavesdropping: It is an operation to learn the aggregate data that is being collected by the entire network.

Threats to Control

Man-in-the-middle attack: In this type of attack, the attacker intrudes into the network and makes an effort to establish an independent connection between a set of sensor nodes and the GW node. The nodes in the network are unaware that the entire flow control is being handled by the attacker.

Replay Attack: This is a common attack in WSN, whereby an attacker is able to intercept user data and re-transmit user data at a later time.

Threat to Availability

Denial of Service(DoS) attack: A DoS attack occurs when an attacker floods the victim with bogus or spoofed packets with the intent to lower the response rate of the victim. In the worst-case scenario, it makes the victim totally unresponsive.

Collision attack: Collision attacks target the MAC layer to create costly exponential back-off. Whenever collision occurs, the nodes should re-transmit packets affected by collision, thus, leading to multiple re-transmissions. The amount of energy expended by the attacker is much less than the energy expended (battery exhaustion) by the sensor nodes.

As attacks on WSNs become more sophisticated, the demand for new security solutions is continually increasing. Hence, an array of new security schemes have been designed and implemented in the past decade [24][11]. Any security suite must ensure authentication, integrity, confidentiality, availability, access control, and non-repudiation. That's why authentication has become one of the important areas to be taken care when we talk about wireless protocols.

2.3 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).

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

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

2.3.1 Authentication in Wireless Sensor Networks

Beneson *et al.*'s [3] talked about the privacy and distinguished between the insider security and the outsider security in WSNs as follows:

- *Insider Security* addresses secure communication in-between the sensors and between the sensors and the GW node(s).
- *Outsider security* addresses secure communication between the WSN (sensors and GW node) and the outside user.

Authentication, which is a part of both insider and outsider security, is a crucial security requirement in WSNs. 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 GW node 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 de-synchronize the network or to disturb the receiver's clock. A countermeasure to this kind of attacks is authentication. Authentication in typical WSNs can be classified into four types:

- GW node to Sensor node authentication,
- Sensor node to other sensor node authentication,
- Sensor Node to User node authentication,
- GW node to User node authentication.

The GW node to sensor nodes authentication has been widely addressed by the current authentication schemes for WSNs [24, 20, 19, 8, 4]. Sensor Node to other sensor node authentication has been dealt with by [30]. 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 WSNs makes it difficult for everyone to deploy own sensor networks to collect data of their interests. The users of the sensor nodes data, thus take use of these deployment agencies of the large scale WSNs to obtain this data. 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.

Providing a secure user access to sensor nodes data requires two basic tasks:

- *User Authentication*: 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.
- *Session key establishment*: This enables secure transmission of confidential sensor nodes data to users after authentication.

2.3.2 User Authentication

User authentication in WSNs may be implemented using some user credentials for instance identity, password, biometric, known only to the user. It might require sensor nodes to store identity, password and Biometric triplet of each user. However, a single compromised node will reveal the passwords and biometric information of all the users. Alternatively, user authentication may be enforced with a Public Key Cryptosystem (PKC) with public and private keys. A simple approach to handle user authentication is a centralized mechanism. The centralized user authentication schemes described in [29, 27, 6, 16] divide the user authentication process into: registration phase, login phase and authentication phase. In a centralized approach, the user sends his login request to a central entity, say a GW node. The GW node, after successful user authentication, forwards the user query to the sensor nodes to obtain the requested data from them. The GW 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 GW node. The GW node then verifies the legitimacy of the user request and decides whether the access should be granted or not. The GW 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 [30].

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 GW node). Assume that the third party authenticates the users and if it fails, then 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 GW 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.

2.3.3 Session Key establishment

To establish a pair-wise key between the user and the sensor nodes, [12] proposes a key establishment scheme based on the self-certified-PKC [25]. In this scheme, the user sends a request along with his identity 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 [30].

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 approach 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 all the previous communication between the same participants, eavesdropped by the adversary.

Chapter 3

Elliptic Curve Cryptography

This chapter reviews the cryptographic tools, primitives and notions, and describes the computational assumptions used in this thesis. It also summaries the various formal definitions that are required to be known for understanding of protocol security.

3.1 Introduction

Elliptic curves have been studied by mathematicians for a long time. They have been used to solve a wide range of problems some of them being the Fermat's Last theorem and Congruent number Problem [13]. In 1985, Koblitz and Miller independently proposed using elliptic curves to design PKCs [21]. Since it has been receiving acceptance throughout the world when standardized institutions across the world specified elliptic curve protocols. The algebraic structures of these elliptic curves form the basis of elliptic curve cryptography (ECC). The fundamental security of this system relies on the difficulty of solving the discrete logarithm problem in the elliptic curve group.

3.2 Cryptography Basics

Cryptography involves the design and analysis of mathematical methods to enable secure communications in the presence of malicious users (adversaries).

Basic Model

In Figure 3.1, entities A (Alice) and B (Bob) are communicating over an unsecured channel. We assume that all the communications take place in the presence of an adversary E (Eve) whose objective is to defeat any security services being provided to A and B.

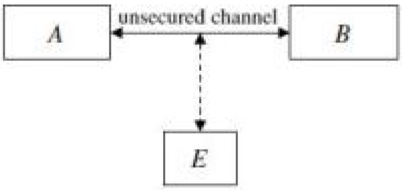

Figure 3.1: Basic communications model

Security Goals

There are few security measures that come into picture while we design any cryptographic protocol.

1. *Confidentiality*: It means that keeping data secret from all but those authorized to see it, messages sent by A to B should not be readable by E.

2. *Data origin Authentication*: It means that verifying the source of data, B should be able to verify that data supposedly sent by A indeed originated with A.

3. *Data Integrity*: It ensuring that data has not been altered by unauthorized means, B should be able to detect when data sent by A has been modified by E.

4. *Entity authentication*: It means that verifying the identity of an entity, B should be convinced of the identity of the other communicating entity.

5. *Non-repudiation*: It means that preventing an entity from denying previous commitments or actions, when B receives a message purportedly from A, not only is B convinced that the message originated with A, but B can convince a neutral third party of this, thus A cannot deny having sent the message to B.

3.3 Weierstrass Equation

Let p be a prime number, and let $\mathbb{F}_q$ denote a finite field of order q and modulo p (i.e., characteristic is p). The non-singular Weierstrass Equation over $\mathbb{F}_q$ is defined as

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{3.1}$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q$. The set $\mathbb{F}_q$ consists of $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ along with point at infinity(∞) [28]

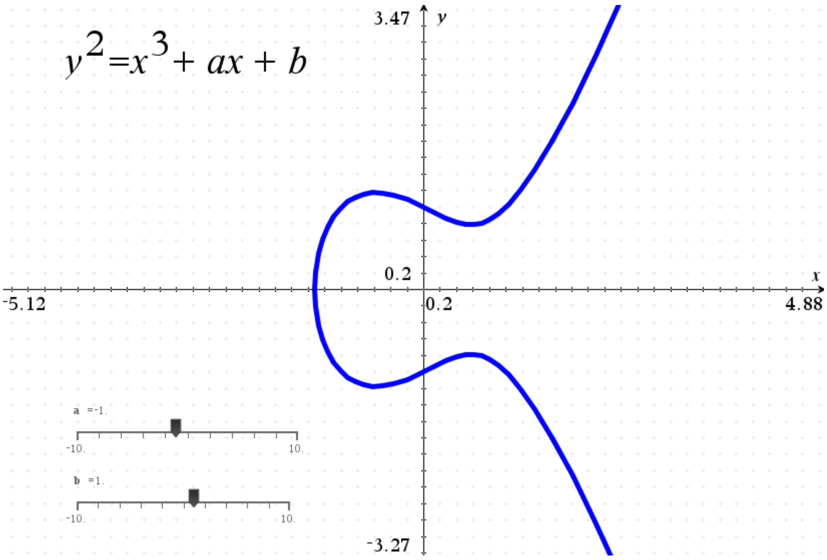

Figure 3.2: A Elliptic Curve described by a Weierstrass Equation

3.3.1 Elliptic Curve groups

Let $\mathbb{F}_q$ be an arbitrary field. An elliptic curve E over a field $\mathbb{F}_q$ is a projective non-singular curve defined over $\mathbb{F}_q$ of genus 1 together with a point $0 \in E$ defined over $\mathbb{F}_q$. A Weierstrass equation for an elliptic curve $E/\mathbb{F}_q$ is an equation of the form:

$$y^2 = x^3 + Ax + B \tag{3.2}$$

where $A,\ B \in \mathbb{F}_p$. As the definition of Elliptic Curve requires it to be non-singular, which means geometrically it has no cusps, self-intersections or isolated points.Algebraically the discriminant has been calculated as $\Delta = -16(4A^3 + 27B^2)$. The curve is non-singular if and only if the discriminant is non-zero or $4A^3 + 27B^2 \not\equiv 0(modp)$. A pair (x, y), where $x, y \in \mathbb{F}_q$, is a point on the curve E if (x, y) satisfies the equation. A example of an elliptic curve has been shown in Figure 3.2.

3.4 Discrete Logarithm

Definition 1. *Let y be an arbitrary integer relatively prime to n. Let g be the primitive root of n. Let $(G, *)$ be a multiplicative cyclic group with generator g. Let x be an integer randomly selected from the interval $[1, \phi(n) - 1]$ where $\phi(n)$ is the totient function.*

$$y = g^x(modn) \tag{3.3}$$

The number x is called the discrete logarithm of y with respect to base g modulo n.

Corollary 1. *The **discrete logarithm problem** in G is defined as the problem of determining x given y, g and n [22].*

3.4.1 Discrete Logarithm Systems

The first discrete logarithm (DL) system was the key agreement protocol proposed by Diffie and Hellman in 1976 [7]. In discrete logarithm systems, a key pair is associated with a set of public domain parameters (p, q, g). Here, p is a prime, q is a prime divisor of $p-1$, and $g \in [1, p-1]$ has order q ($t = q$ is the smallest positive integer satisfying $g^t \equiv 1(modp)$). A private key is an integer x that is selected uniformly at random from the interval $[1, q{-}1]$, and the corresponding public key is $y = g^x \ modp$. The problem of determining x given domain parameters (p, q, g) and y is the discrete logarithm problem (DLP).

3.4.2 Elliptic Curve Discrete Log Problem (*ECDLP)*

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Let P be a point in $E(\mathbb{F}_q)$, and suppose that P has prime order n. Then the cyclic subgroup of $E/\mathbb{F}_q$ generated by P is $\langle P \rangle = \{\infty,\ P,\ 2P,\ 3P, \cdots,\ (n-1)P\ \}$. The prime q, the equation of the elliptic curve E, and the point $N \in \langle P \rangle$ and its order n, are the public domain parameters.

Let $Q \in \mathbb{F}_q$ of order q. A private key is an integer $d \in \mathbb{Z}$ that is selected uniformly at random from the interval $[1, n-1]$, $Q = dN$.

The problem of determining d given the domain parameters and Q is the elliptic curve discrete logarithm problem (ECDLP). The integer d is called the discrete logarithm of Q base N, where $d = \log_N Q$.

3.5 Diffie-Hellman Problem

The **Diffie-Hellman problem (*DHP*)** is a mathematical problem first proposed by Whitfield Diffie and Martin Hellman [7] in the context of cryptography. The problem is stated as follows.

Definition 2. *Let G be a cyclic group of order of a large prime p (in practice it is better to choose the number p such that $(p-1)/2$ is also prime) generated by an another prime element g, and g being the primitive root of p. Given g and the values g^x and g^y, computing the value of g^{xy} is the Diffie-Hellman Problem.*

3.5.1 Computational Diffie-Hellman Assumptions

Definition 3. *Let G be a finite cyclic group of prime order p generated by g. The CDH assumption states that, given (g, g^x, g^y) for a randomly chosen generator g and random $x, y \in \{0, 1, 2, \cdots, p-1\}$, it is computationally intractable to compute the value g^{xy}.*

3.5.2 Elliptic Curve Diffie-Hellman Problem (*ECDHP*)

Corollary 2. *The (computational) Elliptic Curve Diffie-Hellman Problem is stated as: Given an elliptic curve E defined over a finite field $\mathbb{F}_q$, a point $P \in E(\mathbb{F}_q)$ of order n, and points $A = aP$, $B = bP \in \langle P \rangle$, calculation of abP is hard.*

If ECDLP can be solved on $\langle P \rangle$, then ECDHP can also be solved efficiently solved by first finding a from (P,A) and then computing aB. Thus, thr ECDHP is no harder than ECDLP to solve.

Corollary 3. *The (decisional) Elliptic Curve Diffie-Hellman Problem is stated as: Given an elliptic curve E defined over a finite field $\mathbb{F}_q$, a point $P \in E(\mathbb{F}_q)$ of order n, and points $A = aP$, $B = bP \in \langle P \rangle$, the probability that the adversary ($\mathscr{A}$) succeeds in where the value of C is abP.*

Remark 1. *Elliptic Diffie-Hellman key exchange requires Alice and Bob to exchange points on an elliptic curve. A point Q in $E(\mathbb{F}_p)$ consists of two coordinates $Q = (x_Q, y_Q)$, where x_Q and y_Q are elements of the finite field $\mathbb{F}_p$, so it appears that Alice must send Bob two numbers in $\mathbb{F}_p$. However, those two numbers modulo p do not contain as much information as two arbitrary numbers, since they are related by the formula $y_Q^2 = x_Q^3 + Ax_Q + B \ \ in \ \ \mathbb{F}_p$*

Note that Eve knows A and B, so if she can guess the correct value of x_Q, then there are only two possible values for y_Q, and in practice it is not too hard for her to actually compute the two values of y_Q.

There is thus little reason for Alice to send both coordinates of Q_A to Bob, since the y-coordinate contains so little additional information. Instead, she sends Bob only the x-coordinate of Q_A. Bob then computes and uses one of the two possible y-coordinates. If he happens to choose the "correct" y, then he is using Q_A, and if he chooses the "incorrect" y (which is the negative of the correct y), then he is using Q_A. In any case, Bob ends up computing one of $\pm n_B Q_A = \pm(n_A n_B)P$. Then Alice and Bob use x-coordinate as their shared secret value, since that x-coordinate is the same regardless of which y they use.

3.6 Advantages of ECC over others

The major advantage that Elliptic Curve Cryptography provides is the reduction in the key size, while providing the same level of security as RSA [17]. Thus provides faster computations, reduced power consumption and reduced storage.

An elliptic curve over a 163-bit prime field currently gives the same level of security as 1024-bit RSA modulus or Diffie-Hellman primes. The difference increases with increase in desired security levels. 571-bit ECC is currently equivalent to 15,360-bit RSA/DH/DSA.

This growing difference in key length for equivalent security levels accounts for performance advantages to be obtained from substituting RSA with ECC in Public Key Cryptographic protocols.

Table 3.1: Table showing key sizes for equivalent security levels.[14]

Symmetric	ECC	DH/DHA/RSA
80	163	1024
128	283	3072
192	409	7680
256	571	15,360

At 163-bit ECC/1024-bit RSA security level, an elliptic curve exponentiation for general curves over arbitrary prime fields is roughly 5 to 15 times faster than RSA operations, depending upon the platform and optimization. At 256-bit ECC/3072-bit RSA, the ratio has increased to between 20 to 60 times. For securing 256-bit AES key, 521-bit ECC is expected to have an advantage of being 400 times faster than 15,360-bit RSA.

Table 3.2: Sample Elliptic curve exponentiation and RSA timings(*in milliseconds*).[14]

Processor	MHz	163-bit ECC	1024-RSA
Ultra SPARC II	450	6.1	33.8
StrongARM	200	22.9	199.5

Chapter 4

Proposed Method

In this chapter, we propose our user authentication protocol for WSNs. This protocol is a biometric and password based authenticated key agreement scheme using ECC for wireless sensor networks. The proposed scheme has four phases: registration phase, login phase, key-agreement phase and Password Change phase.

4.1 Notations

The following notations are used throughout in the proposed scheme.

Table 4.1: Different notations and their meanings

Notations	Descriptions
U	User
ID_U	Identity of U
PW_U	Password of U
B_U	Biometric template of U
GW	Gateway node of WSN
ID_{GW}	Identity of GW
S_j	Sensor node
s	Secret master key of GW node
y	Secret known to only GW node sensor node S_j
$h(\cdot)$	A secure one-way hash function (e.g., SHA-1)
$E_k(\cdot)$	A symmetric encryption function using AES [2] with key k
$d(\cdot)$	A parametric distance function
τ	A predetermined threshold for biometric verification
F_p	A finite field
E	An elliptic curve over F_p
$E(F_P)$	Set of all the points on E
P	Base point of $E(F_P)$ of order n
$\oplus$	Bitwise XOR operation
$\|\|$	Message concatenation operator

We introduce a new global variable *free* for the GW node, which is visible to all the other nodes in the network. The variable *free* is set to 1 (one), if the GW node is free for service i.e, not busy and GW node accepts the registration or login requests. After accepting the registration or login requests, the GW node sets *free* variable to 0 (zero). This factor is takes care of certain aspects of the DoS attacks. Suppose if the *free* value is 1 still the GW node is not providing service, the other nodes can understand that the network has been compromised by a DOS attacker.

4.2 Registration Phase

Before the user is able to use the sensor node for its purposes for the first time, it needs to register itself with the GW node of the network. The following steps are performed.

User U/Smartcard		**GW Node**
ID_U, PW_U, B_U Selects a random number b_U $V = h(PW_U \| \| b_U)$ $Q = h((B_U \oplus ID_U)) \| \| b_U)$		(s, y)
	$\xrightarrow{\langle ID_U, V, Q \rangle}$ (via a insecure channel)	
		If $free \neq 1$, reject, else Update $free = 0$ $K_U = h(ID_U \| \| s)$ $L_U = h(ID_U \| \| V \| \| Q)$ $W_U = h(ID_U \oplus (V \| \| Q)) \oplus K_U$ Generate a random number n $MID = E_s(ID_U \| \| n)$ Update $free = 1$
	$\xleftarrow{\langle \text{Smartcard}(ID_{GW}, T_U, W_U, MID, h(\cdot)) \rangle}$ (via a insecure channel)	
Store b_U, ID_U into the smartcard's memory Smart card holds $\langle ID_U, ID_{GW}, b_U, L_U, W_U, MID, h(\cdot) \rangle$		

Figure 4.1: Registration Phase

1. U inputs his/her identity ID_U, password PW_U and biometric B_U. U also generates a random number b_U and computes $V = h(PW_U||b_U)$ and $Q = h((ID_U \oplus B_U))||b_U)$. Then U sends $\langle ID_u, V, Q \rangle$ to GW node in a secure channel.

2. On receiving $\langle ID_u, V, Q \rangle$, if GW is free for receiving request of service, it computes $K_U = h(ID_U||s)$. It also computes $L_U = h(ID_U||V||Q)$ and $W_U = h(ID_U \oplus (V||Q)) \oplus K_U$. The GW node generates a a random number n and computes $MID = (ID_U||n)_s$.

3. The GW node then stores $\langle ID_{GW}$, L_U, W_U, MID, $h(\cdot)\rangle$ on a new smartcard and sends the smartcard to U over a secure channel, and opens itself up for further requests. On

receiving the smartcard, U stores b_U, Q, ID_U on the smartcard. Finally, the smart card contains the information $\langle ID_U, ID_{GW}, , b_U, L_U, W_U, MID, h(\cdot)\rangle$

4.3 Login Phase

For U to access services and data from the WSN, it needs to login, performing the following steps:

1. U inserts his smartcard into the card reader and inputs his ID_U, password PW_U and personal Biometric B_U^*.
2. The smart card computes $Q^* = h((ID_U \oplus B_U^*)||b_U)$ and read Q from the smartcard. If $d(Q^*, Q) > \tau$, then rejects the session message. Otherwise, the smart card compute $V^* = h(PW_U||b_U)$ and hence compute $L_U^* = h(ID_U||V^*||Q^*)$ and checks if $L_U^* = L_U$. If true then the smartcard executes the following operations.
    - Compute $K_U = W_U \oplus h(ID_U \oplus (V||Q))$.
    - Generate $\alpha \in F_P$.
    - Compute $A = \alpha P$.
    - Choose a current timestamp T_U.
    - Compute $M = h(ID_U||A||K_U||T_U)$.
    - Send the login message $\langle MID, A, M, T_U\rangle$ to S_j over a public channel.

4.4 Key Agreement Phase

1. When S_j receives the login request at time T_U^*, it performs the following steps:
    - Check if the flag $free \neq 0$, then reject, otherwise proceed to next step.
    - Verify if $(T_U^* - T_U) < \Delta T$. If it fails the authentication phase is aborted, where ΔT is the expected time interval for the transmission delay of WSN. If on the contrary, it proceeds to the next step.
    - Change the $free$ flag to 0.
    - Generate a random number $\beta \in F_P$ and choose a current time stamp T_S.
    - Compute $B = \beta P$.
    - Compute $N = h(y||MID||A||T_U||M||B||T_S||ID_S)$.

User U/Smartcard	**Sensor S_j**	**GW-Node**
Insert smartcard Input ID_U, PW_U, B_U $Q^* = h((B_U^* \oplus ID_U)\|\|b_U)$ $V* = h(PW_U^*\|\|b_U)$ $L_U^* = h(ID_U^*\|\|V^*\|\|Q^*)$ If $d(L_U^*, L_U) > \tau$ Reject $K_U = W_U \oplus h(ID_U \oplus (V^*\|\|Q^*))$ Generate $\alpha \in \mathbb{F}_P$ $A = \alpha P$ Choose T_U $M = h(MID\|\|A\|\|K_U\|\|T_U)$		
$\xrightarrow{\langle MID, M, A, T_U \rangle}$ (via a secure channel)		
	Verify $(T_U^* - T_U) \leq \Delta T$ Generate $\beta \in \mathbb{F}_P$ $B = \beta P$ Choose T_S $N = h(y\|\|MID\|\|M\|\|A\|\|B\|\|T_S\|\|ID_S)$ $M_2 = E_y(MID, M, N, T_U, A, B, ID_S)$	
	$\xrightarrow{\langle M_2, T_S \rangle}$ (via a secure channel)	
		If $free \neq 1$, reject Else Update $free = 0$ Decrypt M_2 using master key y Verify $(T_S^* - T_S) \leq \Delta T$ Decrypt MID using master key s and get ID_U $N^* = h(y\|\|MID\|\|M\|\|A\|\|B\|\|T_S\|\|ID_U)$ Verify $N^* = N$ $K_U^* = h(ID_U\|\|s)$ $M^* = h(MID\|\|A\|\|K_U^*\|\|T_U)$ If $M^* \neq M$ Reject Generate n' $MID' = E_s(ID_U\|\|n')$ Choose T_G $\gamma = h(y\|\|MID'\|\|A\|\|T_U\|\|M\|\|ID_S\|\|B\|\|T_S\|\|T_G\|\|ID_{GW})$ $\delta = h(ID_U\|\|A\|\|K_U\|\|T_U\|\|B\|\|T_S\|\|ID_{GW})$ $M_3 = E_y(\gamma, h(\delta), MID')$ Update $free = 1$
		$\xleftarrow{\langle M_3, T_G \rangle}$ (via a secure channel)
	Decrypt M_3 using master key y Verify $(T_G^* - T_G) \leq \Delta T$ $\gamma^* = h(y\|\|MID'\|\|A\|\|T_U\|\|M\|\|ID_S\|\|B\|\|T_S\|\|T_G\|\|ID_{GW})$ If $\gamma^* \neq \gamma$ Reject $K = \beta A = \alpha\beta P$ $sk = h(MID'\|\|ID_S\|\|ID_{GW}\|\|A\|\|B\|\|(K \oplus M))$ $\rho = h(B\|\|T_S\|\|h(\delta)\|\|sk)$	
	$\xleftarrow{\langle MID', B, T_S, T_G, h(\delta), \rho \rangle}$ (via a secure channel)	
Verify $(T_S^* - T_S) \leq \Delta T$ Verify $(T_G^* - T_G) \leq \Delta T$ $K^* = \alpha B = \alpha\beta P$ $\delta^* = h(ID_U\|\|A\|\|K_U\|\|T_U\|\|B\|\|T_S\|\|ID_{GW})$ If $h(\delta^*) \neq h(\delta)$ Reject $sk^* = h(MID'\|\|ID_S\|\|ID_{GW}\|\|A\|\|B\|\|(K^* \oplus M)$ $\rho^* = h(B\|\|T_S\|\|h(\delta*)\|\|sk^*)$ If $\rho^* \neq \rho$ Reject Update MID by MID' in smartcard		

Figure 4.2: Login and authentication phase of the proposed scheme

- Send the message $M_2 = E_y(MID, m, T_U, A, B, T_S, ID_S, N)$, T_S to the GW node over a public channel.

2. The GW node then decrypts the MID to get ID_U. It computes $N^* = h(y||MID||M||A||B||T_S||ID_U)$ and checks if $N = N^*$. If false then the process stops immediately otherwise,

    - Compute $K_U = h(ID_U||s)$ and subsequently $M^* = h(ID_U||A||K_U||T_U)$. Check if $M^* = M$, if false, the authentication process is stopped, otherwise,
    - Generate a random number n', a time stamp T_G and compute $MID' = (ID_U||n')_s$
    - Compute $\gamma = h(y||MID'||A||T_U||M||ID_S||B||T_S||T_G||ID_{GW})$ and $\delta = h(ID_U||A||K_U||T_U||B||T_S||ID_{GW})$
    - Update $free = 1$.
    - Send the message $M_3 = E_y(\gamma, h(\delta), MID')$, T_G to S_j over a public channel.

3. S_j receives $\gamma, h(\delta), T_G, MID'$ from the GW node, it needs to verify its origin. It follows the following steps:

    - Verify if $(T_G^* - T_G) < \Delta T$. If it fails the authentication phase is aborted, where ΔT is the expected time interval for the transmission delay of WSN. If on the contrary, it proceeds to the next step.
    - Compute $\gamma^* = h(y||MID'||A||T_U||M||ID_S||B||T_S||T_G||ID_{GW})$ and verify if $\gamma^* = \gamma$. If not then the process is aborted immediately, otherwise, it follows in to the next step.
    - A key $K = \beta A = \alpha\beta P$ is evaluated.
    - Session key is evaluated as $sk = h(MID'||ID_S||ID_{GW}||A||B||(K \oplus M))$.
    - A mutual authenticator $\rho = h(B||T_S||h(\delta)||sk)$ is computed an then it sends (MID', B, T_S, T_G, $h(\delta)$, ρ) to U.

4. Assume that, U receives the message (MID', B, T_S, T_G, $h(\delta)$, ρ) at time T_S^* and then he doses as follows:

    - U verifies the two time stamps T_S and T_G.
    - The key K is calculated again, but this time by the user node. Then the mutual authenticator δ is computed.
    - Evaluate $h(\delta^*) = h(\delta)$. If this is false, the the process is aborted and no session key is established, otherwise it continues to the next step.

- The session key sk is computed by the user as $sk^* = h(MID'||ID_S||ID_{GW}||A||B||(K^* \oplus M))$.
- The second authenticator ρ is computed and crosschecked against the received value from the sensor. If it matches , the user updates the new MID' over the old MID.

4.5 Password and Biometric Change Phase

In this section, we give the password change/update phase. In the password or biometric change phase , when the user wants to change his password PW_U to a new password PW_U^*, or his biometric B_U to B_U^*, he inserts his smartcard into the terminal and enters his ID_U and password. The following procedure is followed:

User U	**Smartcard**
Enters ID_U and PW_U	
	Compute $Q^* = h((B_U \oplus ID_U)||b_U)$.
	Compute $V^* = h(PW_U^*||b_U)$.
	Evaluate $L_U^* = h(ID_U^*||V^*||Q^*)$
	If $d(L_U^*, L_U) > \tau$
	Reject
	Ask for new password and biometric from U
Enters PW_U^n and B_U^n	
	Compute $Q^n = h((B_U^n \oplus ID_U)||B_U^n)$
	Compute $V^n = h(PW_U^n||B_U^n)$
	Compute $L_U^n = h(ID_U||V^n||Q^n)$.
	Compute $W_U^n = W_U \oplus h(ID_U \oplus (V||Q)) \oplus h(ID_U \oplus (V^n||Q^n))$
	Update L_U^n and W_U^n in place of L_U and W_U respectively.

Figure 4.3: Password and Biometric change Phase

1. Smartcard validates his ID_U from the data stored in the card.
2. It computes $Q^* = h((B_U \oplus ID_U)||b_U)$.
3. Compute $V^* = h(PW_U^*||b_U)$.
4. Evaluate $L_U^* = h(ID_U^*||V^*||Q^*)$.
5. Compare if $d(L^*, L) < \tau$. If false process stops otherwise proceed to the next step.
6. The user is asked for new password PW_U^n and Biometric B_U^n.
7. Smartcard computes $Q^n = h((B_U^n \oplus ID_U)||B_U^n)$ and $V^n = h(PW_U^n||B_U^n)$.
8. Computes $L_U^n = h(ID_U||V^n||Q^n)$.

9. Compute $W_U^n = W_U \oplus h(ID_U \oplus (V||Q)) \oplus h(ID_U \oplus (V^n||Q^n))$

10. Store L_U^n and W_U^n in place of L_U and W_U respectively.

Chapter 5

Analysis and Comparisons

This chapter presents the formal analysis of the protocol. We also discuss the various security properties and compare the performance of our scheme with other related schemes.

5.1 Security Analysis

5.1.1 Adversary Capabilities

In this section, we define the semantic security for the session key and anonymity of user's identity. We define an adversary by $\mathscr{A}$, and a participant as P, where a participant can be any user or a sensor node.

Assuming there exists $\mathscr{A}$ running in a probabilistic polynomial time (PPT) in the security parameter k, which represents the bit-length of session keys. We note that the size of the dictionary $|D|$ is a fixed constant that is independent of the security parameter k. $\mathscr{A}$ has control of all communications between entities, can request for access to session keys and long-term keys, and can use side channel analysis to extract users' information stored on the smart card. These capabilities of $\mathscr{A}$ are modeled via the following oracle queries which $\mathscr{A}$ is allowed to make to model the capabilities .

- On a hash query $h(q)$ for which there exists a record $(q,\ r)$ appears in Λ_h, return r. Otherwise, choose an element $r \in \mathbb{Z}_p^*$, add the record $(q,\ r)$ to the list Λ_h and return r.
- on a query $Send(U, start)$, assuming U is in the correct state, we proceed as described in the protocol and the query is answered by $(MID,\ M, A,\ T_U)$.
- On a query $Send(S_i(MId,\ M,\ A,\ T_U))$, assuming S_i is in correct state, we proceed as described in the protocol and the query is answered by $(MID,\ M,\ T_U, A, B,\ T_S,\ ID_S,\ N)$.
- On a query $Send(G(MID,\ M,\ T_U, A, B,\ T_S,\ ID_U,\ N))$ assuming G is in the correct state, we proceed as described in the protocol and the query is answered by $(\gamma, h(\delta),\ T_G,\ MID')$.

- On a query $Send(S_i(\gamma, h(\delta), T_G, MID'))$ assuming S_i is in correct state , we proceed as described in the protocol and the query is answered by $(MID', B, h(\delta))$.

- On a query $Reveal(U_i)$, we proceed as follows: for an instance U has accepted, the query is answered with U returning the session key sk.

- On a query $Execute(U_i, S_i, G)$, we proceed using the simulation of the Send queries and the query is answered with the transcript $((MID, M, A, T_U), (MID, M, T_U, A, B, T_S, ID_U, N), (\gamma, h(\delta), T_G, MID'), (MID', B, h(\delta)))$.

- $CorruptLL(U)/CorruptSC(U)$: This query models the corruption capability of the adversary. The former returns the password of U while the latter returns the information stored in the smart card of U.

- $CorruptLL(S_i)$: It outputs specific sensor nodes' secret key y.

- $CorruptLL(GW)$: This models the privileged insider attacks.

- $TestAKE(P)$: This query is used for determining whether the protocol achieves authenticated key exchange or not. If P has accepted, then depending on a random bit b chosen by the oracle, $\mathscr{A}$ is given either the real session key sk if $b = 1$ or a random key drawn from the session-key space if $b = 0$.

- $TestAnon(U_i, ID_0, ID_1)$: This query is not used to simulate the adversary attack, but to define anonymity of user identity. After querying the oracle, the transcript of U's identity ID_0 or ID_1 will be returned according to a pre-defined random bit c. If $c = 1$, the adversary will learn the transcript of U with identity ID_1, otherwise with ID_0. This query can be called only once.

5.1.2 User Anonymity

The definition of user anonymity is based on the notion of cleanness.

Definition 4. *(Cleanness) A user U is said to be clean if none of the following occurs:*

1. *$\mathscr{A}$ queries both CorruptLL(U) and CorruptSC(U).*

2. *$\mathscr{A}$ queries CorruptLL(GW)*

User anonymity is formalized in the context of over the following set of experiments with a few limitations:
Experiment **ExpUA**$_0$:

1. $\mathscr{A}$ is not allowed to make the $TestAnon(U)$ query if the user U is not clean.

2. $\mathscr{A}$ is not allowed to corrupt GW and U if it has already made the $TestAnon(U)$ query.

3. $\mathscr{A}$ is not allowed to access to the $TestAKE(P)$ oracle.

4. $\mathscr{A}$ now outputs a bit b' as a guess on the hidden bit b chosen by the $TestUA$ oracle. $\mathscr{A}$ is said to succeed if $b = b'$.

Let $SuccUA_0$ be the event that $\mathscr{A}$ succeeds in the experiment $ExpUA_0$, and $Adv_P^{UA}(A)$ denote the advantage of $\mathscr{A}$ in attacking the user anonymity of protocol P. Then, we define

$$Adv_P^{UA}(\mathscr{A}) = 2 \cdot Pr_{P,\mathscr{A}}[SuccUA_0] - 1 \tag{5.1}$$

Definition 5. *(User Anonymity) An authentication and key exchange protocol P provides user anonymity if $Adv_P^{UA}(\mathscr{A})$ is negligible for any PPT adversary $\mathscr{A}$.*

Theorem 1. *Our authentication and key exchange protocol, provides user anonymity in the random oracle model under the ECDHP assumption in G.*

Proof. Let $\mathscr{A}$ be a PPT adversary against the user anonymity property of protocol. We prove the theorem by making a series of modifications to the original experiment $\mathbf{ExpUA}_0$, bounding the difference in the success probability of A between two consecutive experiments, and ending up with an experiment where A has a success probability of $\frac{1}{2}$ (meaning, $\mathscr{A}$ has no advantage). Let $SuccUA_i$ denote the event that $\mathscr{A}$ correctly guesses the hidden bit b chosen by the $TestAnon(U)$ oracle in experiment $\mathbf{ExpUA}_i$. Let t_i^{UA} be the maximum time required to perform the experiment $\mathbf{ExpUA}_i$ involving $\mathscr{A}$.

- $\mathbf{ExpUA}_1$. In this experiment, we simulate the random oracle h as follows: For each h query on a string s, the simulator first checks if an entry of the form (s, l) is in a list called $HList$ which contains all the input-output pairs of h. If such an entry exists in $HList$, the simulator returns l as the output of the h query. Otherwise, the simulator chooses a random n-bit string l', returns l' in response to the query, and adds the entry (s, l') to $HList$. For all other oracle queries of $\mathscr{A}$, the simulator answers them as in the original experiment $\mathbf{ExpUA}_0$. $\mathbf{ExpUA}_1$ is perfectly indistinguishable from $\mathbf{ExpUA}_0$ and therefore, the following claim is true.

$$Pr_{IA,\mathscr{A}}[SuccUA_1] = Pr_{P,\mathscr{A}}[SuccUA_0] \tag{5.2}$$

- $\mathbf{ExpUA}_2$. Here, we modify the experiment so that A (the elliptic curve point) is computed as follows:

  1. The simulator chooses a random exponent $x \in \mathbb{Z}_p$ and computes $X = xP$.

2. For each user instance, the simulator chooses a random $\alpha \in \mathbb{Z}_p$ and sets $A = \alpha X$. As a result of the modification, each K is set to $\alpha x \beta P$ for some random $\alpha \in \mathbb{Z}_p$. Since the view of A is identical between **ExpUA**$_2$ and **ExpUA**$_1$, it follows that:

$$Pr_{IA,\mathscr{A}}[SuccUA_2] = Pr_{P,\mathscr{A}}[SuccUA_1]. \tag{5.3}$$

- **ExpUA**$_3$. We next modify the computations of A and B as follows:

1. The simulator chooses two random elements $X, Y \in G$ and sets $B = Y$.
2. For each instance of clean users, the simulator chooses a random $\alpha \in \mathbb{Z}_p$ and sets $A = \alpha X$. For other instances, the simulator computes X as in **ExpUA**$_2$.
3. For each instance of clean users, the simulator sets each sk to a random n-bit string. For other instances, the simulator computes sk as in **ExpUA**$_2$. Since SK is set to a random n-bit string, for clean users, the success probability of $\mathscr{A}$ may be different between **ExpUA**$_3$ and **ExpUA**$_2$ if it makes an H($MID'||ID_S||ID_{GW}||A||B||(K \oplus M)$) query. This difference can be asserted by the claim that

$$|Pr_{P,\mathscr{A}}[SuccUA_3] - Pr_{P,\mathscr{A}}[SuccUA_2]| \leq 1/q_h \cdot Adv_G^{ECDHP}(t_U^3 A) \tag{5.4}$$

where q_h is the number of queries made to the h oracle. The objective of $\mathscr{A}_{ECDHP}$ is to compute and output the value $W = uvP \in G$ when given an ECDHP-problem instance $(U = uP, V = vP) \in G^2$. $\mathscr{A}_{ECDHP}$ runs $\mathscr{A}$ as a subroutine while simulating all the oracles on its own.

$\mathscr{A}_{ECDHP}$ handles all the oracle queries of $\mathscr{A}$ as specified in experiment **ExpUA**$_3$, but using U and V in place of A and B. When $\mathscr{A}$ outputs its guess b_0, $\mathscr{A}_{ECDHP}$ chooses an entry of the form $(MID'||ID_S||ID_{GW}||A||B||(K \oplus M))$ at random from $HList$ and terminates outputting $K/x \oplus M$. From the simulation, it is clear that $\mathscr{A}_{ECDHP}$ outputs the desired result $W = uvP$ with probability at least $1/q_h$ if $\mathscr{A}$ makes a H($MID'||ID_S||ID_{GW}||A||B||(K \oplus M)$) query for some instance of a clean user U. This completes the proof of this claim.

4. **ExpUA**$_4$. We finally modify the experiment so that, for each clean user U, a random identity ID'_U drawn from the identity space is used in place of the true identity ID_U in generating MID. We see that $\mathscr{A}$ gains no information on the hidden bit b chosen by the $TestAnon$ oracle because the identities of all clean users are chosen uniformly at random from the identity space. It, therefore, follows that $Pr_{IA,\mathscr{A}}[SuccUA_4] = \frac{1}{2}$. Thus the last results along with the results 5.2-5.4, we get the statement of the Theorem 1 and also arrive to the equation 5.1, following which we can choose to check the user anonymity.

□

5.1.3 Authenticated Key Exchange (AKE) Security

We need to make sure that our key exchange scheme is resistant to all kinds of known attacks to authentication protocols. A method of provable security is used. The security proof is based on a random oracle model [1].

Theorem 2. *In the the key agreement phase of the proposed protocol, the mobile user and the foreign agent correctly generates the same session key between them.*

Proof. In Step 3 of key agreement phase, S_j generates the session key as $SK = h(MID' \parallel ID_S \parallel ID_{GW} \parallel A \parallel B \parallel (K \oplus M))$ and $K = \beta A = \alpha\beta P$. In Step 4, the smartcard of the user generates the session key $SK^* = h(MID' \| ID_U \| ID_{GW} \| A \| B \| (K^* \oplus M))$ and $K^* = \alpha B = \alpha\beta P$. Therefore, we have $SK = SK^*$. Thus, in each session, U and S_j always establishes a fresh and common session key between them. □

Lemma 1. *(Birthday Paradox) For a positive integer N, and say q elements $y_1, y_2, \cdots, y_q$ are chosen uniformly and independently from set of size N. The probability that there exists distinct i, j with $y_i = y_j$ is at most $\frac{q^2}{2N}$.*

Proof. Let **Col** denote an event of collision, i.e there exists i, j with $y_i = y_j$, and let $Col_{i,j}$ denote the event $y_i = y_j$. Then it is clear that Pr[$Col_{i,j}$]= $1/N$ for any distinct i, j. Also $Col = \vee_{i \neq j} Col_{i,j}$ and so $Pr[Col] = Pr[\vee_{i \neq j} Col_{i,j}] \leq \Sigma_{i \neq j} Pr[Col_{i,j}] = \frac{q}{2} \cdot \frac{1}{N} \leq \frac{q^2}{2N}$ □

Theorem 3. *Let G represent group and D uniformly distributed dictionary of size $|D|$. Let K be our proposed protocol. Let $\mathscr{A}$ be an adversary against a semantic security within a time bound t, with less than q_{send} Send queries and q_{exe} Execution queries, and making less than $(q + h)$ random oracle queries. Then we have*

$$Adv_{K,D}(\mathscr{A}) \leq \frac{2q_{send}}{|D|} + 2q_h Adv_G^{cdh}(t + (q_{send} + q_{exe} + 1)\tau_G) + \frac{2q_{send}}{p} + \frac{q_h^2 + (q_{send} + q_{exe})^2}{p} \quad (5.5)$$

where τ_G denotes he exponential computation time in G

Proof. The proof defines a series of hybrid experiments, starting with a real attack Exp_0 and ending with a experiment Exp_4 in which $\mathscr{A}$ had no advantage. For each experiment Exp_i, we define an even $Succ_i$ depicting the case where the adversary correctly guess the bit b involved in the *Test*-query. At the end of the experiments we measure the probability $\Delta_i = Pr[Succ_{i+1}] - Pr[Succ_i]$. Using each difference of the probability, we get the result of Theorem 1.

- $\mathbf{Exp_0}$. The experiment corresponds to the real attack, in the random oracle model model [1]. By definition, we have $Adv_{K,D}(\mathscr{A}) = 2Pr[Succ_0] - 1$. So,

$$Adv_{K,D}(\mathscr{A}) = 2Pr[Succ_4] - 1 + 2(Pr[Succ_0 - Pr[Succ_4) \leq 2Pr[Succ_4] - 1 + 2\Sigma_{i=0}^{3}\Delta_i \tag{5.6}$$

- $\mathbf{Exp_1}$. In this experiment, we simulate the random oracles(h, but also some additional random oracles h' that will appear in Exp_4) as usual by maintaining hash lists Λ_h and $\Lambda_{h'}$. The *Execute*, *Reveal*, *Send*, *Corrupt* and *Test* oracles are also simulated as in the real attack. One can easily see that this experiment is perfectly indistinguishable from the real experiment. Hence

$$\Delta_0 = 0 \tag{5.7}$$

  - On a hash query $h(q)$ for which there exists a record $(q,\ r)$ appears in Λ_h, return r. Otherwise choose an element $r \in \mathbb{Z}_p^*$, add the record $(q,\ r)$ to the list Λ_h and return r.
  - on a query $Send(U{,}start)$, assuming U is in the correct state, we proceed as described in the protocol and the query is answered by $(MID,\ M, A,\ T_U)$.
  - On a query $Send(S_i(MId,\ M, A,\ T_U))$, assuming S_i is in correct state, we proceed as described in the protocol and the query is answered by $(MID,\ M,\ T_U, A,\ B{,}T_S,\ ID_U,\ N)$.
  - On a query $Send(G(MID,\ M,\ T_U, A,\ B,\ T_S,\ ID_U,\ N))$ assuming G is in the correct state, we proceed as described in the protocol and the query is answered by $(\gamma,\ h(\delta),\ T_G,\ MID')$.
  - On a query $Send(S_i(\gamma,\ h(\delta),\ T_G,\ MID'))$ assuming S_i is in correct state , we proceed as described in the protocol and the query is answered by $(MID',\ B,\ h(\delta))$
  - On a query $Reveal(U_i)$, we proceed as follows: for an instance U has accepted, the query is answered with the session key.
  - On a query $Execute(U_i,\ S_i,\ G)$, we proceed using the simulation of the Send queries and the query is answered with the transcript $((MID,\ M, A,\ T_U),\ (MID,\ M,\ T_U, A,\ B,\ T_S,\ ID_U, N),\ (\gamma, h(\delta),\ T_G,\ MID'),\ (MID',\ B,\ h(\delta)))$
  - On a query $Test(U_i)$, we proceed as follows, get SK from $Reveal(U_i)$ and flip a coin b. If $b = 1$, we return the value of the session key SK, otherwise, we return a random value with the same length.

- $\mathbf{Exp_2}$. In this experiment, we simulate all the oracles as in $\mathbf{Exp_1}$, except that we halt all executions in which a collision occurs in the transcript $((MID,\ M, A,\ T_U),\ (MID,\ M,\ T_U,\ A,\ B,\ T_S,\ ID_U,\ N),\ (\gamma,\ h(\delta),\ T_G,\ MID'),\ (MID',\ B,\ h(\delta)))$. According to the Lemma 1, the probability of collisions in the output of the h oracle is at most $q_h^2/2p$. Similarly, the probability of collisions in the transcript is at most $(q_{send} + q_{exe})^2/(2p)$, since A, b_U is chosen uniformly at random. Consequently,

$$\Delta_1 \leq \frac{q_h^2 + (q_{send} + q_{exe})^2}{2p} \tag{5.8}$$

- **Exp$_3$**. In this experiment, we abort the executions wherein $\mathscr{A}$ may have been lucky in guessing the authentication values M, N, ω, γ. The experiments Exp_3 and Exp_2 are indistinguishable unless the participants reject a valid authentication value:

$$\Delta_2 \leq \frac{q_{send}}{p}. \tag{5.9}$$

- **Exp$_4$**. In this experiment, we do not compute the session key using the oracle h, but using the private oracle h', so that the values are completely independent from A, B. we proceed with the $Execute$ queries $((MID, M, A, T_U), (MID, M, T_U, A, B, T_S, ID_U, N), (\gamma, h(\delta), T_G, MID'), (MID', B, h(\delta)))$. The experiments Exp_3 and Exp_4 are completely indistinguishable unless the following event **AskH** occurs: the adversary $\mathscr{A}$ queries the hash function h. In addition, whatever the bit b involved in the $Test$ query, the answer is random and independent for all session. So

$$\Delta_3 \leq Pr[AskH] \tag{5.10}$$

$$Pr[Succ_4] = \frac{1}{2} \tag{5.11}$$

To compute the experiment Exp_4, we simulate the executions using the random self-reducabilty of the Elliptic Curve Computational Diffie-Hellman Problem, given one ECDHP instance $(\alpha P, \beta P)$. We don't need to know the values of α and β since the values of K is not longer needed to compute the session key. Additionally, for every transcript there only one password which can be tested by the adversary,so the probability there is $q_{send}/|D|$. So we can conclude that :

$$Pr[AskH] \leq \frac{q_{send}}{|D|} + q_h Adv_G^{ecdhp}(t + (q_{send} + q_{exe} + 1) \cdot \tau_G)$$

So adding 5.6-5.11, we get the result 5.5.

□

5.2 Security Properties

1. **Mutual Authentication** Our scheme provides mutual authentication, where all the entities are mutually authenticating each other. Like when GW node receives the message $(MID, M, T_U, A, B, T_S, ID_U, N)$ it can make sure that the user message (MID, M, T_U, A) is

included in the sensor message to GW node. Similarly the message $(\gamma,\ h(\delta),\ T_G,\ MID')$ when sent by the GW node to S_j, the sensor S_j sensor can authenticate it and also further U can authenticate it being from a real GW node and sensor node.

2. **Replay Attack** Our scheme is resistant to replay attack, because the authenticity of the four messages is validated by checking the timestamps. Suppose adversary $\mathscr{A}$ intercepts a login message $(MID,\ M,\ T_U, A)$ and attempts to access the sensor node by replaying the same message. The verification fails since the time difference expires.
3. **Denial of Service Attacks** Our scheme is resistant to DoS attacks because the GW node has the parameter *free* which changes its value once it gets a request. So if the *free* parameter is 1 and the GW node has not received any request, the system will detect an intrusion.
4. **Impersonation Attack**
   - *User:* An attacker can't impersonate the user U. Suppose he forges the login message (MID, M, T_U, A). Now he tries to login into the system with a modified message (MID, M^*, T_U, A^*). However, the attacker cant forge M^* without knowing K or the master key s as he will be faced with ECDLP.
   - *Sensor:* As long as the attacker does not possess the secret key y , he cannot impersonate the sensor and generate N. Similarly, the adversary cannot generate ρ, as he does not know K to generate δ from that.
   - *GW node:* As long as the attacker does not know y, he cannot generate a valid message for the sensor node.
5. **Stolen Verifier attack** An attacker who steals the password verifier(like hashed passwords) from the gateway can use the stolen verifiers to impersonate a legal user to login into the system. But in our scheme, no such information is stored at the server , by which the adversary can make fabricated login request to the server.
6. **Guessing Attack**(Online/Offline) Our scheme can resist password guessing attack since in login user password is not the only thing that is required. It also required Biometric guessing. An adversary can't guess the B_U from $Q = h(B_U \oplus ID_U)$ because of one way hash. Similarly guessing attack on the master key s and y is tough as both are sent as a digest of secret information.
7. **Insider Attack** It may be possible when the GW manager or system administrator can use user password PW_U, it impersonate the user U through any network GW. Our scheme doesn't allow even privileged insiders, as in registration phase the user passes $V = h(PW_U||b_U)$ instead of plain PW_U. here b_U is high entropy number not known to GW. Also in registration phase the smartcard computes $Q = h(ID_U \oplus B_U)$, so the adversary guessing the correct B_U is not easy because of the one-way hash function. So, this scheme provides security against insider attacks.

8. **Redirection Attacks** When the attacker redirects one's smartcard's communication message to another one, the sensor node. IN has no information of the session key SK without mutual authentication and key agreement. Hence our scheme is safe from redirection attacks.

9. **Perfect Forward Secrecy** In our scheme DH key exchange algorithm based on ECC is used to generate key for session $SK = \alpha\beta P$, perfect forward secrecy is ensured because an attacker with a compromised all secret key (B_U, PW_U, s, y) is only able to obtain $A = \alpha P$ and $B = \beta P$. Over that computational in-feasibility to obtain session key $\alpha\beta P$ from αP and βP as it is ECDHP (Elliptic Curve Dille-Helfman Problem). Hence it provides perfect forward secrecy.

10. **Known-Session specific temporary information attack (KSSTIA)** Our Scheme is can resist this kind of attack. We assume that another adversary $\mathscr{A}$ knows a and b. However A still cant know the session key sk. We see that sk includes the factor K which is $\alpha\beta P \oplus M$. So even if the adversary can calculate $\alpha\beta P$ he can no way know M. So cannot compute random points α and β to know sk.

5.3 Security and Performance Comparisons

5.3.1 Storage Cost

In our protocol, the only storage that is involved is the smartcard in the registration phase. To estimate the storage cost, we assume the following

1. the size of point in the cyclic group of point P used in the ECC scheme is 320 bits.
2. the block cipher text for AES encryption/decryption is 128 bits.
3. the digest message size of the hash function (SHA-1) is 160 bits.
4. the identity size is 80 bits.
5. and the random number size is 160 bits.

In the registration phase, the GW node computes L_U, W_U and MID and stores them in the smartcard. The size of L_U and W_U are both 160 bits as both of them are digest messages of the hash function $h(\cdot)$. MID is a symmetric key encryption cipher text, hence its size is 128 bits. Now when the smartcard reaches the user, the user appends the random number b_u to the smartcard, whose size being another 160 bits. So the total cost can be calculated as to be : $160 + 160 + 128 + 80 + 160 = 688$ bits.

Table 5.1: Smartcard storage cost Comparisons of proposed scheme with other schemes

	Das [6]	Li et al. [18]	Das [5]	Shi[26]	Proposed
Smartcard Storage Cost	560 bits	400 bits	1344 bits	640 bits	688 bits

5.3.2 Security comparison

We can recall that protocols of Yuan et al.[32], Yoon et al.[31] do not provide for mutual authentication and can also be vulnerable to Insider attacks. The security comparisons between our protocol and some related schemes are summarized in the table 5.2.

Table 5.2: Security Comparisons of proposed scheme with other schemes

Security prop.	Yuan et al. [32]	Das [6]	Yoon et al.[31]	Das [5]	Shi *et al.*'s [26]	Proposed
Stolen verifier Attacks	Secure	Insecure	Secure	Secure	Secure	Secure
Guessing Attacks	Secure	Secure	Secure	Secure	Secure	Secure
Impersonation Attacks	Insecure	Insecure	Secure	Secure	Secure	Secure
Replay Attacks	Secure	Secure	Secure	Secure	Secure	Secure
DoS Attacks	Insecure	Insecure	Insecure	Secure	Insecure	Secure
Insider Attacks	Insecure	Insecure	Secure	Secure	Secure	Secure
Redirection Attacks	Insecure	Insecure	Insecure	Secure	Insecure	Secure
KSSTIA	Insecure	Insecure	Insecure	Insecure	Insecure	Secure
Mutual Authentication	Not Provided	No Provided	Not Provided	Not Provided	Provided	Provided
Perfect Forward Secrecy	Not Provided	Not Provided	Not Provided	Provided	Provided	Provided
Session Key agreement	Not Provided	Not Provided	Not Provided	Provided	Provided	Provided
Message Confidentiality	Not Provided	Provided	Provided	Provided	Provided	Provided
Security Factor	Three-factor	Two-factor	Two-factor	Three-factor	Two-factor	Three-factor
Password Change Phase	Provided	Not Provided	Not Provided	Not Provided	Provided	Provided

5.3.3 Performance Comparison

We evaluate the performance of the proposed scheme in the terms of the computations and communication costs and compare it with related schemes. To estimate the computation costs, we define the following notations: t_m is the time needed for elliptic curve scalar multiplication, t_a is the time for elliptic curve scalar addition, t_{enc} is the time required for Symmetric key encryption /decryption , t_h is the time for one-way hash function , t_{grn} for generating a random random using a random number generator and t_{fe} is the time for executing fuzzy extractor function(used in Das' Scheme [5]). The execution time computed by Farash *et al.*'s[10] for the cryptographic operations on different hardware platform is shown in table 5.3. The execution times have been calculated on an amd64 processor

Operations	t_m	t_a	t_{enc}	t_h	t_{grn}	t_{fe}
Execution Time	0.23 ms	1.14 ms	4.196 ms[23]	0.001988 ms	< 0.0001 ms	**PLse give a value, I didnt find**

Table 5.3: The execution time of cryptographic operations

To estimate the computational cost , we assume the following

1. the size of n used in the ECC scheme is 160 bits,
2. the block cipher text for AES encryption/decryption is 128 bits,
3. the digest message size of the hash function (SHA-1) is 160 bits,
4. and the identity size is 80 bits.

To compute the communication cost, we consider all exchanged messages during a session. In our scheme , the exchanges messages are $((MID, M, A, T_U), (MID, M, T_U, A, B, T_S, ID_S, N)$,$(\gamma, h(\delta), T_G, MID')$,$(MID', B, h(\delta)))$. These message for login phase has a communication cost of 80 bit of identity, 160 bits of message digest, a block cipher text containing the MID and an elliptic curve point P of 320 bits. So the overall cost for the login phase comes out to be 608 bits. For the key agreement and mutual authentication phase, we have a communication cost of $3(128) + 5(160) + 3(320) + 80 = 2224$ bits.

[**Yuan Paper for the table below is not available online for free...not able to download for calculating values, also on internet values not available. Pleasse sir, if you have the paper, can you send me.**]

	Yuan et al.[32]	AK Das [5]	Shi *et al.*'s [26]	Proposed Protocol
Registration phase		$5t_h + 2t_{grn} + t_{fe}$	$4t_h + t_m + t_{grn}$	$5t_h + 2t_{grn} + t_{enc}$
Login phase		$7t_h + t_{grn} + t_{fe}$	$4t_h + 2t_m + t_{grn}$	$5t_h + t_m + t_{grn}$
Key agreement phase		$8t_h + 2t_{enc}$	$12t_h + 4t_m + t_{grn}$	$12t_h + 3t_m + 2t_{grn} + 2t_{enc}$
Password Change Phase		Not Provided	$5t_h$	$8t_h$

Table 5.4: A comparison of execution times of various phases in various three factor Authentication systems

[**Sir please I need some some values for the execution times , otherwise I am not able to calculate the correct estimated calculation times for all the protocols. ALso I am searching desparately for three-factor protocols on the internet but I am not able to find anymore. Please sir if you can send me some more, then I will be able to comapre the performances.**
]

Communication Cost	Das [5]	Shi *et al.*'s [26]	Proposed Protocol
Registration Phase	400 bits	240 bits	400 bits
Login phase	400 bits	560 bits	608 bits
Key Agreement Phase	528 bits	1760 bits	1024 bits
Password Change Phase	–	–	–

Table 5.5: The communication costs of cryptographic operations

The performance of the three phases , that is the login phase the key agreement phase and mutual authentication phase, of the proposed protocol and a comparison with the related schemes are summarized in table 5.3.3. Although the communication costs of our protocol is little bit more than Das's Scheme, but as summarised in the table 5.2, that protocol fails to provide mutual authentication between each node, which is a huge disadvntage as it opens up the possibilty of many masquerading attacks. Our protocol used a 163-bit secure Elliptic Curve Cryptosystem unlike Das's scheme which provided 1024-bit security. Also our protocol provides better security because of three-factor authentication and higher efficiency than the rest in terms of execution time, providing almost speed up by 50%.

Chapter 6

Conclusion and Further Work

This thesis provides a detailed description of an Authentication protocol for Wireless Sensor Networks which provides User Anonymity and Secure Key Exchange and agreement. We also study the application of Elliptic Curve Cryptography on WSNs. We also provide a formal analysis of the security of the protocol. We show that our protocol provides protection against every kind of wireless sensor attacks known. Also efficiency performance against similar works have proved to be much better.

Further Work

It would be an interesting area of research to look into formal analysis of authentication protocols. Any formal analysis till date does not talk about security against various specific attacks. Any new work proposing such an security proof that takes in account protection against all the known attacks on WSNs might be a great future work. Also implementing Functional Encryption in WSNs for securing message transmissions from Users to Sensors to GW node is a new path to venture into. Any adversary even if he eavesdrops messages cannot break the ciphers from the function as he doesn't have the authority to do so.

Bibliography

[1] Bellare, M., and Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security* (1993), ACM, pp. 62–73.

[2] Bellare, M., and Rogaway, P. Optimal asymmetric encryption. In *Advances in Cryptology-EUROCRYPT'94* (1995), Springer, pp. 92–111.

[3] Benenson, Z., Gärtner, F. C., and Kesdogan, D. User authentication in sensor networks. In *GI Jahrestagung (2)* (2004), Citeseer, pp. 385–389.

[4] Cao, X., Kou, W., Dang, L., and Zhao, B. Imbas: Identity-based multi-user broadcast authentication in wireless sensor networks. *Computer communications 31*, 4 (2008), 659–667.

[5] Das, A. K. A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wireless Personal Communications* (2015), 1–28.

[6] Das, M. L. Two-factor user authentication in wireless sensor networks. *Wireless Communications, IEEE Transactions on 8*, 3 (2009), 1086–1090.

[7] Diffie, W., and Hellman, M. E. New directions in cryptography. *Information Theory, IEEE Transactions on 22*, 6 (1976), 644–654.

[8] Drissi, J., and Gu, Q. Localized broadcast authentication in large sensor networks. In *Networking and Services, 2006. ICNS'06. International conference on* (2006), IEEE, pp. 25–25.

[9] Fabbri, F., Buratti, C., and Verdone, R. A multi-sink multi-hop wireless sensor network over a square region: Connectivity and energy consumption issues. In *GLOBECOM Workshops, 2008 IEEE* (2008), IEEE, pp. 1–6.

[10] Farash, M. S., Attari, M. A., Atani, R. E., and Jami, M. A new efficient authenticated multiple-key exchange protocol from bilinear pairings. *Computers & Electrical Engineering 39*, 2 (2013), 530–541.

[11] Healy, M., Newe, T., and Lewis, E. Security for wireless sensor networks: A review. In *Sensors Applications Symposium, 2009. SAS 2009. IEEE* (2009), IEEE, pp. 80–85.

[12] Jiang, C., Li, B., and Xu, H. An efficient scheme for user authentication in wireless sensor networks. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on* (2007), vol. 1, IEEE, pp. 438–442.

[13] Koblitz, N. *Introduction to elliptic curves and modular forms*, vol. 97. Springer Science & Business Media, 1993.

[14] Lauter, K. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless communications 11*, 1 (2004), 62–67.

[15] Lee, C.-C., Lin, T.-H., and Tsai, C.-S. A new authenticated group key agreement in a mobile environment. *annals of telecommunications-annales des télécommunications 64*, 11-12 (2009), 735–744.

[16] Lee, T.-H. Simple dynamic user authentication protocols for wireless sensor networks. In *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on* (2008), IEEE, pp. 657–660.

[17] Lenstra, A., Tromer, E., Shamir, A., Kortsmit, W., Dodson, B., Hughes, J., and Leyland, P. Factoring estimates for a 1024-bit rsa modulus. In *Advances in Cryptology-ASIACRYPT 2003*. Springer, 2003, pp. 55–74.

[18] Li, C.-T., and Hwang, M.-S. An efficient biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications 33*, 1 (2010), 1–5.

[19] Liu, D., Ning, P., et al. Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In *NDSS* (2003).

[20] Liu, D., Ning, P., Zhu, S., and Jajodia, S. Practical broadcast authentication in sensor networks. In *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on* (2005), IEEE, pp. 118–129.

[21] LÃşpez, J., and Dahab, R. An overview of elliptic curve cryptography. Tech. rep., 2000.

[22] Patil, H. K., and Szygenda, S. A. *Security for wireless sensor networks using identity-based cryptography*. CRC Press, 2012.

[23] Paul, R., Saha, S., Sau, S., and Chakrabarti, A. Design and implementation of real time aes-128 on real time operating system for multiple fpga communication. *arXiv preprint arXiv:1205.2153* (2012).

[24] Perrig, A., Szewczyk, R., Tygar, J., Wen, V., and Culler, D. E. Spins: Security protocols for sensor networks. *Wireless networks 8*, 5 (2002), 521–534.

[25] Petersen, H., and Horster, P. Self-certified keys-concepts and applications. In *Proc. Communications and Multimedia Security* (1997), vol. 97, pp. 102–116.

[26] Shi, W., and Gong, P. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. *International Journal of Distributed Sensor Networks 2013* (2013).

[27] Tseng, H.-R., Jan, R.-H., and Yang, W. An improved dynamic user authentication scheme for wireless sensor networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE* (2007), IEEE, pp. 986–990.

[28] Washington, L. C. *Elliptic curves: number theory and cryptography.* CRC press, 2008.

[29] Wong, K. H., Zheng, Y., Cao, J., and Wang, S. A dynamic user authentication scheme for wireless sensor networks. In *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on* (2006), vol. 1, IEEE, pp. 8–pp.

[30] Yasmin, R. *An efficient authentication framework for wireless sensor networks.* PhD thesis, University of Birmingham, 2012.

[31] Yoon, E.-J., and Yoo, K.-Y. A new biometric-based user authentication scheme without using password for wireless sensor networks. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2011 20th IEEE International Workshops on* (2011), IEEE, pp. 279–284.

[32] Yuan, J., Jiang, C., and Jiang, Z. A biometric-based user authentication for wireless sensor networks. *Wuhan University Journal of Natural Sciences 15*, 3 (2010), 272–276.