PASSWORD BASED SECURE USER AUTHENTICATION PROTOCOLS IN WIRELESS SENSOR NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

UĞUR ŞEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2019

Approval of the thesis:

PASSWORD BASED SECURE USER AUTHENTICATION PROTOCOLS IN WIRELESS SENSOR NETWORKS

submitted by **UĞUR ŞEN** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. Ömür Uğur
Director, Graduate School of **Applied Mathematics** _____

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** _____

Assoc. Prof. Dr. Ali Doğanaksoy
Supervisor, **Mathematics Department, METU** _____

Assist. Prof. Dr. Pelin Angın
Co-supervisor, **Computer Engineering Department, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Fatih Sulak
Mathematics Department, Atılım University _____

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics Department, METU _____

Assoc. Prof. Dr. Murat Cenk
Cryptography, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: UĞUR ŞEN

Signature :

ABSTRACT

PASSWORD BASED SECURE USER AUTHENTICATION PROTOCOLS IN WIRELESS SENSOR NETWORKS

Şen, Uğur

M.S., Department of Cryptography

Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

Co-Supervisor : Assist. Prof. Dr. Pelin Angın

September 2019, 44 pages

A wireless sensor network (WSN) is a network, which consists of resource-constrained devices like sensors. Using these sensors, it is possible to monitor and track wide environments. WSNs have become widespread as a promising technology in the context of Internet of Things. One of the biggest disadvantages of these networks, which are used in many different systems and environments is the difficulty of providing their security. In WSNs, standard algorithms used for encryption, authentication and data integrity are too complex to be used in sensors. Designing algorithms and protocols to ensure the safety of WSNs by working efficiently on resource constrained devices such as sensors is one of the top agenda items of information security. In the literature, there are many studies and analyses on the subject. WSN security is analyzed under six main aspects and one of the most important of these is user authentication. Generally prescribed methods for user authentication are examined under two main headings: asymmetric encryption and password-based protocols. In this thesis, password based user authentication protocols in WSNs are examined. The security features of these protocols, the attacks on the protocols and the measures that can be taken against these attacks are mentioned. We also investigate the security proof of a designed protocol and the implementation of automatic security validation. Finally, an exemplary protocol from the literature is described and its weaknesses are examined.

.

Keywords: resource constrainedness, Internet of Things security, password based authentication

ÖZ

KABLOSUZ SENSÖR AĞLARINDAKİ PAROLA TABANLI GÜVENLİ KULLANICI DOĞRULAMA PROTOKOLLERİ

Şen, Uğur
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Pelin Angın

Eylül 2019, 44 sayfa

Kablosuz Sensör Ağları(KSA), sensörler gibi kaynak kısıtlı cihazların oluşturduğu ağlardır. Bu ağlarda bulunan sensörler ile geniş çevreleri izlemek ve takip etmek mümkündür. Giderek yaygınlaşan KSA, nesnelerin interneti bağlamında gelecek vaadeden bir teknolojidir. Birçok farklı sistemde ve ortamda kullanılabilen bu ağların en büyük dezavantajı güvenliktir. Şifreleme, kimlik doğrulama, veri bütünlüğü gibi alanlarda kullanılan standart algoritmalar KSAda yer alan sensörlerde kullanılamayacak kadar büyüktür. Sensörler gibi kısıtlı cihazlarda verimli şekilde çalışarak KSA güvenliğini sağlayacak algoritma ve protokoller tasarlanması bilgi güvenliğinin gündem maddelerinden biridir. Literatürde konu ile ilgili birçok çalışma ve analiz yapılmaktadır. KSA güvenliği altı ana parçaya ayrılır ve bunların en önemlilerinden birisi ise kullanıcı yetkilendirme özelliğidir. Kullanıcı yetkilendirme için genel olarak öngörülen yöntemler, asimetrik şifreleme ve parola tabanlı protokoller olmak üzere iki ana başlıkta incelenir. Bu tezde KSA üzerinde parola tabanlı kullanıcı yetkilendirme protokolleri üzerinde durulmuştur. Bu protokollerin güvenlik özellikleri, protokollere yapılan saldırılar, bu saldırılara karşı alınabilecek önlemlere değinilmiştir. Ayrıca, tasarlanmış bir protokolün güvenlik ispatını ve otomatik güvenlik değerlendirmesinin gerçeklenmesinden de bahsedilecektir. Son olarak, literatürden örnek bir protokol verilip bu protokoldeki zafiyetler incelenecektir.

Anahtar Kelimeler: kaynak kısıtlılık, Nesnelerin İnterneti, parola tabanlı kimlik denetleme

To My Family

ACKNOWLEDGMENTS

I would like to express my very great appreciation to my thesis supervisor Assoc. Prof. Dr. Ali Doğanaksoy for his patient guidance, enthusiastic encouragement during this thesis.

I wish to express my sincere appreciation to my co-supervisor Asst. Prof. Dr. Pelin Angın for many ideas, contributions and guidance.

I would like to thank Dr. Onur Koçak for his treasured advices and supports.

I also would like to have thanks to Dilara Özel for her patience and supports.

My sincere thanks go to FAME CRYPT and my colleagues for their valuable supports.

I wish to thank all my friends for their motivation. I also would like to thank all members of the IAM - METU.

In addition, I thank my cats Erdinç and Memoli who always greet me with joy.

Lastly, I would like to express my gratefulness to my parents and brother who always respect my opinions and supports me under all circumstances.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

IoT	Internet of Things
WSN	Wireless Sensor Networks
DWSN	Distributed Wireless Sensor Networks
HWSN	Hierarchical Wireless Sensor Networks
RSA	Rivest, Shamir, Adleman Cryptosystem
ECC	Elliptic Curve Cryptosystem
GWN	Gateway
OTP	One-time-password
GPS	Global Position System
C4ISRT	Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance and Targeting
NBC	Nuclear, Biologic and Chemical
GPRS	General Packet Radio Service
ID	Identitiy
UA	User Authentication
XOR	Exclusive Or Operations
MAC	Message Authentication Code
DH	Diffie-Hellman Key Exchange

CHAPTER 1

INTRODUCTION

The first signal was transmitted from hundreds of kilometers via the Internet 50 years ago. Initially our computers, then our mobile phones and now our things plug into the Internet in such a way that they never leave. The things that are connected to the Internet has given rise to the Internet of Things (IoT) notion. According to the predictions of authorities, in 2020, 20 billion devices will be connected to the Internet [2]. Wireless sensor networks (WSNs) form a part of the IoT network of disorderly tiny devices called sensors, which can sense pyhsical conditions such as temperature, pressure, light, humidity, sound etc. In addition to our daily lives, wireless sensor networks (WSNs) are used in agriculture, health-care, business and manufactoring, military, automation and supply chain [9], in short wherever there is information for monitoring as a part of the IoT environment. WSNs are fruitful to minimize the human error and maximize the efficiency by collecting lots of information. All these advantages of WSNs bring along some security shortcomings since the components of WSN cannot perform the protocols based on algorithms that keep us safe in known networks (for example RSA [32] and ECC [25]) properly. Thereby, efficient security solutions are needed for WSNs.

Security of IoT and WSN is divided into six categories in [17], which are: key agreement, user authentication, device authentication, access control, privacy preservation and identity management. User authentication and key agreement are challenging security features for WSN, since the components of WSN are resource-constrained and communication is wireless. Therefore, WSN may be exposed to various attacks. In other words, a secure user authentication and key agreemet protocol for WSNs requires specific features as follows:

- Resilience against known attacks
- Satisfaction of several security features such as mutual authentication, forward - backward security and anonymity etc.
- Efficiency and storage-friendliness for tiny devices

Finally, there are numerous proposed protocols that try to satisfy these requirements. These protocols can be divided into two categories based on public key cryptography and based on

password or pre-shared key.

In this thesis, efficient passsword-based user authentication protocols that are built upon storage friendly operations such that XOR and secure hash functions were examined in terms of complexity and security proofs.

The remainder of this thesis is organized as follows:

In Chapter 2, cryptographic hash functions and authentication methods that are used in password-based user authentication protocols are described.

In Chapter 3, an overview of wireless sensor networks is provided in term of components, network models and applications.

In Chapter 4, user authentication protocols in WSNs are analyzed from the aspects of taxonomy and anatomy, manageable attacks and security proofs.

In Chapter 5, an example protocol and its analysis involving the authentication scheme, vulnerabilities and complexity are shown.

CHAPTER 2

PRELIMINARIES

2.1 Cryptographic Hash Functions

A hash function is a function that converts inputs of different lengths to a fixed length. Let h be a hash function $h : A \longrightarrow B$ where A is a message s.t. $A = \{0,1\}^*$ and its hash is $B = \{0,1\}^l$ where l is a fixed length. B is also called the digest of A. A cryptographic hash function is a hash function that satisfies four more conditions as follows:

- For a given $m \in A$, $h(m)$ can be computed quickly. In addition, h needs to be deterministic, i.e. the same messages always result in the same hash value.
- For a given $h(m) = y$, it is computationally unattainable to find an m. This property means h is a one-way or preimage resistant function.
- When $h(m1) = b$, it is computationally infeasible to find m_2 such that $m_1 \neq m_2$ and $h(m_2) = b$.
- It is computationally infeasible to find two messages m_1 and m_2, which are different such that $h(m_1) = h(m_2)$. This means h is strongly collision-free.

The term *computationally infeasible* means that there is the lack of solutions at a reasonable time with the technology owned. For instance, finding the preimage of an n-bit hash function requires 2^n operations. This reduces to $2^{n/2}$ with the help of the birthday paradox.
Cryptographic hash functions are used in digital signatures and for message integrity to ensure that transmitted data is not corrupted during transmission. They are also used for fingerprinting of large data.

2.2 Known Authentication Methods and Protocols

Cryptograhpy is a strong tool, which is a branch of science heavily using mathematics and engineering for data security. The objectives are privacy, data integrity and authentication to secure communication among parties.

Privacy

If the messages flow as plaintext in an insecure network, a malicious party can sniff the network and gather critical information. Encryption is an operation that turns plaintext into a ciphertext to create an unreadable message. Therefore malicious parties cannot distinguish the difference between the ciphertext and any random messages. Privacy can be achieved by using encryption algorithms with a secret key through symmetric or public encryption algorithms.

Data Integrity

Cryptography provides tools to ensure that sent messages have not been changed, when malicious parties do not just read messages, but also try to manipulate messages between two parties. Cryptographic hash functions such as MACs are used for providing data integrity.

Authentication

There is no meaning of privacy and data integrity if a participant is not sure about who they are talking with. Cryptography also provides tools for assuring that a malicious party cannot pretend to be a legitimate user or party. Cryptographic primitives such as digital signatures and public-key infrastructure are used for authentication.
Commonly used authentication methods and protocols are mentioned below.

2.2.1 Passwords and OTPs

Passwords are the most common authentication method. Although passwords are stored encrypted in databases, they are still predictable. Also passwords might be eavesdroped if they are not transmitted as encrypted. In addition, using the same passwords in different systems cause a security risk. To avoid this, one-time-passwords (OTPs) are used. There are two types of OTPs, the first one is a large list that contains all the passwords and the second one is a recursive creator that creates an unpredictable new password from the previous password. OTPs overcome reusing problems, but they need a second secure channel for sharing. Both ways, it is expected that the entropy of passwords are high enough for unpredictability.

2.2.2 Public Key Cryptography and Digital Signatures

Public key or asymmetric cryptography is an encryption scheme that uses two related but unidentical keys called public and private key, relying on the complexity of the integer factorization problem. Encryption is performed using the public key and decryption by the private key. Eventually, the party holding the private key can prove himself/herself to other par-

ties. On the other hand, digital signatures use public key encryption to create signed data encrypted with the private key and verification can done using the public key. Although Public Key cryptography and digital signatures are secure and well studied, they are not efficient since operations in encryption or decryption are complex mathematical operations.

CHAPTER 3

WIRELESS SENSOR NETWORKS: FEATURES AND USAGE

A WSN is a network of devices that can communicate the information collected from a monitored or tracked field through wireless connections. The devices involved are sensors that have constrained resources. Therefore these devices are cheap, which is why a large number of sensors can be deployed in a WSN for wide area applications. We have low cost and also low power sensor nodes for wireless communication since the new developments in the area of micro-electro-mechanical systems. Wireless sensor networks are able to track or monitor conditions that include temperature, humidity, pressure, soil makeup, noise levels, and the characteristics such as speed, direction and size of an object [3]. Monitoring these conditions makes WSNs useful in military, healthcare-medicine, environmental monitoring, smart home automation and commercial applications.

3.1 Components of Wireless Sensor Networks

Wireless sensor networks consist of three essential components including sensor nodes, gateways (sinkholes) and users. Sensors are deployed on objects to collect the data for transmitting wirelessly to users when the users require these values from sensors by means of a gateway. WSNs are mainly used for monitoring systems and have been able to overcome the problems of other monitoring systems, because they need no human attendance in the field, provide real-time data from the field and maintain efficient and low-power operations[4]. In order to cover large areas for monitoring, the components of WSNs need to be cheap. Limitations of WSNs stem from this minimal hardware design. That is why sensor nodes are not equipped with a complex technology such that GPS or GPRS for communication, but instead only the gateway (or sink-base station) node can usually afford it.

3.1.1 Sensors

Sensors are tiny devices that sense phenomena such as temperature, acceleration, sound etc. and transfer the gathered data to the interested node. These phenomena may vary according to need.

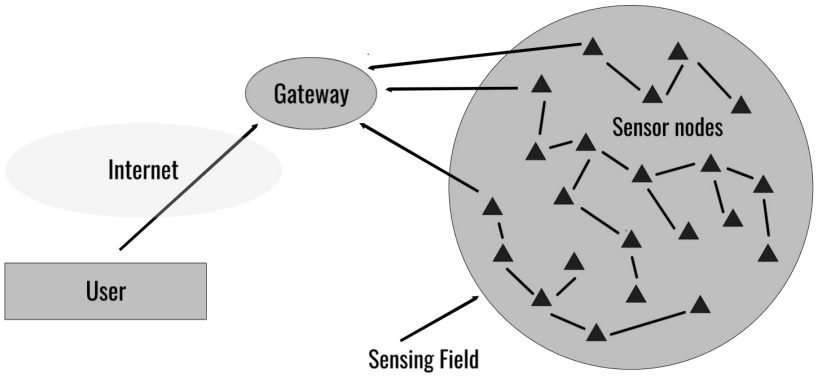

Figure 3.1: The model of wireless sensor networks.

Sensors are divided into two categories, passive sensors and active sensors.
Passive sensors sense the environment without interacting. These sensors do not need any extra power to make measurements. The power is needed when amplifying the analog signal and sending gathered data. Examples for passive sensors include microphones, light sensors and thermometers.

Active sensors work with environmental interaction, for example, to sense the data they need to emit sound or light waves. Therefore active sensors need extra power to make measurements in addition to amplifying the analog signal and sending gathered data.

Some features of sensors are as follows:

- Sensors have limited memory and low computing capabilities. Therefore their prices range between $0.1 - $1. Also the sensor nodes are low-bandwidth capable, which makes it difficult for them to transmit large amounts of data. Finally, they consume extremely low power.

- Since the communication is wireless in WSNs, the sensor nodes are battery operated. Therefore, the protocols and the applications implemented for the sensors need to be efficient to use minimum power for computing and transmission.

- Sensor nodes could be used in hostile or unattended environments such as battlefields. As a consequence, sensor nodes could be captured by enemies or adversaries. Since sensor nodes are not tamper-resistant, the adversary can reach the data in the sensor's memory[7].

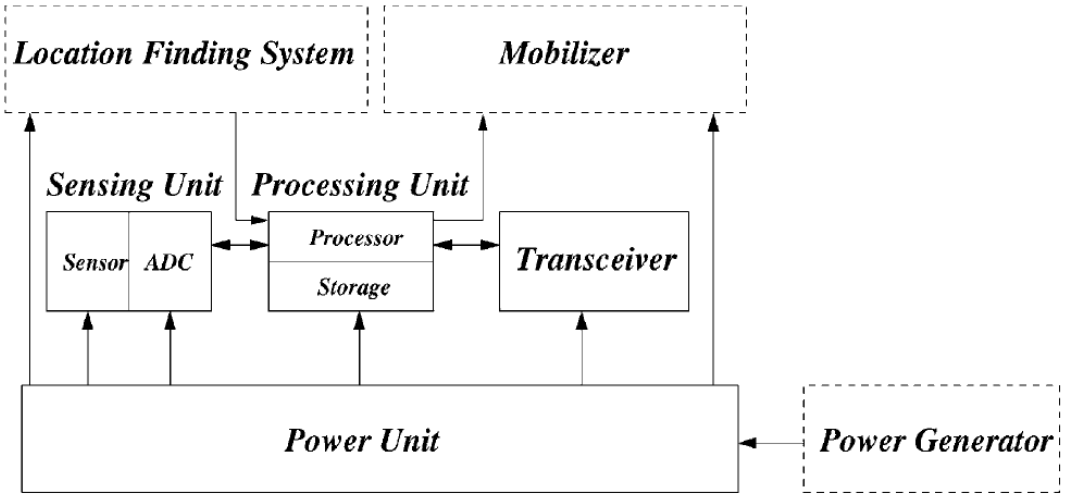

Figure 3.2: The architecture of sensor nodes [3]

- Sensing unit: This unit is the heart of sensor nodes. It is responsible to convert the analog sensor reading to digital and sends it to the processing unit for further operations.
- Transceiver unit: This unit is responsible for wireless communication using radio transmission waves.
- Processing unit: This unit helps in processing the received messages and deciding whether they are valid or not. According to the results, they trigger the sensor unit to capture sensor activity.
- Power unit: This unit provides the energy for the sensor unit to work properly.
- Storage unit: The storage unit stores its identity (ID) and pre-shared keys.

3.1.2 Gateways (Sink holes)

Gateways, or sinkholes, provide internetworking with external networks such as other wireless sensor networks, command or control systems and the Internet. It is very important that a WSN connects to the Internet in the IoT notion. While in WSNs, sensors could be in multitudes, the number of gateways is one or a few more. Gateways are small devices with resources a bit more than sensors. Moreover, in some WSN systems gateways acts as a trusted third party and the security of the network relies on the gateway.

3.1.3 Users

Users receive the collected data by using some instruments such as smartphones, websites or pagers etc. The main goal of WSNs are gathering information from physical environments

via sensors and transferring the collected data to interested users. Within the IoT notion, in order to get the data, no proprietary tool is necessary, it is sufficient to connect to the Internet.

3.2 Network Models of WSNs

There are two types of network modes for WSNs. These are distributed wireless sensor networks (DWSN) and hierarchical wireless sensor networks (HWSN).

3.2.1 Distributed Wireless Sensor Network

In this model, the sensors are deployed randomly in the environment. The only rule is that all the sensors need to be around the GWN. The information is delivered from any sensor node to another sensor node using multi-hop communications.

3.2.2 Hierarchical Wireless Sensor Network

In this model, there is a hierarchical rule among nodes in the network based on their capabilities. A HWSN involves several types of nodes such as sensor nodes, base stations and cluster heads. The smallest devices in HWSN are sensor nodes. A cluster is a set, which consists of many sensors. There is a cluster head inside every cluster and the sensors firstly send the sensed information to the specific cluster head. The cluster heads have more resources compared to sensor nodes and they are able to perform more complex operations. After that, all cluster heads in the network send information to the gateway (GWN).

3.3 Wireless Sensor Network Applications

Wireless sensor networks have gained increasing popularity with the IoT notion. There are many real-time applications, which can be used in critical areas such as military and healthcare applications, as well as commercially, i.e. in manufacturing, business and smart home applications. In this section we categorize the popular applications into military, healthcare, environment, smart home and other commercial areas.

3.3.1 Military Applications

A WSN could be used in military, command, computing, control, communications, intelligence, surveillance, reconnaissance and targeting (C4ISRT) systems, as well as monitoring friendly forces, ammunition and equipment. In case of NBC (Nuclear, Biological and Chem-

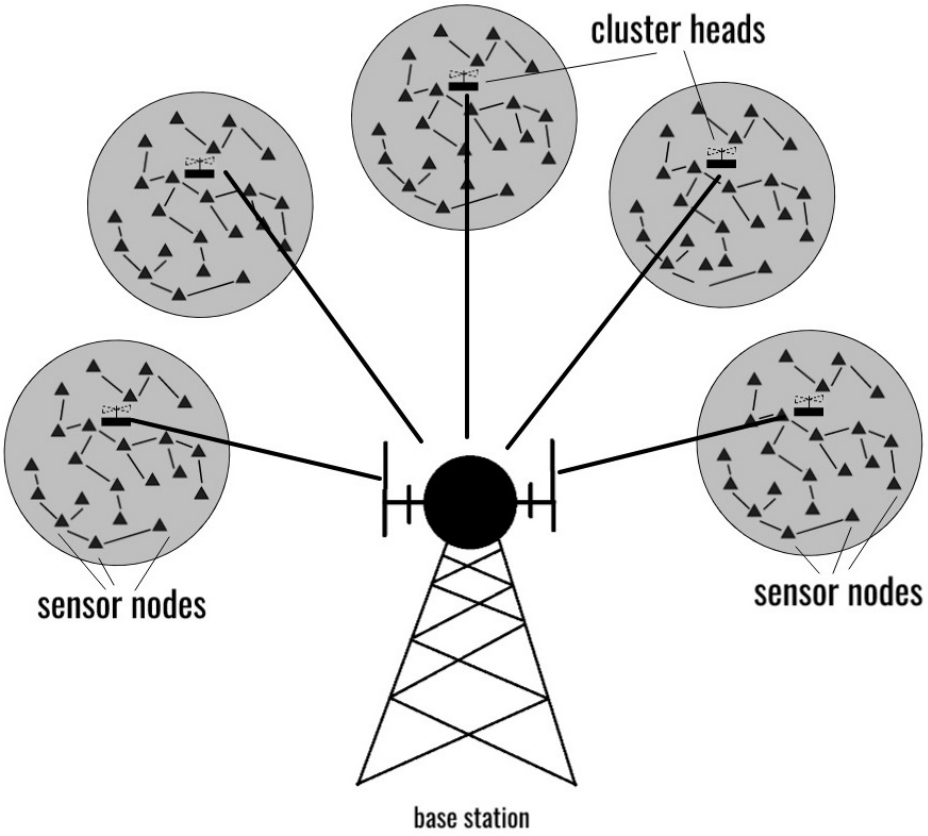

Figure 3.3: The model of hierarchical wireless sensor networks.

ical) attacks WSNs can detect the attacks to provide the friendly forces in a reliable reaction time.

3.3.2 Healthcare Applications

Monitoring patients is significant for treatment and taking necessary precautions. In this area, WSNs can be a part of healthcare. For instance, while one sensor node is measuring the heart rate another can be measuring blood pressure.

3.3.3 Environmental Applications

Tracking wild animals, forest fire detection, mapping bio-complexity and flood detection are possible applications of WSNs. Also, measuring air or water pollution in real time are capabilities of WSNs.

3.3.4 Smarthome Applications

Traditional electronic devices such as refrigerators, air conditions, ovens etc. have been getting smart with WSNs. Among benefits of these smart devices are that they can be controlled remotely and they can provide alerts for any changes planned in advance. For example, refrigerators can remind us to go shopping when the milk runs out.

3.3.5 Commercial Applications

When sensor nodes are deployed into factories, managing inventories and raw materials, monitoring quality of products, monitoring automatisation processes, detecting engine troubles, and tracking company vehicles can increase efficiency by means of WSNs.

CHAPTER 4

USER AUTHENTICATION PROTOCOLS FOR WSNS

4.1 Possible Attacks on WSNs

In this section we provide an overview of common attacks that could be launched against WSNs.

4.1.1 Replay Attack

An attacker can perform the replay attack by intercepting the traffic and replaying the login request of a party A to use the services of B. To avoid this attack, timestamps are used with login requests. If the attacker decides to apply the replay attack, it creates a small transition delay in the network. After B receives the login request message, B could understand this delay from the timestamps in the request and the current timestamps.

4.1.2 Forgery Attack

If an attacker intercepts or eavesdrops a login message, he/she can modify the login message to behave as a legitimate user in order to log in to the network. For a protocol resilient to the forgery attack, login messages of the protocol should not be derived from each other.

4.1.3 Stolen Smart Card and Smart Card Breach Attack

In this attack, somehow the attacker has gained access to a legitimate user's smart card. He/she can gain access to the information inside the smart card since the smart cards are not tamper resistant[26] [29]. For prevention, two and three factor authentication can be used. This is because even if the attacker has a legitimate card, he/she cannot provide the correct credentials for the next factor of authentication, e.g. pasword or biometric information.

4.1.4 Sensor Node Spoofing Attack with Sensor Node Capture

Assume that, S_j and S_t are sensor nodes and an attacker has a connection with S_j. Further, assume that a legitimate user U_i wants to connect to S_t. In this scenario, if the attacker succeeds in masquerading as the sensor node S_t, U_i will think that it is connecting to S_t, but actually connects to the attacker.

4.1.5 Impersonation Attacks

There is a possibility that the adversary gains some information, which is private or public, about a legitimate user. With this piece of information, the attacker can perform an attack called the *impersonation* attack, which impersonates any behaviour as a legitimate user to log in to the system. To avoid this type of attack, private information needs to be encrypted and not derivable from public data.

4.1.6 Denial of Service Attack

If an attacker has nothing to attack, he/she can choose to keep the system occupied by sending lots of requests to system. If the system tried to respond to all the requests regardless of authentication, the attacker can succeed. Consequently, the network needs to understand which request is legitimate or not by executing authentication methods first.

4.1.7 Many Logged-in Users with Same Login-ID Attack

When a legitimate user has logged in to the network with his/her password and ID at the same time as an attacker, the attacker may be able to steal these legitimate credentials. If the protocol allows a second entrance with the same credentials, then the attacker can operate this attack.

4.1.8 Stolen - Verifier Attack

Although it is not desirable, important credentials may be stored in the network in a table. If this kind of table is captured by an attacker, he/she can craft many attacks such as impersonation, and guessing attacks with the other data that the attacker has obtained. To avoid this, user-specific data should not be stored in the network. If it is necessary, data can be stored encrypted or hashed.

4.1.9 Insider Attack

After the malicious user has successfully signed into the system, he/she can launch an attack against another user such as the impersonation attack using his/her own messages. An insider attack is similar to the forgery attack. In this attack, the attacker has more information because he/she finishes the all steps. To avoid this attack, any credentials of a user should not be derivable from another user's credentials.

4.1.10 Privileged-Insider Attack

The privileged-insider attack is similar to the insider attack, however the attacker is privileged in here like a system administrator. With the information provided by the authority, the attacker can try to impersonate users.

4.1.11 Node Capture Attack

Suppose an entity is physically captured by an attacker, he/she may extract information from the memory of this entity. With this information, the attacker can perfrom several attacks on other entities such as impersonation, insider and password guessing attacks. To avoid this attack, extracted information needs to be useless for the rest of the units in the network.

4.1.12 GWN Bypassing Attack

If an attacker can successfully pass the steps of the authentication mechanism without being let by the GWN, it is said to be performing a GWN bypassing attack.

4.1.13 Password Guessing Attack

In secure systems, private information is always stored encrypted or hashed. Therefore, an attacker cannot extract the private information. However, he/she may try to guess and verify. If the entropy of a password is not high enough, the attacker can succeed in performing the password guessing attack by means of a dictionary. To avoid this attack, an attacker should not have the value derived from the password, otherwise he/she can try and validate the password from the computed value. This attack can be performed in two ways: offline and online password guessing attacks.

4.2 WSN Authentication Approaches

This section presents state-of-the-art user authentication protocols for WSNs.

4.2.1 Password-based User Authentication Protocols

Password-based user authentication for WSNs became prominent after considering asymmetric encryption was not realistic for implementation in sensor nodes. In 2004 Watro et al.[39] proposed a user authentication protocol based on the RSA cryptosystem[32] and Diffie-Hellman key exchange[19] named TinyPK. Das[18] and Tseng et al.[35] showed that Watro et al.'s protocol had some vulnerabilities such as the *masquerade attack* and also that it was inconvenient to implement for real applications, since the RSA cryptosystem and Diffie-Hellman key exchange are not suitable for tiny devices. Later, in 2006 Wong et al.[40] proposed a dynamic user authentication protocol, which only uses hash functions and XOR operations. This scheme is promising since its complexity is suitable for resource-constrained environments like WSNs. In response to this protocol, Das[18] showed that the protocol suffers from *many logged-in users with the same login-id attack*, as well as the *stolen verifier attack* and they improved Wong et al.'s scheme by adding a gateway (GWN) as a trusted third party for authentication and proposed an efficient password-based user authentication that used timestamps for verification without the key agreement.

In 2010 Khan and Alghatbar [24] proposed an improved scheme of Das's study by solving its insecured password and mutual authentication problems by introducing pre-shared keys and hash passwords. A similar attempt was presented by Chen and Shih [13], where they provided a robust user authentication protocol for the WSNs, but their protocol was later shown to be vulnerable to *replay, forgery* and *bypassing attacks*. Moreover, Khan and Alghathbar's scheme was found to be vulnerable in Vaidya's study [38]. They showed that Khan and Alghathbar's scheme had vulnerabilities to the *stolen smart card attack* and *forgery attack with node capture attack*. Another improvement on Das's scheme came from Nyang et al. [31] by improving security with the features of efficiency and usability to overcome the *privileged-insider* and *offline-password guessing attacks*. In 2010, He et al. [22] showed that Das's scheme had vulnerabilities such as *insider* and *key impersonation attacks*.

Das et al. [16] and Xue et al.[41] proposed two user authentication and key agreement protocols for WSNs using smart cards. In 2013 Turkanovic et al. [37] showed that Das's scheme had several security shortages and was not feasible for real implementation. In 2014 Turkanovic et al. [37] proposed a novel user authentication and key agreement protocol for WSNs. In 2016 Farash et al. [20] proved that this protocol had security shortages and it was not resilient against some attacks like *stolen smart card*, *man-in-the-middle attack* and proposed an improved version of Turkanovic et al.'s scheme. Although the authors claimed higher security levels and resilience against cryptographic attacks, in 2017 Hamidreza Yazdanpanah et al. [42] showed that Farash et al.'s scheme was not resillient against the *stolen smart card attack* and *stolen sensor node attack*. Also in 2016 Amin et al. [5] showed that Turkanovic et al.'s scheme is insecure and inefficient due to many drawbacks such as the *offline password guessing attack*, *stolen smart card, user impersonation attack* and *sensor node impersonation attack*. In the same study, Amin et al. proposed a secure lightweight scheme for user authentication and key agreement.

4.2.2 ECC-based User Authentication Protocols

ECC can be used for user authentication protocols in WSNs. Although, ECC-based user authentication protocols are not as efficient as password-based solutions, they provide many security features.

In 2011, Yeh et al. [43] proposed the first ECC-based user authentication scheme for WSNs. After that, in 2013 Shi et al. [33] presented an improved authentication scheme. In 2014, Choi et al. [15] presented an enhanced authentication scheme, which is resillient against the *stolen smart card* and *sensor energy exhaustion attack* over Shi et al.'s scheme. However, user anonymity and untraceability were not satisfied by any of these three protocols. In 2014, Nam et al. [30] presented an authentication scheme based on ECC, which provides user anonymity and perfect forward secrecy.

4.3 Taxonomy and Anatomy of WSN User Authentication Protocols

4.3.1 Taxonomy of WSN Security

A security protocol definition is a sequence of operations to achieve certain security goals among communicating parties. Based on Das et al.'s taxonomy, [17] the aims of security protocols for IoT environments are divided into six categories including key agreement, user authentication, device authentication, access/user access control, privacy preservation and identity management. In this section, WSN security protocols that focus on two of these six categories, namely user authentication and secure key agreement, are examined with respect to the used method and complexity.

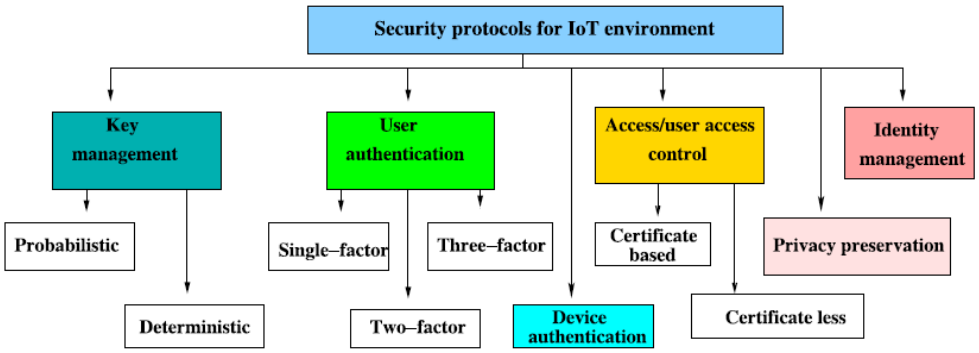

Figure 4.1: The taxonomy of IoT security[17] .

The six categories in the taxonomy can be summarized as follows:

- Key pre-distribution
  Since the sensor nodes are resource-constrained, secure and traditional public key cryptosystems, i.e. RSA [32], Diffie-Hellman key exchange protocol [19] are complicated

and energy-consuming for key pre-distribution in WSNs. In addition to this, trusted third party authentication protocols like Kerberos are not so feasible because of the unpredictable network architecture, short transmission range and intermittent operations of WSNs. Although ECC is feasible for designing security protocols in WSNs with the recent developments, symmetric solutions are still demanding for efficiency.

- User authentication
  In a WSN, sensor nodes should collect data that could be sensitive and send them only to legitimate users registered in the network beforehand. Otherwise, collected data can be captured by adversaries.

- Device Authentication
  This is a security feature that is necessary for the secure communication of two or more sensor devices, which are not controlled by or are interacting with humans.

- Access Control
  WSNs are dynamic networks since the sensor nodes are dispensable. A sensor in the network can be captured or go offline due to its battery draining out or hardware failures. To compensate for these lost sensors, new sensors need to be deployed in the network securely. Otherwise, malicious nodes can enter the network.

- User Access Control
  User access control mechanisms promise access rights to legitimate users for accessing the right information in the WSN. This provides WSN security by preventing illegal users from accessing WSN resources.

- Privacy Preservation
  WSNs can be used in environments where critical information such as military, healthcare, governmental applications should be kept confidential. In order to provide this feature, encryption schemes can be used after the key exchange mechanism is executed securely.

- Identity Management
  Identity is used for representation of a person, device or system from the real world in the digital world. Identity management includes how to assign and manage these IDs in IoT networks, where various IDSs may need to be used. If the attacker is able to change these IDs in a way, it may prevent the system from working correctly.

4.3.2 Anatomy of Password Based User Authentication Protocols for WSN

All the parties in a WSN have credentials and secrets for providing security features such as authentication and secure key agreement. Authentication of a party in the network is defined by proving these credentials and secret set uniquely. This means that there is no other legitimate party that can provide these credentials. For instance, in a user authentication protocol,

before the user sends queries to any sensor, the user must register with the system using its credentials securely. There credentials are as shown below.

- User identity; this must be secret, otherwise user anonymity is not provided.
- User password; this must be secret too, otherwise an illegal person that gathered user passwords has a chance to impersonate a legitimate user.
- Sensor identity; identification number of a sensor.
- Gateway secret password.
- Shared passwords among gateway-sensors and gateway-users.
- Secret nonces; these are picked by parties to mask the values that cannot be transferred in plaintext.
- Secret session key; this is obtained by the user and the sensor node securely.

Password-based user authentication schemes consist of four phases for authentication: pre-deployment phase(setup), registration phase, login phase and authentication phase and two secondary phases for node addition and password change.

1. **Pre-deployment (setup) phase**

   The first phase of a password-based user authentication protocol is the pre-deployment (setup) phase. In this phase, each sensor node is deployed into the network with its identity. Then the gateway node generates the shared keys between itself and each sensor. If necessary, the gateway generates its own secret password. Thereafter, each sensor node stores its own shared keys and the gateway node stores all the shared keys between the gateway and sensors. As we mentioned before, gateways arericher in resources, therefore they can store all the shared keys of all the sensor nodes.

2. **Registration Phase**

   Credentials of users and sensors may be captured or stolen in transmission. For this reason, in this phase, they are masked before transmitting. Also, for authentication, these raw credentials are customized by involving shared keys and the gateway's password hashing and XORing operations to use in later phases.
   The registration phase is performed in two steps, user and sensor registration. After this phase, the gateway stores the processed credentials and in some cases writes to a smart card or secure cloud.

3. **Login Phase**
   After the registration phase, all preparations for user authentication have been completed. For a protocol to provide user authentication, only legal users should be allowed

to log in to the system. Therefore, a user that wants to log in to the network first needs to enter his/her password.
At this point, some user authentication protocols provide two or three-factor authentication for higher security. This means that the user has a smart card or a biometric identity to use in the login phase. In this situation, the user needs to enter his/her password, smart card and biometrics. Next, the terminal or gateway tries to verify this inserted data. If verification is successful, the user chooses a sensor that the user wants to connect to and authentication as the last phase is started by the user.

4. **Authentication Phase**

   In this phase, authentication messages visit all parties of the WSN and the parties authenticate each other. The parties send an XOR of secret values with data known by receivers, therefore receivers can extract a secret value by the XOR operations. Hence, parties can authenticate each other by verifying sent authentication messages. Also eavesdropers cannot gain any information from the flowing messages because all messages are masked by XOR operations.

4.4 Security Features of WSN

Every security protocol is designed for a specific purpose. In this section these goals will be examined. The comparison of the protocols in the literature are provided in table 4.1.

4.4.1 Mutual Authentication

As we mentioned before, each user has his/her ID and password, each sensor has its ID and both have their private keys and shared keys. Parties use these credentials to log in to networks. Authentication ensures that any illegal party cannot pretend as a legitimate user and cannot log in to the system by using those credentials. For example, if the gateway authenticates a sensor S, it means that the gateway is sure that the sensor S cannot be an illegal node pretending to be sensor S.

If the two parties authenticate each other, it is called mutual authentication. In the previous example, mutual authentication is performed, if the sensor S authenticates the gateway too.

Mutual authentication is one of the highly important security features in WSNs because of the architecture of WSN, since all communication is wireless and units of WSN such as sensor nodes are considered as resource-constrained. These reasons make WSNs easy to manipulate in terms of the traffic in the wireless network.

4.4.2 Key Agreement

Wireless network systems are also dynamic networks that can contain hundreds of resource-constrained sensor nodes communicating with each other. Therefore, temporary keys are generated from the initial shared keys in every user login. It is obvious that key agreement needs to be secure. Since nodes are resource-constrained, secure key agreement in WSN is a little more difficult than known secure key agreement. Finally, privacy is achieved, after success of secure key agreement, all traffic flow can be encrypted.

4.4.3 Password and ID Protection

When a user wants to connect to a specific sensor in a WSN, most of the time, this operation is done in a public (insecure) network. If the passwords and IDs are not encrypted or masked they could be intercepted or stolen by malicious parties. Malicious parties can use this information to impersonate a legitimate user.

4.4.4 Forward and Backward Secrecy

Forward secrecy ensures the confidentiality of future communication between the entities, for instance if a node or entity leaves the network they are not be able to read the communication anymore.

Backward secrecy ensures the confidentiality of previous communication between the entities, for instance if a node or entity joins the network they are not be able to read the communication before joining the network.

4.4.5 User Anonymity and Untraceability

If a protocol satisfies the user authentication security features, outsiders are not capable of discovering the owners of the messages since the users' identities are not public. Because of this, user or sensor identity needs to be transmitted encrypted or masked.

Untraceability is similar to anonymity except that outsiders cannot discover the owners of the messages, but they can distinguish between the owners of the messages. If an attacker detects that a certain user logged in to two systems at different times, we rule that the protocol does not satisfy the untraceability feature.

4.4.6 Sensor Node Anonymity

If the identity of a sensor node or entity is sent unmasked over an insecure channel, an adversary can have the opportunity to distinguish between sensors. Hence the adversary can collect data about this specific sensor node and perform an attack with this information. The protocol should not allow the adversary to distinguish between the sensors, which is provided by encrypted or masked identity of sensors.

4.4.7 Secure Password Update

User authentication schemes need to allow a registered user to change its password securely any time the user wants. Otherwise, an illegal party can start a legitimate user's password change phase and impersonate this legitimate user by recovering the new password. This is also called a *password changing attack*.

Table 4.1: Security comparison table

Security Features	[41]	[20]	[16]	[34]	[15]	[36]	[43]	[6]	[33]
Mutual Authentication	x	✓	x	✓	✓	✓	✓	✓	✓
Secure Key Agreement	✓	✓	✓	✓	✓	x	✓	✓	✓
User Anonymity	x	x	✓	✓	x	x	x	✓	x
Untraceability	x	x	✓	✓	x	x	x	x	x
Stolen Smart Card Attack	x	✓	✓	✓	x	x	x	✓	x
Impersonation Attack	x	✓	x	✓	✓	x	✓	✓	?
Insider Attack	x	x	x	✓	?	✓	✓	✓	✓
Stolen Verifier Attack	✓	x	✓	✓	x	x	✓	✓	?
Man-in-the-Middle Attack	?	?	?	✓	?	x	✓	?	?
Replay Attack	✓	✓	x	✓	✓	✓	✓	✓	✓
Offline Guessing Attack	x	x	x	✓	✓	x	?	?	?

✓ : resilient against the attack x : not resilient against the attack ? : not observed

4.5 Complexity of Protocols

Not only security, but efficiency is a demanding feature for WSNs, as WSNs can be used in critical areas such as military or health monitoring, where data flow must continue uninterrupted. With the resource-constrained units of the WSNs, lightweight solutions are more suitable for these dynamic and crowded networks.

There are numerous user authentication schemes for WSN in the literature. The table in 4.2 provides a comparison of the protocols in terms of time complexity.

Table 4.2: Complexity comparison table

Protocols	User	Sensor Node	Gateway	Total
[15]	$7T_h + 3T_{ECC}$	$4T_h + 2T_{ECC}$	$4T_h + T_{ECC}$	$15T_h + 6T_{ECC}$
[43]	$T_h + 2T_{ECC}$	$4T_h + 4T_{ECC}$	$3T_h + 2T_{ECC}$	$8T_h + 8T_{ECC}$
[44]	$5T_h$	$2T_h$	$6T_h$	$13T_h$
[36]	$7T_h$	$5T_h$	$7T_h$	$19T_h$
[20]	$11T_h$	$7T_h$	$14T_h$	$32T_h$
[27]	$17T_h$	$9T_h$	$18T_h$	$44T_h$
[28]	$9T_h$	$6T_h$	$11T_h$	$26T_h$
[41]	$7T_h$	$5T_h$	$10T_h$	$22T_h$
[34]	$18T_h$	$11T_h$	$24T_h$	$53T_h$
[21]	$7T_h$	$3T_h$	$9T_h$	$19T_h$
[23]	$8T_h$	$5T_h$	$12T_h$	$25T_h$
[6]	$12T_h$	$5T_h$	$15T_h$	$32T_h$
[38]	$6T_h$	$2T_h$	$5T_h$	$13T_h$

T_{ECC} is the cost of an ECC encryption/decryption operation

T_h is the cost of a hashing operation

T_{SE} is the cost of a symmetric encryption/decryption operation

4.5.1 Encryption Based UA Protocols

Protocols based on RSA or DH are not realistic because of high memory overhead and computational time costs. Even if these problems were came over, mathematical operations consume a lot of power, which is not good for battery-powered devices. In the beginning, ECC-based protocols looked infeasible like others, but with the developments such as efficient point doubling, ECC-based protocols have become more feasible for use in this area.

4.5.2 Password Based UA Protocols

Firstly, Wong et al.[40] noticed that encryption-based authentication is not appropriate for user authentication and key agreement for WSNs. They proposed a novel authentication scheme that just uses XOR and hash operations. Since this scheme involves XOR and hash operations, it is very efficient and low-cost compared to encryption-based schemes.

4.6 Security Test and Proofs

Authentication protocols promise that parties can mutually be assured about each other's identity. In the literature, there are lots of authentication protocols that are characteristically described by listing the messages sent among the parties. It is important that a protocol should not contain any security shortcomings. Therefore, formal analysis is necessary to detect any

vulnerabilities or shortcomings.

4.6.1 BAN Logic

In 1989, the first attempt of formal analysis came from Burrows et al. with their contribution "BAN Logic" [12]. It analyzes protocols formally based on the aspect of authentication and not other issues like secrecy etc.

Basic Notations

- A, B, S denote parties.
- K_{ab}, K_{as} , K_{bs} denote shared passwords
- K_a, K_b , K_s denote public keys
- N_a, N_b , N_s denote statements
- P, Q, R range over parties
- X, Y range over statement
- K ranges over encryption keys

Conjunction

- $P \mid\equiv X$ P believes X, so P acts as if X is true.
- $P \triangleleft X$ X is seen by P
- $P \mid\sim X$ P once said X
- $P \Rightarrow X$ P has control over X
- $\#(X)$ The Statement X is fresh, so it has never been used
- $\{X\}_K$ X has been encrypted with the key K
- $\langle X \rangle_Y$ The statement X is combined with Y
- $P \overset{k}{=\!\!=} Q$ X is a secret except P and Q
- $P \overset{k}{\longleftrightarrow} Q$ K is a shared key between P and Q
- $\overset{k}{\mapsto} P$ P knows the public key K

Logical Rules

1. **Message-Meaning Rule**

   The message meaning rules for shared keys

$$\frac{P \mid\equiv Q \overset{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$$

   If P believes that K is shared with Q and also sees the statement X is encrypted with the key K, P believes that Q once said the statement X. This rule helps reduce logical results when forwarded messages are encrypted with the shared key K.

   The message meaning rules for shared secrets

$$\frac{P \mid\equiv Q \overset{K}{=\!=} P, P \triangleleft \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}$$

   If P believes that K is shared with Q and also sees $\langle X \rangle_Y$, P believes that Q once said X. This rule helps reduce logical results when shared secrets are included in the messages.

2. **The Nonce-Verification Rule**

$$\frac{P \mid\equiv \#(()X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

   If P believes that the statement X could have been said only lately and that Q once said X, P believes that Q believes X.

3. **The Jurisdiction Rules**

$$\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

   If P believes that Q has jurisdiction over X, P trusts Q on the truth of X.

4. **Freshness Rule**

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}$$

   If X is fresh therefore any message involving X is also fresh.

5. **Other rules**

   The essential rule of the belief operator $\mid\equiv$ is that P believes a set of statements if and only if P believes each individual statement separately.

$$\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X, Y)}$$

$$\frac{P \mid\equiv (X, Y)}{P \mid\equiv (X)}$$

$$\frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X}$$

4.6.1.1 An Application of BAN Logic

In this part, the example usage of BAN logic is presented by a toy protocol, which aims to distribute a new session key betweeen A and B who want to share a long-term key.

Table 4.3: Toy protocol for BAN logic

A		B
$A, \{M_a\}_{K_{ab}}$	$\longrightarrow$	
	$\longleftarrow$	$\{M_a + 1, M_b\}_{K_{ab}}$
$\{M_b + 1\}_{K_{ab}}$	$\longrightarrow$	
	$\longleftarrow$	$\{K_{ab}^{'}, M_b^{'}\}_{K_{ab}}$

Initial Assumptions
A believes $A \overset{K_{ab}}{\longleftrightarrow} B$ and B believes $A \overset{K_{ab}}{\longleftrightarrow} B$ since K_{ab} is pre-shared key between A and B.
A believes (B controls $A \overset{K_{ab}^{'}}{\longleftrightarrow} B$) since $K_{ab}^{'}$ is generated by B.
A believes $fresh(M_a)$ and B believes $fresh(M_b)$ since M_a is generated by A and M_b is generated by B.
B believes $fresh(M_b^{'})$.

Analysis the Protocol

In the last step of this protocol, A must believe that $K_{ab}^{'}$ shares K_{ab} with B and B must believe that $K_{ab}^{'}$ shares K_{ab} with A.
$A \mid\equiv A \overset{K_{ab}^{'}}{\longleftrightarrow} B$ and $B \mid\equiv A \overset{K_{ab}^{'}}{\longleftrightarrow} B$

From initial assumptions;
$A \mid\equiv (B \Rightarrow A \overset{K_{ab}^{'}}{\longleftrightarrow} B)$...[1].

From $A \mid\equiv fresh(A \overset{K_{ab}^{'}}{\longleftrightarrow} B)$, $A \mid\equiv B$ said $A \overset{K_{ab}^{'}}{\longleftrightarrow} B$
$A \mid\equiv B \mid\equiv A \overset{K_{ab}^{'}}{\longleftrightarrow} B$...[2]

From jurisdiction rules of [1] and [2];

$A \mid\equiv A \overset{K'_{ab}}{\longleftrightarrow} B ...[3]$

From initial assupmtions;
$A \mid\equiv A \overset{K_{ab}}{\longleftrightarrow} B ...[4]$

From message 4 and decomposition rule; $A \triangleleft \{A \overset{K'_{ab}}{\longleftrightarrow} B, M'_b\}_{K_{ab}}$
$A \triangleleft \{A \overset{K'_{ab}}{\longleftrightarrow} B\}_{K_{ab}} ...[5]$

From $[4]$ and $[5]$ message meaning;
$A \mid\equiv B$ said $A \overset{K'_{ab}}{\longleftrightarrow} B$

Finally, A believes fresh $fresh(A \overset{K'_{ab}}{\longleftrightarrow} B)$ cannot be proven. In other words, if K'_{ab} is changed, A will not know this replacement. Therefore, this protocol has security flaws, i.e. an attacker T can perform a replay attack as follows:

$A \longrightarrow B : A, \{M_a\}_{K_{ab}}$
$B \longrightarrow A : \{M_a + 1, M_b\}_{K_{ab}}$
$A \longrightarrow B : \{M_b + 1\}_{K_{ab}}$
$B \longrightarrow T : \{K'_{ab}, M'_b\}_{K_{ab}}$
$T \longrightarrow A : \{K''_{ab}, M''_a\}_{K_{ab}}$

Table 4.4: Modified protocol is resilient to replay attack

A		B
A, M_a	$\longrightarrow$	
	$\longleftarrow$	$\{M_a, K'_{ab}\}_{K_{ab}}$
$\{M_a\}_{K'_{ab}}$	$\longrightarrow$	
	$\longleftarrow$	M_b

4.6.2 AVISPA Tool

AVISPA (Automated Validation of Internet Security Protocol and Applications) is an automatic tool for security verification of applications and protocols with a high-level language specification [1]. It is important to point that the AVISPA tool can only capture replay and man-in-the-middle attacks.

AVISPA can validate security protocols in four different techniques as seen in Figure 4.2. First, a security problem is described to HLPSL(High Level Protocols Specification Language) by the analyzer or developer of the protocol. In addition, protocol analyzer takes advantage of modular data stuctures, several models for intruders, complicated security features and several cryptographic structures and algebraic features of this language. After then, the desribed model is sent to the HLPSL2IF module to translate previous definitions into the IF (Intermediate Format). AVISPA tool's logic only can process IF descriptions. Finally, four analysis techniques are used in this tool as follows:

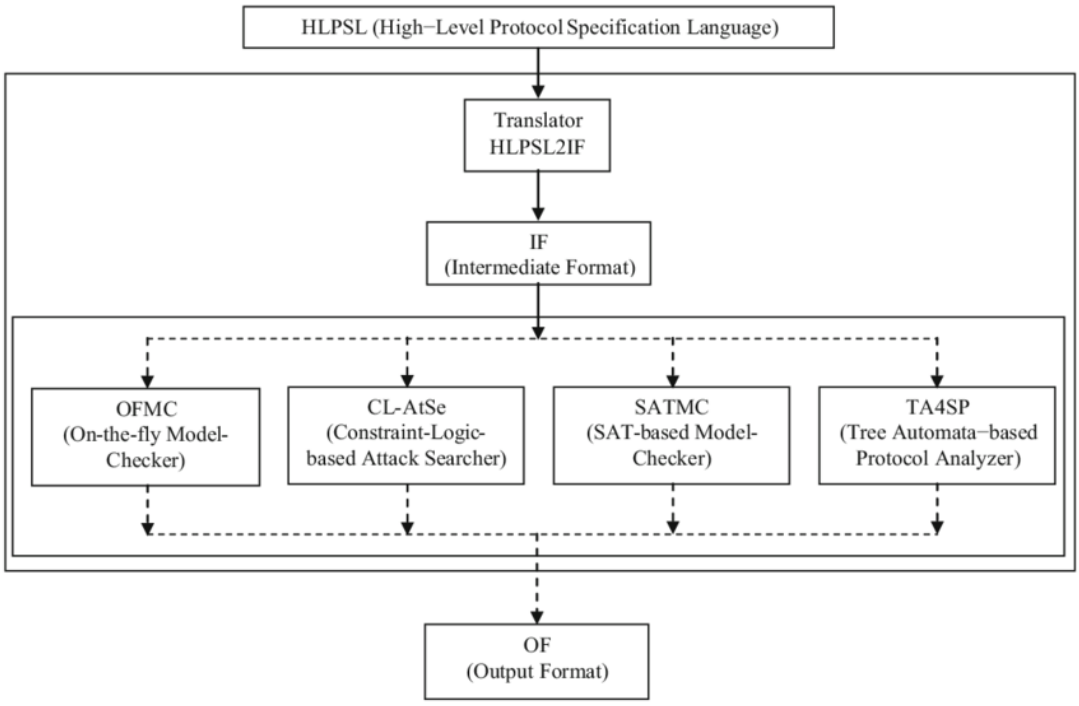

Figure 4.2: The architecture of AVISPA Tool.

1. On-the-fly Model Checker (OFMC)
   It searches simply all the known attacks and tries to discover a new attack on the protocol through testing [10].

2. Constraint Logic Based Attack Searcher(CL-AtSe)
   Constraint solutions are performed in this model to process unbounded sessions in the test protocol[14].

3. SAT-based Model Checker (SATMC)
   This model is performed to process bounded analysis of the security problems in the protocol that involves an intruder[8].

4. Tree Automata based on Automatic Approximations for Analysis of Security (TA4SP)
   This model is used to try to predict intrusion or malicious information by means of languages of a regular tree[11].

CHAPTER 5

EXAMPLE OF PASSWORD BASED USER AUTHENTICATION PROTOCOL AND ITS ANALYSIS

5.1 Definition of Password-Based UA Scheme

An example for password-based user authentication and key agreement protocols is the protocol proposed by Turkanovic et al. [36] in 2014. This protocol consists of four main phases, which are the pre-deployment (setup) phase, the registration phase, the login phase and the authentication phase. There are two additional phases; password change phase and dynamic node addition phase. Also, this protocol uses a smart card in addition to passwords for authentication.

In this protocol; the user U_i and the sensor S_j do not authenticate each other directly. In brief, mutual authentication is achieved as follows. First, the gateway authenticates the user U_i and the sensor S_j. Next, the sensor S_j authenticates the gateway and finally the user U_i authenticates the gateway. In short, the gateway sets the mutual authentication with the user U_i and the sensor S_j. Finally, the user U_i and the sensor S_j trust each other for gateway's sake.
The notations used in the definition of the example protocol are provided in Figure 5.1

5.1.1 Pre-Deployment Phase

In this phase, a sensor node is identified with its identity S_j, $1 \leq j \leq n$ and a pre-shared key X_{GWN-S_j}. This pair is stored in the sensor node. Likewise, the gateway is identified by its own secret password X_{GWN}.

5.1.2 Registration Phase

This phase is separated into two parts; registration of user and registration of sensor node.

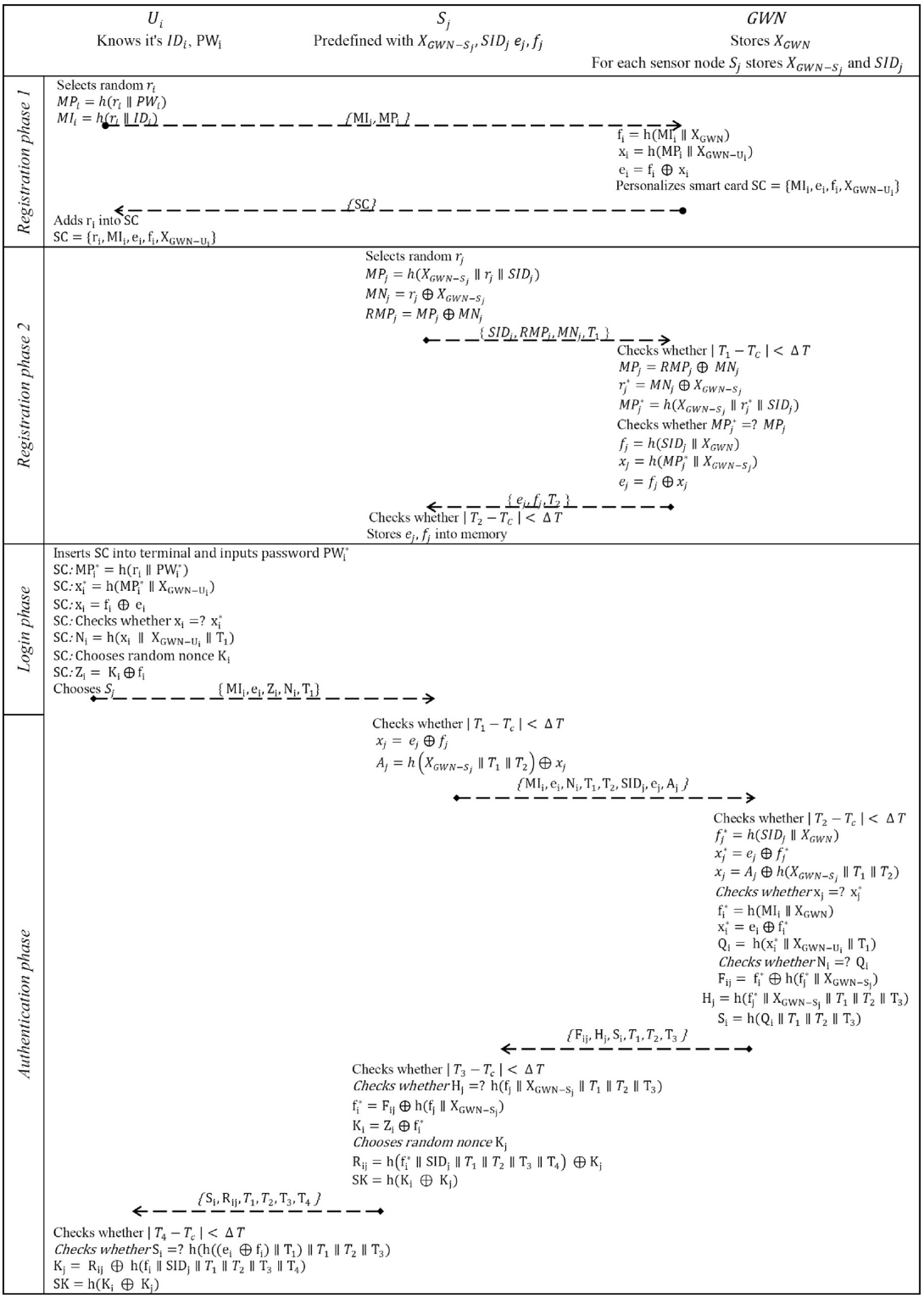

Figure 5.1: Recommended scheme by Turkanovic et al. [20]

Table 5.1: Notations

Notations	Description
U_i	User i
S_j	Sensor node j
GWN	Gateway/sinkhole
ID_i	Identity of user i
PW_i	Password of user i
X_{GWN-U_i}	Pre-shared key between GWN and user i
X_{GWN-S_j}	Pre-shared key between GWN and sensor node j
X_{GWN}	Secret password of the GWN
r_i	High entropy nonce generated by user
r_j	High entropy nonce generated by sensor
$h(.)$	Cryptographic hash functions
SC	Smart card
$T_.$	Timestamps
ΔT	Time interval of transmission delay
$\|$	Concatenation operation
$\oplus$	XOR operation
SK	Session key defined among user and sensor node
MP_i	Masked password

User Registration

The registration process of a user U_i in the network is as follows. First, the user U_i picks a randomly generated high entropy nonce r_i and computes the masked password $MP_i = h(r_i \| PW_i)$ and masked identity $MI_i = h(r_i \| ID_i)$. After the computation, the user sends this pair to the GWN from the secure channel. The GWN figures out $f_i = h(MI_i \| X_{GWN})$ and $x_i = (MP_i \| X_{GWN-U_i})$. After the computation, GWN calculates $e_i = x_i \oplus f_i$. After that, the GWN personalizes the smart card (SC) of the user with these parameters: $\{MI_i, f_i, e_i, X_{GWN-U_i}\}$. After personalization of the smart card of the user U_i, the GWN stores the masked identity of user MI_i and the pre-shared password X_{GWN-U_i} in its memory. At the end, the user U_i adds random nonce r_i into the smart card. With this adding operation, registration of user U_i is done. The protocol must perform these registrations for each deployed user.

Sensor Registration

The registration process of a sensor S_j in the network is as follows. The sensor S_j picks a randomly generated nonce r_j and computes the masked password $MP_j = h(X_{GWN-S_j} \| r_j \| SID_j)$ and masked nonce $MN_j = r_j \oplus X_{GWN-S_j}$. The sensor node computes RMP_j by XORing

these two values previously computed. Next, the sensor node S_i sends $\{SID_j, MN_j, RMP_j, T_1\}$ to the GWN node over a public channel, where T_1 is the actual timestamp which is used for preventing the replay attack. After obtaining the values from S_j, the GWN node checks the timestamp $\mid T_1 - T_c \mid < \Delta T$ where T_c is actual timestamp of the GWN. If the transmission delay, namely ΔT is not small enough, the GWN aborts the registration of the sensor S_j process. Otherwise, the GWN computes MN_j by XORing RMP_j and MN_j in order to check the validity of the sensor. The GWN computes $r_j^* = MN_j \oplus X_{GWN-S_j}$ and computes its own edition of MP_j^* = $h(X_{GWN-S_j} \| r_j^* \| SID_j)$ then compares the calculated MP_j^* and receiving MP_j. If the equation does not hold, the GWN aborts the process. If it does, the GWN figures out $f_j = h(SID_j \| X_{GWN})$, $x_j = h(MP_j^* \| X_{GWN-S_j})$ and $e_j = f_j \oplus x_j$. After performing the computations, the GWN sends e_j, f_j and T_2 via the public channel to the S_j, where T_2 is the current timestamp of the GWN. Finally, the sensor checks the transmission delay $\mid T_2 - T_c \mid < \Delta T$. If the equation holds, the sensor stores e_j and f_j, thereby the registration phase of the sensor S_j ends successfully.

5.1.3 Login Phase

After the registration phase, to initiate the authentication stage firstly the user U_i needs to log in to the network. The user U_i inserts his/her smart card into the station with his/her password PW_i^*. Then, SC checks whether the user U_i is legitimate or not by verifying the password as follows. SC computes the $x_i^* = h(h(r_i \| PW_i^*) \| X_{GWN-U_i})$ and extracts the original $x_i = f_i \oplus e_i$ from SC. If the equation $x_i = x_i^*$ does not hold, the login phase is aborted. If the equality holds, it means that the user's password is valid. Then the smart card computes messages of authentication to the selected sensor S_i; $\{e_i, MI_i, N_i, Z_i, T_1\}$ where $N_i = h(x_i \| X_{GWN-U_i} \| T_1)$ and $Z_i = K_i \oplus f_i$, where K_i is randomly chosen as the first part of the session key.

5.1.4 Authentication Phase

The final phase is the authentication process, which is started by the user U_i after sending authentication messages to the interested sensor S_j. First, the sensor node S_j checks the time for preventing the replay attack. Next, the sensor node S_j computes $A_i = h(X_{GWN-S_j} \| T_1 \| T_2) \oplus x_j$, where x_j comes from the e_j and f_j stored by the sensor S_j. S_j sends $\{MI_i, e_i, N_i, T_1, T_2, SID, e_j, A_j\}$ to the GWN, where the sensor S_j just forwards MI_i, e_i, N_i and T_1 directly. The GWN checks the time, by using T_2 and its current timestamp. If time verification is provided, the GWN calculates $f_j^* = h(SID_j \| X_{GWN})$ using its secret password X_{GWN} and the identity of sensor S_j. Now the GWN can compute $x_j^* = e_j \oplus f_j^*$. On the other hand, the GWN can extract the original x_j by XORing A_i and $h(X_{GWN-S_j} \| T_1 \| T_2)$, where the $\{X_{GWN-S_j}, T_1, T_2\}$ tuple is known by the GWN. The GWN verifies the equality, which means the GWN authenticates S_j and goes on for authentication of the user U_i. The GWN calculates $f_i^* = h(MI_i \| X_{GWN})$, where MI_i was sent by sensor S_j before and X_{GWN}

is the secure password of the GWN. Next, it calculates x_i^* by XORing e_i sent by the sensor and f_i^*. Now, the GWN can compute $Q_i = h(x_i \| X_{GWN-U_i} \| T_1)$. After computing Q_i, the GWN compares Q_i and N_i, if these are the same, the user U_i is authenticated by the GWN successfully. Now, the GWN needs to create several values in order to complete the mutual authentication and key agreement. One of these values is F_{ij}, which consists of $f_i^* \oplus h(f_j^* \| X_{GWN-S_j})$. F_{ij} is to be used for extraction of the first part of the session key by the sensor node. The GWN then computes $H_j = h(f_j^* \| X_{GWN-S_j} \| T_1 \| T_2 \| T_3)$ and $S_i = h(Q_i \| T_1 \| T_2 \| T_3)$. Having calculated the values, the GWN sends $\{F_{ij}, H_j, S_i, T_1, T_2, T_3\}$ to the sensor S_j over the public channel. With this step, the GWN has completed its task. After receiving the message to sensor node S_j, it firstly checks the transmission delay using T_3 and its current timestamp. Then, the sensor S_j tries to authenticate the GWN by calculating the value $h(f_j \| X_{GWN-S_j} \| T_1 \| T_2 \| T_3)$. If this value equals H_j, the GWN is verified by the sensor S_j. Now, the sensor node S_j needs to extract the first part of the session key K_i. To do this, the sensor node computes $f_i^* = F_{ij} \oplus h(f_j \| X_{GWN-S_j})$, in which both f_i and the pre-shared X_{GWN-S_j} are known by the sensor S_j. Finally, the sensor S_j extracts the K_i by XORing Z_i, which was sent by user U_i in the first authentication message. Then it chooses K_j , where K_j is the second part of the session key. Later, the sensor node S_j masks the K_j by $R_{ij} = h(f_i^* \| SID_j \| T_1 \| T_2 \| T_3 \| T_4)$ in a way that only the user U_i can extract. The sensor S_j has obtained the session key $SK = h(K_i \oplus K_j)$. Eventually, the sensor S_j sends $\{S_i, R_{ij}, T_1, T_2, T_3, T_4\}$, where S_i and T_1, T_2, T_3 are forwarded from the GWN over the insecure channel. The user U_i checks the time using T_4 and its current timestamp. If transmission time is small enough, the user U_i calculates the value $h(h((e_i \oplus f_i) \| T_1) \| T_1 \| T_2 \| T_3)$. If this calculated value is the same as the one received from S_i, it means that the GWN is verified by the user U_i. And, now the user U_i needs to extract the K_j as the last steps. In order to obtain the K_j, the user U_i XORs the received R_{ij} with $(f_i \| SID_j \| T_1 \| T_2 \| T_3 \| T_4)$. In the end, the user U_i has obtained the temporary session key $SK = h(K_i \oplus K_j)$ from his/her credentials. The authentication phase ends with this last operation.

5.2 Security Analysis of The Example Protocol

In this section, we perform the security analysis of the example authentication protocol in terms of several attacks and security features.

Mutual Authentication

The example protocol achieves mutual authentication, in other words every party in the network authenticates each other directly or indirectly. Firstly, the GWN authenticates the user and the sensor node in the authentication phase. Later, the sensor node authenticates the GWN. Lastly, the user authenticates the GWN in last step of the authentication phase. Although the sensor node and the user node do not authenticate each other directly, the user and the sensor node set mutual authentication with the GWN, where the GWN is playing the role of a trusted third party.

Key Agreement

Turkanovic's protocol provides key agreement among the user and the sensor node at the end of the authentication phase. The first part of the session key SK is K_i, generated by the user. K_i is transported along the authentication route, masked with f_i such that the sensor node can extract K_i. Afterwards, the sensor node obtains the K_i and choose the second part of the session key K_j. Then the sensor node sends the masked K_j, namely $R_{ij} = h(f_i^* \| SID_j \| T_1 \| T_2 \| T_3 \| T_4)$ to the user. Finally, the user has information to extract the K_j.

Replay Attack

Turkanovic's protocol is resilient against the replay attack, since the receivers check timestamps for each message that they received from the public channel.

User Anonymity and Traceability

If an attacker has a chance to separate a user in distinct login sessions, the user is traceable. In Turkanovic's protocol, the attacker can obtain MI_i, e_i, Z_i, N_i, T_1 from the login message of a user. Since MI_i, e_i are the same for all sessions of U_i, the attacker can identify the specific user U_i.

Anonymity of Sensor Node

In the authentication process of Turkanovic's protocol $MI_i, e_i, N_i, Z_i, T_1, T_2, SID_j, e_j, A_j$ are sent by the sensor node S_j to GWN via the public channel ,which is visible to the attacker. Therefore, the sensor node identity SID_j is open. This means the protocol does not provide anonymity for the sensor nodes.

Stolen Smart Card attack

According to [26] and [29] the attacker can extract the information in any SC. In this protocol, the attacker can obtain $r_i, MI_i, e_i, f_i, X_{GWN-U_i}$ by this way. Attacker can perform an offline password guessing attack as follows,

First, the attacker calculates $x_i = f_i \oplus e_i$ by using e_i and f_i from the smart card. Then, it chooses a password PW_i and computes $x_i^* = h(h(r_i \| PW_i) \| X_{GWN-U_i})$. If x_i is equal to x_i^* the attacker succeeds in guessing the password. Otherwise the attacker chooses another password to try until the equation holds.

Disclosure of Secret Parameters

As we mentioned in the stolen smart card attack, the attacker can obtain sensitive information by using a stolen smart card, such as $r_i, MI_i, e_i, f_i, X_{GWN-U_i}$.

- The attacker captures whole messages among the sensor node S_j and the gateway GWN, which are, $\{MI_i, e_i, N_i, Z_i, T_1, T_2, SID_j, e_j, A_j\}$ and $\{F_{ij}, H_j, S_j, T_1, T_2, T_3\}$. Hence forth, the attacker can calculate $h(f_j \| X_{GWN-S_j}) = f_i \oplus F_{ij}$, since $F_{ij} = f_i \oplus h(f_j \| X_{GWN-S_j})$.

- Secondly, the attacker observes the session of each user U_k and captures $\{MI_k, e_k, Z_k, N_k, T_1^{'}, T_2^{'}, SID_j, e_j, A_j^{'}\}$ and $\{F_{kj}, H_j^{'}, S_k, T_1^{'}, T_2^{'}, T_3^{'}\}$. Later, the attacker can obtain secret parameters of U_k f_k, which is equal to $f_k = F_{kj} \oplus h(f_j \| X_{GWN-S_j})$. In the end, $x_k = f_k \oplus e_k$.

Disclosure of the Session Key

The attacker has an opportunity to attain the session key among S_j and U_k with the information secret parameter f_k of U_k as mentioned before as follows.

- Firstly, the attacker captures messages between U_k and the sensor node S_j $\{MI_k, e_k, Z_k, N_k, T_1\}$ and $\{R_{kj}, S_k, T_1, T_2, T_3, T_4\}$ by exploiting the wireless communication.
- Later, the attacker can calculate $k_k = Z_k \oplus f_k$ and $K_j = R_{kj} \oplus h(f_k \| SID_j \| T_1 \| T_2 \| T_3 \| T_4)$.
- In the end, the attacker obtains the session key $SK = h(K_k \oplus K_j)$.

Man-in-the-Middle Attack

Turkanovic et al.'s protocol is not resilient against man-in-the-middle attack as follows.

- At first, the attacker captures and intercepts the login message $\{MI_k, e_k, Z_k, N_k, T_1\}$, which is sent by U_k to sensor node S_j. Then, the parameter Z_k as $\overline{Z}_k = Z_K \oplus K_k^{'}$ is modified by the attacker and the changed message $\{MI_k, e_k, \overline{Z}_k, N_k, T_1\}$ is sent to S_j.
- When the sensor node S_j replies with the message $\{R_{kj}, S_k, T_1, T_2, T_3, T_4\}$ to U_k the attacker, who intercepts the message again and changes the parameter R_{kj} as $\overline{R}_{kj} = R_{kj} \oplus K_J^{'}$ and sends the changed message $\{\overline{R}_{kj}, S_k, T_1, T_2, T_3, T_4\}$ to U_k. Turkanovic's scheme has no verification of the parameters R_{kj} and Z_k. Therefore, both the user U_k and the sensor S_j approve these messages as legitimate messages.
- As a result of the previous step, the user U_k calculates the session key $SK = h(K_k \oplus \overline{K}_j)$, where $\overline{K}_j$ is $\overline{K}_j = \overline{RK}_{kj} \oplus h(f_k \| SID_j \| T_1 \| T_2 \| T_3 \| T_4)$, which is equal to $K_j \oplus K_j^{'}$. Thus, U_k calculates the session key $SK_{U_k} = h(K_k \oplus K_j \oplus K_j^{'})$.
- The sensor node S_j also calculates the session key $SK = h(\overline{K}_k \oplus K_j)$, where $\overline{K}_j$ is $\overline{K}_k = \overline{Z}_{kj} \oplus f_k$, which is equal to $K_k \oplus K_k^{'}$. Thus, S_j calculates the session key $SK_{U_k} = h(K_k \oplus K_j \oplus K_k^{'})$.
- As a summary, a man-in-the-middle attack is performed by the attacker. The malicious party provides U_k and S_j calculates the different session keys. The malicious party is able to compute both session keys as follows:
  The attacker takes K_k and K_j and calculates the user U_k's session key as $SK_{U_k} = h(K_k \oplus K_j \oplus K_j^{'})$ and the sensor node S_j's session key as S_j is $SK_{U_k} = h(K_k \oplus K_j \oplus K_k^{'})$.

As a consequence, the sensor node S_j and the user U_k wrongfully accept that both of them performed secure key agreement. In fact, shared key agreement was done by the parties with the attacker. That is to say, the attacker can control the traffic among the sensor node S_j and the user U_k.

Sensor node impersonation attack

The malicious party can conduct a sensor node impersonation attack with secret information of U_k as mentioned in secret parameters' disclosure as follows:

- The malicious party intercepts the message of login $\{MI_k, e_k, N_k, Z_k, T_1\}$ and executes the following steps

  Pick a high entropy nonce $\overline{K}_j$

  Calculate $\overline{R}_{kj} = h(f_k \| SID_j \| T_1 \| T_2 \| T_3 \| T_4) \oplus \overline{K}_j$.

  Compute $\overline{S}_k = h(N_k \| T_1 \| T_2 \| T_3)$

  Send the message $\{\overline{S}_k, \overline{R}_{kj}, T_2, T_3, T_4\}$ to user U_k.

- After the user U_k receives the message $\{\overline{S}_k, \overline{R}_{kj}, T_2, T_3, T_4\}$, checks the timestamps and verifies $\overline{S}_k$ and $\overline{R}_{kj}$ as follows. $\overline{S}_k = ?h(h((e_i \oplus f_i) \| T_1) \| T_1 \| T_2 \| T_3)$ and $\overline{R}_{kj} = ?h(f_k \| SID_j \| T_1) \| T_1 \| T_2 \| T_3 \| T_4) \oplus \overline{K}_j$. It is obvious that the equation needs to hold as follows.
  $\overline{S}_k = h(N_k \| T_1 \| T_2 \| T_3)$
  $= h(h(x_i) \| X_{GWN-U_i} \| T_1) \| T_1 \| T_2 \| T_3)$
  $= h(h((e_i \oplus x_i) \| X_{GWN-U_i}) \| T_1) \| T_1 \| T_2 \| T_3)$
  Therefore, the user U_k believes that the received message is a legitimate message and there is no hindrance to calculate the session key $SK = h(K_k \| \overline{K}_j)$.

- Consequently, the attacker obtains the nonce $K_k = Z_k \oplus f_k$ and calculates the session key $SK = h(K_k \| \overline{K}_j)$.

5.3 Complexity Analysis of Example Protocol

As mentioned in the computational comparison, which is shown in table 4.2, the example protocol uses only hash operations and XOR operations, which makes the protocol efficient and storage-friendly.
Despite the fact that user authentication is not continuously performed operations, efficiency of this process is still important for low energy consumption. In this scheme, the user performs seven hash operations, the sensor node performs five hash operations and gateway node performs seven hash operations.

CHAPTER 6

CONCLUSION

In this thesis, efficient authentication and key agreement protocols for wireless sensor networks were examined. These protocols can work on devices that lack resources in computing and storage in wireless sensor networks.

In the first chapter, we introduced the IoT notion and WSN security, security features on demand and types of authentication and key agreement protocols.

In the second chapter, we gave definitions of hash functions and known authentication methods that are used in efficient authentication and key agreement protocols.

In the third chapter, wireless sensor networks from the aspect of components of WSN, network model and applications are considered.

In the fourth chapter, we presented an up-to-date historic development of password-based user authentication protocols in the literature. We focused on not just user authentication, but also other security needs for WSNs 4.1. Then, twenty security features involving attacks for user authentication protocols were considered with a comparative statement of several proposed protocols in the literature 4.1. We touched on the complexity of protocols for password-based user authentication with a comparison table 4.2. Although public key cryptosystems can make the protocols resilient against many attacks, because of storage and battery life limitations, password-based solutions are more suitable for WSNs. Lastly, we briefly mentioned how to test a protocol with BAN logic [12] and AVISPA tool [1].

In the fifth chapter, we gave an example of a password-based user authentication protocol with its definition, security analysis and complexity analysis.

To conclude, WSNs are an emerging technology in IoT environments. In the near future, these kinds of networks will be more widespread. WSNs may be used in many areas, especially critical areas such as military, business and healthcare. Consequently, security of WSNs is significant. At this point, user authentication becomes important in the security needs of WSNs. When considering the resource limitations of sensors, password-based user authenti-

cation protocols are efficient and suitable for units of WSNs and other resource-constrained devices in IoT environments. These protocols are exposed to various attacks because of their capabilities and type of communication. Therefore, we put forward the necessary guidelines for developing a secure password-based user authentication protocol and methods to test their security and complexity.

REFERENCES

[1] The avispa project, http://www.avispa-project.org/, accessed:2019-08-16.

[2] Gartner, https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf, accessed:2019-08-16.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: a survey, Computer networks, 38(4), pp. 393–422, 2002.

[4] F. Al-Turjman, *Wireless Sensor Networks: Deployment Strategies for Outdoor Monitoring*, CRC Press, 2018.

[5] R. Amin and G. Biswas, A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks, Ad Hoc Networks, 36, pp. 58–80, 2016.

[6] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, L. Leng, and N. Kumar, Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks, Computer Networks, 101, pp. 42–62, 2016.

[7] R. Anderson and M. Kuhn, Tamper resistance-a cautionary note, in *Proceedings of the second Usenix workshop on electronic commerce*, volume 2, pp. 1–11, 1996.

[8] A. Armando and L. Compagna, Satmc: a sat-based model checker for security protocols, in *European workshop on logics in artificial intelligence*, pp. 730–733, Springer, 2004.

[9] L. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, Computer networks, 54(15), pp. 2787–2805, 2010.

[10] D. Basin, S. Mödersheim, and L. Vigano, Ofmc: A symbolic model checker for security protocols, International Journal of Information Security, 4(3), pp. 181–208, 2005.

[11] Y. Boichut, P.-C. Héam, O. Kouchnarenko, and F. Oehl, Improvements on the genet and klay technique to automatically verify security protocols, in *Proc. AVIS*, volume 4, pp. 1–84, 2004.

[12] M. Burrows, M. Abadi, and R. M. Needham, A logic of authentication, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 426(1871), pp. 233–271, 1989.

[13] T.-H. Chen and W.-K. Shih, A robust mutual authentication protocol for wireless sensor networks, ETRI journal, 32(5), pp. 704–712, 2010.

[14] Y. Chevalier and L. Vigneron, Automated unbounded verification of security protocols, in *International conference on computer aided verification*, pp. 324–337, Springer, 2002.

[15] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography, Sensors, 14(6), pp. 10081–10106, 2014.

[16] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, A dynamic password-based user authentication scheme for hierarchical wireless sensor networks, Journal of Network and Computer Applications, 35(5), pp. 1646–1656, 2012.

[17] A. K. Das, S. Zeadally, and D. He, Taxonomy and analysis of security protocols for internet of things, Future Generation Computer Systems, 89, pp. 110–125, 2018.

[18] M. L. Das, Two-factor user authentication in wireless sensor networks, IEEE transactions on wireless communications, 8(3), pp. 1086–1090, 2009.

[19] W. Diffie and M. Hellman, New directions in cryptography, IEEE transactions on Information Theory, 22(6), pp. 644–654, 1976.

[20] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment, Ad Hoc Networks, 36, pp. 152–176, 2016.

[21] P. Gope and T. Hwang, A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks, IEEE Transactions on Industrial Electronics, 63(11), pp. 7124–7132, 2016.

[22] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, An enhanced two-factor user authentication scheme in wireless sensor networks., Ad hoc & sensor wireless networks, 10(4), pp. 361–371, 2010.

[23] Q. Jiang, S. Zeadally, J. Ma, and D. He, Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks, IEEE Access, 5, pp. 3376–3392, 2017.

[24] M. K. Khan and K. Alghathbar, Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks', Sensors, 10(3), pp. 2450–2459, 2010.

[25] N. Koblitz, Elliptic curve cryptosystems, Mathematics of computation, 48(177), pp. 203–209, 1987.

[26] P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, in *Annual International Cryptology Conference*, pp. 388–397, Springer, 1999.

[27] S. Kumari, A. K. Das, M. Wazid, X. Li, F. Wu, K.-K. R. Choo, and M. K. Khan, On the design of a secure user authentication and key agreement scheme for wireless sensor

networks, Concurrency and Computation: Practice and Experience, 29(23), p. e3930, 2017.

[28] C.-T. Li, C.-Y. Weng, and C.-C. Lee, An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks, Sensors, 13(8), pp. 9589–9603, 2013.

[29] T. Messerges, E. Dabbish, and R. Sloan, Examining smart-card security under the thread of power analysis, IEEE Trans. Computers, 51, pp. 541–522, 2002.

[30] J. Nam, M. Kim, J. Paik, Y. Lee, and D. Won, A provably-secure ecc-based authentication scheme for wireless sensor networks, Sensors, 14(11), pp. 21023–21044, 2014.

[31] D. Nyang and M.-K. Lee, Improvement of das's two-factor authentication protocol in wireless sensor networks., IACR Cryptology ePrint Archive, 2009, p. 631, 2009.

[32] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 21(2), pp. 120–126, 1978.

[33] W. Shi and P. Gong, A new user authentication protocol for wireless sensor networks using elliptic curves cryptography, International Journal of Distributed Sensor Networks, 9(4), p. 730831, 2013.

[34] S. Shin and T. Kwon, Two-factor authenticated key agreement supporting unlinkability in 5g-integrated wireless sensor networks, IEEE Access, 6, pp. 11229–11241, 2018.

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

[36] M. Turkanović, B. Brumen, and M. Hölbl, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion, Ad Hoc Networks, 20, pp. 96–112, 2014.

[37] M. Turkanovic and M. Holbl, An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks, Elektronika ir Elektrotechnika, 19(6), pp. 109–116, 2013.

[38] B. Vaidya, D. Makrakis, and H. T. Mouftah, Improved two-factor user authentication in wireless sensor networks, in *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 600–606, IEEE, 2010.

[39] R. Watro, D. Kong, S.-f. Cuti, C. Gardiner, C. Lynn, and P. Kruus, Tinypk: securing sensor networks with public key technology, in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pp. 59–64, ACM, 2004.

[40] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, A dynamic user authentication scheme for wireless sensor networks, in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, volume 1, pp. 8–pp, IEEE, 2006.

[41] K. Xue, C. Ma, P. Hong, and R. Ding, A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks, Journal of Network and Computer Applications, 36(1), pp. 316–323, 2013.

[42] H. Yazdanpanah, M. H. Ahangar, M. Azizi, and A. Ghafouri, A secure user authentication and key agreement scheme for hwsn tailored for the internet of things environment., IACR Cryptology ePrint Archive, 2017, p. 574, 2017.

[43] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, A secured authentication protocol for wireless sensor networks using elliptic curves cryptography, Sensors, 11(5), pp. 4767–4779, 2011.

[44] S. G. Yoo, H. Lee, and J. Kim, A performance and usability aware secure two-factor user authentication scheme for wireless sensor networks, International Journal of Distributed Sensor Networks, 9(5), p. 543950, 2013.