Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería

Modelo de Autenticación Configurable en Redes IoT Acorde al Poder Computacional de sus Nodos

TESIS

que para obtener el grado de

Doctora en Ciencias

Presenta:

Evangelina Lara Camacho

Director de Tesis

Leocundo Aguilar Noriega

Tijuana B.C. **Agosto 2021**

Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

Folio No.316
Tijuana, B.C., a 21 de septiembre del 2021

C. Evangelina Lara Camacho
Pasante de: Doctorado en Ciencias
Presente.-

El tema de trabajo y/o tesis para su examen profesional, en la Opción: TESIS

Es propuesto, por el C. Dr. Leocundo Aguilar Noriega

Quien será el responsable de la calidad del trabajo que usted presente, referido al tema: "Modelo de Autenticación Configurable en Redes IoT Acorde al Poder Computacional de sus Nodos",

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

I. Introducción.
II. Marco Teórico.
III. Un modelo unificado de protocolos de autenticación.
IV. Protocolos de autenticación generados del modelo unificado.
V. Integración de los protocolos de autenticación producidos por el modelo unificado.
VI. Conclusiones y trabajo futuro.
VII. Referencias.
VIII. Apéndices.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS QUÍMICA E INGENIERÍA CAMPUS TIJUANA

Dra. Ana Alejandra Ramírez Rodríguez
Sub-Directora

Dr. Leocundo Aguilar Noriega
Director de Tesis

M.C. Roberto Alejandro Reyes Martínez
Director Provisional

Resumen

Internet de las Cosas (Internet of Things, IoT) consiste de objetos físicos habilitados con sensores, software e interfaces de red, lo cual les permite compartir información, coordinar acciones, usar recursos eficientemente y responder prontamente a cambios en el ambiente sin intervención humana. Un amplio rango de dominios de aplicación se puede beneficiar de la tecnología IoT para atender problemas sociales. Algunos son monitoreo ambiental, monitoreo de salud, sistemas de notificación de emergencias, control de tráfico, monitoreo de cadenas de suministro, detección de fallas en la maquinaria, entre otros. Sin embargo, debido a que los objetos físicos se conectan a Internet, están expuestos a todos los problemas de seguridad y privacidad existentes en los sistemas de cómputo tradicionales. Además, en dispositivos IoT los ataques a la seguridad pueden revelar la información privada del usuario tal como estado de salud, ubicación y actividades rutinarias. Y pueden poner en peligro la salud física y mental del usuario si el adversario ataca los dispositivos vestidos por o implantados en el cuerpo del usuario. La diversidad de casos de uso en IoT resulta en una heterogeneidad de dispositivos, tecnologías de red, protocolos y modelos de comunicación, lo cual reduce la interoperabilidad de los dispositivos y hace difícil la implementación de servicios de seguridad compatibles con todos los dispositivos en un sistema. Buscando atender los problemas de seguridad e interoperabilidad en IoT, se propone un modelo unificado que permite la generación de diferentes protocolos de autenticación. Los esquemas proveen autenticación mutua entre dispositivos IoT que quieren compartir información en un canal de comunicación inseguro, con un uso eficiente de recursos y resistencia en contra de ataques potenciales en IoT. Las técnicas y operaciones criptográficas usadas son configuradas de acuerdo a las capacidades de cómputo de los dispositivos IoT; por lo tanto, los protocolos no impactan los limitados recursos de cómputo de los dispositivos. Se evaluó la seguridad de los protocolos usando métodos formales e informales. Los cuales probaron que los protocolos proveen autenticación mutua a las partes comunicantes, y que resisten ataques en IoT tales como reenvío y hombre en el medio (Man in the Middle, MITM), entre otros. Se evaluó el rendimiento de los protocolos en términos de los recursos de cómputo necesarios para su ejecución. Los resultados mostraron que son apropiados a las capacidades de cómputo de los dispositivos IoT. Adicionalmente, se comparó el rendimiento y seguridad de los protocolos con propuestas similares en la literatura, lo cual demostró que los esquemas logran más atributos de seguridad y tienen un rendimiento apropiado para IoT. Finalmente, se presenta un caso de uso de la integración de los protocolos de autenticación para proveer seguridad a la comunicación de sistemas IoT heterogéneos, el cual corresponde a una propuesta de un sistema de autenticación adaptativo.

Abstract

Internet of Things (IoT) consists of physical objects enabled with sensors, software, and network interfaces that allow them to share information, coordinate actions, use resources efficiently and respond promptly to environmental changes without human intervention. A wide range of application domains can benefit from IoT technology to address social problems. They include environment monitoring, health-care monitoring, emergency notification systems, road-traffic control, supply chain monitoring, machine failure detection, among others. However, because physical objects connect to the Internet, they are exposed to all the security and privacy problems existing in traditional computing systems. Further, security attacks on IoT devices can disclose the user's private information, such as health status, location, and routine activities. And can jeopardize the user's physical and mental health if the adversary attacks the devices worn by or implanted on the user's body. The diversity of use cases in IoT results in the heterogeneity of devices, network technologies, protocols, and communication models, which reduces the interoperability of the devices and makes it difficult to implement security services compatible with all the devices in a system. Intending to address the security and interconnectivity problems, a unified model is proposed that permits the generation of different authentication protocols. The schemes provide mutual authentication between IoT devices that want to share information on an insecure communication channel, with efficient resource usage and resistance against potential security attacks on IoT. The used cryptographic techniques and operations on the protocols are configured according to the computational capabilities of the IoT devices; therefore, the protocols do not impact the limited computing resources of the devices. The security of the protocols is assessed using formal and informal methods. Which proved they provide mutual authentication to the communicating parties, and they resist attacks in IoT such as replay and Man in the Middle (MITM), among others. The performance of the protocols is evaluated in terms of the computing resources needed for their execution. The results showed that they are suitable for the computing capabilities of IoT devices. Additionally, the security and performance of the protocols is compared with similar proposals in the literature, which demonstrated the protocols achieve more security features and have a proper performance for IoT. To conclude, a use case of the authentication protocols' integration to secure the communication of heterogeneous IoT systems is presented, which corresponds to a proposal of an adaptive authentication system.

Dedication

to

my MOTHER and FATHER,

and Jesús

with love

Acknowledgements

I want to thank the following people for their support of this work: my thesis director for his time, insight, wisdom, and knowledge imparted to me. And for his absolute commitment to improving the quality of education of the undergraduate and postgraduate computer programs. Thank the members of my thesis committee for their time, insight, and suggestions. And for providing an inspirational academic environment. Finally, thank my family for their infinite love, support, experience, and wisdom trusted with me.

This work was supported in part by CONACYT under grant number 536467.

The icons used in this document were created by:

- arduino nano by Vectors Point from the Noun Project.
- Wireless by beth bolton from the Noun Project.
- device by Ilham Fitrotul Hayat from the Noun Project.
- Computer by Danny Sturgess from the Noun Project.
- Router by Nociconist from the Noun Project.
- Temperature by Mint Shirt from the Noun Project.
- Speedometer by Greg Beck from the Noun Project.

General Index

List of Tables

List of Figures

Chapter 1

Introduction

Internet of Things (IoT) consists of physical objects enabled with sensors, actuators, software, network interfaces, and virtual representations. They can share information and resources, coordinate actions, auto-organize, use resources efficiently, and respond promptly to environmental changes without human intervention [1]. A wide range of application domains can benefit from the connectivity of everyday objects to address social issues. They include health-care monitoring, elder care monitoring, emergency notification systems, road-traffic control, environmental monitoring, supply chain monitoring and optimization, machine failure detection, etc. [2]. However, the diversity of use cases results in the heterogeneity of devices, network technologies, protocols, and communication models, which reduces the interoperability of the devices and makes it difficult for the elaboration of value-added services.

Furthermore, the interconnection of smart things creates new security issues. Due to the physical objects have a connection to the Internet, they inherit all the security and privacy problems that exist in traditional computing systems. Security attacks on IoT devices can expose users' private information such as health status, routine activities, location, financial data, etc. Further, because some smart devices are worn by the user or even implanted in his/her body, attacks on the devices can damage the user's physical and mental health [3, 4].

1.1 Problem statement

IoT is intended for a wide range of application environments; thus, different network technologies, energy sources, mobility patterns, and communication models are used according to the specific use-cases. For example, some devices are energy-limited such as those that harvest energy, have small batteries, or have batteries difficult to recharge. It can cause that the devices sleep and be unreachable for long periods to conserve their energy. Moreover, some devices can have intermittent connections to the Internet due to their mobility or remote locations, causing the devices to be sometimes reachable and sometimes not. In contrast, some devices are mains-powered and always connected to the Internet.

Additionally, resource-constrained devices might not support transport protocols such as TCP and have to use lightweight alternatives [5]. Therefore, IoT is a heterogeneous information network. IoT devices

have diverse hardware performances, such as different capabilities in computation, communication, and data storage. Furthermore, IoT systems employ a plurality of protocols, platforms, and policies. The heterogeneity in IoT causes an absence of common security services, reduces the interoperability of the devices, and generates additional costs in performance and money for the interpretation between devices. Additionally, it makes it difficult to implement security policies compatible with all the devices in a system [6]. Consequently, to achieve common security services, a unified security model for IoT has to be developed.

1.2 Hypothesis

It is possible to have mutual authentication between two IoT devices that want to exchange information on an insecure communication channel, with efficient resource usage and resistance against potential security attacks on IoT; using an authentication model in which the cryptographic techniques can be configured to obtain different authentication protocols.

1.3 Solution proposal

A solution to the lack of interoperability between IoT devices with heterogeneous computing capabilities is services that can be configured according to the available computing resources. It can exist a unified model that establishes the activities the services have to perform. However, the processes involved in each activity are selected according to the capabilities of the device implementing the service.

In authentication protocols, this solution consists of defining the protocol phases needed to achieve mutual authentication. Then, the cryptographic functions that compose each phase are determined according to the devices' computing capabilities. Resource-constrained devices implement only symmetric key cryptography functions, whereas devices with fewer restrictions implement symmetric and asymmetric key functions. The configurable functions are encryption algorithms, hash functions, and arithmetic and logic operations.

1.4 Objectives

General objective.

Design and validate an authentication model for IoT networks. The model allows obtaining different authentication protocols configured according to the computational capabilities of the IoT nodes. Also, the protocols achieve mutual authentication and resist potential attacks on IoT.

Specific objectives.

- To analyze current authentication protocols for traditional networks.

- To analyze current proposals of authentication protocols for IoT networks.
- To determine the protocol phases needed to achieve mutual authentication.
- To determine the cryptographic functions that compose each phase according to the devices' computing capabilities.
- To validate the authentication protocols.

1.5 Methodology

- To review current authentication protocols for traditional networks and analyze their viability for IoT networks.
- To review state-of-the-art authentication protocols for IoT networks of different performances.
- To analyze the security requirements that the authentication protocols should comply with to achieve mutual authentication.
- To analyze the phases that compose the authentication protocols.
- To analyze the cryptographic functions that constitute each phase. Determine which can be configured according to the devices' computing capabilities.
- To research how to evaluate the authentication protocols' security.
- To validate the authentication protocols.

1.6 Outline

The contents of this document are organized into five chapters. They are briefly describe below.

- **Chapter 1:** Contains the introduction, problem statement, hypothesis, solution proposal, objectives, and methodology of this work.
- **Chapter 2:** Provides background information about the IoT paradigm, cryptographic techniques, formal methods to assess the security of authentication protocols, and related works.
- **Chapter 3:** Introduces the proposed unified model of authentication protocols.
- **Chapter 4:** Presents authentication protocols generated from the unified model.
- **Chapter 5:** Describes a use case of the authentication protocols' integration to secure the communication of heterogeneous IoT systems.
- **Chapter 6:** Presents the conclusions and future work.

Chapter 2

Background

2.1 Internet of Things (IoT)

Today's society is orienting towards a more connected world. IoT is a technology that enables physical objects with virtual representations that allow them to interchange contextual data. The physical objects can use the information to coordinate actions, respond promptly and better to changes in their environments, and use their resources efficiently. Furthermore, the contextual data allows users to make informed decisions [7, 8]. Many applications are interested in the benefits that smart environments can provide. They include industry, smart cities, connected medical services, smart agriculture, smart retail, and smart homes [9]. IoT devices are composed of sensors and actuators, allowing devices to be aware of their environments and respond to their changes. Additionally, they have a communication way to interact with other devices [10].

Ideally, IoT devices are everywhere, and they are always on and available. Therefore, through the state of the devices' sensors and actuators, users can achieve a better understanding of the current context [8, 11].

The physical devices now have an Internet connection with the IoT technology, either directly or through gateways. The connectivity makes them accessible anytime to their service clients and from anywhere in the world. Nonetheless, the devices are also available to adversaries who can be anywhere in the world. They can attack IoT devices to obtain private information. Even they can use actuators to perform actions that damage the system or harm the user [12, 13, 14].

2.1.1 IoT benefits

Some of the benefits that IoT brings are the new market segments made possible by the ability to interact with physical devices through digital interfaces. IoT makes it possible to optimize the business process by using advanced analysis techniques on the IoT data streams. Additionally, it creates new business opportunities for transversal services based on common communication platforms. Some of the key areas where IoT solutions can provide competitive advantages versus traditional solutions are the following [7, 10].

- *Smart homes and buildings:* Integrating IoT technologies into these environments can help to reduce the consumption of resources such as water, electricity, and gas. The impact of the reduction is economical

and social because it decreases the users' expenses and reduces the carbon footprint associated with buildings, which are significant contributors to the emission of greenhouse gasses. The sensors in IoT devices have an essential role in smart homes and buildings because they are necessary to monitor the resource usage and proactively detect the current users' needs, e.g., turn on or off the heating, lights, etc.

- *Smart cities:* The implementation of advanced communication services and infrastructure at a city level can help to optimize the use of the current physical infrastructure, including highways and water and electricity networks. IoT technologies can monitor traffic on roadways and provide recommendation systems to drivers to avoid road congestions. Smart parking can improve urban mobility by monitoring available parking spaces and give recommendations to drivers. Additionally, sensors can monitor the level of environmental pollution and send information to health agencies.

- *Environment monitoring:* The capacity of measuring and communicating the state of natural phenomena such as temperature, rain, wind, fire, earthquakes, and rivers' depth, allows the prompt detection of anomalies that endanger human and animal lives. An extensive sensor deployment and the capacity of real-time accessing the environmental data let implementing efficient coordination strategies to mitigate damage and reduce the disaster level. Further, sensors can monitor critical regions where human operators are not viable, such as volcanic areas.

- *Smart agriculture:* Extensive sensor networks implemented on fields allow farmers to monitor earth conditions that need special attention. Further, enable the smart packaging of seeds and fertilizers and the prompt deployment of pest control mechanisms that respond to local conditions. Agricultural productivity increases because of a better understanding of climate variability, earth conditions, and plant growth models.

- *Health-care:* Patients can have sensors that monitor the intake of medicine and body parameters such as blood pressure, temperature, the glucose level in the blood, etc., to have an integral image of the patient's health status. Further, accelerometers and gyroscopes allow monitoring of users' activities, including burned calories, steps walked, exercise performed, accidents, etc. Aggregated information is transmitted to medical centers to have a prompt response in case of a medical emergency.

- *Inventory and product managing:* Through Radio-frequency identification (RFID) technologies, goods can be identified and tracked. RFID labels are placed on products to monitor and manage their movement through supply chains, enabling interoperability between RFID-based applications of the different actors that administer products during the products' life cycle. In retail commerce, IoT

technologies allow accurate real-time monitoring of merchandise in inventory. Additionally, after users purchase the products, they can automatically obtain the items' information.

2.1.2 IoT architecture

Typically, the IoT architecture is divided into three basic layers: perception, network, and application [15]. They are illustrated in Figure 2.1.

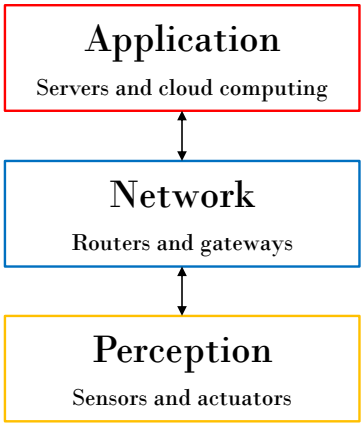


Figure 2.1 IoT architecture of three layers.

- *Perception:* The main objectives of this layer are the physical objects' connection to the IoT network and the measurement, collection, and processing of information about the objects' state. Then, the transmission of the information to the next layer. Things interact through smart devices, such as sensors, actuators, and RFID devices.

- *Network:* Receives information processed by the perception layer and determines routes to transmit them through integrated networks to hubs, applications, and devices. This layer incorporates several devices, such as cloud computing, commuters, hubs, gateways, etc. Through diverse communication technologies such as WiFi, Bluetooth, etc. The network layer transmits information through gateways or interfaces between heterogeneous networks, using several communication technologies and protocols.

- *Application:* Receives information transmitted by the network layer and uses them to provide operations and services, such as the storage in a database. It can use the information to predict the future state of physical devices. The latter involves procedures such as data mining, data analysis, etc.

2.1.3 IoT heterogeneity

An embedded systems classification according to the computing capabilities and available resources is in RFC 7228 [16]. It contains distinct classes concerning the devices' capabilities of computing, network, and energy. Table 2.1 presents the systems' classifications involving the computing capabilities, Table 2.2 the classes regarding the energy resources availability. Finally, Table 2.3 shows the categories of network capacities.

Concerning the classifications in Table 2.1, the C0 devices are very limited in processing and memory capacities; thus, it is probable that they do not have enough resources to communicate directly to the Internet in a secure manner. The devices require intermediates such as gateways, proxies, and servers to communicate through the Internet. They collect the information sent by the C0 devices in their networks using their protocols and communication technologies and redirect the data to services on the Internet to make them available to users and systems.

The C1 devices are also limited in processing and memory capacities. Consequently, they cannot implement full Internet protocol stacks such as Transport Layer Security (TLS), Hypertext Transfer Protocol (HTTP), etc. However, they have enough capacity to communicate to the Internet without intermediaries and implement protocol stacks designed or adapted to systems with limited resources. An example of this type of protocol is Constrained Application Protocol (CoAP), which uses the User Datagram Protocol (UDP) to achieve a communication type without connection. C1 devices can integrate into IP networks, but they need to be careful with the memory, code space, and energy usage.

Finally, C2 devices are less limited in computing capabilities, and they are usually able to implement the same protocol stacks as traditional computing systems. Nonetheless, the devices can benefit from lightweight communication and security protocols that use few network and energy resources. Thus, more resources are available to applications, and the devices' battery lives extend.

In addition to the limitations in computing capabilities, some devices are limited in energy resources. RFC 7228 proposes four classes to categorize devices concerning their energy limitations. They are E0, E1, E2, and E9, as shown in Table 2.2. An E0 device has limited energy, and it is only available during a specific event, e.g., only when pressing down an energy-harvesting interrupter button. E1 devices have energy limitations for particular periods. Some examples are devices charged by solar panels with a limited amount of energy for their night operation. Also, devices are connected manually to chargers, and there is a period between recharges. E2 devices are limited in the total energy available during their life cycle. I.e., the device's battery is not replaceable; thus, once the battery is empty, the device is discarded. Finally, E9 devices do not have relevant energy limitations.

One of the more energy-consuming activities in devices is wireless data transmission. Devices use different connection strategies concerning the frequency they need to transmit information and the energy source and

capacity available. RFC 7228 proposes three approaches which are usually off, low power, and always on. Their name classes are P0, P1, and P9, respectively. Table 2.3 shows the strategies. P0 devices have many energy limitations, but they need to communicate frequently. Techniques used to minimize the energy consumption include decreasing the work related to the communication reestablishment tasks after the device wakes up from low-power mode and tuning the communication frequency appropriately. P1 devices sleep for prolonged periods. Therefore, once they wake up, they perform all the network reinsertion processes. In this case, the optimization goal is to minimize the effort required in the reinsertion. Finally, P9 devices do not have relevant energy-saving requirements in their communications; therefore, they can be connected to the network all the time.

Table 2.1 Embedded systems' classifications concerning the computing capacities

Class name	Data memory	Program memory
Class 0 (C0)	$\ll$ 10 KiB	$\ll$ 100 KiB
Class 1 (C1)	$\sim$ 10 KiB	$\sim$ 100 KiB
Class 2 (C2)	$\sim$ 50 KiB	$\sim$ 250 KiB

Table 2.2 Embedded systems' classifications concerning the energy resources.

Class name	Energy limitation type	Energy source example
E0	Energy limited for event.	Energy harvesting based on events.
E1	Energy limited for period.	Battery that is periodically recharged or replaced.
E2	Energy limited for life cycle.	Main battery not replaceable.
E9	Without direct quantitative energy limitations.	Connected to an stable power grid permanently.

Table 2.3 Embedded systems' classifications concerning the network capacities.

Class name	Strategy	Communication capacity
P0	Normally off.	Reconnection when required.
P1	Low power.	Appears as connected, may have high latency.
P9	Always on.	Always connected.

2.1.4 IoT security

Due to their close relationship with the physical world, IoT technologies must be secure from design and have mechanisms to preserve information privacy. IoT devices collect information about all the aspects of the user's life, both physical and conceptual. For example, location, preferences, medical information, activity schedule, etc. The advantage of knowing this information is that it helps to understand a context or situation. However, at the same time, it increments the security threats because adversaries can maliciously or inappropriately use the context. It can be exposed the user's identity, location, activities, and behavior. Therefore, utmost care should be taken when collecting, modeling, and storing information.

Systems should ensure some properties to achieve security in data, services, and IoT systems. They include confidentiality, integrity, authentication, authorization, non-repudiation, availability, and privacy. They are major challenges due to IoT environment characteristics such as IoT devices with limited computing resources and the heterogeneity of devices, networks, and services. It is improbable a large-scale adoption of IoT solutions if there is no assurance of confidentiality, authenticity, and privacy [7, 8, 9, 17]. The security properties that IoT environments need to achieve are describe below.

- *Confidentiality:* It is difficult for some applications to ensure data confidentiality because they are related to the physical environment. It is relevant in the business context because data can represent goods that need to be protected to preserve competitiveness and market value. In IoT, authorized users and objects can access data. Conventional solutions to assure data confidentiality cannot always apply directly to these contexts because of scalability problems due to the large amount of data generated by the systems and because of the necessity of controlling the data access in a flexible and online manner, with access rights that change during run-time and that apply to dynamic data streams.

  Furthermore, to avoid unauthorized access to the systems' lower layers, such as wireless communications and hardware components, should be combined access control mechanisms with proper data protection techniques. Some examples are mechanisms to suppress data or to make them look random or indistinguishable from others.

- *Privacy:* It is a fundamental IoT requirement because it defines the rules that designate which of the data related to a particular user can be accessed. The challenges in IoT data privacy preservation classify in data collection policies and data anonymity. The first describes the strategies that specify the types of data an application can gather and the access control. It can be restricted the type and quantity of collected and stored information through data collection policies; therefore, preserving the user's privacy. The second challenge concerns cryptographic data protection and masking of data relationships to remove direct connections between data and their user owners.

- *Trust:* Trust negotiations concern the process of credentials exchange. It allows an entity that requests a resource or service to provide the required credentials to obtain it. Negotiations depend on peer-to-peer interactions and comprise the iterative demonstration of digital credentials representing certified statements from trusted entities. They allow verifying the owner's properties to establish mutual trust. In such scenarios, access to resources is only possible after the trust negotiation completes. The ability to achieve the trust requirements is strictly related to identity management and access control. An effective trust model should also consider the highly distributed nature of IoT and the requirements related to computational complexity and applications' response times.

2.1.5 IoT security challenges

One of the main challenges of IoT applications is the security and privacy of users' data. IoT applications transmit data through insecure wireless networks exposed to risks such as message interception, eavesdropping, and modification. Many IoT applications send sensitive information. Its unauthorized disclosure causes social and familiar problems, psychological distress, the risk of users losing their jobs, or even risk their lives. Examples of situations that negatively impact the users are the unauthorized disclosure of information related to their financial and health status.

Basic requirements in any IoT application are security and privacy. Users will not share confidential information unless the network is trustworthy. Security solutions of traditional networks are sufficiently developed; however, they are infeasible for IoT contexts. The reasons are the size and heterogeneity of IoT networks and the limited computing and energy resources in IoT devices. Some of the challenges traditional network security solutions face when trying to implement them in IoT contexts are described below [18].

- *Cryptographic techniques:* Cryptographic algorithms currently used were designed for the computing, energy, and network capacities of traditional computing systems. The symmetric key encryption algorithms such as Advanced Encryption Standard (AES). Hash functions like Secure Hash Algorithms (SHA). And asymmetric-key cryptography, including Rivest–Shamir–Adleman (RSA), Diffie–Hellman (DH), and Elliptic Curve Cryptography (ECC), are security mechanisms for traditional networks that proved their security and effectiveness. However, these solutions use many computing and energy resources [19, 20], making their conventional manner infeasible for all IoT applications. Therefore, there exists the necessity to develop security solutions adapted to the characteristics and limitations of IoT. Or to optimize the current solutions to make them viable to battery-operated IoT devices with limited resources.

- *Denial of service:* Denial-of-service (DoS) attacks provoke catastrophic effects in IoT because they can risk users' lives if adversaries execute them in critical systems such as health-care, smart vehicles, and

smart grid. Further, the DoS attack detection and mitigation techniques used in traditional networks could not be appropriate to IoT applications because IoT devices have limited resources and cannot afford the reception of even ten packages to detect and block attacks.

- *Authentication and access control:* Many of the communication done in IoT is of machine-to-machine (M2M) type. Node authentication is significant in this case to assure the security and privacy of the transmitted information. The nodes have to authenticate mutually before exchanging sensitive information. However, there is no efficient authentication mechanism for the massive quantity of IoT devices that considers the devices' limitations in computing and energy resources.

- *Key management:* The administration of keys is the process of generation, exchange, and storage of cryptographic keys. Efficient key management is vital in node authentication, allowing confidentiality and integrity to the transmitted information. Additionally, it enables the blocking of unauthorized access to data and resources. Cryptographic key management in IoT is an open challenge because traditional techniques are not proper for IoT systems due to the massive number of connected devices [21].

2.1.6 Potential attacks in IoT

Some IoT applications work with sensitive user information, which needs to be protected from unauthorized uses that risk the user. Some examples of applications are Telecare Medicine Information Systems (TMIS) and financial. However, similarly to other network-based applications, IoT applications are vulnerable to diverse attacks performed by internal or external adversaries. If the attack is successful, the adversary can access private information such as personal, medical, location, and financial data. The adversary can also use actuators to damage the system or risk the user's life.

In this subsection, common attacks in the layers of the IoT architecture are described [15, 22]. The notation $\mathscr{A}$ represents the adversary.

1. Perception layer.

   (a) *Node capture attack:* Adversary captures and controls an IoT node by physically replacing the device or manipulating its hardware. The exposed device can disclose sensitive information to $\mathscr{A}$, such as group communication keys. Further, $\mathscr{A}$ can copy the data to a malicious node and pretend it is a valid node when joining the IoT network.

   (b) *Malicious code injection attack:* Besides capturing a device, the adversary can control nodes by inserting malicious code into the devices' memories. The code can enable $\mathscr{A}$ to have access to the system or even gain absolute control of it.

(c) *Injection of false data attack:* After $\mathscr{A}$ captures an IoT node, he/she can introduce untruthful information in the network instead of the accurate data measured by the device before its capture. Consequently, the IoT applications give erroneous feedbacks and reactions, which affect the system's effectiveness.

(d) *Forwarding attack (attacks on data freshness):* The adversary uses a malicious node to forward legitimate information that he/she has captured. The adversary intention is to earn the trust of the IoT system.

(e) *Cryptanalysis and side-channel attacks:* Cryptanalysis attacks use ciphertexts and plaintexts to deduce the secret key used to encrypt. In contrast, side-channel attacks use techniques such as time analysis of the encryption algorithm's execution time to obtain the key.

(f) *Eavesdropping and jamming:* Most IoT applications use wireless networks to communicate. Therefore, they become exposed to potential risks of eavesdropping on the transmitted data. Further, $\mathscr{A}$ can send noise in the communication channel to block legitimate transmissions.

(g) *Sleep deprivation attack:* IoT devices have limited energy resources, and most of them are battery-operated. Therefore, the devices are programmed to follow sleep routines that set them in low power consumption and prolong the batteries' lives. This attack inhibits sleep routines, keeping devices awake all the time until consumption of its energy source.

2. Network layer.

(a) *DoS:* This attack consumes all the available resources in the IoT device or system through attacking network protocols or inundating the network with a massive number of packets, causing the system to be unavailable to legitimate users.

(b) *Impersonation attack:* The purpose is for the adversary to obtain full access to the IoT system and sending malicious information. Some examples are IP impersonation, where $\mathscr{A}$ stores IP addresses of legitimate users and sends dishonest data in the system using the obtained IP to make them appear valid. And RFID impersonation, where $\mathscr{A}$ stores information about authentic RFID labels and transmits fraudulent information using their ID.

(c) *Sinkhole attack:* An exposed or malicious node pretends to have exceptional computing, communicating, and energy capacities, more than its neighbors. Consequently, the other network devices select it as a forwarding node in the process of data routing. The dishonest node can increase the number of obtained data before sending them to the system. The attack not only violates the confidentiality of the data transmitted, but it can also be the base step before the adversary performs attacks such as DoS.

(d) *Wormhole attack:* Two dishonest devices that are in separate locations can initiate the attack. The nodes exchange routing information on private channels to achieve a fake one-hop transmission between them, even if the devices are in distant locations from each other. Therefore, the devices transmit more data than the other nodes due to the false reduction of the forwarding hops. The attack causes damages similarly to the sinkhole attack.

(e) *Man in the Middle Attack, (MITM):* A malicious device controlled by the adversary is virtually in between two communicating parties. The device is in the middle of the communication and stores and forwards the data sent between the nodes. The adversary performs the attack by coping with the identification information of the legitimate parties. The devices cannot detect the presence of the dishonest node. They believe they have direct communication. The attack compromises the confidentiality, integrity, and privacy of the transmitted data. The adversary controls the communication channel used by the legitimate entities. Therefore, the adversary can monitor, eavesdrop, and modify data. The launching of MITM attacks depends only on the communication protocols used in the IoT network, in contrast to node capture attacks that need physical contact with the device.

(f) *Routing information attack:* The attack centers on the routing protocols used in the IoT systems. The adversary modifies the routing information to create routing loops in the data transmission. Therefore, the paths extend, and the end-to-end transmission delay increments.

(g) *Sybil attack:* A dishonest device, denoted as Sybil, affirms that it has many legitimate identities and impersonates the authentic nodes. A Sybil node sends fake information in the network that is undoubtedly accepted by honest devices because the node uses legitimate identities. Further, routes that select the Sybil node as forwarding node can believe that there exist multiple transmission routes when there is only one and uses the Sybil node. Therefore, the adversary can use the path to perform DoS or jamming attacks.

(h) *Unauthorized access:* Many RFID-based devices integrate IoT networks. However, the majority of the RFID labels do not have appropriate authentication mechanisms. Consequently, the adversary can access RFID labels and obtain, modify, and erase the information stored in them.

3. Application layer.

(a) *Phishing attack:* $\mathscr{A}$ obtains users' confidential data such as identities and passwords when supplanting authentication credentials through infected e-mails and phishing websites.

(b) *Virus and worms:* The adversary can infect an IoT system by spreading Trojan horses, worms, etc., to obtain and modify confidential data.

(c) *Malicious scripts:* The adversary adds, modifies, and erases scripts in the software to damage the functions of the IoT system. $\mathscr{A}$ deceives users to execute malicious scripts when requesting services on the Internet. The scripts can filter confidential information or cause a system crash.

2.2 Authentication

Authentication is one of the most significant objectives in information security. Until the mid-1970s, researchers believed that data secrecy and authentication were essentially connected. However, the development of hash functions and digital signatures showed that they are independent security objectives [23].

The following situations exemplify the unrelated aspects of authentication. There are two contexts if two entities, named A and B, want to ensure the identity of the other.

1. A and B communicate without a significant delay, i.e., both are active in real-time.
2. A and B exchange messages, the communication has a significant delay. The data are routed through several networks, stored, and forwarded sometime later.

In the first context, A and B have to verify their identities in real-time. To achieve it, A sends a challenge that only B can correctly respond to and vice versa. This type of authentication is generally named entity authentication.

In the second context, it is unsuitable sending a challenge and wait for a response. Further, the communication could be one-way only. The name of this identification process is data origin authentication.

This document proposes a unified model for entity authentication. The purpose of the mechanisms for entity authentication is to guarantee to one or more parties the identity of another. They perform it through the collection of probative evidence. Additionally, they assure that the party was active during the acquisition or creation of the proof. The authentication generally does not involve significant messages other than those used to attest that an entity is someone in particular.

The general context of an authentication protocol comprises a petitioner and an examiner. The latter knows beforehand the claimed identity of the petitioner. The intention is to prove that it is his/her true identity.

2.3 Cryptographic techniques

This section describes the cryptographic techniques used in the proposed unified authentication model.

2.3.1 Cryptographic hash functions

Protocols use widely one-way cryptographic hash functions to provide integrity to the transmitted data. The hash functions are very susceptible to changes in their inputs, even if they are minimal. If two very similar inputs enter into the hash function, the outputs should not be correlated.

Let S_a and S_b be two values of arbitrary size input to a hash function and let $h(S_a)$ and $h(S_b)$ be the outputs generated by the function. A one-way cryptographic hash function that is collision-resistant has the following properties [24, 25].

- An insignificant variation in S_a generates a massive variation in $h(S_a)$.
- It is computationally easy to compute $h(S_a)$ knowing S_a. However, it is infeasible to compute S_a knowing only $h(S_a)$.
- It is infeasible to found a pair $\{S_a,\ S_b\}$ that fulfills $h(S_a) = h(S_b)$. The name of this property is collision resistance.

A one-way collision-resistant cryptographic hash function can be formally defined as follows.

Let $h : \{0,1\}^* \rightarrow \{0,1\}^n$ be a one-way hash function. When $h(.)$ receives an input of arbitrary size, it generates a fixed-size output of n bits. The output is named message digest or hash output. Let $Adv_{\mathscr{A}}^{HASH}(t)$ be the advantage the adversary has in detecting a hash collision in a polynomial time t. Therefore, $Adv_{\mathscr{A}}^{HASH}(t) = Pr[\mathscr{A}(S_a, S_b) \in_R \mathscr{A} : S_a \neq S_b \text{ and } h(S_a) = h(S_b)] \leq \epsilon$. The advantage $Adv_{\mathscr{A}}^{HASH}(t)$ is negligible.

2.3.2 Elliptic Curve Cryptography

An introduction to ECC is presented following. Authentication protocols use the mathematical properties of ECC to verify the entities' identities. Also, for the key exchange.

Let p and q be two large prime numbers and E an elliptic curve over the finite field F_p: $E = y^2 = x^3 + ax + b \mod p$, $a, b \in_R F_p$ and $4a^3 + 27b^2 \neq 0$. The points on E comprise an additive cyclic group G of order q. P is the group's generator.

The elliptic curve has the following group properties [26].

- **Point addition:** Let P and Q be points on $E_p(a, b)$. The addition of the points is $P + Q = R$. The point $-R$ is on the line that joins P and Q, in the position where the line intersects $E_p(a, b)$. The $-R$'s reflection on the x-axis is R.

- **Point doubling:** Let P be a point on $E_p(a, b)$. The doubling of the point is $Q = 2.P$. $-Q$ is on the tangent line to P, in the position where the line intersects $E_p(a, b)$. The $-Q$'s reflection on the x-axis is Q.

- **Scalar point multiplication:** Let $x \in Z_q^*$ and P be a point on $E_p(a, b)$. The scalar point multiplication is $Q = x.P = \{P + P + ... + P \ (x \text{ times the point addition})\}$. It is the addition of P x times.

Figure 2.2 shows an example of an elliptic curve. It shows the curve $y^2 = x^3 - 3x + 5$ and the points P and Q addition, resulting in point R.

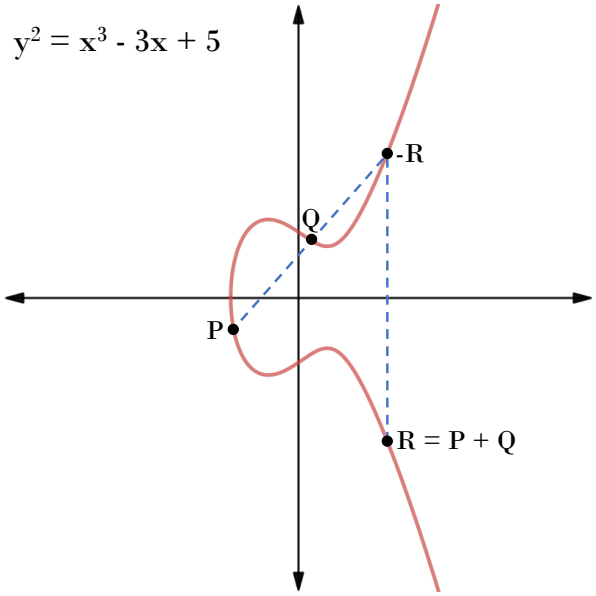


Figure 2.2 Addition of two points on an elliptic curve.

2.3.3 Computational intractable problems

Authentication protocols that use ECC base their security on the following computational intractable problems [27, 24].

- **Elliptic Curve Discrete Logarithm Problem (ECDLP):** Let P and Q be points on $E_p(a, b)$, $x \in Z_q^*$ and $Q = x.P$. $\mathscr{A}$ can compute x knowing $\{P, Q\}$ with a probability $Adv_{\mathscr{A}}^{ECDLP}(t) = Pr[\mathscr{A}(P, Q = x.P) = x : x \in Z_q^*] \leq \epsilon$. The probability $Adv_{\mathscr{A}}^{ECDLP}(t)$ is negligible.

- **Elliptic Curve Computational Diffie-Hellman Problem (ECDHP):** Let P, Q, and R be points on $E_p(a, b)$, $x, y \in Z_q^*$, $Q = x.P$, and $R = y.P$. $\mathscr{A}$ can compute $xy.P$ knowing $\{P, Q, R\}$ with a

probability $Adv_{\mathscr{A}}^{ECDHP}(t) = Pr[\mathscr{A}(P, Q = x.P, R = y.P) = xy.P : x, y \in Z_q^*] \leq \epsilon$. The probability $Adv_{\mathscr{A}}^{ECDHP}(t)$ is negligible.

2.4 Formal methods to assess the security of authentication protocols

Two formal methods were used to study the security properties of the schemes generated by the unified model of authentication protocols. They are the AVISPA tool and BAN logic.

2.4.1 AVISPA tool

The AVISPA tool performs automated verifications on cryptographic protocols. It has four back-ends to examine if attacks are possible on the security properties of the scheme under analysis. The back-ends are Constraint-Logic-based Attack Searcher (CL-AtSe), Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP), SAT-based Model-Checker (SATMC), and On-the-fly Model-Checker (OFMC) [28]. Figure 2.3 shows them. The AVISPA tool uses the High-Level Protocol Specification Language (HLPSL) in the protocol description. HLPSL is a role-based language. Each role details the information the principal knowns initially, which include cryptographic algorithms and pre-shared keys, also, the roles interactions, and the conditions for state transitions [29]. The AVISPA tool supports a protocol animator named the SPAN tool. It creates message sequence charts of both the scheme and the attacks found interactively [30].

The AVISPA tool uses a communication channel subject to a Dolev-Yao (DY) adversary. The intruder can modify, intercept, reassemble, and decompose the transmitted messages. Nonetheless, the DY threat model assumes the use of perfect cryptography; thus, the adversary can see the content of the messages only if he/she has the appropriated decryption keys [31, 32]. The tool performs three verifications. It first examines that the non-trivial HLPSL specification can run to completion; hence, the specification can enter states where attacks can exist. Secondly, the tool verifies the possibility of replay attacks by giving the adversary the information of regular sessions between legitimate parties, checking if the parties can run the scheme by searching for a passive adversary, and determining if the replay attack is possible. Finally, the tool does the DY verification in which the back-ends check if a MITM attack exists. After the three verifications, the AVISPA tool indicates whether it concludes the protocol as safe, unsafe, or the analysis is inconclusive [33]. The verification output includes the sections presented below [34].

- SUMMARY: Mentions the analysis decision, whether the AVISPA tool concluded the protocol as safe, unsafe, or the result is inconclusive.

- DETAILS: Describes the analysis conclusion, explaining why the protocol is safe, why the result is inconclusive, or when the AVISPA tool finds attacks, under what conditions they can happen.
- PROTOCOL: It displays the protocol's file path.
- GOAL: Mentions the security goals specified in the HLPSL protocol description. When the AVISPA tool finds attacks, the section displays the unachieved goals.
- BACKEND: Indicates the back-end used in the verification.
- The final section shows state statistics and the duration of the analysis. If attacks exist, it displays the vulnerabilities trace.

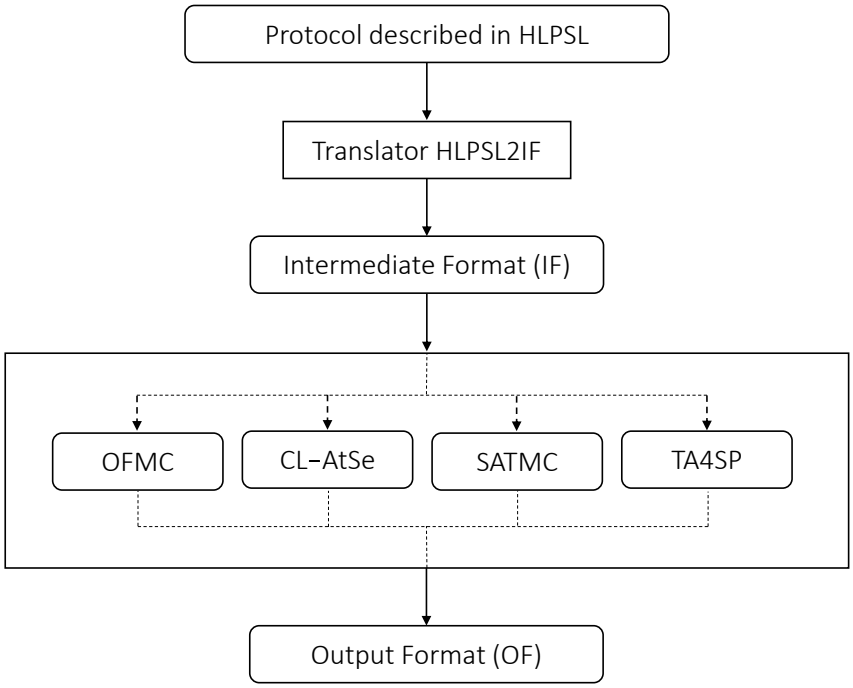


Figure 2.3 Back-ends of the AVISPA tool.

2.4.2 BAN logic

BAN logic comprises a set of postulates and rules used for verifying the trustworthiness of the transmitted information in a protocol [35]. It can be used to reason concerning what principals believe about each other. Table 2.4 introduces the BAN logic notations, and Table 2.5 the rules.

Table 2.4 BAN logic notations

Symbol	Description
P, Q	Principals.
K	Encryption key.
X, Y	Statements.
$P \triangleleft X$	P sees X.
$P \mid\equiv X$	P believes X.
$P \mid\sim X$	P once said X.
$P \overset{K}{\longleftrightarrow} Q$	K is a shared key, P and Q may use it in their communication.
$P \Rightarrow X$	P has jurisdiction on X.
$\overset{K}{\longmapsto} P$	K is P's public key.
$\sharp(X)$	X has not been used in old protocol executions, it is fresh.
$P \overset{X}{=\!=} Q$	X is a secret between P and Q. Principals trusted by them may also know it.
$\langle X \rangle_Y$	X is combined with Y. Y is a secret, it proves the originator's identity.
$\{X\}_K$	X encrypted using K.

Table 2.5 BAN logic rules

	Symbol	Description
(1)	$\frac{P \mid\equiv Q \overset{K}{\longleftrightarrow} P, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$ $\frac{P \mid\equiv Q \overset{Y}{=\!=} P, P \triangleleft \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}$	Message meaning rule.
(2)	$\frac{P \mid\equiv \sharp(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$	Nonce verification rule.
(3)	$\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$	Jurisdiction rule.
(4)	$\frac{P \mid\equiv \sharp(X)}{P \mid\equiv \sharp(X,Y)}$	A formula is fresh if one or more of its elements is fresh.
(5)	$\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X,Y)}$ $\frac{P \mid\equiv (X,Y)}{P \mid\equiv X}$ $\frac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv X}$	Belief rule.
(6)	$\frac{P \mid\equiv \overset{K}{\longmapsto} P, P \triangleleft \{X\}_K}{P \triangleleft X}$ $\frac{P \mid\equiv \overset{K}{\longmapsto} Q, P \triangleleft \{X\}_{K^{-1}}}{P \triangleleft X}$ $\frac{P \triangleleft \langle X \rangle_Y}{P \triangleleft X}$ $\frac{P \mid\equiv Q \overset{K}{\longleftrightarrow} P, P \triangleleft \{X\}_K}{P \triangleleft X}$	A principal sees the formula components if he/she posseses the required keys.

2.5 Related works

In this section, current proposals of authentication protocols for IoT are briefly analyzed.

In [36], the authors propose an authentication protocol for ad hoc wireless sensor networks (WSN). They designed the scheme for heterogeneous environments with resource-constrained sensor nodes and robust gateways. They use xor operations and hash functions. The protocol is analyzed in [37], it is found unsafe to off-line identity and password guessing, impersonation, and stolen smart card attacks. The authors propose a modified version; however, [38] shows that the new version does not achieve mutual authentication. Additionally, it is unsafe to node and user forgery, tracking, and session key disclosure attacks.

The protocol in [36] is studied again in [39]. The authors demonstrate that if the adversary compromises a device, he/she can obtain the session key exchanged by the user and another node. Then, the authors present an authentication scheme for ad hoc WSN. The protocol is lightweight since it only uses xor operations and hash functions. Nonetheless, the authors of [40] analyze the proposal and demonstrate it does not resist node capture and unknown key-share attacks and is vulnerable to node information secrecy. Then, they present an authentication scheme for IIoT. The protocol does not add timestamps to the messages. Since synchronizing time is a challenging task in distributed networks. However, the authors of [41] show that it is possible to achieve DoS and impersonation attacks because of a protocol's independence of time.

The work in [42] proposes two protocols for ad hoc WSN. One scheme is lightweight. The other uses public-key cryptography. They designed the last-mentioned for adverse environments that have a high probability of node capture attacks. The protocols are analyzed in [43], the work shows the schemes do not resist tracking and stolen smart card attacks. The protocols do not achieve mutual authentication. And, the user identity is not verified; hence, the protocol executes even if the user enters an invalid identity. Then, the authors propose a new authentication protocol to solve the insecurities and provide user anonymity and security against mobile device loss attacks. The scheme uses public-key cryptography on the user's mobile device. Therefore, it is not suitable for resource-constrained systems [44, 45].

The authors of [46] propose an authentication, authorization, and accounting (AAA) system for IIoT. It adopts the Next Generation Access Control (NGAC) standard [47], which is a flexible infrastructure used to provide access control in different domains. The proposal gives access control to heterogeneous IIoT devices in a local cloud with interoperability and security. However, the proposal uses X.509 certificates in the authentication mechanism. Therefore, the solution is not suitable for resource-constrained devices because the management of public-key certificates causes performance bottlenecks, and the validation of public-keys exhaust the devices' resources [48].

Some works propose authentication protocols based on PKI, such as [49, 50, 51]. PKI systems need the certificate authority (CA) to create certificates that bind users' identities with their public keys. The

certificate needs to be verified prior to running the authentication. However, managing certificates becomes difficult as the number of users in the system grows [52].

The authors of [53] propose a certificateless authentication protocol for Wireless body area networks (WBANs). The scheme intends to have anonymity in transmitting a patient's medical information with an Application Provider (AP). Additionally, network managers and APs cannot impersonating users or disclosing users' identities. The work in [54] presents another certificateless authentication protocol for WBANs in health-care applications. Its features are non-repudiation, client anonymity, key escrow resistance, and revocability. Nonetheless, both schemes use bilinear pairing operations; therefore, they have a high execution time, which is not suitable for resource-constrained devices [55].

Moreover, the authors of [56] analyzed the scheme in [53]. They reveal that the protocol does not resist the stolen verifier attack and does not provide anonymity. They propose a new protocol using Identity-based cryptography (IBC). The proposal advantages are that it does not use bilinear pairing operations, the AP does not need a verifier table, and the scheme is scalable to the addition of clients. Nevertheless, [52] indicates that the protocol does not provide anonymity and resistance to tracking attacks. [52] proposes a protocol to achieve true anonymity and resolve the security problems of the proposals in [53] and [56]. However, [57] detects that the scheme is insecure against impersonation attacks. The protocol lets legal entities and intruders impersonate legal entities. Later, [57] proposes an anonymous authentication protocol that has better performance than the scheme in [52]. Unfortunately, [58] discovers that an AP can impersonate a client effortlessly in the protocols [57] and [52]. And [59] indicates that the protocol in [57] does not achieve mutual authentication, and it is insecure against impersonation attacks.

The work in [55] proposes an anonymous authentication protocol for WBAN. In the proposal, the parties use only one round of communication, which results in low energy consumption and high computational efficiency. Unfortunately, the work in [60] demonstrates that the protocol is insecure to stolen-verifier, DoS, and key compromise impersonation attacks. Next, the authors propose an improved version of the scheme to resolve the weakness. However, the authors of [61] discover that the new version does not resist MITM and replay attacks.

The previously discussed works were studied when designing the proposed authentication protocols to aim for schemes without the mentioned security vulnerabilities and the use of cryptographic operations suitable to the computational resources of IoT devices.

Chapter 3

A unified model of authentication protocols

The proposed unified model of authentication protocols is composed of two phases:

1. Registration.
2. Authentication.

The Registration phase executes when a device wants to join an IoT environment. The device registers on the Authentication Server, Network Manager, or Gateway responsible for the list of members in its environment. For simplicity, this entity is denoted as the server in the rest of this document. This phase consists of the device and the server exchanging secret values that the entities created using their keys and identities. The values are unique in the environment and are only known by them; thus, the device can use them to demonstrate its identity to the server and vice versa. The communication in this phase performs on a secure channel to keep the data exchanged a secret. The device executes this phase only once.

The Authentication phase executes every time two entities in the IoT environment want to communicate securely over an open channel. This phase consists of two entities that prove their identities to each other. If the proof is successful, the entities create the same session key, which they use to secure their communication. The entities use the secret values exchanged in the Registration phase to prove their identities without revealing any value.

The two phases are described below. The symbol $\mathscr{A}$ represents a polynomial-time-bounded adversary.

3.1 Registration phase

In this phase, the device and the server create the secret values b_0 and b_1, respectively. Additionally, the entities agree on a mechanism that they will use to generate secure keys.

This phase consists of the next steps. In Figure 3.1, the phase is illustrated.

1. The device creates b_0 using a secure hash function, identity, and a random number or key and sends b_0 to the server. The server is considered an honest-but-curious entity, which means it will execute the protocol properly but will try to learn and record all possible information from the received messages [62]. Therefore a one-way and collision-resistant hash function is needed to prevent the server from obtaining the device's credentials. Additionally, the random number is necessary to avoid tracing, replay, and impersonation attacks.

2. The server creates b_1 using its credentials. The value b_1 can be a hash of the server's identity and a random number or key, or it can be the encryption of the device's b_0 with the server's key.

3. The server selects a mechanism to generate secure keys. The choice is made based on the computing capabilities of the devices that compose the network. If the devices have limited computing resources, the server selects a mechanism that does not require many resources, such as a symmetric-key mechanism. Instead, if the devices do not have significant computing resource restrictions, the server can choose one that demands more resources, such as an asymmetric-key mechanism. Some examples are key pools for symmetric-key mechanisms and ECC public-key certificates for asymmetric-key.

4. The server sends to the device b_1 and either the chosen mechanism or the information necessary to generate it securely. The latter applies when the mechanism consists of public-key certificates. In this case, the server creates the device's certificate, but the device's private key has to be generated by the device to preclude the key-escrow problem [63].

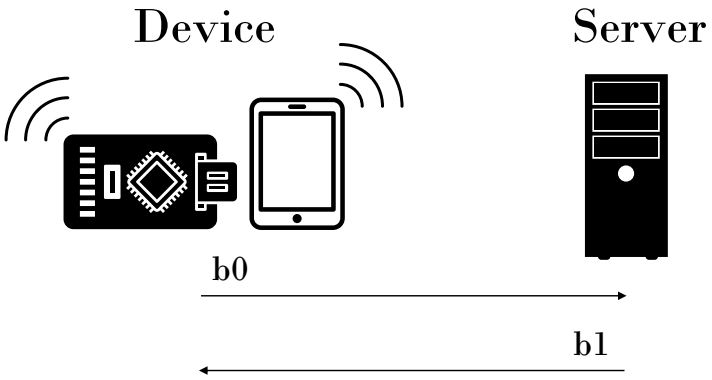


Figure 3.1 Registration phase of the unified authentication model.

3.2 Authentication phase

In this phase, two entities prove their identities to each other, exchange ephemeral secrets, and generate a session key that they will use to secure their communication. The authentication can be performed either between the device and the server, the device and an AP, or between two devices if they have public-key certificates.

The phase consists of the following steps. For simplicity, the entity that starts the authentication process is denoted as Entity a, and the other as Entity b. Figure 3.2 illustrates the authentication phase.

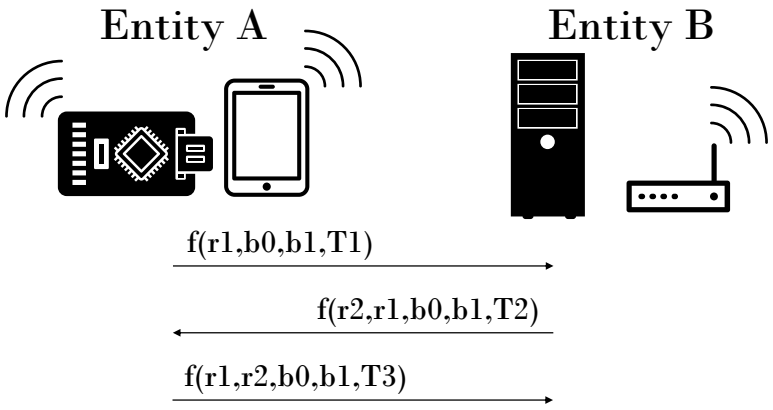


Figure 3.2 Authentication phase of the unified authentication model. The expression f() symbolizes a transformation applied to the values inside the parenthesis. The notations T1, T2, and T3 represent three different time-stamps.

1. Entity a creates a random number named r_1. This value is an ephemeral secret that will be used to generate the session key. Entity a ciphers r_1 to keep it a secret from $\mathscr{A}$. Entity a also hashes r_1 to prevent r_1 from being maliciously modified by $\mathscr{A}$ when transmitted. Entity a creates a message with the ciphered and the hashed r_1 and includes the secrets that the entities created in the Registration phase, i.e., either the shared secrets b_0 and b_1 or the private and public keys derived from them. The intention behind including the values is to certify that the legitimate Entity a originated the message. The values are added in a non-invertible manner to prevent Entity b or another entity from computing them and reusing them to impersonate Entity a in another communication. Entity a also adds a time-stamp to the message to prevent its reuse at a later time. Finally, Entity a sends the message to Entity b.

2. After Entity b receives the message, Entity b verifies the time-stamp against the allowed transmission delay. If the time-stamp has a longer delay, Entity b discards the message and terminates the communication because the message could be a replay from a previous session. Otherwise, Entity b uses

the values it created in the Registration phase to obtain r_1 or a representation of it. Further, Entity b verifies that r_1 or its representation has not been maliciously modified by $\mathscr{A}$, through computing the hash of the value and comparing the result with the hash received in the message. If they are different, Entity b aborts the communication as the received information is not trustworthy.

3. Entity b creates a random number named r_2, which will be used with r_1 or its representation to generate the session key. As in step 1, Entity b ciphers and hashes r_2 and adds the results to a new message. Entity b includes in a non-invertible manner the secrets it created in the Registration phase to certify the message's legitimacy. Entity b also adds r_1 or its representation in the message. Only the legitimate Entity b can obtain this value from the message that Entity a sent in step 1. Therefore, by adding the value to the message, Entity b proves to Entity a that it is the authentic party. Besides, Entity b can also add any more information necessary to generate the session key. For example, if the entities use a key-pool to create the session key, Entity b can select an index of a key and include it in the message. Finally, Entity b adds a time-stamp and sends the message to Entity a.

4. After Entity a receives the message, Entity a verifies the time-stamp as described in step 2. Then, Entity a obtains r_2 or its representation using the values Entity a created in the Registration phase. To verify the value's trustworthiness, Entity a computes the value's hash and compares the result with the hash received in the message. If they are different, Entity a terminates the communication. Additionally, Entity a verifies that the sender is the authentic Entity b by comparing the r_1 or its representation in the message against the value that Entity a created and shared in step 1. If they are equal, it means the sender is the legitimate Entity b. Otherwise, Entity a aborts the communication.

5. Entity a creates a new message that contains the hash of r_1 and r_2 or their representations. Any additional information shared by the entities to construct the session key, e.g., an index in the key-pool, is also included. In this message, Entity a confirms to Entity b that it correctly received the shared values. Therefore, Entity a believes in the identity of Entity b. Further, by sending r_2 or its representation, Entity a also wants to prove its identity to Entity b because only the legitimate Entity a can obtain it. Finally, Entity a includes a time-stamp and sends the message to Entity b.

6. After Entity b receives the message, Entity b verifies the time-stamp as described in step 2. Further, Entity b validates that the received information is correct. Entity b computes the hash of r_1 and r_2 or their representations and compares it with the hash received in the message. If they are different, Entity b aborts the communication. Otherwise, it means the sender is the authentic Entity a because only the legitimate party can obtain r_2 or its representation from the message that Entity b transmitted in step 3. At this moment, both entities believe in each other identity and have shared the information

required to create the session key. In some mechanisms for the session-key generation, it might be necessary that Entity b sends another message confirming some values. Entity a and Entity b can now create the session key and use it to communicate securely in an open channel.

3.3 Mechanisms to generate keys

Common initial set-ups used as mechanisms for the session key agreement are briefly discussed below [64].

- Private and public key pair: Each entity has a private and public key pair. The owner keeps the private key a secret and makes the public key available to the other entities. Certificates are used to verify the ownership of public keys by entities. Due to the mathematical relationship between the keys, only the private key can decrypt messages encrypted with the public key and vice versa.
- High-quality shared secrets: Two entities create and share values uniformly chosen from a high-entropy distribution. The entities share the values on a secure channel to keep them a secret. Therefore, they can use them to conduct an authentication session.
- Low-quality shared secrets: The mechanism is similar to the previously described; however, the shared values can come from small distributions with low entropy. An example is a password that comes from a small dictionary.

3.4 Classification of mechanisms based on their use of computing resources

The mechanisms previously presented have different computer system requirements. The work in [65] presents an evaluation of the resource usage that security algorithms have on the family of microcontrollers ARM Cortex-M family [66], which are devices frequently found in programmable controllers. For the evaluation, the authors chose standard algorithms implemented on a certified cryptographic library. Therefore, they can be sure that the algorithms' implementations are correct, and they provide the security level that the standards claim. The algorithms evaluated were the cipher AES 128 in counter mode [67], the hash function SHA-256 [68], and the Elliptic Curve Digital Signature Algorithm (ECDSA) [69]. The cryptographic library used was STMicroelectronics X-CUBE-CRYPTOLIB [70], which is certified by the NIST Cryptographic Algorithm Validation Program [71]. The resource usage was evaluated in terms of the execution time, power consumption, and memory resource occupation that each algorithm had when executed in the microcontrollers Cortex-M0, M3, M4, and M7. Table 3.1 presents the performance of the three algorithms

when executed on the Cortex-M0. The latter is a microcontroller optimized for low cost, area, and power applications.

Table 3.1 Resource usage of the algorithms AES 128, SHA-256, and ECDSA on the Cortex-M0 microcontroller

Algorithm	Execution time	Power consumption	Memory occupation	
			RAM	Flash
AES 128	486.470 μs ± 0.039 μs	072.462 mW ± 8.339 mW	1.67 kB	08.28 kB
SHA-256	397.000 μs ± 0.046 μs	057.538 mW ± 6.019 mW	1.61 kB	06.32 kB
ECDSA	016.600 s ± 0.000 s	132.308 mW ± 8.153 mW	1.57 kB	25.04 kB

Based on the resource usage reported in [65] and on the classification of embedded systems in the RFC 7228 [16] presented in Chapter 2, the mechanisms are classified concerning their requirements of computing resources. Two classes are created, which are low and high resource-usage mechanisms. According to Table 3.1, the symmetric-key algorithms (AES 128 and SHA-256) had a short running time, low power consumption, and a small memory footprint. In comparison, the asymmetric-key algorithm (ECDSA) had a significantly longer running time, higher power consumption, and a larger flash memory footprint. Therefore, the high-quality and low-quality shared secrets mechanisms, based on symmetric-key algorithms, belong to the first class. And the private and public key pair mechanism, based on asymmetric-key algorithms, belongs to the second. Table 3.2 shows the classification. Mechanisms' classes can be related to embedded systems' classes in the RFC 7228 to associate devices with their most suitable security mechanism concerning computer system requirements. Table 3.3 presents the correspondence between the classifications of mechanisms and embedded systems, which is also illustrated in Figure 3.3.

Table 3.2 Classification of mechanism to generate keys

Class	Mechanism
Low resource-usage mechanism	High-quality shared secrets
	Low-quality shared secrets
High resource-usage mechanism	Private and public key pair

Table 3.3 Relationship between devices and mechanisms classes

Embedded system class	Low resource-usage mechanism	High resource-usage mechanism
Computing-capability class	C0	C2
	C1	
Energy-limitation class	E0	E1
	E2	E9
Network-capability class	P0	P9
	P1	

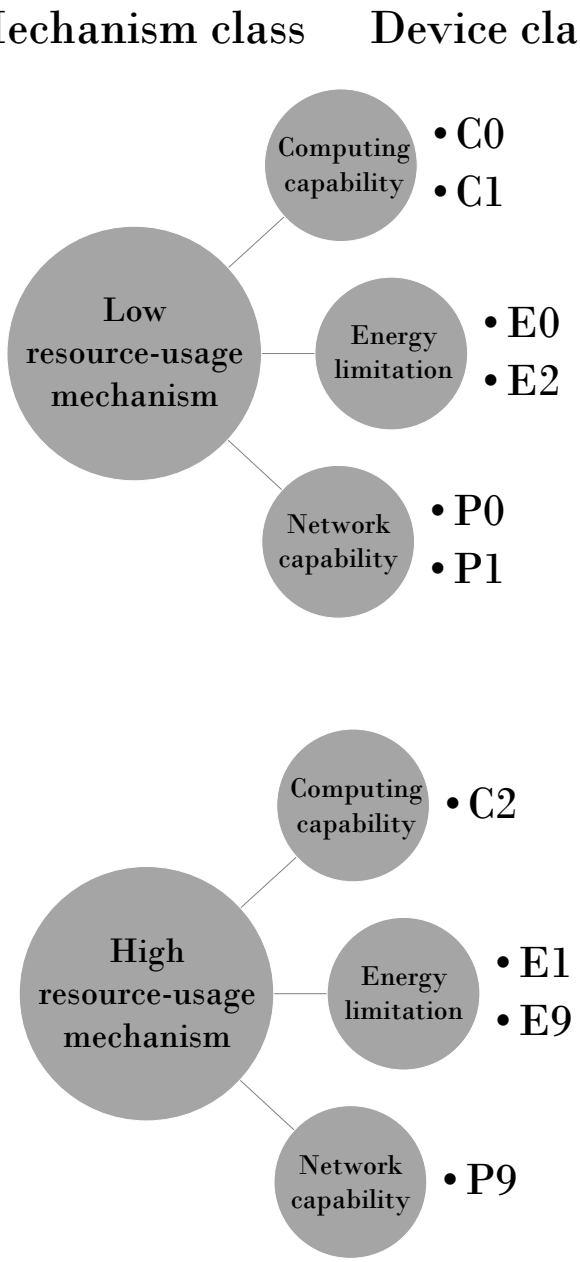


Figure 3.3 Association of mechanisms and devices classes.

As predictability in execution time and real-time operation are essential requirements to IoT devices, cryptographic functions must not delay the system in meeting its deadlines [65]. According to the relationship between classes of devices and mechanisms to generate the session-key showed in Table 3.3. It can be deduced that different authentication protocols are needed for each mechanism class to be appropriate to

the devices' computing capabilities and resource availabilities of the embedded system's classes. The unified authentication model can be used to produce authentication protocols that differ in the mechanism used. Consequently, the protocols vary in performance, making each of them suitable to an embedded system's class.

Currently, two authentication protocols have been generated from the unified authentication model. The first scheme uses only symmetric-key cryptographic operations; therefore, it belongs to the low resource-usage mechanism class. The second scheme is composed of ECC operations; thus, it classifies as a high resource-usage mechanism. The protocols were evaluated in terms of security and performance. The schemes' security was assessed through the well-known formal methods AVISPA tool and BAN logic. Besides, informal analyzes were made concerning the achievement of security properties and attack resistance. Furthermore, the protocols' performances were analyzed in terms of execution time and communication cost. The evaluations and analyzes showed that the schemes achieved more security features than similar protocols in the literature and have an appropriate performance for IoT networks.

In the next chapter, the two proposals are described.

Chapter 4

Authentication protocols generated from the unified model

The protocol classified as a low resource-usage mechanism is named Lightweight Authentication and Key Distribution (LAKD) [72]. It was published in the Sensors journal [73] in January 2020. LAKD is proposed for (M2M) communication in Industrial IoT (IIoT) [74, 75, 76]. The participants are a resource-constrained sensor node and a gateway. LAKD is based on the lightweight operations addition, subtraction, xor, and a hash function.

The scheme classified as a high resource-usage mechanism is named Two-Party Lightweight Authentication Protocol (TLAP) [77]. It was published in the IEEE Access journal [78] in May 2021. TLAP is proposed for WBANs in health-care applications [79, 80, 81]. The participants are a portable personal terminal, such as a smartphone or a personal digital assistant (PDA), and an application provider (AP). TLAP uses ECC self-certified public keys, ECC scalar point multiplication, symmetric key encryption, xor, and hash function.

The notations used in the protocols' descriptions are presented in Table 4.1. In the sections below, LAKD and TLAP protocols are introduced.

Symbol	**Description**
k_g	Gateway's secret key.
ID_g	Gateway's identity.
k_s	Sensor node's secret key.
ID_s	Sensor node's identity.
AID	Sensor node's pseudonym.
$b0$, $b1$	Secret values shared by the entities during the Registration phase.
KP	Sensor node's key-pool.
idx	An index in the key-pool.
p, q	Large prime numbers.

Symbol	Description
$E_p(a, b)$	An elliptic curve.
P	Group G's generator.
Client a, ID_a	A client and its identity, respectively.
AP b, ID_b	An AP and its identity, respectively.
Entity i, ID_i	An entity and its identity, respectively.
s, Pub_s	The private and public key pair of the Network Manager (NM).
i, Pub_i	The private and public key pair of Entity i.
ΔT	Maximum admissible delay for message reception.
H	Secure hash function.
$E_k()$, $D_k()$	Symmetric-key encryption and decryption using k as the key, respectively.
$\|\|$	Data concatenation operator.
$\oplus$	Xor function. Given the points P and Q on $E_p(a, b)$, the xor of their x and y coordinates is represented as $P \oplus Q$ [82]. Given $x, y \in Z_q^*$, the xor in Z_q^* of the values is symbolized as $x \oplus y$.

Table 4.1 Protocols' notations.

4.1 LAKD protocol

IIoT is comprised of sensors, networks, and services to connect and remotely manage production systems. The benefits of IIoT are the monitoring of supply chains, detection of machine failures, real-time supervising of production systems, inventory tracking, tracking of health and safety indicators to prevent accidents and injuries to the workers, etc. However, integrating IoT into industries brings new security problems as sensitive and critical business information is being remotely shared and analyzed. Some of the issues include exposure of employees' and customers' private data, industrial espionage, and machine infiltration that could disrupt the production capabilities and physically endanger workers [83, 84, 85].

Authentication protocols for IIoT have been proposed to address security problems. However, some proposals use many computing resources in the authentication; therefore, they are not viable on legacy

resource-constrained devices in industries. One of the challenges of IIoT is integrating recent technology into legacy devices. Machines on production floors have a lifespan of several decades because it is costly to replace all of the legacy equipment every time a new technology is developed. Consequently, authentication protocols should also be designed for legacy devices, which are usually resource-constrained [86, 87, 88].

LAKD is a lightweight protocol that uses only symmetric-key algorithms in authentication. It intends for the communication between a resource-constrained sensor node and a gateway. The former gathers and sends contextual data to the gateway. The latter stores the data in buffers and transmits them to control nodes, which compile data from several gateways and send them to cloud services. LAKD follows the IIoT network architecture proposed in [89, 90], comprised of sense domains, networks with RESTful services, and cloud servers. Sensor nodes, gateways, and control nodes are part of the sense domain. The advantage of having a three-layered hierarchy in the sense domain is that the workload and energy used by the massive quantity of sensor nodes are distributed, which increases the IIoT system lifetime and energy efficiency to help to accomplish a green IIoT [89]. Figure 4.1 illustrates the network architecture.

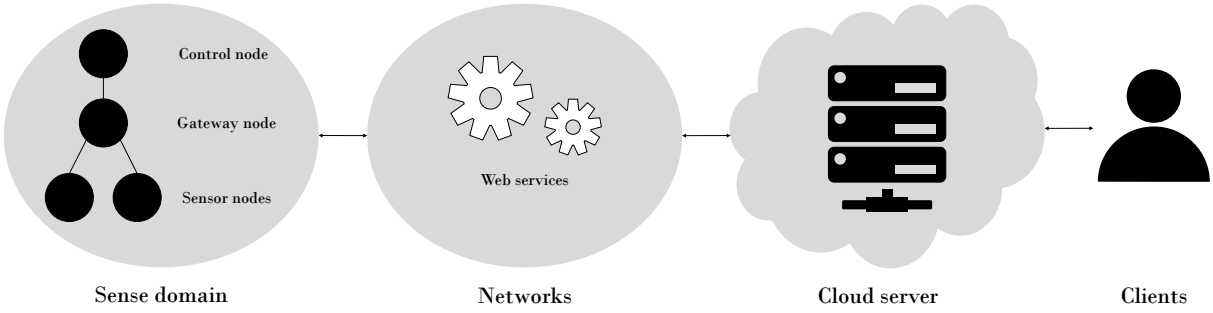


Figure 4.1 Energy-efficient IIoT network architecture used in LAKD protocol.

There are security requirements that LAKD must meet to be suitable for IIoT. They include the achievement of perfect forward secrecy and resistance to the following attacks: known session-specific temporary information, DoS, replay, impersonation, de-synchronization, identity compromise impersonation, man-in-the-middle, un-linkability, key-offset, stolen device, and known key [91]. The LAKD's initial assumptions and the threat model with the possible security threats and vulnerabilities the entities can encounter during the LAKD execution are present below.

- The channel used in the Registration phase is private and secure.
- The hash function has the collision-resistant property.
- Sensor node and gateway devices have tampering protection.
- $\mathscr{A}$ is unable to guess keys and random numbers in polynomial time.

- $\mathscr{A}$ can replay messages intercepted from previous protocol executions in the current session.
- $\mathscr{A}$ can tamper with captured messages.
- $\mathscr{A}$ can link a sensor node behind different protocol executions.
- $\mathscr{A}$ can impersonate a legitimate sensor node or gateway.
- $\mathscr{A}$ can send fraudulent messages.

LAKD protocol comprises two phases. Registration phase in which sensor node and gateway share the secrets b_0, b_1, and KP. And Authentication phase, where the entities use the shared secrets to prove their identities and create a session key. The two phases are described below.

4.1.1 Registration phase

Sensor node and gateway perform the following steps for the registration.

1. Sensor node chooses its first pseudonym AID, selects a random number r_0, and uses r_0 to compute $b0 = H(ID_s||k_s||r_0)$. Finally, sensor node sends to gateway AID, r_0, and $b0$.
2. Gateway selects a random number r_1 and uses it to compute $b1 = H(ID_g||k_g||r_1)$. Further, the gateway selects a key pool KP. The number of keys in KP is dependent on the available computing resources in the sensor node. Three keys can be appropriate if the device has limited resources. The gateway sends $b1$ and KP to the sensor node.
3. The two entities store securely $\{b0, b1, KP\}$.

Figure 4.2 illustrates the messages the entities exchange in the registration phase.

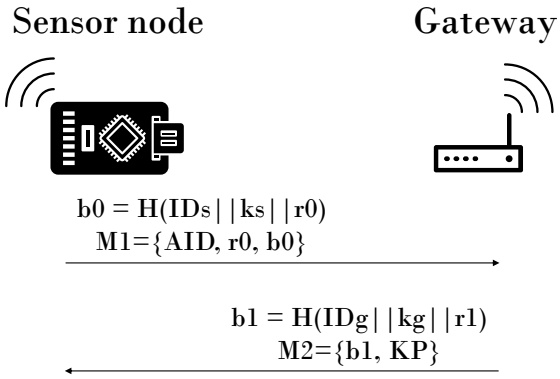


Figure 4.2 LAKD's registration phase.

4.1.2 Authentication phase

Sensor node and gateway perform the following steps to authenticate mutually and generate the session key SK.

1. The sensor node selects the random number and the time-stamp r_1 and T_1, respectively, computes $D_1 = H(AID||b1||T_1) \oplus r_1$ and $D_2 = H(r_1||T_1||b0)$, and sends $M_1 = \{T_1, AID, D_1, D_2\}$ to the gateway.

2. Gateway selects the time-stamp T_2 and verifies $|T_2 - T_1| \leq \Delta T$. If true, the gateway computes $r_1 = D_1 \oplus H(AID||b1||T_1)$ and verifies $D_2 \stackrel{?}{=} H(r_1||T_1||b0)$. If any verification is false, the gateway terminates the communication. Otherwise, it selects the random number r_2 and idx, which is a valid index in KP. Gateway computes $D_3 = H(r_1||T_2) \oplus r_2$, $D_4 = (idx + r_1 + r_2) \oplus H(b1||r_2)$, and $D_5 = H(idx||r_2||b0||r_1)$. Finally, the gateway sends $M_2 = \{T_2, D_3, D_4, D_5\}$ to the sensor node.

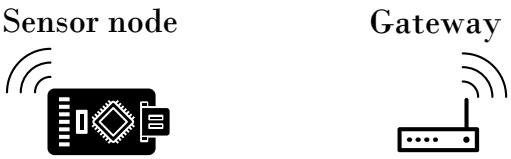
3. The sensor node selects the time-stamp T_3 and verifies $|T_3 - T_2| \leq \Delta T$. If true, the sensor node computes $r_2 = D_3 \oplus H(r_1||T_2)$ and $idx = D_4 \oplus H(b1||r_2) - r_1 - r_2$. Then, the sensor node verifies $D_5 \stackrel{?}{=} H(idx||r_2||b0||r_1)$. If any verification is false, the sensor node terminates the communication. Otherwise, it obtains $KP(idx)$, which is the key stored in KP at index idx and uses it to compute $D_6 = H(b1||r_1||T_3||r_2||KP(idx))$. Finally, the sensor node sends $M_3 = \{T_3, D_6\}$ to the gateway.

4. Gateway selects the time-stamp T_4 and verifies $|T_4 - T_3| \leq \Delta T$. If it holds, the gateway verifies $D_6 \stackrel{?}{=} H(b1||r_1||T_3||r_2||KP(idx))$. If any verification is false, the gateway terminates the communication. Otherwise, it computes $D_7 = H(b0||r_1||T_4||r_2||KP(idx))$ and $SK = H(r_1||r_2||AID||KP(idx))$. Finally, the gateway sends $M_4 = \{T_4, D_7\}$ to the sensor node.

5. The sensor node selects the time-stamp T_5 and verifies $|T_5 - T_4| \leq \Delta T$. If it holds, the sensor node verifies $D_7 \stackrel{?}{=} H(b0||r_1||T_4||r_2||KP(idx))$. If any verification is false, the sensor node terminates the communication. Otherwise, the sensor node computes $SK = H(r_1||r_2||AID||KP(idx))$.

6. Sensor node and Gateway compute the new sensor node's pseudonym $AID = H(AID||b0||r_1||r_2)$.

Figure 4.3 illustrates the messages the entities exchange in the authentication phase.



Select: r1 and T1
Compute:
$D1 = H(AID||b1||T1) \oplus r1$
$D2 = H(r1||T1||b0)$
$M1 = \{T1, AID, D1, D2\}$

Select: T2
Verify: $|T2 - T1| \leq \Delta T$
Compute: $r1 = D1 \oplus H(AID||b1||T1)$
Verify: $D2 ?= H(r1||T1||b0)$
Select: r2 and idx
Compute:
$D3 = H(r1||T2) \oplus r2$
$D4 = (idx+r1+r2) \oplus H(b1||r2)$
$D5 = H(idx||r2||b0||r1)$
$M2 = \{T2, D3, D4, D5\}$

Select: T3
Verify: $|T3 - T2| \leq \Delta T$
Compute:
$r2 = D3 \oplus H(r1||T2)$
$idx = D4 \oplus H(b1||r2) - r1 - r2$
Verify: $D5 ?= H(idx||r2||b0||r1)$
Compute: $D6 = H(b1||r1||T3||r2||KP(idx))$
$M3 = \{T3, D6\}$

Select: T4
Verify:
$|T4 - T3| \leq \Delta T$
$D6 ?= H(b1||r1||T3||r2||KP(idx))$
Compute:
$D7 = H(b0||r1||T4||r2||KP(idx))$
$SK = H(r1||r2||AID||KP(idx))$
$M4 = \{T4, D7\}$

Select: T5
Verify:
$|T5 - T4| \leq \Delta T$
$D7 ?= H(b0||r1||T4||r2||KP(idx))$
Compute: $SK = H(r1||r2||AID||KP(idx))$

Figure 4.3 LAKD's authentication phase.

4.1.3 Security analysis

The security of LAKD was analyzed using well-known formal and informal methods. The first consisted of the AVISPA tool and BAN logic. The second of examining the security features achieved by LAKD and its resistance to potential attacks.

The informal analysis comprises the study of how LAKD achieves the security attributes of mutual authentication, integrity, confidentiality, sensor node anonymity, known session key security, and perfect forward and backward secrecy. Further, the analysis of LAKD resistance to potential attacks in IIoT. They include DoS, impersonation, injection, and traceability. Key disclosure, de-synchronization, off-line identity guessing, known session-specific temporary information, privileged insider, MITM, and replay attacks. The informal analysis is not included in this document to keep this subsection concise. The reader can refer to the study details in the article [72].

The analysis results of the AVISPA tool and BAN logic are described below.

AVISPA tool

The LAKD's modeling in the HLPSL language is present in Appendix A. The security goals stipulated were secrecy of SK and mutual authentication. If any goal is unachieved, the AVISPA tool concludes that the protocol is unsafe. The achievement of the mutual authentication goal implicates that the agents believe correctly that the intended participant is present in the current session, reached some state, and agrees on a single-use value. The LAKD's modeling was evaluated through the CL-AtSe and OFMC back-ends due to their support of the xor function [92]. The two back-ends concluded that LAKD resists replay and MITM attacks. Further, LAKD achieved the secrecy of SK and the mutual authentication of the principals. Figure 4.4 shows the evaluation results.

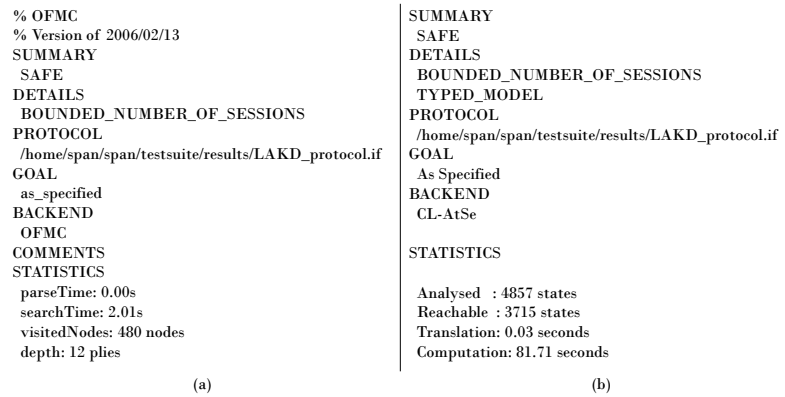


Figure 4.4 LAKD's analysis results through the AVISPA tool: (a) Using the OFMC back-end. (b) Using the CL-AtSe back-end.

BAN logic

LAKD has to meet the following security goals in BAN logic to demonstrate it achieves mutual authentication. The BAN logic proof uses the notations E_g and E_s to represent the gateway and the sensor node, respectively.

Goal 1: $E_g \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

Goal 2: $E_s \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

Goal 3: $E_g \mid\equiv E_s \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

Goal 4: $E_s \mid\equiv E_g \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

The LAKD's idealized form is as follows:

Message 1: $\{r_1, T_1\}_{b1}, \left\{\langle E_g \overset{r_1}{\rightleftharpoons} E_s\rangle_{b0}, T1\right\}_{b0}$.

Message 2: $\{r_2, T_2\}_{r_1}, \left\{\langle E_g \overset{r_2}{\rightleftharpoons} E_s\rangle_{r_1}\right\}_{b1}, \left\{\langle E_g \overset{KP(idx)}{\rightleftharpoons} E_s\rangle_{r_1}, r_2\right\}_{b0}$.

Message 3: $\left\{E_g \overset{SK}{\longleftrightarrow} E_s, T_3\right\}_{b1}$.

Message 4: $\left\{E_g \overset{SK}{\longleftrightarrow} E_s, T_4\right\}_{b0}$.

The LAKD's assumptions about its initial state are presented below:

1. $E_g \mid\equiv E_s \Rightarrow (r_1, T_1, T_3)$.
2. $E_g \mid\equiv \sharp(T_1, T_3)$.
3. $E_g \mid\equiv E_g \overset{KP(idx)}{\longleftrightarrow} E_s$.
4. $E_g \mid\equiv E_g \overset{b0}{\longleftrightarrow} E_s$.
5. $E_g \mid\equiv E_g \overset{b1}{\longleftrightarrow} E_s$.
6. $E_s \mid\equiv E_g \Rightarrow (r_2, T_2, T_4)$.
7. $E_s \mid\equiv \sharp(T_2, T_4)$.
8. $E_s \mid\equiv E_g \overset{KP(idx)}{\longleftrightarrow} E_s$.
9. $E_s \mid\equiv E_g \overset{b0}{\longleftrightarrow} E_s$.
10. $E_s \mid\equiv E_g \overset{b1}{\longleftrightarrow} E_s$.

The LAKD's BAN logic proof is described following. It demonstrates that LAKD accomplishes the four mutual authentication goals presented previously.

From Message 1 of LAKD's idealized form, we obtain:

Step 1: $E_g \triangleleft \{r_1, T_1\}_{b1}, \left\{\langle E_g \overset{r_1}{\rightleftharpoons} E_s\rangle_{b0}, T1\right\}_{b0}$.

Step 2: From Step 1, using Rule 1 and Assumptions 4 and 5, we obtain: $E_g \mid\equiv E_s \mid\sim (r_1, T_1, \langle E_g \overset{r_1}{\rightleftharpoons} E_s\rangle_{b0}, T_1)$.

Step 3: From Step 2, using Rule 4 and Assumption 2, we obtain: $E_g \mid\equiv \sharp(r_1, T_1)$.

Step 4: From Step 3, using Rule 2, we obtain: $E_g \mid\equiv E_s \mid\equiv (r_1, T_1)$.

Step 5: From Step 4, using Rule 3 and 5 and Assumption 1, we obtain: $E_g \mid\equiv r_1$.

Step 6: From Step 2, using Rule 4 and Assumption 2, we obtain: $E_g \mid\equiv \sharp(\langle E_g \overset{r_1}{\rightleftharpoons} E_s\rangle_{b0}, T_1)$.

Step 7: From Step 6, using Rule 2, we obtain: $E_g \mid\equiv E_s \mid\equiv (E_g \overset{r_1}{\rightleftharpoons} E_s, T_1)$.

Step 8: From Step 7, using Rule 3 and 5 and Assumption 1, we obtain: $E_g \mid\equiv E_g \overset{r_1}{\rightleftharpoons} E_s$.

From Message 2, we obtain:

Step 9: $E_s \triangleleft \{r_2, T_2\}_{r_1}, \left\{\langle E_g \overset{r_2}{\rightleftharpoons} E_s\rangle_{r_1}\right\}_{b1}, \left\{\langle E_g \overset{KP(idx)}{\rightleftharpoons} E_s\rangle_{r_1}, r_2\right\}_{b0}$.

Step 10: From Step 9, using Rule 1 and Assumptions 9 and 10, we obtain:
$E_s \mid\equiv E_g \mid\sim (r_2, T_2, \langle E_g \overset{r_2}{\rightleftharpoons} E_s\rangle_{r_1}, \langle E_g \overset{KP(idx)}{\rightleftharpoons} E_s\rangle_{r_1}, r_2)$.

Step 11: From Step 10, using Rule 4 and Assumption 7, we obtain: $E_s \mid\equiv \sharp(r_2, T_2)$.

Step 12: From Step 11, using Rule 2, we obtain: $E_s \mid\equiv E_g \mid\equiv (r_2, T_2)$.

Step 13: From Step 12, using Rule 3 and 5 and Assumption 6, we obtain: $E_s \mid\equiv r_2$.

Step 14: From Step 10, using Rule 4 and the assumption $E_s \mid\equiv \sharp r_1$ because it is its originator, we obtain: $E_s \mid\equiv \sharp\langle E_g \overset{r_2}{\rightleftharpoons} E_s\rangle_{r_1}$.

Step 15: From Step 14, using Rule 2, we obtain: $E_s \mid\equiv E_g \mid\equiv E_g \overset{r_2}{\rightleftharpoons} E_s$.

Step 16: From Step 15, using Rule 3 and Assumption 6, we obtain: $E_s \mid\equiv E_g \overset{r_2}{\rightleftharpoons} E_s$.

Step 17: From Step 10, using Rule 4 and the assumption $E_s \mid\equiv \sharp r_1$, we obtain: $E_s \mid\equiv \sharp(\langle E_g \overset{KP(idx)}{\rightleftharpoons} E_s\rangle_{r_1}, r_2)$.

Step 18: From Step 17, using Rule 2, we obtain: $E_s \mid\equiv E_g \mid\equiv \langle E_g \overset{KP(idx)}{\rightleftharpoons} E_s\rangle_{r_1}$.

From Message 3, we obtain:

Step 19: $E_g \triangleleft \left\{E_g \overset{SK}{\longleftrightarrow} E_s, T_3\right\}_{b1}$.

Step 20: From Step 19, using Rule 1 and Assumption 5, we obtain: $E_g \mid\equiv E_s \mid\sim (E_g \overset{SK}{\longleftrightarrow} E_s, T_3)$.

Step 21: From Step 20, using Rule 4 and Assumption 2, we obtain: $E_g \mid\equiv \sharp(E_g \overset{SK}{\longleftrightarrow} E_s, T_3)$.

Step 22: From Step 21, using Rule 2, we obtain: $E_g \mid\equiv E_s \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s, T_3)$.

Step 23: From Step 22, breaking the conjunction, we obtain: $E_g \mid\equiv E_s \mid\equiv E_g \overset{SK}{\longleftrightarrow} E_s$. **(Goal 3)**

Step 24: From Step 23, using Rule 3, Assumptions 1 and 3, and the deductions of steps 5 and 8, we obtain: $E_g \mid\equiv E_g \overset{SK}{\longleftrightarrow} E_s$. **(Goal 1)**

From Message 4, we obtain:

Step 25: $E_s \triangleleft \left\{ E_g \overset{SK}{\longleftrightarrow} E_s, T_4 \right\}_{b0}$.

Step 26: From Step 25, using Rule 1 and Assumption 9, we obtain: $E_s \mid\equiv E_g \mid\sim (E_g \overset{SK}{\longleftrightarrow} E_s, T_4)$.

Step 27: From Step 26, using Rule 4 and Assumption 7, we obtain: $E_s \mid\equiv \sharp(E_g \overset{SK}{\longleftrightarrow} E_s, T_4)$.

Step 28: From Step 27, using Rule 2, we obtain: $E_s \mid\equiv E_g \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s, T_4)$.

Step 29: From Step 28, breaking the conjunction, we obtain: $E_s \mid\equiv E_g \mid\equiv E_g \overset{SK}{\longleftrightarrow} E_s$. **(Goal 4)**

Step 30: From Step 29, using Rule 3, Assumptions 6 and 8, and the deductions of steps 13 and 16, we obtain: $E_s \mid\equiv E_g \overset{SK}{\longleftrightarrow} E_s$. **(Goal 2)**

4.1.4 Performance analysis

The methodology presented in [93] was followed to calculate the LAKD's performance. It was evaluated the time needed to complete a protocol execution and the number of bits that are transmitted. The results were compared with five published protocols with a similar architecture to LAKD and aimed at M2M communications. The protocols are the following, Esfahani *et al.* [94], Han *et al.* [95], Qiu *et al.* [96], Renuka *et al.* [97], and Joshitta *et al.* [98]. The protocols' execution times analyzes are presented first, then the analyzes of the protocols' communication costs.

When the protocols' execution times were evaluated, it was calculated the necessary time to execute the operations included in the authentication phases. The functions in the registration phases were not considered because the principals run the phase once only. Two metrics were used in the evaluation, the first contains the operations' execution times presented by Zhou *et al.* [99], and the second of Whu *et al.* [38]. The first metric is denoted as Case 1 and the second Case 2. Table 4.2 shows the metrics. The execution times are in milliseconds (ms). The symbol T_h denotes the execution time of the hash function and T_e of the AES algorithm. The execution times of the addition, xor, concatenation, and subtraction operations are not included in the analysis because they are negligible compared to the other functions.

Table 4.2 Metrics used for the evaluation of LAKD's execution time.

	Operation	**Execution time**
Case 1	T_h	0.0051700 ms
	T_e	0.0214800 ms
Case 2	T_h	0.0000328 ms
	T_e	0.0214385 ms

Table 4.3 shows the operations that comprise the protocols' authentication phases and their execution times, separated for the entity that executes them. The table includes the two metrics. About Case 1

results, LAKD's execution time is lower than the other schemes, except for the protocols of Esfahani *et al.* and Joshitta *et al.* The difference between LAKD and the Esfahani *et al.*'s protocol is that the sensor node in LAKD executes an additional hash function that causes an increment of 12.5% of execution time. The difference with Joshitta *et al.* is that their protocol mainly uses encryption. The medical device of Joshitta *et al.*'s protocol has 48.07% less execution time than LAKD's sensor node, and the Joshitta *et al.*'s authentication server has 23.07% less execution time than LAKD's gateway. Concerning Case 2 results, only Esfahani *et al.*'s protocol has less execution time than LAKD, with a difference of 12.5% due to the additional hash function on LAKD's sensor node.

Table 4.3 Results of the evaluation of LAKD's execution time.

Protocol proposed by	**Entity**	**Operations**	**Case 1**	**Case 2**
Esfahani *et al.* [94]	Sensor node	$7T_h$	0.03619 ms	0.0002296 ms
	Router	$8T_h$	0.04136 ms	0.0002624 ms
Han *et al.* [95]	Device 1	$3T_e$	0.06444 ms	0.0643155 ms
	Device 2	$3T_e$	0.06444 ms	0.0643155 ms
Qiu *et al.* [96]	Host	$5T_h + 4T_e$	0.11177 ms	0.0859180 ms
	Router	$T_h + 2T_e$	0.04813 ms	0.0429098 ms
	Edge router	$6T_h + 6T_e$	0.15990 ms	0.1288278 ms
Renuka *et al.* [97]	Sensor C	$4T_e$	0.08592 ms	0.0857540 ms
	Sensor D	$3T_e$	0.06444 ms	0.0643155 ms
	Gateway	$3T_e$	0.06444 ms	0.0643155 ms
Joshitta *et al.* [98]	Medical device	T_e	0.02148 ms	0.0214385 ms
	Authentication server	$2T_h + T_e$	0.03182 ms	0.0215041 ms
Us (LAKD)	Sensor node	$8T_h$	0.04136 ms	0.0002624 ms
	Gateway	$8T_h$	0.04136 ms	0.0002624 ms

LAKD was designed to achieve high security and low resource usage, so it is viable for legacy IIoT devices with limited computing resources. As shown in Table 4.3, LAKD achieved an appropriate execution time when it is compared with similar protocols. Some schemes have a lower execution time. Nonetheless, LAKD accomplished more security features than them. Furthermore, two protocols have significant security problems, as is detailed in the subsection Security features comparison.

The protocols' communication costs were evaluated by adding the number of bits that the entities transmit during the authentication phases, similar to the works in [100, 101]. The data transmitted in the registration phases were not added because the principals run the phase once only, and the communication operates in a private channel. The sizes of the cryptographic data sent in the protocols were obtained and added to calculate the communication cost. Two metrics were used in the evaluation, they are named Case 1 and Case 2. The first metric considers that the size of hash function outputs, keys, timestamps, random numbers,

and identities are 128 bits. The second uses data sizes of 256 bits. Both metrics consider that encryption outputs are 128 bits per block [93]. The metrics follow the conventional data size in security.

In the proposal of Esfahani *et al.*, the data transmitted between sensor node and router are M_1, M_2, f_{3i}, AID_i, M_1', M_2', AID_j, and M_1''. M_1, M_1', M_1'', and f_{3i} are random numbers, M_2 and M_2' are hash function outputs, and AID_i and AID_j are pseudonyms. Therefore, the communication cost of Esfahani *et al.*'s protocol is 1024 bits for the Case 1, and 2048 for the Case 2.

In the scheme of Han *et al.*, the data transmitted between the devices are *Device* 1_ID, $r1$, *Device* 2_ID, $E_{PSK}(r2 \oplus r1)$, $r2$, and $E_{SK}(r2||r1)$. *Device* 1_ID and *Device* 2_ID are identities, $r1$ and $r2$ are random numbers, and $E_{PSK}(r2 \oplus r1)$ and $E_{SK}(r2||r1)$ are encryption outputs. Therefore, the communication cost of Han *et al.*'s protocol is 896 bits for the Case 1, and 1792 for the Case 2.

In the protocol of Qiu *et al.*, the data transmitted between the host, router, and edge router are $Hello$, ID_{6LH}, $(ID_{6LH}, N_{6LH1})K_{6LH\_6LER}$, $MAC1$, $Hello$, ID_{6LR}, $(ID_{6LR}$, N_{6LR}, ID_{6LH}, (ID_{6LH}, N_{6LH1}) K_{6LH_6LER}, $MAC1)K_{6LR\_6LER}$, $MAC2$, *Response*, $(ID_{6LR}$, N_{6LH1}, $N_{6LER1})K_{6LH\_6LER}$, $MAC3$, ID_{6LH}, $(ID_{6LH}$, ID_{6LR}, N_{6LER1}, $N_{6LH2})K_{6LH\_6LER}$, $MAC4$, $(ID'_{6LH}$, N_{6LER2}, $(ID'_{6LH}$, N_{6LH2}, $N_{6LER2})$ $K_{6LH\_6LER})K_{6LR\_6LER}$, $MAC5$, $(ID'_{6LH}$, N_{6LH2}, $N_{6LER2})K_{6LH\_6LER}$, and $MAC5$. The values that start with ID are identities, N are random numbers, and MAC are hash function outputs. The notation $()K$ denotes an encryption with the key K. Therefore, the communication cost of Qiu *et al.*'s protocol is 5376 bits for the Case 1, and 10752 for the Case 2.

In the proposal of Renuka *et al.*, the data transmitted between the sensors and the gateway are ID_{sd}, r_s, ID_{Sc}, ID_{GW}, $E_{k(Sc)GW}(ID_{Sc}$, ID_{Sd}, ID_{GW}, r_{Sc}, $r_{Sd})$, ID_{GW}, ID_{Sc}, $E_{k(Sd)GW}(ID_{Sc}$, r_{Sd}, $k_{Sc}^{(Sd)})$, $E_{k(Sc)GW}(ID_{Sd}$, r_{Sc}, $k_{Sc}^{(Sd)}$, $K)$, ID_{Sc}, ID_{Sd}, $E_{k(Sd)GW}(ID_{Sc}$, r_{Sd}, $k_{Sc}^{(Sd)})$, $E_{k(Sd)Sc}(r_{Sd}$, $r_{Sc})$, ID_{Sd}, ID_{Sc}, and $E_{k(Sd)Sc}(r_{Sc})$. The values that start with ID are identities, r are random numbers, and K are keys. The notation $E_K()$ denotes an encryption with the key K. Therefore, the communication cost of Renuka *et al.*'s protocol is 3584 bits for the Case 1, and 7168 for the Case 2.

In the scheme of Joshitta *et al.*, the data transmitted between the medical device and the authentication server are Y_i, $MEPC_i$, M_i^1, IP_i, $E_{AS}[SK_i]$, and T_1. Y_i is a hash function output, $MEPC_i$ and M_i^1 are random numbers, IP_i is an IP address, $E_{AS}[SK_i]$ denotes encryption, and T_1 is a time-stamp. Therefore, the communication cost of Joshitta *et al.*'s protocol is 768 bits for the Case 1, and 1408 for the Case 2.

In LAKD, the data transmitted between the sensor node and the gateway are T_1, AID, D_1, D_2, T_2, D_3, D_4, D_5, T_3, D_6, T_4, and D_7. T_1, T_2, T_3, and T_4 are time-stamps, AID is a pseudonym, D_1, D_3, and D_4 are random numbers, and D_2, D_5, D_6, and D_7 are hash function outputs. Therefore, the communication cost of LAKD protocol is 1536 bits for the Case 1, and 3072 for the Case 2.

Figure 4.5 shows a comparison of the communication cost of the protocols. The schemes proposed by Esfahani *et al.*, Han *et al.*, and Joshitta *et al.* have a communication cost lower than LAKD. The

protocol of Esfahani *et al.* transmits 33.33% fewer bits, the scheme of Han *et al.* sends 41.67% fewer bits, and the protocol of Joshitta *et al.* transmits 50% fewer bits for Case 1 and 54.17% fewer bits for Case 2. LAKD sends only four messages to achieve mutual authentication. The messages include timestamps, message authentication codes, and random numbers. By adding these data, the protocols can avoid attacks, including replay, impersonation, modification, and DoS [41]. The protocols with a lower communication cost than LAKD do not include this information in their messages. Therefore, they do not resist some attacks. The details are in the subsection Security features comparison. The LAKD's communication cost is low compared to Qiu *et al.*'s and Renuka *et al.*'s cost, and it is a reasonable cost for IIoT compared to Esfahani *et al.*'s, Han *et al.*'s, and Joshitta *et al.*'s cost. LAKD's is not the protocol with the lowest communication cost. However, it does achieve more security features than the other protocols. In the design of LAKD, security was prioritized over the message sizes.

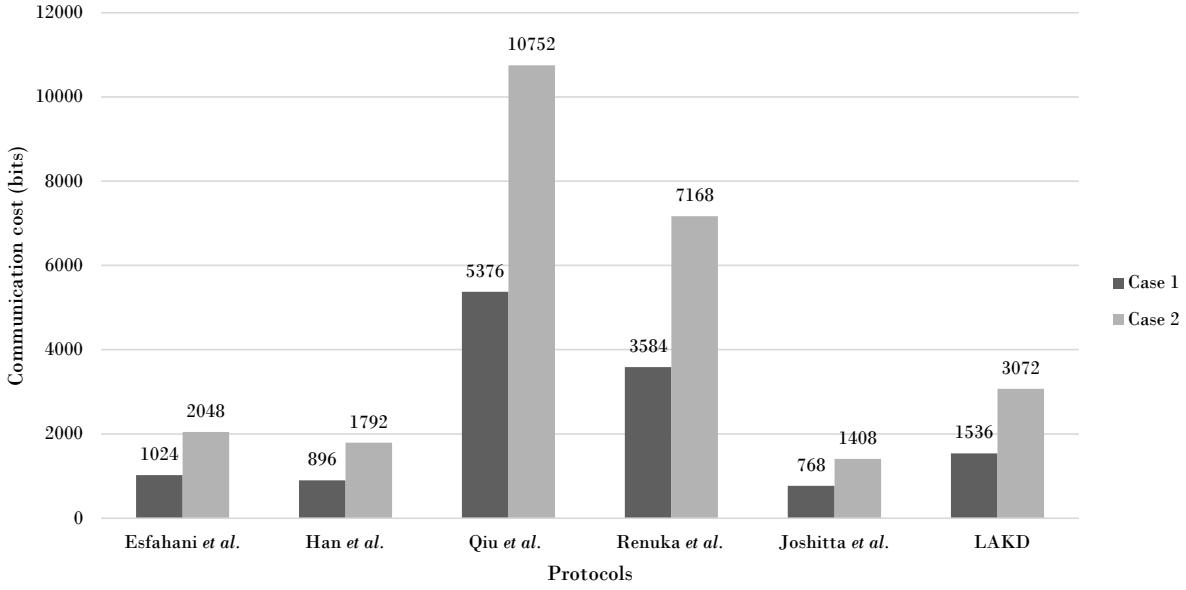


Figure 4.5 Comparison of the communication costs of LAKD and similar protocols.

4.1.5 Security features comparison

The security properties of LAKD, and the protocols of Esfahani *et al.*, Han *et al.*, Qiu *et al.*, Renuka *et al.*, and Joshitta *et al.* were analyzed. The works in [41, 102, 103] were reviewed during the analysis. The schemes were examined concerning the resistance to potential attacks in IIoT. Including traceability, off-line identity guessing, and impersonation. MITM, privileged insider, replay, known session-specific temporary information, DoS, modification, and key disclosure. The procedures for the security analysis demonstrated

in various articles were followed. Table 4.4 shows the analyzed attacks and the works on which the authors based on when performing the investigation.

Table 4.4 Works with procedures for analyzing attack resistance in authentication protocols.

Attack	**Work**
DoS	Aghili *et al.* [41] Malekzadeh *et al.* [104]
Impersonation	Aghili *et al.* [41] Safkhani *et al.* [105]
Traceability	Aghili *et al.* [41] Song *et al.* [106] Phan *et al.* [107]
Key disclosure	Aghili *et al.* [41]
Off-line identity guessing	Jiang *et al.* [108] Khan *et al.* [109]
Known session-specific temporary information	Meng *et al.* [110] Sarvabhatla *et al.* [111]
Modification	Kuo *et al.* [112]
Privileged insider	Khan *et al.* [109]
MITM	Guha *et al.* [113]

Table 4.5 shows the analysis results of the protocols' attack resistance. Schemes with lower execution times and communication costs than LAKD have security vulnerabilities that LAKD does not have. E.g., Esfahani et al.'s proposal does not resist the attacks key-disclosure, impersonation, modification, privileged insider, tracing, DoS [41, 102, 103], among others. The security problems in Joshitta et al.'s protocol are using the same session key in different authentication sessions and the disclosure of the session key.

Table 4.5 Comparative of the resistance to attacks for LAKD and related schemes.

Attack	Esfahani et al.	Han et al.	Qiu et al.	Renuka et al.	Joshitta et al.	LAKD
SA1	x	x	x	x	x	✓
SA2	x	✓	✓	✓	x	✓
SA3	x	x	✓	x	x	✓
SA4	x	✓	✓	✓	x	✓
SA5	x	x	✓	x	x	✓
SA6	x	✓	✓	✓	x	✓
SA7	x	x	✓	✓	x	✓
SA8	x	✓	✓	x	x	✓
SA9	✓	✓	✓	✓	x	✓
SA10	✓	✓	✓	✓	x	✓

✓: The scheme is resistant to the attack. x: The protocol is not resistant to the attack.
SA1: DoS; SA2: Impersonation; SA3: Traceability; SA4: Key disclosure; SA5: Off-line identity guessing; SA6: Known session-specific temporary information; SA7: Modification; SA8: Privileged insider; SA9: MITM; SA10: Replay.

4.1.6 Discussion and conclusion

This section presented the LAKD protocol designed for IIoT devices. LAKD intends for resource-constrained devices, i.e., Class 0 devices in the RFC 7228 classification. Thus, the protocol does not use cryptographic operations that require many computing resources, such as public-key cryptography [44, 45]. Further, LAKD does not use encryption algorithms. The proposal bases on lightweight operations such as addition, subtraction, xor, and a hash function. In [99] the authors report that the execution time of a hash function is 75.93% lower than an encryption algorithm. And according to [38], the difference is 99.85%. LAKD ciphers session random numbers using xor operations. And use hash functions to compute message authentication codes (MACs) to ensure the integrity of the transmitted messages.

The security of LAKD was analyzed using formal and informal methods. The AVISPA tool was used to formally assess the security of LAKD in a Dolev-Yao (DY) channel. In the DY threat model, $\mathscr{A}$ can seize, modify, and reassemble any transmitted message. The AVISPA tool concluded that LAKD resists replay and MITM attacks in these conditions. BAN logic was also used to evaluate the trustworthiness of the transmitted data formally. LAKD achieved secure authentication when it met the goals below.

$E_g \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

$E_s \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

LAKD achieved a strong authentication when it accomplished the goals below [114].

$E_g \mid\equiv E_s \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

$E_s \mid\equiv E_g \mid\equiv (E_g \overset{SK}{\longleftrightarrow} E_s)$.

Informal security analyzes were conducted where it was examined how LAKD achieves security attributes. They include mutual authentication, integrity, confidentiality, sensor node anonymity, known session key

security, and perfect forward and backward secrecy. Moreover, it was analyzed how LAKD resists potential attacks in IIoT. They include DoS, impersonation, injection, traceability, key disclosure, and desynchronization. Off-line identity guessing, known session-specific temporary information, privileged insider, MITM, and replay attacks. The reader can refer to the results in the article [72].

But as with many cryptographic protocols, the are limitations in LAKD. They include the requirement of the gateway storing a key pool per sensor node that belongs to its network. The gateway device has many computational resources; however, the memory cost can be notable if the network is large. A solution is to store the key pools in an authentication server. The gateway retrieves the key pool from the authentication server when it receives a communication request from the sensor node. The gateway can have a cache memory in which it stores the key pools of sensor nodes that communicate frequently, similar to the cache proposed in [115]

LAKD was compared with similar schemes for M2M communication in terms of the computing requirements and the security features achieved. Table 4.3 shows the comparison concerning the execution times of the protocols. Figure 4.5 about the communication cost. And Table 4.5 compares the security features provided by the proposals. Figure 4.6 and Figure 4.7 show the protocols' execution times with the increase in number of users. The first figure uses Case 1 criteria, and the second Case 2. The running times increase linearly with the growth in the number of users in the system. Nonetheless, the increase of LAKD's running time is slow for both criteria. Figure 4.8 and Figure 4.9 shows the communication cost with the increase in number of users, for Case 1 and Case 2 criteria, respectively. Nevertheless, the increase of LAKD's communication cost is slow for both criteria. LAKD's is not the protocol with the lowest execution time and communication cost. However, it does achieve more security features than the other schemes. When LAKD was designed, the main priority was security. From the analysis results, it can be concluded that LAKD achieves a security level similar to protocols that use more computational resources. E.g., the proposal of Qiu *et al.* resists many attacks. However, it has a higher execution time and communication cost than LAKD.

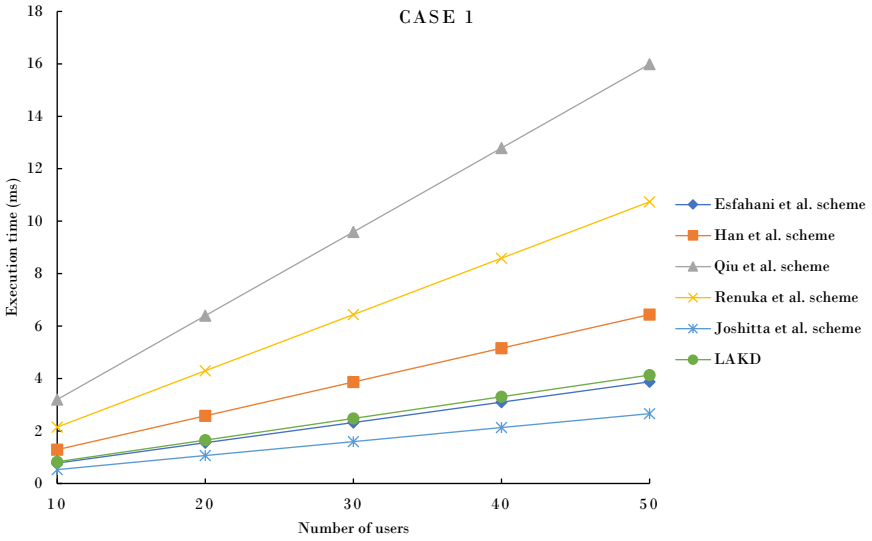


Figure 4.6 Computation overhead of LAKD and similar protocols for Case 1 criteria, with regard to the growth in the number of users.

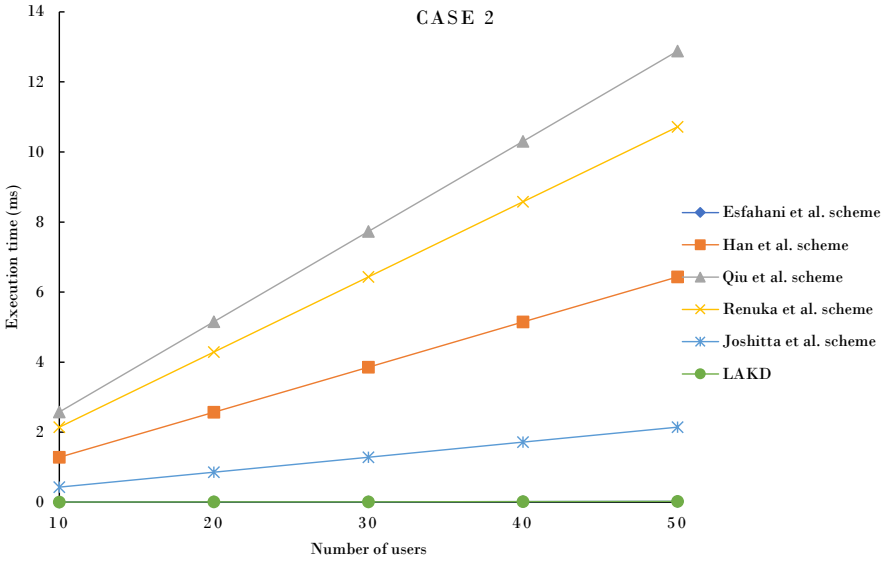


Figure 4.7 Computation overhead of LAKD and similar protocols for Case 2 criteria, with regard to the growth in the number of users.

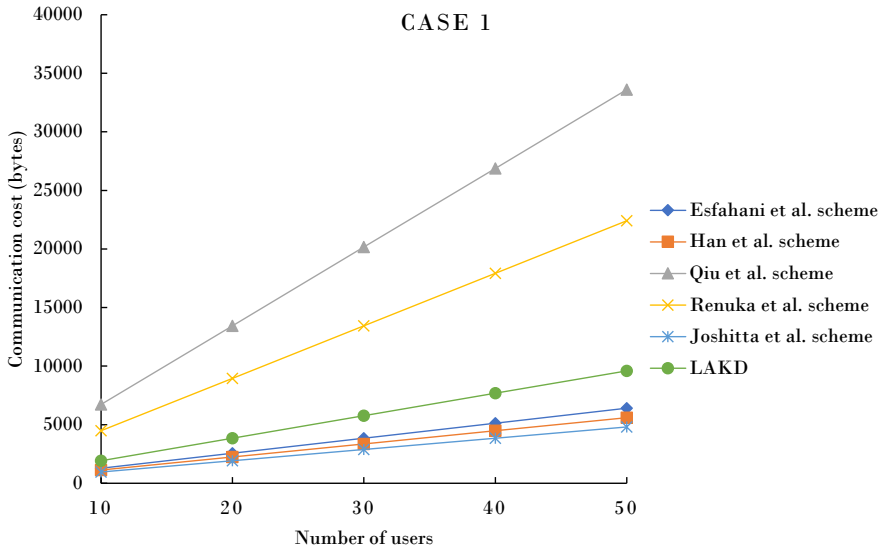


Figure 4.8 Communication overhead of LAKD and similar protocols for Case 1 criteria, with regard to the growth in the number of users.

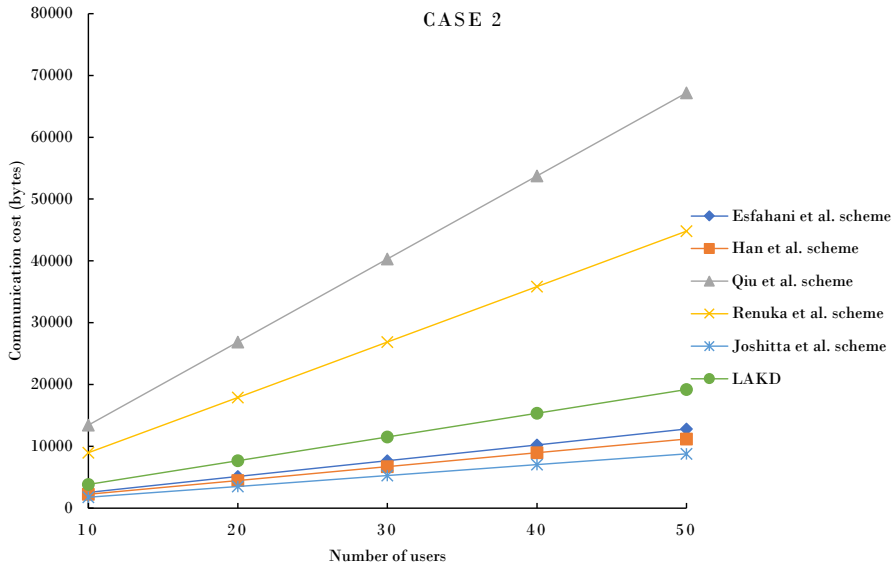


Figure 4.9 Communication overhead of LAKD and similar protocols for Case 2 criteria, with regard to the growth in the number of users.

4.2 TLAP

Health care is one of the current challenges of humanity. The world population is fast-growing, as well as the aging population. However, the number of medical facilities is not growing in proportion to the population sizee [116]. Further, sedentary behaviors and inadequate diets cause increments in chronic diseases that medical professionals need to monitor continuously to prevent risks to the patient's life [117]. Moreover, timely disease diagnostics can reduce the costs of medical systems [118].

Consequently, it is imperative to build proactive, efficient, accurate, intelligent, and affordable methods for continuous medical monitoring and early risk detection that help to decrease the workload in medical facilities [119]. Medical-related services can adopt the wireless body area network (WBAN) technology to improve patient monitoring and transmission of health parameter status. WBANs are wireless communication networks composed of remote control centers, various biomedical sensors, and portable personal terminals [120]. Biomedical sensors are in different body areas, either inside the patients' skin or on them. They measure body parameters, including weight, temperature, heart rate, blood glucose, blood pressure, and respiration rate [121, 122]. Portable personal terminals gather health data from sensors and deliver them to remote control centers. The latter make the information accessible to medical professionals. WBANs can erect ubiquitous systems that facilitate medical services to patients anywhere and anytime to relieve the pressure in health centers [123].

A significant challenge of medical applications is the privacy and security of the patients' health data. The information is transmitted through insecure wireless networks and can encounter security attacks such as data modification, interception, and eavesdropping. The Health Insurance Portability and Accountability Act (HIPAA) obliges medical professionals to secure patients' medical records [123, 124]. Health information is confidential due to its unauthorized exposure can cause social conflicts to patients. Further, the malicious fabrication and alteration of medical data can jeopardize the patients' life [116]. WBAN technology has a high potential to build proactive, efficient, and affordable health-care services. Nonetheless, WBAN systems must protect patient medical information effectively to achieve the confidence of health-care professionals and patients. Some of the requirements WBAN systems must meet are authentication and data integrity, confidentiality, and freshness [121].

Many authentication protocols for WBANs have been proposed to address the security problems. Some works use Public Key Infrastructure (PKI) [125], including [126, 49, 50, 127]. Nevertheless, security mechanisms based on PKI consume many computational resources in the management, storage, processing, and communication tasks. Thus, PKI is not appropriated for WBANs [128]. Other proposals are based on bilinear pairing operations and map-to-point hash functions [129, 130], such as the schemes [131, 132, 133, 134, 135]. These cryptographic operations use many computational resources, which affect the WBAN devices' batter-

ies [136, 137]. Some devices are inside the patient's body; thus, their batteries are difficult to replace and recharge. Therefore, bilinear pairing operations and map-to-point hash functions are not suitable for this type of device.

In contrast, TLAP protocol uses only ECC scalar point multiplication, symmetric-key encryption, conventional one-way hash function, and xor operation. The operations are lightweight since the computational cost of the map-to-point hash function and bilinear pairing operation are many times costlier than ECC scalar point multiplication [137]. Consequently, TLAP does not impact highly on the WBAN devices' computing and battery resources.

The main contributions of the proposal are summarized below.

1. TLAP authentication protocol based on ECC self-certified public keys, ECC scalar point multiplication, symmetric key encryption, xor, and hash function. TLAP does not require computational complex operations such as bilinear pairing, map-to-point hash function, or PKI-based. Therefore, TLAP has low running time and communication cost.

2. TLAP does not need public-key certificates. It uses self-certified public keys, which avoid the computational cost of managing certificates [26].

3. Key escrow problem avoidance. Protocols based on IBC suffer from the key escrow problem because a third party named the private key generator (PKG) creates the parties' private keys. Malicious PKG could do MITM attacks using the parties' keys [26, 138]. In TLAP, the principals create their private keys, preventing the key escrow problem.

4. The security of TLAP was formally assessed through the AVISPA tool and BAN logic. Additionally, informal analyzes of the proposal were made. The findings that resulted from the analyzes show that TLAP achieves mutual authentication and key agreement, data integrity, confidentiality, client anonymity, known session key security, and perfect forward and backward secrecy. Further, TLAP resists potential attacks in WBAN such as tracing, impersonation, off-line identity guessing, injection, privileged insider, MITM, replay, DoS, known session-specific temporary information, key disclosure, and de-synchronization attacks.

5. The TLAP's computing requirements were analyzed concerning the execution time and communication cost. The performance of TLAP was compared with similar protocols, which showed that TLAP uses fewer computing resources.

6. The TLAP's security features were compared with similar protocols, which showed that TLAP has more security features and better resistance to attacks.

4.2.1 Network model

TLAP assumes the typical network model of WBANs composed of sensors, a controller, an NM, and APs. Figure 4.10 illustrates the model. The sensors measure physiological parameters and send the information to a portable personal terminal known as the controller. The latter gathers measurements from various sensors and delivers them to telemedicine APs, which supply them to medical professionals. The health information is communicated in two phases. First, the sensors transmit it to the controller. Second, the latter transmits the information to APs. TLAP is proposed to secure the second communication phase. The first phase can be authenticated with LAKD since the sensor nodes are resource-constrained devices.

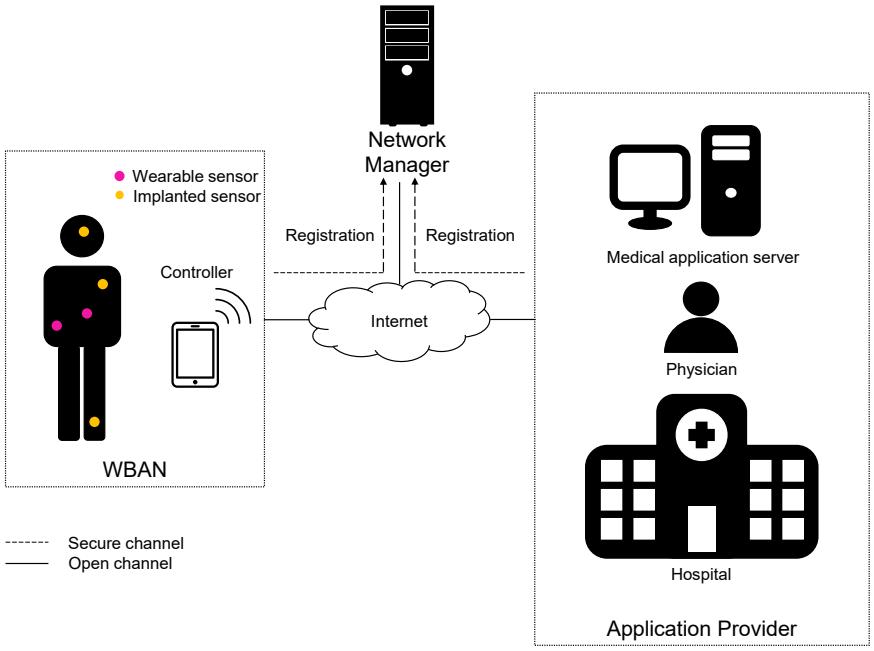


Figure 4.10 WBAN network architecture used in TLAP protocol.

There are three types of participants in TLAP. They are described below.

- WBAN Client: It acts as the WBAN controller. It can be a smartphone, PDA, or medical device. Patients use it to access health services provided by APs.
- NM: It acts as the management server in the WBAN. It generates the system's parameters, enrolls WBAN Clients and APs, and creates the public keys of WBAN Clients and APs. Since NMs can be commercial organizations, they can illegally access and collect private data to get commercial benefits. TLAP avoids these behaviors because it does not have the key escrow problem. Therefore, NMs cannot impersonate APs and WBAN Clients to access private information [54].

- AP: It supplies remote services to authorized users. The services include patient monitoring, physician consults, and medical treatments. APs are entities such as clinics, hospitals, and physicians.

4.2.2 Threat model

A telecare medicine information system (TMIS) should meet the security requirements described below [133].

- Provide mutual authentication and key agreement on an insecure channel securely and efficiently.
- The mutual authentication procedure should provide un-traceability and anonymity.
- Session keys should be authenticated to prevent attacks.
- The mutual authentication procedure should use few computing resources because IoT devices are resource-constrained.

The threat model for IoT environments examined in [139] is adopted in TLAP. A brief discussion of the model is presented below.

The threat model considers a DY channel in which a passive or active $\mathscr{A}$ controls the communication channel. $\mathscr{A}$ can seize, modify, and reassemble any transmitted message. Nonetheless, $\mathscr{A}$ can decrypt messages only if he/she has the appropriate keys.

Furthermore, TLAP also applies the CK-adversary model [140, 141], in which $\mathscr{A}$ can expose secret information such as keys and session states. Thus, authentication protocols have to ensure that the secret information of other parties' is not exposed if ephemeral secrets are leaked. [139]. $\mathscr{A}$ can also physically capture smart devices to extract their secret credentials because the devices can be in unattended environments.

Gateways and servers are trusted and reliable entities. $\mathscr{A}$ cannot compromise them.

TLAP bases its security on the ECC operations and computationally intractable problems introduced below.

Let p and q be large prime numbers, E is a non-singular elliptic curve on the finite field F_p: $E = y^2 = x^3 + ax + b \mod p$, $a, b \in_R F_p$ and $4a^3 + 27b^2 \neq 0$ holds. G is an additive cyclic group formed by the points on the curve. G has order q. Point P is the group generator.

Elliptic curve has the group properties introduced below [26].

- **Point addition:** If P and Q are two points on $E_p(a, b)$, the point addition is $P + Q = R$. Point $-R$ is the intersection on $E_p(a, b)$ of the line that joins P and Q. Point R is the reflection of $-R$ on the x-axis.

- **Point doubling:** If P is a point on $E_p(a,b)$, the point doubling is $Q = 2.P$. Point $-Q$ is the intersection on $E_p(a,b)$ of the tangent line at P. Point Q is the reflection of $-Q$ on the x-axis.

- **Scalar point multiplication:** If P is a point on $E_p(a,b)$ and $x \in Z_q^*$, the scalar point multiplication is $Q = x.P = \{P + P + ... + P \ (x \text{ times})\}$. It adds the point P x times.

TLAP builds its security on the computationally intractable problems described below [27, 24]. The notation t denotes a polynomial-time boundary.

- **Elliptic Curve Discrete Logarithm Problem (ECDLP)**: Let P and Q be two points on $E_p(a,b)$, where $Q = x.P$ and $x \in Z_q^*$. $\mathscr{A}$ can compute x from $\{P, Q\}$ with the probability $Adv_{\mathscr{A}}^{ECDLP}(t) = Pr[\mathscr{A}(P, Q = x.P) = x : x \in Z_q^*] \leq \epsilon$. $\mathscr{A}$ has an advantage $Adv_{\mathscr{A}}^{ECDLP}(t)$ negligible.

- **Elliptic Curve Computational Diffie-Hellman Problem (ECDHP)**: Let P, Q, and R be three points on $E_p(a,b)$, where $Q = x.P$, $R = y.P$, and $x, y \in Z_q^*$. $\mathscr{A}$ can compute $xy.P$ from $\{P, Q, R\}$ with the probability $Adv_{\mathscr{A}}^{ECDHP}(t) = Pr[\mathscr{A}(P, Q = x.P, R = y.P) = xy.P : x, y \in Z_q^*] \leq \epsilon$. $\mathscr{A}$ has an advantage $Adv_{\mathscr{A}}^{ECDHP}(t)$ negligible.

- **Elliptic Curve Decisional Diffie–Hellman Problem (ECDDHP)**: Let P, $x.P$, $y.P$, and $z.P$ be four points on $E_p(a,b)$, where $x, y, z \in Z_q^*$. The challenge is to decide whether $z = xy$ or it is a uniform value. When p is large, e.g., p is chosen at least as a 160-bit prime number [142], ECDDHP is computationally infeasible.

TLAP assumes the security properties presented below.

1. The entities perform the registration phase in a secure communication channel.

2. TLAP uses a one-way hash function that is collision-resistant.

3. NM is a secure entity. $\mathscr{A}$ cannot jeopardize it.

TLAP comprises the following phases: setup, registration, and mutual authentication. In the setup phase, the NM generates the system's parameters. It runs one time when the environment starts up. In the registration phase, the entities' private and public key pairs are created. It executes every time a new entity joins the environment. The mutual authentication phase runs each time two entities need to communicate securely. Figure 4.11 illustrates the participants and working flow in TLAP.

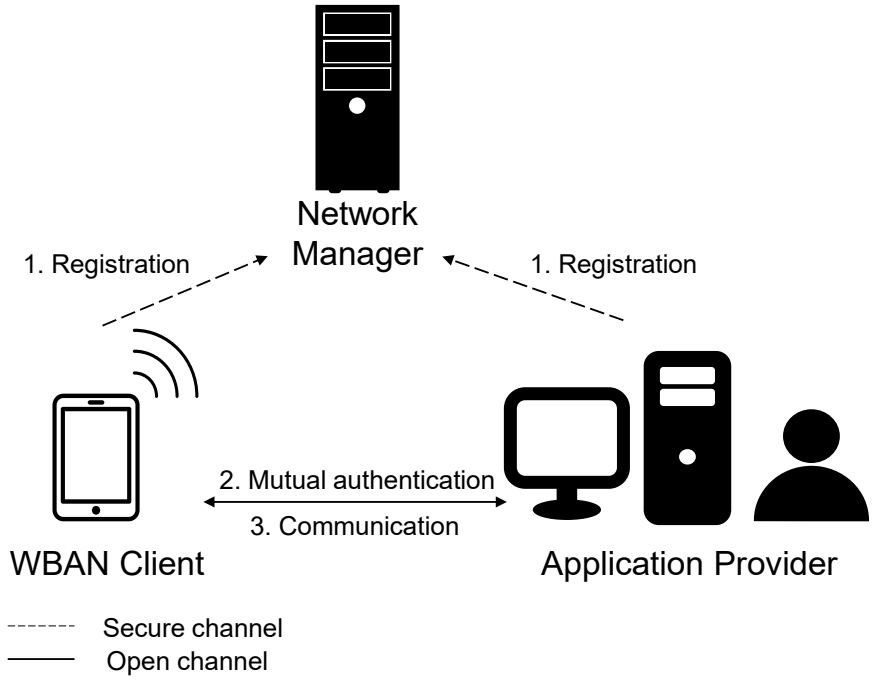


Figure 4.11 Participants and working flow in TLAP.

4.2.3 Setup phase

NM generates the following system parameters. NM selects the random number $s \in_R Z_q^*$ and sets it as its private key. Then, NM computes its public key $Pub_s = s.P$. NM chooses a one-way collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$, $k \in Z^+$. Lastly, NM publishes $\{P, p, q, Pub_s, H\}$. NM keeps s a secret.

4.2.4 Registration phase

A principal, named Entity i, registers with the NM. The NM generates the Entity i's public key, denoted Pub_i, whereas Entity i creates its private key, labeled i, to avoid the key escrow problem. The steps of the phase are described below. Figure 4.12 illustrates the messages the entities exchange in the registration phase.

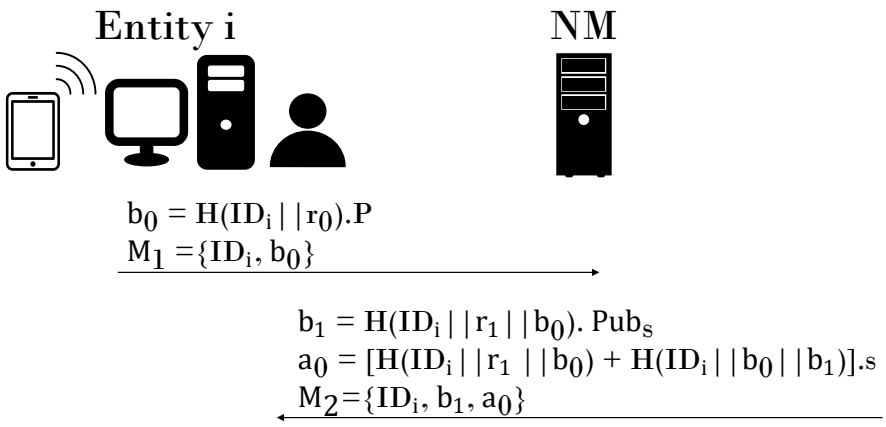


Figure 4.12 TLAP's registration phase.

1. Entity i selects a random number $r_0 \in_R Z_q^*$ and its identity ID_i. It computes $b_0 = H(ID_i||r_0).P$ and sends ID_i and b_0 to NM.

2. NM chooses a random number $r_1 \in_R Z_q^*$. It computes $b_1 = H(ID_i||r_1||b_0).Pub_s$, $a_0 = [H(ID_i||r_1||b_0)+ H(ID_i||b_0||b_1)].s$, and $Pub_i = b_1 + H(ID_i||b_0||b_1).Pub_s + b_0$ and sends ID_i, b_1, a_0 to Entity i.

3. Entity i computes $i = a_0 + H(ID_i||r_0)$ and verifies the correctness of (ID_i, b_1, a_0) via $i.P = b_1 + H(ID_i||b_0||b_1).Pub_s + b_0$. Entity i adopt $\{Pub_i, i\}$ as its public and private key pairs if the expression holds.

The Entity i's private key is verified as described below.

$i.P = \{[H(ID_i||r_1||b_0) + H(ID_i||b_0||b_1)].s + H(ID_i||r_0)\}.P$

$i.P = H(ID_i||r_1||b_0).s.P + H(ID_i||b_0||b_1).s.P + H(ID_i||r_0).P$

$i.P = H(ID_i||r_1||b_0).Pub_s + H(ID_i||b_0||b_1).Pub_s + H(ID_i||r_0).P$

$i.P = H(ID_i||r_1||b_0).Pub_s + H(ID_i||b_0||b_1).Pub_s + b_0$ $i.P = b_1 + H(ID_i||b_0||b_1).Pub_s + b_0$

4.2.5 Mutual authentication phase

The mutual authentication phase comprises the four steps described below. For convenience, AP is denoted as AP b, and WBAN client as Client a. Figure 4.13 illustrates the steps of this phase.

Select: T_1 and $r_1 \in_R Z_q^*$
Compute:
$D_1 = (r_1 \oplus a).P$
$D_2 = (r_1 \oplus a).Pub_b$
$D_3 = E_{H(D_2)}(ID_a \oplus D_2)$
$D_4 = H(D_1||D_2||ID_a||ID_b||T_1)$
$M1 = \{T_1, D_1, D_3, D_4\}$

Select: T_2
Verify: $|T_2 - T_1| \leq \Delta T$
Compute:
$D_2 = b.D_1$
$ID_a = \left(D_{H(D_2)}(D_3)\right) \oplus D_2$
Verify: $D_4 \stackrel{?}{=} H(D_1||D_2||ID_a||ID_b||T_1)$
Select: $r_2 \in_R Z_q^*$
Compute:
$D_5 = (r_2 \oplus b).P$
$D_6 = (r_2 \oplus b).Pub_a$
$D_7 = H(D_5||D_6||D_2||ID_b||ID_a||T_2)$
$M_2 = \{T_2, D_5, D_7\}$

Select: T_3
Verify: $|T_3 - T_2| \leq \Delta T$
Compute: $D_6 = a.D_5$
Verify: $D_7 \stackrel{?}{=} H(D_5||D_6||D_2||ID_b||ID_a||T_2)$
Compute:
$D_8 = H(D_2||D_6||ID_a||ID_b||T_3)$
$K = D_2 \oplus D_6$
$SK = H(K||ID_a||ID_b)$
$M_3 = \{T_3, D_8\}$

Select: T_4
Verify: $|T_4 - T_3| \leq \Delta T$
Verify: $D_8 \stackrel{?}{=} H(D_2||D_6||ID_a||ID_b||T_3)$
Compute:
$K = D_2 \oplus D_6$
$SK = H(K||ID_a||ID_b)$

Figure 4.13 TLAP's authentication phase..

1. Client a selects a random number $r_1 \in_R Z_q^*$ and a timestamp T_1. It computes the next data: $D_1 = (r_1 \oplus a).P$, $D_2 = (r_1 \oplus a).Pub_b$, $D_3 = E_{H(D_2)}(ID_a \oplus D_2)$, and $D_4 = H(D_1||D_2||ID_a||ID_b||T_1)$. Data a is the private key of Client a and Pub_b is the public key of AP b. D_3 uses the hash of D_2 as encryption key to cipher $ID_a \oplus D_2$. Client a sends $M_1 = \{T_1, D_1, D_3, D_4\}$ to AP b.

2. AP b chooses a timestamp T_2 and checks $|T_2 - T_1| \leq \Delta T$. If it holds, AP b computes $D_2 = b.D_1$ and $ID_a = (D_{H(D_2)}(D_3)) \oplus D_2$. Data b is the private key of AP b and $D_{H(D_2)}(D_3)$ denotes the D_3's deciphering using as decryption key the hash of D_2. AP b checks $D_4 \stackrel{?}{=} H(D_1||D_2||ID_a||ID_b||T_1)$. If any of the validations is false, AP b terminates the communication. Otherwise, AP b chooses a random number $r_2 \in_R Z_q^*$, and computes $D_5 = (r_2 \oplus b).P$, $D_6 = (r_2 \oplus b).Pub_a$, and $D_7 = H(D_5||D_6||D_2||ID_b||ID_a||T_2)$. Pub_a is the public key of Client a. AP b sends $M_2 = \{T_2, D_5, D_7\}$ to Client a.

3. Client a chooses a timestamp T_3 and checks $|T_3 - T_2| \leq \Delta T$. If it holds, Client a computes $D_6 = a.D_5$ and checks $D_7 \stackrel{?}{=} H(D_5||D_6||D_2||ID_b||ID_a||T_2)$. If any of the validations is false, Client a terminates the communication. Otherwise, Client a computes the following: $D_8 = H(D_2||D_6||ID_a||ID_b||T_3)$, $K = D_2 \oplus D_6$, and $SK = H(K||ID_a||ID_b)$. SK is the session key. Client a sends $M_3 = \{T_3, D_8\}$ to AP b.

4. AP b chooses a timestamp T_4 and checks $|T_4 - T_3| \leq \Delta T$. If it holds, AP b verifies $D_8 \stackrel{?}{=} H(D_2||D_6||ID_a||ID_b||T_3)$. If any of the validations is false, AP b terminates the communication. Otherwise, it computes $K = D_2 \oplus D_6$ and $SK = H(K||ID_a||ID_b)$. SK is the session key. At this point, the mutual authentication between AP b and Client a is successful. The parties have created the same session key SK which they will use to exchange data with confidentiality.

4.2.6 Security analysis

Formal and informal methods were used to analyze the security of TLAP. The formal methods were the AVISPA tool and BAN logic. Further, it was informally studied how TLAP achieves the security attributes of confidentiality, client anonymity, mutual authentication and key agreement, data integrity, known session key security, and perfect forward and backward secrecy. Moreover, the analysis of TLAP resistance to potential attacks in WBAN. They include tracing, key disclosure, off-line identity guessing, injection, impersonation, privileged insider, known session-specific temporary information, replay, de-synchronization, DoS, and MITM attacks. The informal analysis is not included in the document to keep this subsection concise. The reader can refer to the details of the study in the article.

The AVISPA tool and BAN logic analysis results are described below.

AVISPA tool

The TLAP's modeling in the HLPSL language is presented in Appendix B. The security goals stipulated were secrecy of the nonces used in the session key and mutual authentication. If any goal is unachieved, the AVISPA tool concludes that the protocol is unsafe. The TLAP's modeling was evaluated through the CL-AtSe and OFMC back-ends due to their support of the xor function [92]. The two back-ends concluded that TLAP resists replay and MITM attacks. Moreover, TLAP achieved the secrecy of the nonces used in the session key and the mutual authentication of the principals. Figure 4.14 shows the evaluation results.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/protocol.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 2.48s
 visitedNodes: 518 nodes
 depth: 10 plies
```

(a)

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /home/span/span/testsuite/results/protocol.if
GOAL
 As Specified
BACKEND
 CL-AtSe


STATISTICS
 Analysed  : 144 states
 Reachable  : 108 states
 Translation: 0.08 seconds
 Computation: 0.05 seconds
```

(b)

Figure 4.14 TLAP's analysis results through the AVISPA tool: (a) Using the OFMC back-end. (b) Using the CL-AtSe back-end.

BAN logic

TLAP needs to fulfill the following security goals in BAN logic to demonstrate it achieves mutual authentication. The BAN logic proof uses the notations E_{ca} and E_{apb} to represent the Client a and the AP b, respectively.

Goal 1: $E_{ca} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$.

Goal 2: $E_{apb} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$.

Goal 3: $E_{ca} \mid\equiv E_{apb} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$.

Goal 4: $E_{apb} \mid\equiv E_{ca} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$.

The TLAP's idealized form is as follows:

Message 1: $\{(r_1 \oplus a)\}_{Pub_b}, \{ID_a\}_{(r_1 \oplus a)}, \left\{ID_a, E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, T_1\right\}_{(r_1 \oplus a)}$.

Message 2: $\{(r_2 \oplus b)\}_{Pub_a}, \left\{\langle E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}\rangle_{(r_1 \oplus a)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2\right\}_{(r_2 \oplus b)}$.

Message 3: $\left\{\langle E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}\rangle_{(r_2 \oplus b)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3\right\}_{(r_1 \oplus a)}$.

The TLAP's assumptions about its initial state are presented below:

1. $E_{ca} \mid\equiv E_{apb} \Rightarrow (r_2, T_2)$.
2. $E_{ca} \mid\equiv \sharp(T_2)$.
3. $E_{ca} \mid\equiv \overset{Pub_b}{\longmapsto} E_{apb}$.
4. $E_{apb} \mid\equiv E_{ca} \Rightarrow (r_1, T_1, T_3)$.
5. $E_{apb} \mid\equiv \sharp(T_1, T_3)$.
6. $E_{apb} \mid\equiv \overset{Pub_a}{\longmapsto} E_{ca}$.

The TLAP's BAN logic proof is described below. It shows that TLAP achieves the four mutual authentication goals presented previously.

From Message 1 of TLAP idealized form, we achieve:

Step 1: $E_{apb} \triangleleft \{(r_1 \oplus a)\}_{Pub_b}, \{ID_a\}_{(r_1 \oplus a)}, \left\{ID_a, E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, T_1\right\}_{(r_1 \oplus a)}$.

Step 2: From Step 1, applying Rule 6, we obtain: $E_{apb} \triangleleft (r_1 \oplus a)$.

Step 3: From Step 1 and Step 2, applying Rule 6, we obtain: $E_{apb} \triangleleft (ID_a, (ID_a, E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, T_1))$.

Step 4: From Step 3, applying Rule 4 and Assumption 5, we obtain: $E_{apb} \mid\equiv \sharp(ID_a, E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, T_1)$.

From Message 2, we achieve:

Step 5: $E_{ca} \triangleleft \{(r_2 \oplus b)\}_{Pub_a}, \left\{\langle E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}\rangle_{(r_1 \oplus a)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2\right\}_{(r_2 \oplus b)}$.

Step 6: From Step 5, applying Rule 6, we obtain: $E_{ca} \triangleleft (r_2 \oplus b)$.

Step 7: From Step 5 and Step 6, applying Rule 6, we obtain: $E_{ca} \triangleleft (\langle E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}\rangle_{(r_1 \oplus a)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2)$.

Step 8: From Step 7, applying Rule 4 and Assumption 2, we obtain: $E_{ca} \mid\equiv \sharp(\langle E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}\rangle_{(r_1 \oplus a)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2)$.

Step 9: From Step 7, applying Rule 1 and the assumption $E_{ca} \mid\equiv E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}$ because it is its originator,

we obtain: $E_{ca} \mid\equiv E_{apb} \mid\sim (E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2)$.

Step 10: From Step 9, applying Rule 2, we obtain: $E_{ca} \mid\equiv E_{apb} \mid\equiv (E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2)$.

Step 11: From Step 10, applying Rule 3 and Assumption 1, we obtain: $E_{ca} \mid\equiv (E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_2)$.

Step 12: From Step 10, applying Rule 5, we obtain: $E_{ca} \mid\equiv E_{apb} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$. **(Goal 3)**

Step 13: From Step 11, applying Rule 5, we obtain: $E_{ca} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$.

(Goal 1)

From Message 3, we achieve:

Step 14: $E_{apb} \triangleleft \left\{ \langle E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb} \rangle_{(r_2 \oplus b)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3 \right\}_{(r_1 \oplus a)}$.

Step 15: From Step 14 and Step 2, applying Rule 6, we obtain: $E_{apb} \triangleleft (\langle E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb} \rangle_{(r_2 \oplus b)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3)$.

Step 16: From Step 15, applying Rule 4 and Assumption 5, we obtain: $E_{apb} \mid\equiv \sharp(\langle E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb} \rangle_{(r_2 \oplus b)}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3)$.

Step 17: From Step 15, applying Rule 1 and the assumption $E_{apb} \mid\equiv E_{ca} \overset{(r_2 \oplus b)}{\rightleftharpoons} E_{apb}$ because it is its originator, we obtain: $E_{apb} \mid\equiv E_{ca} \mid\sim (E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3)$.

Step 18: From Step 17, applying Rule 2, we obtain: $E_{apb} \mid\equiv E_{ca} \mid\equiv (E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3)$.

Step 19: From Step 18, applying Rule 3 and Assumption 4, we obtain: $E_{apb} \mid\equiv (E_{ca} \overset{(r_1 \oplus a)}{\rightleftharpoons} E_{apb}, E_{ca} \overset{SK}{\longleftrightarrow} E_{apb}, T_3)$

Step 20: From Step 18, applying Rule 5, we obtain: $E_{apb} \mid\equiv E_{ca} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$.

(Goal 4)

Step 21: From Step 19, applying Rule 5, we obtain: $E_{apb} \mid\equiv (E_{ca} \overset{SK}{\longleftrightarrow} E_{apb})$. **(Goal 2)**

4.2.7 Performance analysis

The performance of TLAP was estimated using the following criteria. The time needed to execute the operations in the authentication phase. The communication cost with respect to the number of bits transmitted by the parties in the authentication phase. And the security features provided by the protocol. Many proposals use these criteria for the performance evaluation, including [142, 82, 143, 144, 145, 146]. Designers can decide about the protocol's viability to secure a device through evaluating the protocol's performance. IoT networks have to satisfy operational requirements such as real-time operation. The last-mentioned relates to the system's capacity to respond predictably and correctly and within deadlines. The addition of cryptographic operations should not stop a system from fulfilling its response deadlines [65].

A comparative study of the performances of TLAP and similar protocols, such as those developed by Das *et al.* [142], Ying *et al.* [143], Hsieh *et al.* [82], Islam *et al.* [144], He *et al.* [145], and Kim *et al.* [146] is presented.

The running times of the operations in the protocols' authentication phases were added for the execution time evaluation. The operations in the registration phase were not considered because the parties run this phase only once.

For ECC-based schemes, it was adopted an additive group G generated by point P with order q, over the non-singular elliptic curve $y^2 = x^3 + ax + b \mod p$. p and q are prime numbers, and they are 160 bits long each one. For bilinear pairing-based protocols, it was adopted a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$. G_1 is an additive group with order $\overline{q}$. Point $\overline{P}$ generates the group over a supersingular elliptic curve $y^2 = x^3 + 1 \mod \overline{p}$. $\overline{p}$ and $\overline{q}$ are prime numbers, whose sizes are 512 bits and 160 bits, respectively [147, 143, 148].

The operations' running times were taken from the experimental results shown in [147, 143]. The environment setup of the experiment was a computer with the processor Intel i7-4770 at 3.40 GHz of clock frequency, the Windows 7 operating system, and 4 gigabytes of memory. The authors got the operations' running times using the cryptographic library Multiprecision Integer and RationalArithmetic Cryptographic Library (MIRACL) [149].

Table 4.6 lists the operations used in the evaluation and their running times. The execution times of the concatenation and xor operations were not considered because their execution times are negligible when compared to the other operations.

Table 4.6 Metrics used for the evaluation of TLAP's execution time.

Symbol	Operation description	Execution time
T_{bp}	Bilinear pairing.	4.21100 ms
T_{sm-bp}	Scalar multiplication, related to the bilinear pairing.	1.70900 ms
T_{pa-bp}	Point addition, related to the bilinear pairing.	0.00700 ms
T_h	Conventional hash function.	0.00010 ms
$T_{e/d}$	Symmetric key encryption or decryption function.	0.00020 ms
T_{mtp}	Hash-to-point function.	4.30200 ms
T_{mul}	Multiplication.	0.00001 ms
T_{sm-ecc}	Scalar point multiplication.	0.44200 ms
T_{pa-ecc}	Point addition.	0.00180 ms

The running times for TLAP and related schemes are shown in Table 4.7. TLAP has the lowest running time. The protocol of Hsieh *et al.* has the highest. The reason for this is that they use bilinear pairing operations and hash-to-point functions. The proposals of Ying *et al.*, Islam *et al.*, and Kim *et al.* have close running times.

Table 4.7 Results of the evaluation of TLAP's execution time.

Protocol proposed by	Entity	Operations	Execution time
Das *et al.* [142]	Device i	$7T_{sm-ecc} + 3T_{pa-ecc} + 6T_h + T_{mul}$	3.1000 ms
	Device j	$7T_{sm-ecc} + 3T_{pa-ecc} + 6T_h + T_{mul}$	3.1000 ms
Ying *et al.* [143]	User	$4T_{sm-ecc} + 2T_{pa-ecc} + 7T_h$	1.7723 ms
	Server	$4T_{sm-ecc} + 2T_{pa-ecc} + 3T_h$	1.7719 ms
Hsieh *et al.* [82]	User	$T_{mtp} + 7T_{sm-bp} + T_{pa-bp} + 8T_h$	16.2728 ms
	Server	$T_{mtp} + 2T_{bp} + 5T_{sm-bp} + T_{pa-bp} + 3T_h$	21.2763 ms
Islam *et al.* [144]	User A	$4T_{sm-ecc} + 2T_{pa-ecc} + 4T_h + 3T_{mul}$	1.7720 ms
	User B	$4T_{sm-ecc} + 2T_{pa-ecc} + 4T_h + 3T_{mul}$	1.7720 ms
He *et al.* [145]	Entity A	$5T_{sm-ecc} + 3T_{pa-ecc} + 2T_h$	2.2156 ms
	Entity B	$5T_{sm-ecc} + 3T_{pa-ecc} + 2T_h$	2.2156 ms
Kim *et al.* [146]	Entity A	$4T_{sm-ecc} + 3T_{pa-ecc} + 2T_h$	1.7736 ms
	Entity B	$4T_{sm-ecc} + 3T_{pa-ecc} + 2T_h$	1.7736 ms
Us (TLAP)	Client a	$3T_{sm-ecc} + 5T_h + T_{e/d}$	1.3267 ms
	AP b	$3T_{sm-ecc} + 5T_h + T_{e/d}$	1.3267 ms

The protocols' communication costs were analyzed by adding the number of bits that the entities transmit in the authentication phases, similar to the works in [142, 150]. The data sent in the registration phases were not added since the parties run the phase once only, and the communication operates in a private channel.

The sizes of the elements in G and G_1 are following described, which are based on the lengths of p and $\overline{p}$ indicated previously in the evaluation of the execution time. Points over an elliptic curve consist of $P = (x_P, y_P)$, x_P and y_P are the x and y coordinates, respectively. Hence, an element in G has a size of 160 x 2 = 320 bits. A 160-bit ECC cryptosystem has the same security as a 1024-bit RSA cryptosystem [148]. Lastly, an element in G_1 has a size of 512 x 2 = 1024 bits.

A hash function has an output of 160 bits if SHA-1 is employed. Timestamps are 32 bits long. Random numbers and identities have a length of 160 bits [142, 150].

Table 4.8 shows the length of the cryptographic data types sent in the protocols' messages.

Table 4.8 Metrics used for the evaluation of TLAP's communication cost.

Data	Size
Identity.	160 bits
Timestamp.	32 bits
Random number.	160 bits
Hash-function output.	160 bits
ECC point (element in G).	320 bits
Map-to-point hash-function output. The function consists of $H : \{0,1\}^* \rightarrow G_1$, where the element in G_1 is of 1024 bits [143, 151].	1024 bits
RSA modular parameters.	1024 bits

In the scheme proposed by Ying *et al.*, the user and server transmit the messages σ_{Ui}, A^*_{Ui}, DID_{Ui}, F_{Ui}, B_{Sj}, σ_{Sj}, F_{Sj}, and ID_{Sj}. Where $\sigma_{Sj}, \sigma_{Ui} \in Z^*_q$, F_{Ui}, A^*_{Ui}, F_{Sj}, $B_{Sj} \in G$, ID_{Sj} is an identity, and DID_{Ui} is a hash-function output. Therefore, the communication cost of Ying *et al.*'s scheme is 1920 bits.

In the proposal of Hsieh *et al.*, the user and server transmit the messages $xAuth_i$, M_i, C_m, B_{ij}, $Auth_{ji}$, R_i, R_j, K_{ji}, and $Auth_{ij}$. Where C_m, $xAuth_i$, M_i, R_i, B_{ij}, R_j, $K_{ji} \in G_1$, and $Auth_{ij}$, $Auth_{ji} \in Z^*_q$. Therefore, the communication cost of Hsieh *et al.*'s proposal is 7488 bits.

In the protocol of Islam *et al.*, the users transmit the messages T_A, ID_A, R_A, ID_B, S_A, T_B, S_B, and R_B. Where ID_B, ID_A, S_B, $S_A \in Z^*_q$, and R_A, R_B, T_A, $T_B \in G$. Therefore, the communication cost of Islam *et al.*'s protocol is 1920 bits.

In the scheme proposed by He *et al.*, the entities transmit the messages R_A, ID_A, ID_B, T_A, T_B, and R_B. Where T_A, T_B, R_A, $R_B \in G$, and ID_B and ID_A are identities. Therefore, the communication cost of He *et al.*'s scheme is 1600 bits.

In the proposal of Kim *et al.*, the entities transmit the messages R_A, ID_A, ID_B, T_A, T_B, and R_B. Where T_A, T_B, R_A, $R_B \in G$, and ID_B and ID_A are identities. Therefore, the communication cost of Kim *et al.*'s proposal is 1600 bits.

In the protocol of Das *et al.*, the smart devices transmit the messages A_i, ID_i, T_i, c_i, R_i, z_i, Q_i, A_j, ID_j, T_j, c_j, R_j, z_j, Q_j, SKV_{ij}, T'_i, and SKV'_{ij}. Where ID_j and ID_i are identities, T'_i, T_j, and T_i are timestamps, c_j and c_i are certificates, z_j and z_i are signatures, SKV'_{ij} and SKV_{ij} are hash-function outputs, and R_i, A_i, Q_i, R_j, Q_j, $A_j \in G$. Therefore, the communication cost of Das *et al.*'s protocol is 3296 bits.

In TLAP, the client and AP transmit the messages D_1, D_3, D_4, D_5, D_7, D_8, T_1, T_2, and T_3. Where T_1, T_2, and T_3 are timestamps, D_5, $D_1 \in G$, and D_8, D_7, and D_4 are hash-function outputs. D_3 is the output of the encryption $E_{H(D_2)}(ID_a \oplus D_2)$. An output of the same length as ID_a is obtained by using the AES algorithm in the counter mode. Therefore, the TLAP's communication cost is 1376 bits.

Figure 4.15 shows a comparison of the communication cost of the protocols described previously. The

proposal of Hsieh *et al.* has the highest cost. TLAP has the lowest. The schemes of Islam *et al.* and Ying *et al.* have an equal cost, the same for the protocols of Kim *et al.* and He *et al.*. The communication cost of the last two can be considered low.

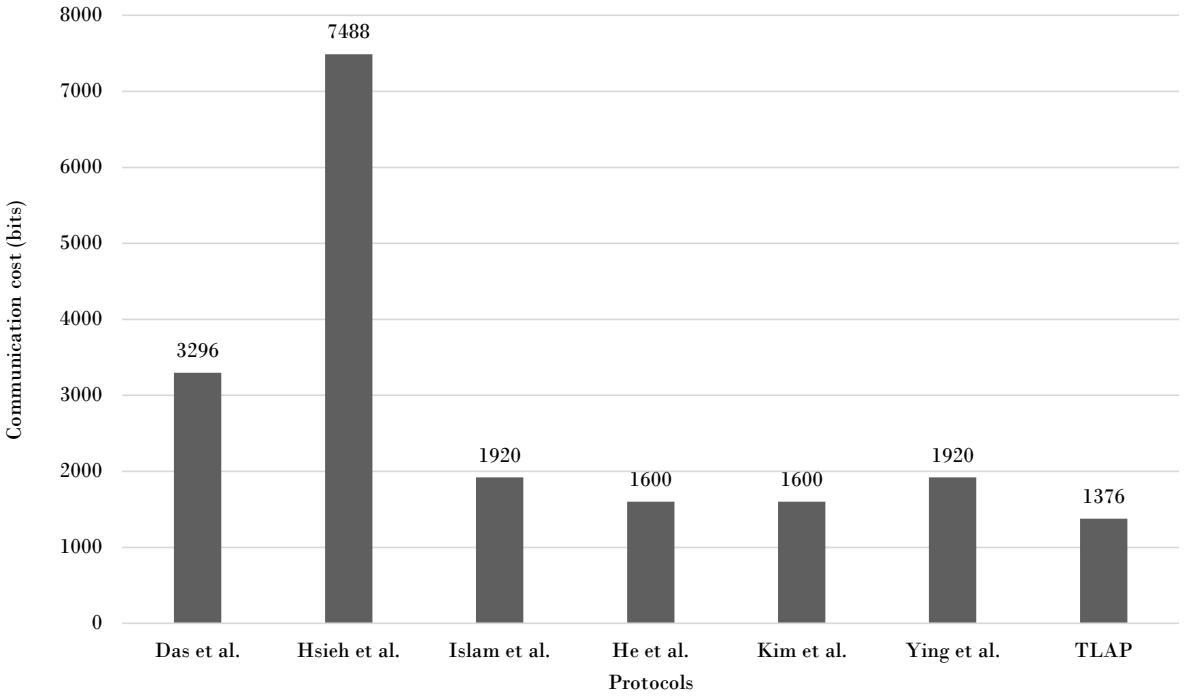


Figure 4.15 Comparison of the communication costs of TLAP and similar protocols.

4.2.8 Security features comparison

The security features of TLAP and the protocols of Das *et al.*, Ying *et al.*, Hsieh *et al.*, Islam *et al.*, He *et al.*, and Kim *et al.* were analyzed. The works in [143, 150, 152, 153, 154, 155, 156] were reviewed during the analysis. The protocols were examined concerning the security properties of un-traceability, forward secrecy, anonymity, and session key agreement. And the resistance against attacks in WBANs including impersonation, off-line identity guessing, insider, session key disclosure, MITM, and replay attacks. Table 4.9 compares the security features that the protocols achieve. TLAP achieves all the mentioned properties. The scheme proposed by Das *et al.* provides many security attributes. However, it accomplishes them at a higher communication cost and execution time than TLAP.

Table 4.9 Comparative of the resistance to attacks for TLAP and related schemes.

Protocol proposed by	SA1	SA2	SA3	SA4	SA5	SA6	SA7	SA8	SA9	SA10
Das *et al.* [142]	✓	✓	✓	✓	✓	x	✓	✓	✓	x
Ying *et al.* [143]	x	x	x	x	x	x	x	x	x	✓
Hsieh *et al.* [82]	x	x	✓	✓	x	✓	✓	✓	✓	x
Islam *et al.* [144]	x	x	✓	✓	✓	✓	x	✓	✓	✓
He *et al.* [145]	x	x	✓	✓	✓	x	x	x	✓	✓
Kim *et al.* [146]	x	x	✓	✓	✓	✓	x	x	✓	✓
Us (TLAP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: The scheme is resistant to the attack. x: The protocol is not resistant to the attack.
SA1: Anonymity; SA2: Un-traceability; SA3: Session key agreement; SA4: Forward secrecy;
SA5: Off-line identity guessing; SA6: Impersonation; SA7: Replay; SA8: Session key disclosure;
SA9: Insider attack; SA10: MITM attack.

4.2.9 Discussion and conclusion

This section described an authentication protocol named TLAP, designed for WBANs in health-care applications. TLAP does not use cryptographic operations that require many computing resources because WBAN devices have limited battery and computational resources. TLAP bases its security on ECC scalar point multiplication, symmetric key encryption, hash function, and xor operation. Consequently, its running time and communication cost are low. TLAP does not need public-key certificates because it uses self-certified public keys. Therefore, the proposal avoids the overhead of managing certificates. Moreover, TLAP does not have the key escrow problem because parties create their private keys rather than a PKG. The latter also impedes malicious PKGs from launching MITM attacks utilizing the parties' keys.

The security of TLAP was assessed using formal and informal methods. The first-mentioned were the AVISPA tool and BAN logic, which concluded that TLAP achieves mutual authentication and is secure against MITM and replay attacks. Moreover, informal analyzes were performed where it was studied how TLAP achieves security attributes. They include confidentiality, client anonymity, mutual authentication and key agreement, data integrity, known session key security, and perfect forward and backward secrecy. Furthermore, it was analyzed how TLAP resists potential attacks in WBANs. They include tracing, key disclosure, off-line identity guessing, injection, impersonation, privileged insider, known session-specific temporary information, replay, de-synchronization, DoS, and MITM attacks. The reader can refer to the results in the article.

The protocols designed for TMIS should ensure the patients' anonymity and un-traceability because medical information is private. Its unauthorized disclosure and modification can have disastrous consequences in the patients' lives. TLAP avoids tracing attacks by not sending the Client a's public key in the open channel. When Client a starts communication with AP b, the latter gets from the NM the Client a's public key. AP b can communicate securely with the NM using TLAP or a conventional cryptosystem.

Moreover, AP b can have a cache memory with the public keys of clients that request services frequently. Equivalent to the cache memory described in [115]. Thus, AP b does not require communicating with the NM to get the Client a's public key each time Client a starts communication with AP b.

The Client a's self-certified public key includes the Client a's identity (ID_a) in a non-invertible form. The NM constructs the public key with the hash of ID_a and the random numbers r_0 and r_1. Hash functions are not invertible; therefore, $\mathscr{A}$ cannot obtain ID_a from the public key. Moreover, $\mathscr{A}$ cannot off-line guess ID_a since r_0 and r_1 are secret, unpredictable, and fresh. Consider that the length of ID_a is n bits and of r_0 and r_1 is m bits. $\mathscr{A}$ can guess the values at the same time with an approximate probability of $\frac{1}{2^{n+2m}}$. It is computationally infeasible to guess two or more parameters in polynomial time [153, 157].

Client a uses ID_a as its identity only in the NM system. ID_a does not have information that could link it with the WBAN human owner. Still, ID_a has to be confidential to impede $\mathscr{A}$ from tracking the behavior patterns of the user. TLAP provides confidentiality to the clients requesting services by not transmitting their ID_a in plain text in the open channel. ID_a is sent only after hashing it or encrypting it. D_4, D_7, and D_8 contain the hash of ID_a, and D_3 its encryption. D_3 represents the Client a's ephemeral pseudonym, which is usable only in the current session. TLAP avoids tracking attacks by not sending constant values during the authentication phases. Because all messages have ephemeral random numbers, there is no similarity between messages of different sessions. Consequently, $\mathscr{A}$ cannot use them to trace the Client a's activities [158].

The performance of TLAP was evaluated concerning its running time, communication cost, and security features because authentication protocols affect the entire data communication process regarding the response time and processor cycles used for their execution. Protocols' performances are significant in applications that comprise frequent transmissions of sensitive information. Quantifying the protocols' security level, data transmission, and execution time assist in selecting the most appropriate protocol for an application [159].

A 160-bit ECC was used in the running time and communication cost evaluations. Thus, the system can have a security level equivalent to 1024-bit RSA [148] but with a significantly shorter key. Executing a 160-bit ECC does not increment the device duty cycle significantly when the volume of data transmitted and stored and the computation complexity are decreased [19].

In Table 4.9, the security features accomplished by TLAP and similar protocols are compared. TLAP achieves more features than the other protocols. Das et al.'s scheme has many security features. Nonetheless, it has more than doubled execution time and communication cost than TLAP.

Table 4.7 has the execution times of TLAP and related protocols. TLAP has the lowest running time. TLAP needs 57.20% less execution time than Das *et al.*, 25.13% less than Ying *et al.*, 92.93% less than Hsieh *et al.*, 25.13% less than Islam *et al.*, 40.12% less than He *et al.*, and 25.20% less than Kim *et al.* Fig. 4.16 presents the protocols' execution times with the increase in the number of users. The running times increase linearly as the number of users grows. Nonetheless, the increment of TLAP's running time is slower than

the other protocols, which improves the authentications' efficiency.

Figure 4.15 presents the communication cost of TLAP and related protocols. TLAP has the lowest communication cost. TLAP sends 58.25% fewer bytes than Das *et al.*, 28.33% fewer bytes than Ying *et al.*, 81.62% fewer bytes than Hsieh *et al.*, 28.33% fewer bytes than Islam *et al.*, 14% fewer bytes than He *et al.*, and 14% fewer bytes than Kim *et al.* Fig. 4.17 presents the protocols' communication costs with the increase in the number of users. The increment of TLAP's communication cost is slower than the other protocols.

The high level of security and performance achieved by TLAP allows medical professionals and patients to be confident that the health-care information transmitted through the protocol is authentic. It is disclosed only to authorized parties, and the adversary does not maliciously modify it during transmission.

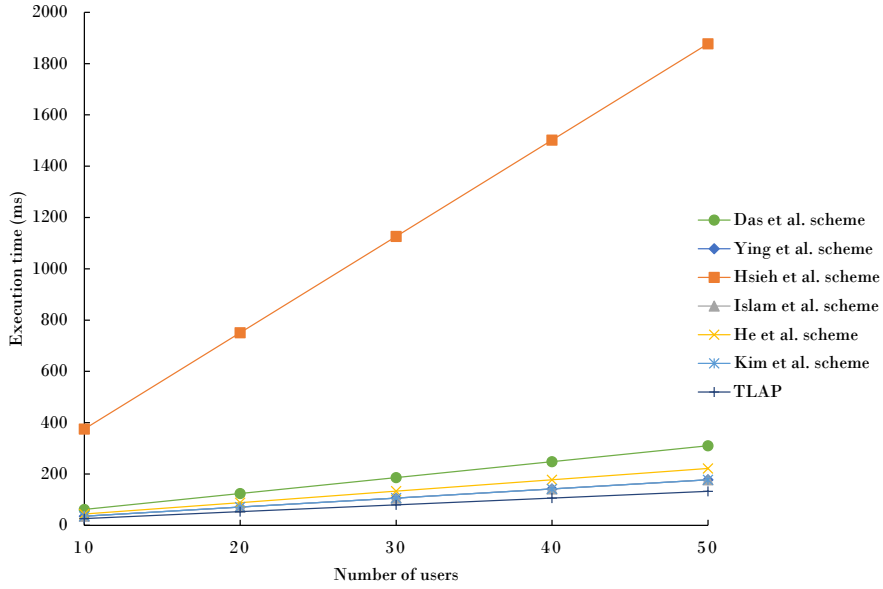


Figure 4.16 Computation overhead of TLAP and similar protocols with regard to the growth in the number of users.

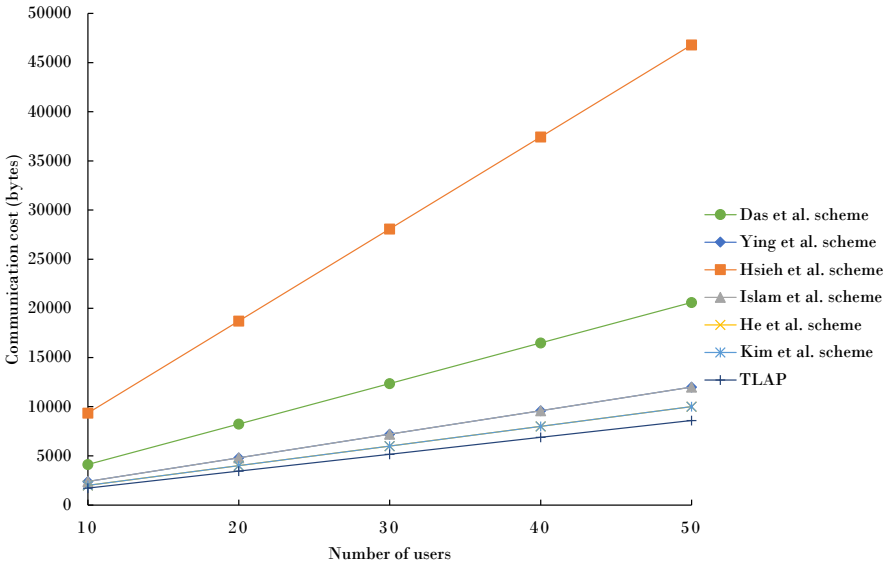


Figure 4.17 Communication overhead of LAKD and similar protocols with regard to the growth in the number of users.

Chapter 5

Integration of the authentication protocols produced by the unified model

The protocols described in the previous chapter can provide authentication to the IoT network architectures presented in [89, 90, 119, 120]. Figure 4.1 and Figure 4.10 illustrate the implementation of the network architectures in the use cases IIoT and WBAN. LAKD protocol can be used to authenticate the communication between the sensor nodes and the gateway or controller. And use TLAP to authenticate the communication between gateways or controllers and APs or web services. Figure 5.1 exemplifies the use of the protocols in IIoT networks and Figure 5.2 in WBANs.

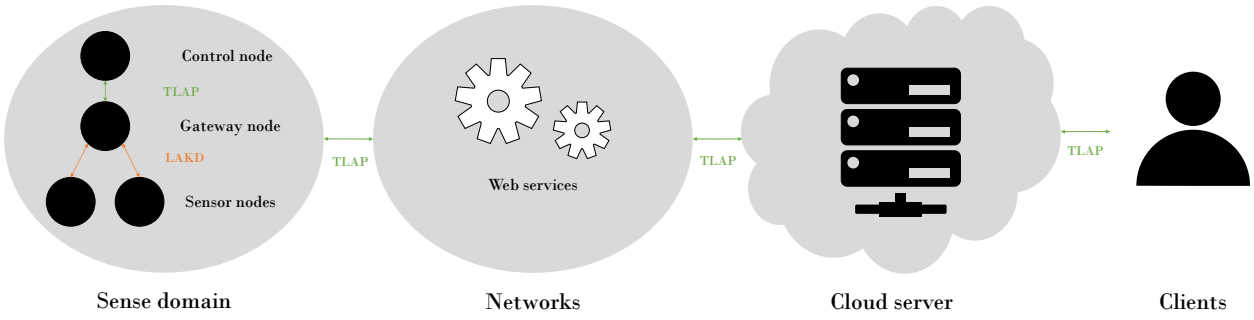


Figure 5.1 Usage of LAKD and TLAP in IIoT networks.

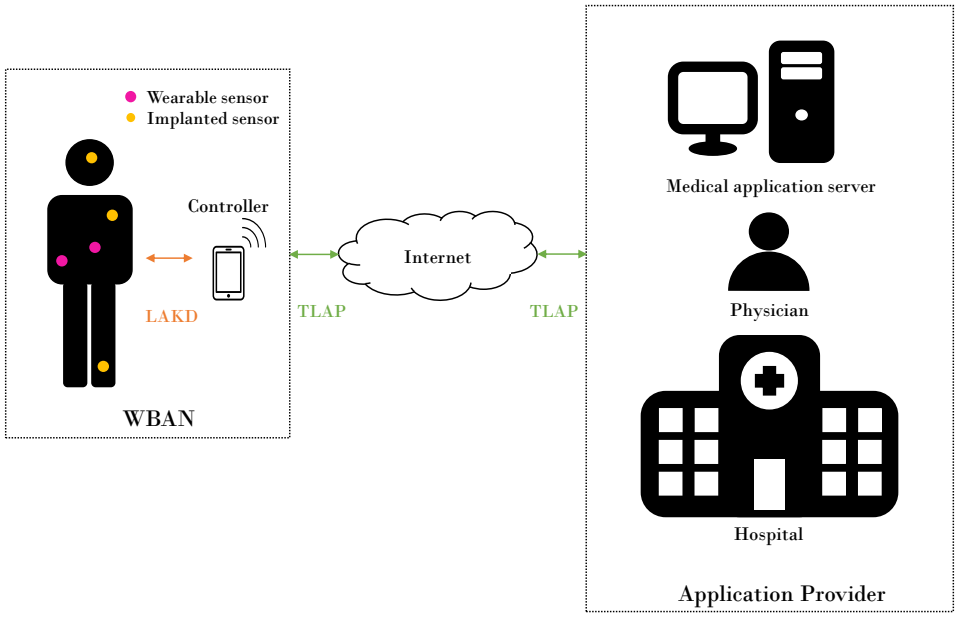


Figure 5.2 Usage of LAKD and TLAP in WBANs.

In this chapter, a use case of the integration of LAKD and TLAP to secure the communication of heterogeneous IoT systems is described.

5.1 Adaptive authentication

Adaptive systems can change their structure and behavior according to their understanding of the environment and system state [160, 161]. Self-adaptation can be applied to many domains because of its effectiveness in managing complexity and designing flexible, configurable, self-optimizing, dependable, and resilient software [160, 162, 163, 164].

Adaptive systems comprehend two aspects, which are the resources to administer and adapt and the adaptation logic. The last-mentioned uses a control structure to achieve the adaptation, where the most well-known is the MAPE-K reference model [165], conceived by IBM in 2005. The name MAPE-K derives from the first letter of the terms of the phases that compose it: Monitor, Analyze, Plan, and Execute. The K represents the Knowledge shared between the steps. Each phase is briefly described following. The monitor phase collects the managed resource's attributes, correlates the information, and determines symptoms that should be handled. The analysis phase studies the data gathered by the previous step and decides if changes are needed. The planning phase receives change requests from the analysis phase and formalizes the actions required to achieve objectives and goals. Finally, the execution phase modifies the managed resource behavior

following the operations selected by the planning phase. Figure 5.3 shows the MAPE-K model with its phases, elements, and feedback loop.

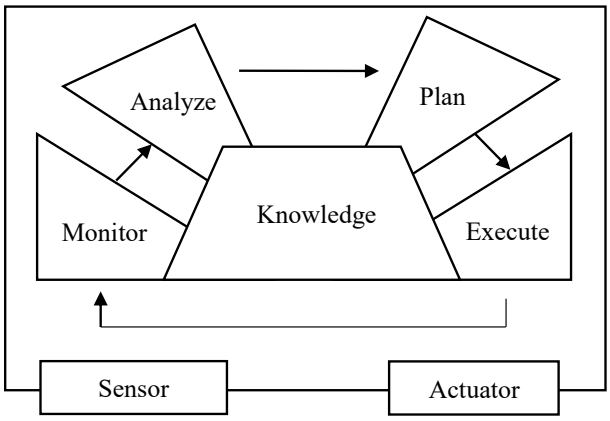


Figure 5.3 MAPE-K reference model for adaptive systems.

In [166], the authors surveyed adaptive security models that use MAPE-K to monitor changes in the system context and environment and adapt the system behavior accordingly to achieve objectives and goals.

Adaptive authentication systems adjust their structure and behavior in response to alterations in their environments automatically [160, 167]. The systems collect user attributes and environmental characteristics and evaluate them in a risk context. The objective is to provide confidence levels that reduce risks when users access sensitive resources. An adaptive authentication system studies the user's behavior patterns based on past login access and system use. If the user keeps the same pattern when logging into and using the system, the authentication process will consider the attempt as low risk. Therefore, it will ask the user for a few simple challenges to prove his/her identity. On the other hand, if the user login under a different context and behavior, the system might require more robust proof of the user identity [168].

In an adaptive authentication system, the authenticators are the resources to administer and adapt. The adaptation logic is the software that selects and uses the authenticators depending on the current context. The system allows diverse types of authenticators to improve flexibility, support applicability to different scenarios, and enhance security and usability [169, 170]. Further, authenticators can be used one time at a particular moment to allow the login to a system, or they can be used in real-time, continuously, and during the session to assure that the legitimate entity is using the service [171]

Authenticators can be a combination of biometric-based, knowledge-based, and token-based mechanisms. Commonly, they are categorized as "something you are," "something you know," and "something you have,"

respectively. Examples of authenticators proposed in the literature are the following. Biometric-based mechanisms can be behavioral as the gait movement, user keystrokes, and mouse patterns. And physiological as iris, face, voice, and fingerprint. Knowledge-based mechanisms can be passwords, PINs, patterns, questions, and one-time passwords (OTPs). Finally, token-based mechanisms can be digital certificates, RFID and NFC tokens, user devices, and USB keys [171].

LAKD and TLAP can be used in adaptive authentication systems to secure the end-to-end communication between the different entities in the IoT environment. Resource-constrained devices can use LAKD in mutual authentication, and devices with more computing capabilities can use TLAP. Figure 5.4 shows an example of the use of these protocols in an adaptive authentication system. The participants of the system and their roles are described below.

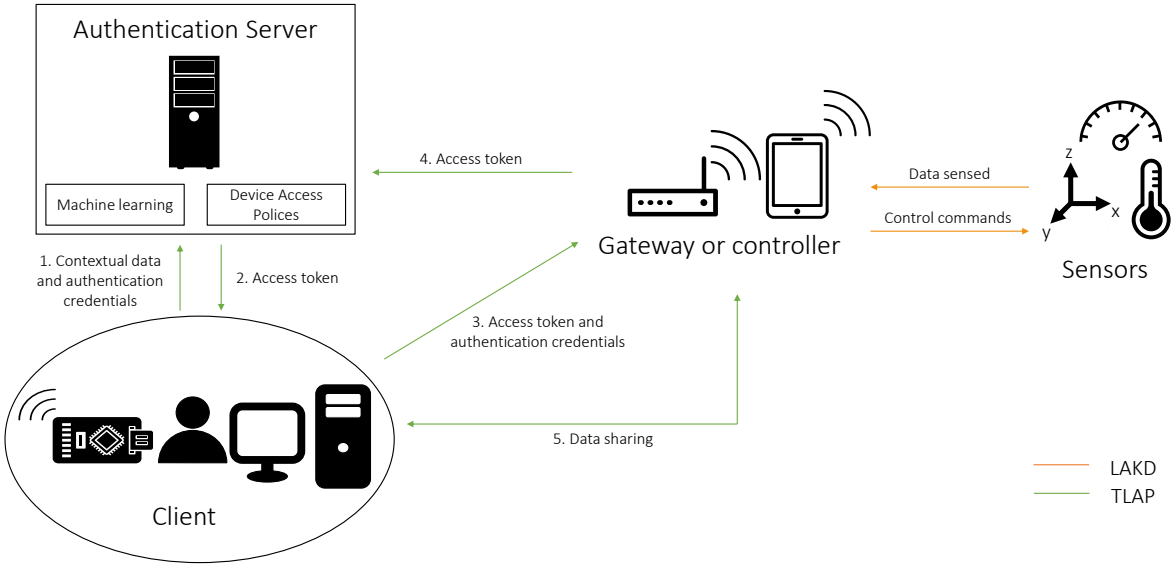


Figure 5.4 Adaptive authentication system.

The client is a user, an AP, or a device that does not have significant computational restrictions. The client wants to access the data perceived by the sensor nodes and perform changes to their state if necessary. The client needs to prove to the gateway its identity to access the sensor nodes' data. The proof of identity comprises two parts: the credentials created during the registration phase of the authentication protocol (private and public key pair) and a token provided by the authentication server. The token demonstrates that the current client's attributes and behaviors and the environmental characteristics coincide with past patterns collected during anterior sessions. Thus, the authentication server considers the recent access attempt as low risk. The client gathers contextual information and sends it to the authentication server to

obtain the token. The latter will provide the token only if the received context data coincide with the client's past context patterns.

The authentication server receives contextual information from a client. It studies the data and determines the risk of allowing the user to access the system. The authentication server uses machine learning algorithms to analyze the context data in terms of the user's past behavioral patterns and environmental characteristics. And it uses policies to determine the consequences of the client accessing the system resources. The policies are behavioral rules associated with specific permissions and usage thresholds. They also contain cost functions that represent the repercussions to the system if a rule is broken [172]. The policies limit clients from accessing the system resources to prevent the disclosure of sensitive information. The authentication server will employ the machine learning algorithms to obtain a coincidence level of the current client's contextual information against its past behavioral patterns. If the policies associated with the requested resource rate the coincidence level as acceptable, the authentication server will provide the client a token that manifests that the risk associated with permitting the user to access system resources is low.

The gateway or controller acts as an intermediary in the communication between clients and sensor nodes. It allows clients to access the sensor nodes' data only if they demonstrate legitimate authentication credentials and an access token. The entity uses TLAP to communicate with the client to verify the client's credentials and receive the client's access token. Then, it uses TLAP to communicate with the authentication server to verify the trustworthiness of the token. Finally, it uses LAKD to communicate with the sensor nodes to obtain their perceived data and state.

The sensor nodes are devices with limited computing resources and a variety of sensors. They are responsible for gathering and sending contextual data to the network gateway or controller.

The contextual information sent by the client contains attributes about itself and its environment. The attributes vary depending on the type of client that requests the access token. If the client is a device, they could be the following: its battery level, surrounding service set identifiers (SSIDs) and their received signal strength indicators (RSSIs), the state of its sensors, its firmware version, the round-trip delay time (RDT), the frequency the device requests access to the sensor nodes, and its system resources such as memory capacity, processor type, and its peripherals [173]. If the client is a user, the attributes could be the following: operating system and browser used to communicate with the authentication server, day and time, location, the applications he/she has used and the percentage of use, websites frequently visited, and biometric data such as fingerprint, veins, and face [170, 174, 175, 176]. The client should obfuscate the last four attributes to protect his/her privacy [177, 178, 179]. He/she can do it through a keyed hash where the key is stored only in the user device. Therefore, the authentication server only knows the hash of the data. The percentages of use do not have to be obfuscated.

The modeling of the client's context comprises the three phases described below. Figure 5.5 shows the process.

1. Initial collection of contextual information to generate a model of the client's attributes and environmental characteristics.
2. Use of a machine-learning algorithm to generate the context model.
3. Adaptability to context changes through replacing old dataset entries with new data of the current client's context.

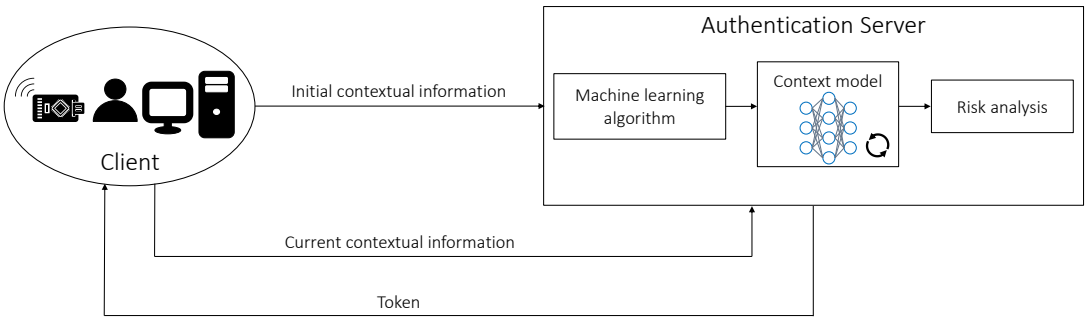


Figure 5.5 Modeling of the client's context.

If the evaluation of the current context is successful, the new data is inserted into the dataset, and old instances are removed gradually. The removal prevents the dataset from growing too large when there are changes in the context, which causes an increase in the training time with the machine learning algorithm. Additionally, it can help to provide to the user the right to be forgotten [180, 181].

5.1.1 Context model testing

A dataset that represents the contextual information of a device was created. It contains environmental and device attributes. The dataset was used to generate a context model that the authentication server uses to calculate a coincidence level of the device's current context against its past patterns. The attributes in these datasets [182, 183, 184, 185, 186, 187, 188, 189] were used to create the contextual information dataset of this work. The attributes included are listed below.

- BLE RSSI (13 attributes).
- WiFi RSSI (7 attributes).

- Accelerometer mean (1 attribute).
- Gyroscope mean (1 attribute).
- Magnetometer magnitude (1 attribute).
- Speed (1 attribute).
- Sound mean (1 attribute).
- Battery voltage in mV (1 attribute).

The authentication server receives many requests from clients to evaluate their contextual information to obtain an access token. The authentication server should not spend much time in the analysis of each petition. Therefore, machine learning algorithms that have fast training and do not use much system memory are preferred. The algorithms decision forest, logistic regression, and averaged perceptron were tested. The Weka tool [190] was employed to build the context model from the dataset using the mentioned algorithms. All of them were able to classify the instances correctly. Due to the logistic regression algorithm is the most computationally simple, the authors consider it the most appropriate. The tests performed on the dataset with the machine learning algorithms are described below.

1. A dataset with 105 instances and four classes was created. Each class represents a different device. It was tested if the algorithms can classify the four classes correctly. The dataset was divided into 70% for training data and 30% for the test.
2. A training dataset with 47 instances and two classes and a test dataset with one instance were created. This was done for each of the four classes. It was tested if the algorithms can classify the samples correctly.
3. A training dataset with 47 instances and two classes and a test dataset with one instance were created. This was done two times; the first time, the instance belonged to the training dataset. The second time the sample did not belong, such that it represented a malicious entity. It was tested if the algorithms can classify the instances correctly.

Figure 5.6 and Figure 5.7 present the result of the first test using logistic regression in Weka. It can be seen that the algorithm was able to classify the four classes correctly. In the adaptive authentication system, it means that the model was able to differentiate the four devices.

```
=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                   Class
Variable                               1                   2                   3
================================================================================
BLE_B2                           -0.0468             -0.0945              0.0463
BLE_B3                           -0.0319             -0.0157             -0.0211
BLE_B4                           -0.0088              0.1127             -0.0694
BLE_B6                            0.0319             -0.0204             -0.0386
WiFi_AP1                         -0.0858              0.4246              0.9361
WiFi_AP2                         -1.7001             -1.2611             -0.2605
WiFi_AP3                         -1.2056             -0.6762              0.0966
WiFi_AP4                         -0.0585              0.3172              1.1711
WiFi_AP5                         -1.2482              -0.233             -1.1596
WiFi_AP6                           0.555              0.6074              0.4622
WiFi_AP7                          0.3618              0.4358             -0.1804
Accelerometer_mean                1.2043              4.5982              7.2642
Gyroscope_mean                    2.3562             -3.2839             16.2269
Sound_mean                       -0.0092              0.0587             -0.0998
Speed_mean                       -0.6722             -0.0431             -1.3651
Vbatt_mVolts                      0.0801              0.0476             -0.0249
Magnetometer_Magnitude         -153.7916            -96.4096             27.4034
Intercept                      -418.1304           -175.4041             53.6963
```

Figure 5.6 Context modeling using logistic regression in the Weka tool.

```
=== Summary ===

Correctly Classified Instances          31               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0004
Root mean squared error                  0.0029
Relative absolute error                  0.1    %
Root relative squared error              0.6607 %
Total Number of Instances               31

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     1
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     2
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     3
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     4
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
  6  0  0  0 |  a = 1
  0 10  0  0 |  b = 2
  0  0  7  0 |  c = 3
  0  0  0  8 |  d = 4
```

Figure 5.7 Accuracy of the device classification.

Chapter 6

Conclusions and future work

IoT technology has a great potential to manage social issues, e.g., medical and environmental monitoring, machine failure detection, and emergency notification systems. Nevertheless, the variety of use cases involves that IoT networks are composed of heterogeneous devices, network technologies, protocols, and communication models. The diversity reduces the devices' interoperability. Moreover, the connection of physical objects to the Internet causes security and privacy problems. Some of the issues are the exposure of users' private information and security attacks on IoT devices that could disrupt their correct functionality. The consequences could be financial losses and even lives at risk.

As a solution to the security problems on heterogeneous IoT networks, it is proposed a unified model of authentication protocols. The model defines the phases needed to achieve mutual authentication. Following there are determined the cryptographic functions that compose each phase according to the computing capabilities of the devices that will implement the authentication protocol. The model can provide different protocols. Each one is configured with cryptographic procedures that efficiently use the computing resources in the IoT devices.

Currently, two protocols have been generated from the unified model. The first, called LAKD, is designed for devices in IIoT applications that are very limited in computing resources. LAKD uses only symmetric-key cryptographic functions. The second, named TLAP, is intended for WBAN devices in health-care applications. TLAP uses self-certified public keys based on ECC, scalar point multiplication, and symmetric-key cryptographic functions. Both protocols have been published in indexed journals.

The protocols have to prove the achievement of the security attributes required in the IoT environment where they will execute. Formal and informal methods were used to demonstrate their security. They concluded that the protocols achieved mutual authentication and satisfied the security requirements.

The protocols' performances were analyzed in terms of the processor's time they use to execute and the number of bytes the parties transmit in the network to authenticate. Both protocols accomplished an appropriate execution and communication cost for IoT. The high performance is significant because it decreases the energy consumption in the IoT network. Due to some IoT devices are difficult to access because they are in remote locations or implanted in the human body, their batteries are difficult to replace or recharge. Thus, the protocols they implement must have an appropriate energy consumption; so the

devices' battery life prolong.

The security and performance of the protocols were compared with current proposals that have a similar architecture to them. It was concluded that the protocols in this work achieved more security features. And have a low execution and communication cost, suitable for IoT.

Since the protocols were designed for devices with different computing resources, they can be used to secure the communication of heterogeneous IoT systems. A use case of the integration of LAKD and TLAP in an adaptive authentication system was described. The latter analyzes behavior patterns according to past login attempts. If the system detects the same pattern, the authentication process considers the attempt as low risk. Thus, it will ask the client for a few simple challenges to prove its identity. On the contrary, if the pattern does not match, the system might require robust proof of the identity. Adaptive authentication systems allow various authenticator types to enhance security, improve flexibility and usability, and support applicability to different scenarios.

Machine learning algorithms were used for modeling and evaluating the client context. The contextual information contains attributes about the client and its environment. If the current context coincides with its past patterns, the authentication server will give it a token that allows it to access the system's resources. LAKD and TLAP can be used in adaptive authentication systems to secure the end-to-end communication between the different entities in the IoT environment. Resource-constrained devices can use LAKD, and devices with more computing capabilities can use TLAP.

In future work, the authors will investigate efficient methods to perform context modeling and evaluation in resource-constrained devices. In this manner, the client's behavioral patterns and environmental information are not transmitted to an authentication server; thus, the privacy of the client's data is preserved. Additionally, the adaptive authentication process might respond faster because it will not need to wait for a response from the authentication server.

References

[1] Madakam, S.; Ramaswamy, R.; Tripathi, S. Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications, vol. 3 no. 5*, **2015**, pp. 164–173. doi: 10.4236/jcc.2015.35021.

[2] Li, S.; Da Xu, L.; Zhao, S. The internet of things: a survey. *Information Systems Frontiers col. 17 no. 2*, **2015**, pp. 243–259, doi:10.1007/s10796-014-9492-7.

[3] Khan, M. A.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems, vol. 82*, **2018**, pp. 395–411, doi:10.1016/j.future.2017.11.022.

[4] Tawalbeh, L. A.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT Privacy and security: Challenges and solutions. *Applied Sciences, vol. 10 no. 12*, **2020**, doi:10.3390/app10124102.

[5] Van den Abeele, F.; Hoebeke, J.; Moerman, I.; Demeester, P. Integration of heterogeneous devices and communication models via the cloud in the constrained internet of things. *International Journal of Distributed Sensor Networks, vol. 11 no. 10*, **2015**, doi: 10.1155/2015/683425.

[6] Oh, S.; Kim, Y., Security Requirements Analysis for the IoT, *2017 International Conference on Platform Technology and Service (PlatCon)*, **2017**, pp. 1–6, doi: 10.1109/PlatCon.2017.7883727.

[7] Miorandi, D.; Sicari, S.; Pellegrini, F.D.; Chlamtac, I., Internet of things: Vision, applications and research challenges. *IEEE Ad. Hoc. Netw.*, **2012**, 1497–1516, doi:10.1016/j.adhoc.2012.02.016.

[8] Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D., Context Aware Computing for The Internet of Things: A Survey. *IEEE Commun. Surv. Tutor.*, **2014**, 414–454, doi:10.1109/SURV.2013.042313.00197.

[9] Borgia, E., The Internet of Things vision: Key features, applications and open issues. *Comput. Commun.*, **2014**, pp. 1–31, doi:10.1016/j.comcom.2014.09.008.

[10] Khan, R.; Khan, S.U.; Zaheer, R.; Khan, S., Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In Proceedings of the 2012 10th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, **17–19 December 2012**, pp. 257–260, doi:10.1109/FIT.2012.53.

[11] Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I., Internet of things IoT security: Current status, challenges and prospective measures. In Proceedings of the 2015 10th International Conference for

Internet Technology and Secured Transactions ICITST, London, UK, **14–16 December 2015**, pp. 336–341.s

[12] Halperin, D.; Heydt-Benjamin, T.S.; Ransford, B.; Clark, S.S.; Defend, B.; Morgan, W.; Fu, K.; Kohno, T.; Maisel, W.H., Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, **18–22 May 2008**, pp. 129–142, doi:10.1109/SP.2008.31.

[13] Arias, O.; Wurm, J.; Hoang, K.; Jin, Y., Privacy and Security in Internet of Things and Wearable Devices. *IEEE Trans. Multi Scale Comput. Syst.*, **2015**, pp. 99–109, doi:10.1109/TMSCS.2015.2498605.

[14] Papp, D.; Ma, Z.; Buttyan, L., Embedded systems security: Threats, vulnerabilities, and attack taxonomy. *Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey*, **July 2015**, pp. 145–152, doi:10.1109/PST.2015.7232966.

[15] Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W., A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal vol. 4, no. 5*, **2017**, pp. 1125–1142, doi:10.1109/JIOT.2017.2683200.

[16] Bormann, C.; Ersue, M.; Keranen, A. RFC 7228-Terminology for Constrained-Node Networks. *Internet Engineering Task Force*, **2014**. Available online: `http://www.ietf.org/rfc/rfc7228.txt` (accessed on 16 December 2020).

[17] Zhang, Z.; Cho, M. C. Y.; Wang, C.; Hsu, C.; Chen, C.; Shieh, S., IoT Security: Ongoing Challenges and Research Opportunities. *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, Matsue*, **2014**, pp. 230–234, doi:10.1109/SOCA.2014.58.

[18] Ali, I.; Sabir, S.; Ullah, Z. Internet of Things Security, Device Authentication and Access Control: A Review. *International Journal of Computer Science and Information Security (IJCSIS), vol. 14, no 8*, **2016**.

[19] Piotrowski, K.; Langendoerfer, P.; Peter, S., How public key cryptography influences wireless sensor node lifetime. *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, **2006**, pp. 169–176.

[20] Saraiva, D.A.F.; Leithardt, V.R.Q.; de Paula, D.; Sales Mendes, A.; González, G.V.; Crocker, P. PRISEC: Comparison of Symmetric Key Algorithms for IoT Devices. *Sensors*, **2019**, doi:10.3390/s19194312.

[21] Manikandan, G. ; Sakthi, U., A Comprehensive Survey on Various key Management Schemes in WSN, *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on, Palladam, India*, **2018**, pp. 378–383, doi: 10.1109/I-SMAC.2018.8653656.

[22] Masdari, M.; Ahmadzadeh, S., A survey and taxonomy of the authentication schemes in Telecare Medicine Information Systems. *Journal of Network and Computer Applications*, **2017**, pp. 1–19.

[23] Menezes, A. J.; Katz, J.; Van Oorschot, P. C., Vanstone, S.A., Handbook of applied cryptography. *CRC Press: 1010 Boca Raton, FL, USA,*, **1996**, pp. 385–424, ISBN 0-8493-8523-7.

[24] Alzahrani, B.A.; Chaudhry, S. A.; Barnawi, A.; Al-Barakati, A.; Shon, T., An Anonymous Device to Device Authentication Protocol Using ECC and Self Certified Public Keys Usable in Internet of Things Based Autonomous Devices, *Electronics vol. 9*, **2020**, doi: 10.3390/electronics9030520.

[25] Banerjee, S.; Odelu, V.; Das, A. K.; Chattopadhyay, S.; Rodrigues, J. J.; Park, Y. Physically secure lightweight anonymous user authentication protocol for internet of things using physically unclonable functions. *IEEE Access*, **2019**, pp. 85627–85644.

[26] Islam, S.H.; Biswas, G.P, Design of Two-Party Authenticated Key Agreement Protocol Based on ECC and Self-Certified Public Keys, *Wireless Pers Commun 82*, **2015**, pp. 2727–2750, doi: 10.1007/s11277-015-2375-5.

[27] Gupta, S.; Kumar, A.; Kumar, N., Design of ECC based authenticated group key agreement protocol using self-certified public keys, *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, **2018**, pp. 1–5, doi: 10.1109/RAIT.2018.8388999.

[28] Vigano, L., Automated Security Protocol Analysis With the AVISPA Tool. *Electronic Notes in Theoretical Computer Science* **2006**, 61–86, doi:10.1016/j.entcs.2005.11.052.

[29] Chevalier, Y.; Compagna, L.; Cuellar, J.; Hankes Drielsma, P.; Mantovani, J.; Moedersheim, S.; Vigneron, L., A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols. **2004**, *Workshop on Specification and Automated Processing of Security Requirements - SAPS'2004* Available online: *https://hal.inria.fr/inria-00099882/* (accessed on 12 May 2021).

[30] Genet, T., A short SPAN+AVISPA tutorial. **2015**, Available online: *https://hal.inria.fr/hal-01213074* (accessed on 12 May 2021).

[31] Dolev, D.; Yao, A., On the security of public key protocols. *IEEE Transactions on Information Theory* **1983**, 198–208, doi:10.1109/TIT.1983.1056650.

[32] Cervesato, I.; Durgin, N. A.; Lincoln, P. D.; Mitchell, J. C.; Scedrov, A., A Meta-Notation for Protocol Analysis. *Proceedings of the 12th IEEE Workshop on Computer Security Foundations* **1999**, 55–69, doi: 10.1109/CSFW.1999.779762.

[33] Wazid, M.; Das, A. K.; Odelu, V.; Kumar, N.; Conti, M.; Jo, M., Design of Secure User Authenticated Key Management Protocol for Generic IoT Networks. *IEEE Internet of Things Journal* **2018**, 269–282, doi:10.1109/JIOT.2017.2780232.

[34] Armando, A. et al., AVISPA: Automated Validation of Internet Security Protocols and Applications. *Future and Emerging Technologies (FET Open)*, Available online: *http://www.avispa-project.org* (accessed on 12 May 2021).

[35] Burrows, M.; Abadi, M.; Needham, R.M., A logic of authentication. *Proc. R. Soc. Lond. A* **1989**, 233—271, doi:10.1098/rspa.1989.0125.

[36] Turkanović, M.; Brumen, B.; Hölbl, M., 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* **2014**, 96–112, doi:10.1016/j.adhoc.2014.03.009.

[37] Amin, R.; Biswas, G.P., A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Networks* **2016**, 58–80, doi:10.1016/j.adhoc.2015.05.020.

[38] Wu, F.; Xu, L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.*, **2017**, pp. 72–85, doi:10.1016/j.jnca.2016.12.008.

[39] Tai, W. L.; Chang, Y. F.; Li, W. H., An IoT notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks. *Journal of Information Security and Applications* **2017**, 133–141, doi:10.1016/j.jisa.2017.04.002.

[40] Eldefrawy, M. H.; Ferrari, N.; Gidlund, M., Dynamic User Authentication Protocol for Industrial IoT without Timestamping. *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)* **2019**, 1–7, doi:10.1109/WFCS.2019.8757984.

[41] Aghili, S.F.; Mala, H. Breaking a Lightweight M2M Authentication Protocol for Communications in IIoT Environment. *IACR Cryptol. ePrint Arch.* **2018**, *891*.

[42] Chang, C.; Le, H., A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks. *IEEE Transactions on Wireless Communications* **2016**, 357–366, doi:10.1109/TWC.2015.2473165.

[43] Li, X.; Peng, J.; Niu, J.; Wu, F.; Liao, J.; Choo, K. R., A Robust and Energy Efficient Authentication Protocol for Industrial Internet of Things. *IEEE Internet of Things Journal* **2018**, 1606–1615, doi:10.1109/JIOT.2017.2787800.

[44] Hummen, R.; Shafagh, H.; Raza, S.; Voig, T.; Wehrle, K. Delegation-based authentication and authorization for the IP-based Internet of Things. 2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Singapore, 30 June–3 July 2014; pp. 284–292, doi:10.1109/SAHCN.2014.6990364.

[45] Suárez-Albela, M.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. A Practical Performance Comparison of ECC and RSA for Resource-Constrained IoT Devices. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; pp. 1–6, doi:10.1109/GIOTS.2018.8534575.

[46] Kolluru, K. K.; Paniagua, C.; van Deventer, J.; Eliasson, J.; Delsing, J.; DeLong, R. J., An AAA solution for securing industrial IoT devices using next generation access control. *2018 IEEE Industrial Cyber-Physical Systems (ICPS)* **2018**, 737–742, doi:10.1109/ICPHYS.2018.8390799.

[47] Ferraiolo, D.; Chandramouli, R.; Kuhn, R.; Hu, V., Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC). *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control* **2016**, 13–24, doi:10.1145/2875491.2875496.

[48] Zhang, Y.; Deng, R. H.; Zheng, D.; Li, J.; Wu, P.; Cao, J., Efficient and Robust Certificateless Signature for Data Crowdsensing in Cloud-Assisted Industrial IoT. *IEEE Transactions on Industrial Informatics* **2019**, 5099–5108, doi:10.1109/TII.2019.2894108.

[49] Lin, H., A secure heterogeneous mobile authentication and key agreement scheme for e-healthcare cloud systems. *PLOS ONE, vol. 13* **2018**, doi:10.1371/journal.pone.0208397.

[50] Asghar, M.; Doss, R.R.M.; Pan, L., A Scalable and Efficient PKI Based Authentication Protocol for VANETs. *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)* **2018**, 1–3, Sydney, NSW, doi:10.1109/ATNAC.2018.8615224.

[51] Sangari, A.S.; Manickam, J.M.L., Public key cryptosystem based security in wireless body area network. *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]* **2014**, 1609–1612, doi:10.1109/ICCPCT.2014.7054788.

[52] Wang, C.; Zhang, Y., New Authentication Scheme for Wireless Body Area Networks Using the Bilinear Pairing. *J Med Syst, vol. 39, no. 11* **2015**, 136, doi:10.1007/s10916-015-0331-2.

[53] Liu, J.; Zhang, Z.; Chen, X.; Kwak, K.S., Certificateless Remote Anonymous Authentication Schemes for Wireless Body Area Networks. *IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2* **2014**, 332–342, doi:10.1109/TPDS.2013.145.

[54] Xiong, H.; Qin, Z., Revocable and Scalable Certificateless Remote Authentication Protocol With Anonymity for Wireless Body Area Networks. *IEEE Transactions on Information Forensics and Security, vol. 10, no. 7* **2015**, 1442–1455, doi:10.1109/TIFS.2015.2414399.

[55] Liu, J.; Zhang, L.; Sun, R., 1-RAAP: An Efficient 1-Round Anonymous Authentication Protocol for Wireless Body Area Networks. *Sensors, vol. 16, no. 5* **2016**, doi:10.3390/s16050728.

[56] Zhao, Z., An Efficient Anonymous Authentication Scheme for Wireless Body Area Networks Using Elliptic Curve Cryptosystem. *J Med Syst, vol. 38, no. 2* **2014**, 13, doi:10.1007/s10916-014-0013-5.

[57] Wu, L.; Zhang, Y.; Li, L.; Shen, J., Efficient and Anonymous Authentication Scheme for Wireless Body Area Networks. *J Med Syst, vol. 40, no. 6* **2016**, 134, doi:10.1007/s10916-016-0491-8.

[58] Omala, A.A.; Kibiwott, K.P.; Li, F., An Efficient Remote Authentication Scheme for Wireless Body Area Network. *J Med Syst, vol. 41, no. 2* **2017**, 25, doi:10.1007/s10916-016-0670-7.

[59] Chen, R.; Peng, D., Analysis and Improvement of a Mutual Authentication Scheme for Wireless Body Area Networks. *J Med Syst, vol. 43, no. 2* **2019**, 19, doi:10.1007/s10916-018-1129-9.

[60] Li, X.; Peng, J.; Kumari, S.; Wu, F.; Karuppiah, M.; Choo, K.K.R., An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. *Computers & Electrical Engineering, vol. 61* **2017**, 238–249, doi:10.1016/j.compeleceng.2017.02.011.

[61] Izza, S.; Benssalah, M.; Ouchikh, R., Security Improvement of the Enhanced 1-round Authentication Protocol for Wireless Body Area Networks. *2018 International Conference on Applied Smart Systems (ICASS)* **2018**, 1–6, doi:10.1109/ICASS.2018.8652036.

[62] Paverd, A. J.; Martin, A.; Brown, I. Modelling and automatically analysing privacy properties for honest-but-curious adversaries, *Univ. Oxford Tech. Rep.*, **2014**.

[63] Ren, Y.; Zhu, F.; Qi, J.; Wang, J.; Sangaiah, A. K. Identity management and access control based on blockchain under edge computing for the industrial internet of things. *Applied Sciences, vol. 9 no. 10*, **2019**, doi:10.3390/app9102058.

[64] Goldreich, O.; Lindell, Y. Session-key generation using human passwords only. *Annual International Cryptology Conference*, **2001**, pp. 408–432, Springer, Berlin, Heidelberg, doi: 10.1007/3-540-44647-8_24.

[65] Ledwaba, L. P.; Hancke, G. P.; Venter, H. S.; Isaac, S. J. Performance costs of software cryptography in securing new-generation Internet of energy endpoint devices. *IEEE Access, vol. 6*, **2018**, pp. 9303–9323, doi: 10.1109/ACCESS.2018.2793301.

[66] Arm Ltd. Cortex-M. *Arm Developer*. Available online: `https://developer.arm.com/ip-products/processors/cortex-m` (accessed on 17 February 2021).

[67] Standard, NIST-FIPS. Announcing the advanced encryption standard (AES). *Federal Information Processing Standards Publication, vol. 197 no. 1-51*, **2001**, pp. 3–3.

[68] FIPS-PUB, N. I. S. T. Secure Hash Standard (SHS). *Federal Information Processing Standards Publication*. Available online: `https://csrc.nist.gov/publications/detail/fips/180/4/final` (accessed on 3 March 2021).

[69] Kerry, C. F.; Gallagher, P. D. Digital signature standard (DSS). *FIPS PUB*, **2013**, pp. 186-4, doi:10.6028/NIST.FIPS.186-4.

[70] STMicroelectronics. X-CUBE-CRYPTOLIB. Available online: `https://www.st.com/en/embedded-software/x-cube-cryptolib.html` (accessed on 17 February 2021).

[71] S. Keller. Cryptographic Algorithm Validation Program. *NIST*, **2018**. Available online: `https://www.nist.gov/programs-projects/cryptographic-algorithm-validation-program` (accessed on 17 February 2021).

[72] Lara, E.; Aguilar, L.; Sanchez, M. A.; García, J. A. Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial Internet of Things. *Sensors, vol. 20 no. 2*, **2020**, doi 10.3390/s20020501.

[73] Sensors MDPI. Sensors. **2021**. Available online: `https://www.mdpi.com/journal/sensors` (accessed on 19 February 2021).

[74] Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics, vol. 14 no. 11*, **2018**, pp. 4724–4734, doi:10.1109/TII.2018.2852491.

[75] Serpanos, D.; Wolf, M. Industrial internet of things. *In Internet-of-Things (IoT) Systems*, **2018**, pp. 37–54. Springer, Cham, doi:10.1007/978-3-319-69715-4_5.

[76] Zhou, L.; Yeh, K. H.; Hancke, G.; Liu, Z.; Su, C. Security and privacy for the industrial internet of things: An overview of approaches to safeguarding endpoints. *IEEE Signal Processing Magazine, vol. 35 no. 5*, **2018**, pp. 76–87, doi:10.1109/MSP.2018.2846297.

[77] Lara, E.; Aguilar, L.; García, J. A., Lightweight Authentication Protocol Using Self-Certified Public Keys for Wireless Body Area Networks in Health-Care Applications. *IEEE Access, vol. 9*, **2021**, pp. 79196–79213, doi:10.1109/ACCESS.2021.3084135.

[78] IEEE Access. IEEE Access: The Multidisciplinary Open Access Journal. *IEEE Access*. **November 20 2020**. Available online: `https://ieeeaccess.ieee.org/` (accessed on 19 February 2021).

[79] Ghamari, M.; Janko, B.; Sherratt, R. S.; Harwin, W.; Piechockic, R.; Soltanpur, C. A survey on wireless body area networks for ehealthcare systems in residential environments. *Sensors, vol. 16 no. 6*, **2016**, doi:10.3390/s16060831.

[80] Olatinwo, D. D.; Abu-Mahfouz, A.; Hancke, G. A survey on LPWAN technologies in WBAN for remote health-care monitoring. *Sensors, vol. 19 no. 23*, **2019**, doi:10.3390/s19235268.

[81] Mucchi, L.; Jayousi, S.; Martinelli, A.; Caputo, S.; Marcocci, P. An overview of security threats, solutions and challenges in wbans for healthcare. *In 2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, **2019**, pp. 1–6, doi:10.1109/ISMICT.2019.8743798.

[82] Hsieh, W. B.; Leu, J. S. An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures. *The Journal of Supercomputing, vol. 70 no. 1*, **2014**, pp. 133–148, doi:10.1007/s11227-014-1135-8.

[83] Xu, H.; Yu, W.; Griffith, D.; Golmie, N. A survey on industrial Internet of Things: A cyber-physical systems perspective. *IEEE Access, vol. 6*, **2018**, pp. 78238–78259, doi:10.1109/ACCESS.2018.2884906.

[84] Boyes, H.; Hallaq, B.; Cunningham, J.; Watson, T. The industrial internet of things (IIoT): An analysis framework. *Computers in industry, vol 101*, **2018**, pp. 1–12, doi:10.1016/j.compind.2018.04.015.

[85] Khan, W. Z.; Rehman, M. H.; Zangoti, H. M.; Afzal, M. K.; Armi, N.; Salah, K. Industrial internet of things: Recent advances, enabling technologies and open challenges. *Computers & Electrical Engineering, vol. 81*, **2020**, doi:10.1016/j.compeleceng.2019.106522.

[86] Chowdhury, A.; A Raut, S. Benefits, Challenges, and Opportunities in Adoption of Industrial IoT. *International Journal of Computational Intelligence & IoT, vol. 2 no. 4*, **2019**.

[87] Sadeghi, A.; Wachsmann, C.; Waidner, M., Security and privacy challenges in industrial Internet of Things. *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, **2015**, pp. 1–6, doi:10.1145/2744769.2747942.

[88] Forsström, S.; Butun, I.; Eldefrawy, M.; Jennehag, U.; Gidlund, M., Challenges of Securing the Industrial Internet of Things Value Chain. *2018 Workshop on Metrology for Industry 4.0 and IoT*, **2018**, pp. 218–223, doi:10.1109/METROI4.2018.8428344.

[89] Wang, K.; Wang, Y.; Sun, Y.; Guo, S.; Wu, J., Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective. *IEEE Communications Magazine*, **2016**, pp. 48–54, doi: 10.1109/MCOM.2016.1600399CM

[90] Abedin, S. F.; Alam, M. G. R.; Haw, R.; Hong, C. S., A system model for energy efficient green-IoT network. *2015 International Conference on Information Networking (ICOIN)*, **2015**, pp. 177–182, doi: 10.1109/ICOIN.2015.7057878

[91] Paliwal, S. Hash-based conditional privacy preserving authentication and key exchange protocol suitable for industrial internet of things. *IEEE Access, vol. 7*, **2019**, pp. 136073–136093, doi:10.1109/ACCESS.2019.2941701.

[92] Turuani, M. The CL-Atse Protocol Analyser. *Term Rewriting Appl.*, **2006**, pp. 277–286, doi:10.1007/11805618_21.

[93] Martínez-Peláez, R.; Toral-Cruz, H.; Parra-Michel, J.R.; García, V.; Mena, L.J.; Félix, V.G.; Ochoa-Brust, A. An Enhanced Lightweight IoT-based Authentication Scheme in Cloud Computing Circumstances. *Sensors*, **2019**, doi:10.3390/s19092098.

[94] Esfahani, A.; Mantas, G.; Matischek, R.; Saghezchi, F.B.; Rodriguez, J.; Bicaku, A.; Maksuti, S.; Tauber, M.G.; Schmittner, C.; Bastos, J. A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment. *IEEE Internet Things J.*, **2019**, pp. 288–296, doi:10.1109/JIOT.2017.2737630.

[95] Han, J. Kim, J. A lightweight authentication mechanism between IoT devices. *In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea, **2017**, pp. 1153–1155, doi:10.1109/ICTC.2017.8190883.

[96] Qiu, Y.; Ma, M. An authentication and key establishment scheme to enhance security for M2M in 6LoWPANs. *In Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW)*, **2015**, pp. 2671–2676, doi:10.1109/ICCW.2015.7247582.

[97] Renuka, K.; Kumari, S.; Zhao, D.; Li, L. Design of a Secure Password-Based Authentication Scheme for M2M Networks in IoT Enabled Cyber-Physical Systems. *IEEE Access*, **2019**, pp. 51014–51027, doi:10.1109/ACCESS.2019.2908499.

[98] Joshitta, R.S.M.; Arockiam, L. Device authentication mechanism for IoT enabled healthcare system. *In Proceedings of the 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, **2017**; pp. 1–6, doi:10.1109/ICAMMAET.2017.8186646.

[99] Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Gener. Comput. Syst.*, **2019**, pp. 244–251, doi:10.1016/j.future.2018.08.038.

[100] Wazid, M.; Das, A.K.; Shetty, S.; JPC Rodrigues, J.; Park, Y. LDAKM-EIoT: Lightweight Device Authentication and Key Management Mechanism for Edge-Based IoT Deployment. *Sensors*, **2019**, *19*, doi:10.3390/s19245539.

[101] Taher, B. H.; Jiang, S.; Yassin, A. A.; Lu, H. Low-Overhead Remote User Authentication Protocol for IoT Based on a Fuzzy Extractor and Feature Extraction. *IEEE Access*, **2019**, pp. 148950–148966, doi:10.1109/ACCESS.2019.2946400.

[102] Adeel, A.; Ali, M.; Khan, A.N.; Khalid, T.; Rehman, F.; Jararweh, Y.; Shuja, J. A multi-attack resilient lightweight IoT authentication scheme. *Trans. Emerg. Tel. Tech.*, **2019**, doi:10.1002/ett.3676.

[103] Limbasiya, T.; Soni, M.; Mishra, S. K. Advanced formal authentication protocol using smart cards for network applicants. *Comput. Electr. Eng.*, **2018**, pp. 50–63, doi:10.1016/j.compeleceng.2017.12.045.

[104] Malekzadeh, M.; Ghani, A. A. A.; Desa, J.; Subramaniam, S. An experimental evaluation of DoS attack and its impact on throughput of IEEE 802.11 wireless networks. *IJCSNS, vol. 8 no. 8*, **2008**, p. 1.

[105] Safkhani, M.; Bagheri, N.; Naderi, M.; Luo, Y.; Chai, Q. Tag impersonation attack on two RFID mutual authentication protocols. *2011 Sixth International Conference on Availability, Reliability and Security*, **2011**, pp. 581–584, doi:10.1109/ARES.2011.87.

[106] Song, B.; Mitchell, C. J. RFID authentication protocol for low-cost tags. *Proceedings of the first ACM conference on Wireless network security*, **2008**, pp. 140–147, doi:10.1145/1352533.1352556.

[107] Phan, R. C. W. Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI. *IEEE Transactions on Dependable and secure Computing, vol. 6 no. 4*, **2008**, pp. 316–320, doi:10.1109/TDSC.2008.33.

[108] Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl. vol. 8 no. 6*; **2015**, pp. 1070–1081, doi:10.1007/s12083-014-0285-z.

[109] Khan, M. K.; Alghathbar, K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors, vol. 10 no. 3*, **2010**, pp. 2450–2459, doi:10.3390/s100302450.

[110] Meng, G.; Futai, Z. Key-compromise impersonation attacks on some certificateless key agreement protocols and two improved protocols. *2009 First International Workshop on Education Technology and Computer Science, vol. 2*, **2009**, pp. 62–66, doi:10.1109/ETCS.2009.276.

[111] Sarvabhatla, M.; Reddy, M. C. M.; Vorugunti, C. S. A robust remote user authentication scheme resistant to known session specific temporary information attack. *2015 Applications and Innovations in Mobile Computing (AIMoC)*, **2015**, pp. 164–169, doi:10.1109/AIMOC.2015.7083847.

[112] Kuo, W. C.; Lee, Y. C. Attack and improvement on the one-time password authentication protocol against theft attacks. *2007 International Conference on Machine Learning and Cybernetics vol. 4*, **2007**, pp. 1918–1922, doi: 10.1109/ICMLC.2007.4370461.

[113] Guha, R. K.; Furqan, Z.; Muhammad, S. Discovering man-in-the-middle attacks in authentication protocols. *MILCOM 2007-IEEE Military Communications Conference*, **2007**, pp. 1–7, doi:10.1109/MILCOM.2007.4455039.

[114] Juang, W.-S.; Lei, C.-L.; Chang, C.-Y. Anonymous channel and authentication in wireless communications. *Comput. Commun.* **1999**, 1502–1511, doi:10.1016/S0140-3664(99)00108-5.

[115] Fan, K.; Gong, Y.; Liang, C.; Li, H.; Yang, Y. Lightweight and ultralightweight RFID mutual authentication protocol with cache in the reader for IoT in 5G. *Secur. Commun. Netw.* **2016**, 3095–3104, doi:10.1002/sec.1314.

[116] Xie, Y.; Zhang, S.; Li, X.; Li, Y.; Chai, Y., CasCP: Efficient and Secure Certificateless Authentication Scheme for Wireless Body Area Networks with Conditional Privacy-Preserving. *Security and Communication Networks* **2019**, doi:10.1155/2019/5860286.

[117] World Health Organization,'Global report on diabetes. *World Health Organization*, **2016**.

[118] Leifer, B.P., Early Diagnosis of Alzheimer's Disease: Clinical and Economic Benefits. *Journal of the American Geriatrics Society* **2003**, S281–S288, doi:10.1046/j.1532-5415.5153.x.

[119] Anwar, M.; Abdullah, A.H.; Qureshi, K.N.; Majid, A.H., Wireless body area networks for healthcare applications: An overview. *Telkomnika, vol. 15, no. 3* **2017**, 1088–1095.

[120] Shen, J.; Gui, Z.; Ji, S.; Shen, J.; Tan, H.; Tang, Y., Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *Journal of Network and Computer Applications, vol. 106* **2018**, 117–123, doi:10.1016/j.jnca.2018.01.003.

[121] Al-Janabi, S.; Al-Shourbaji, I.; Shojafar, M.; Shamshirband, S., Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. *Egyptian Informatics Journal, vol. 18, no. 2* **2017**, 113–122, doi:10.1016/j.eij.2016.11.001.

[122] Negra, R.; Jemili, I.; Belghith, A., Wireless Body Area Networks: Applications and Technologies *Procedia Computer Science, vol. 83* **2016**, 1274–1281, doi:10.1016/j.procs.2016.04.266.

[123] Kasyoka, P.; Kimwele, M.; Mbandu Angolo, S., Certificateless pairing-free authentication scheme for wireless body area network in healthcare management system *Journal of Medical Engineering & Technology, vol. 44, no. 1* **2020**, 12–19, doi:10.1080/03091902.2019.1707890.

[124] United States Department of Health and Human Services, Health Insurance Portability and Accountability Act (HIPAA). Available online: `https://www.hhs.gov/hipaa/for-professionals/privacy/index.html` (accessed on 14 April 2021).

[125] Lozupone, V., Analyze encryption and public key infrastructure (PKI). *International Journal of Information Management, vol. 38, no. 1* **2018**, 42–44, doi:10.1016/j.ijinfomgt.2017.08.004.

[126] Yang, H.; Yang, B., A blockchain-based approach to the secure sharing of healthcare data. *Nisk Journal* **2017**, 100–111.

[127] Giri, D.; Maitra, T.; Amin, R.; Srivastava, P.D., An Efficient and Robust RSA-Based Remote User Authentication for Telecare Medical Information Systems. *J Med Syst, vol. 39, no. 1* **2015**, 145, doi:10.1007/s10916-014-0145-7.

[128] Monshizadeh, M.; Khatri, V.; Koskimies, O.; Honkanen, M., IoT Use Cases and Implementations *IoT Security* **2020**, 225–245, doi:10.1002/9781119527978.ch12.

[129] Dutta, R.; Barua, R.; Sarkar, P., Pairing-based cryptography: A survey. *Cryptology ePrint Archive* **2004**, Available online: `http://eprint.iacr.org/2004/064.pdf` (accessed on 14 April 2021).

[130] Boneh, D.; Lynn, B.; Shacham, H., Short Signatures from the Weil Pairing. *Journal of cryptology, vol. 17, no. 4,* **2004**, 297–319.

[131] Jiang, Q.; Lian, X.; Yang, C.; Ma, J.; Tian, Y.; Yang, Y., A bilinear pairing based anonymous authentication scheme in wireless body area networks for mHealth. *Journal of Medical Systems, vol. 40* **2016** doi:10.1007/s10916-016-0587-1.

[132] Jia, X.; He, D.; Kumar, N.; Choo, K.R., Authenticated key agreement scheme for fog-driven IoT healthcare system. *Wireless Networks, vol. 25* **2019**, doi:10.1007/s11276-018-1759-3.

[133] Khatoon, S.; Rahman, S.M.M.; Alrubaian, M.; Alamri, A., Privacy-Preserved, Provable Secure, Mutually Authenticated Key Agreement Protocol for Healthcare in a Smart City Environment. *IEEE Access, vol. 7* **2019**, 47962–47971, doi:10.1109/ACCESS.2019.2909556.

[134] Al Hamid, H.A.; Rahman, S.M.M.; Hossain, M.S.; Almogren, A.; Alamri, A., A Security Model for Preserving the Privacy of Medical Big Data in a Healthcare Cloud Using a Fog Computing Facility With Pairing-Based Cryptography. *IEEE Access, vol. 5* **2017**, 22313–22328, doi:10.1109/ACCESS.2017.2757844.

[135] Tang, W.; Zhang, K.; Ren., J.; Zhang, Y.; Shen, X., Flexible and Efficient Authenticated Key Agreement Scheme for BANs Based on Physiological Features. *IEEE Transactions on Mobile Computing, vol. 18, no. 4* **2019** 845–856, doi:10.1109/TMC.2018.2848644.

[136] Karati, A.; Islam, S.H.; Biswas, G.P., A pairing-free and provably secure certificateless signature scheme. *Information Sciences, vol. 450* **2018**, 378–391, doi:10.1016/j.ins.2018.03.053.

[137] Islam, S.H.; Biswas, G.P., A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks. *Annals of télécommunications-annales des telecommunications, vol. 67, no. 11-12* **2012**, 547–558, doi:10.1007/s12243-012-0296-9.

[138] Su, P.; Xie, Y.; Liu, P., Anonymous and Efficient Certificateless Multirecipient Signcryption Scheme for Ecological Data Sharing. *Journal of Sensors* **2020**, doi:10.1155/2020/5132861.

[139] Zeadally, S.; Das, A.K.; Sklavos, N., Cryptographic technologies and protocol standards for Internet of Things. *Internet of Things* **2019**, doi:10.1016/j.iot.2019.100075.

[140] Canetti, R.; Krawczyk, H., Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. *Pfitzmann B. (eds) Advances in Cryptology — EUROCRYPT 2001. EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045* **2001**, Berlin, Heidelberg, doi:10.1007/3-540-44987-6_28.

[141] Canetti, R.; Krawczyk, H., Universally composable notions of key exchange and secure channels. *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques–Advances in Cryptology (EUROCRYPT'02)* **2002**, 337—351, Amsterdam, The Netherlands.

[142] Das, A. K.; Wazid, M.; Yannam, A. R.; Rodrigues, J. J. P. C.; Park, Y., Provably Secure ECC-Based Device Access Control and Key Agreement Protocol for IoT Environment. *IEEE Access, vol. 7* **2019**, 55382–55397, doi:10.1109/ACCESS.2019.2912998.

[143] Ying, B.; Nayak, A., Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. *Journal of Network and Computer Applications, vol. 131* **2019**, 66–74, doi:10.1016/j.jnca.2019.01.017.

[144] SK Hafizul Islam; Biswas, G.P., A pairing-free identity-based two-party authenticated key agreement protocol for secure and efficient communication. *Journal of King Saud University - Computer and Information Sciences, vol. 29* **2017**, 63–73, doi:10.1016/j.jksuci.2015.01.004.

[145] He, D.; Padhye, S.; Chen, J., An efficient certificateless two-party authenticated key agreement protocol. *Computers & Mathematics with Applications, vol. 64* **2012**, 1914–1926, doi:10.1016/j.camwa.2012.03.044.

[146] Kim, Y. J.; Kim, Y. M.; Choe, Y. J.; Hyong, O., An efficient bilinear pairing-free certificateless two-party authenticated key agreement protocol in the eCK model. *Journal of Theoretical Physics and Cryptography, vol. 3* **2013**, 1—10.

[147] He, D.; Zeadally, S.; Xu, B.; Huang, X., An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks. *IEEE Transactions on Information Forensics and Security, vol. 10, no. 12* **2015**, 2681–2691, doi:10.1109/TIFS.2015.2473820.

[148] Barker, E., NIST Special Publication 800–57 Part 1 Revision 5, Recommendation for Key Management: Part 1 – General. *National Institute of Standards and Technology (NIST)* **2020**, doi:10.6028/NIST.SP.800-57pt1r5.

[149] MIRACL, MIRACL SDK: Multiprecision Integer and RationalArithmetic Cryptographic Library. **2020**. Available online: `https://github.com/miracl/MIRACL` (accessed on 20 April 2021).

[150] Chaudhry, S. A.; Yahya, K.; Al-Turjman, F.; Yang, M. -H., A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. *IEEE Access, vol. 8* **2020**, 139244–139254, doi:10.1109/ACCESS.2020.3012121.

[151] He, D.; Zeadally, S.; Kumar, N.; Wu, W., Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server Architectures. *IEEE Transactions on Information Forensics and Security, vol. 11, no. 9* **2016**, 2052–2064, doi:10.1109/TIFS.2016.2573746.

[152] ul haq, I.; Wang, J.; Zhu, Y., Secure two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5G networks. *Journal of Network and Computer Applications* **2020**, 1084–8045, doi:10.1016/j.jnca.2020.102660.

[153] Amin, R.; Biswas, G.P., Design and Analysis of Bilinear Pairing Based Mutual Authentication and Key Agreement Protocol Usable in Multi-server Environment. *Wireless Pers Commun 84* **2015**, 439–462, doi:10.1007/s11277-015-2616-7.

[154] Ogundoyin, S. O., A privacy-preserving certificateless two-party authenticated key exchange protocol without bilinear pairing for mobile-commerce applications. *Journal of Cyber Security Technology, 3:3* **2019**, 137–162, doi:10.1080/23742917.2019.1595357.

[155] Cheng, Q., Cryptanalysis of an efficient certificateless two-party authenticated key agreement protocol. *IACR Cryptol. ePrint Arch. 2012* **2012**, 725.

[156] Bala, S.; Sharma, G.; Verma, A. K., Impersonation attack on CertificateLess key agreement protocol. *International Journal of Ad Hoc and Ubiquitous Computing, vol. 27, no. 2* **2018**, 108–120, doi:10.1504/IJAHUC.2018.089580.

[157] Chang, YF.; Yu, SH.; Shiao, DR., A Uniqueness-and-Anonymity-Preserving Remote User Authentication Scheme for Connected Health Care. *J Med Syst, vol. 37* **2013**. doi:10.1007/s10916-012-9902-7.

[158] He, D.; Kumar, N.; Khan, M. K.; Wang, L.; Shen, J., Efficient privacy-aware authentication scheme for mobile cloud computing services. *IEEE Systems Journal, vol. 12, no. 2* **2016**, 1621–1631. doi:10.1109/JSYST.2016.2633809.

[159] Mokhtarnameh, R.; Muthuvelu, N.; Ho, S. B.; Chai, I., A Comparison Study on Key Exchange-Authentication protocol. *International journal of computer applications, vol. 7, no. 5* **2010**, 5–11.

[160] Cheng, B. HC, et al., Software engineering for self-adaptive systems: A research roadmap **2009**, doi:10.1007/978-3-642-02161-9_1.

[161] Kephart, J. O.; Chess, D. M., The vision of autonomic computing. *Computer, vol. 36, no. 1* **2003**, 41–50. doi:10.1109/MC.2003.1160055.

[162] De Lemos, R., et al., Software engineering for self-adaptive systems: A second research roadmap. *Software Engineering for Self-Adaptive Systems II. Springer, Berlin, Heidelberg* **2013**, 1–32. doi:10.1007/978-3-642-35813-5_1.

[163] Kramer, J.; Magee, J., Self-Managed Systems: an Architectural Challenge. *Future of Software Engineering (FOSE '07)* **2007**, 259–268. doi:10.1109/FOSE.2007.19.

[164] Oreizy, P., et al., An architecture-based approach to self-adaptive software. *IEEE Intelligent Systems and their Applications, vol. 14, no. 3* **1999**, 54–62. doi:10.1109/5254.769885.

[165] IBM, An architectural blueprint for autonomic computing. *IBM White Paper* **2005**. Available online: `https://www-03.ibm.com/autonomic/pdfs/ACBlueprintWhitePaperV7.pdf`. (accessed on 5 May 2021).

[166] Lara E.; Aguilar L.; Sanchez M.A.; García J.A., Adaptive Security Based on MAPE-K: A Survey. *Applied Decision-Making. Studies in Systems, Decision and Control, vol 209. Springer, Cham* **2019**, 157–183. doi:doi.org/10.1007/978-3-030-17985-4_7.

[167] Krupitzer, C.; Roth, F. M.; VanSyckel, S.; Schiele, G.; Becker, C., *A survey on engineering approaches for self-adaptive systems. Pervasive and Mobile Computing, vol 17* **2015**, 184–206. doi:10.1016/j.pmcj.2014.09.009.

[168] Bakar, K. A. A.; Haron, G. R., Adaptive authentication: Issues and challenges. *2013 World Congress on Computer and Information Technology (WCCIT)*, **2013**, 1–6, doi: 10.1109/WCCIT.2013.6618657.

[169] Patel, V. M.; Chellappa, R.; Chandra, D.; Barbello, B., Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine, vol. 33, no. 4* **2016**, 49–61. doi:10.1109/MSP.2016.2555335.

[170] Jorquera Valero, J.M.; Sánchez Sánchez, P.M.; Fernández Maimó, L.; Huertas Celdrán, A.; Arjona Fernández, M.; De Los Santos Vílchez, S.; Martínez Pérez, G., Improving the Security and QoE in Mobile Devices through an Intelligent and Adaptive Continuous Authentication System. *Sensors, vol. 18, no. 11* **2018**. doi:10.3390/s18113769

[171] Arias-Cabarcos, P.; Krupitzer, C.; Becker, C., A survey on adaptive authentication. *ACM Computing Surveys (CSUR), vol. 52, no. 4* **2019**, 1–30. doi:10.1145/3336117.

[172] Bailey, C.; Chadwick, D. W.; Lemos R. d. Self-Adaptive Authorization Framework for Policy Based RBAC/ABAC Models. *J 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing* **2011**, doi:10.1109/DASC.2011.31

[173] Liu, Y.; Seet, B.-C.; Al-Anbuky, A. An Ontology-Based Context Model for Wireless Sensor Network (WSN) Management in the Internet of Things. *J. Sens. Actuator Netw.* **2013**, 653–674. doi:10.3390/jsan2040653.

[174] Huadong Wu; Siegel, M.; Ablay, S., Sensor fusion for context understanding. *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276), vol. 1* **2002**, 13–17. doi:10.1109/IMTC.2002.1006808.

[175] Abuhamad, M.; Abusnaina, A.; Nyang, D.; Mohaisen, D., Sensor-Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey. *IEEE Internet of Things Journal, vol. 8, no. 1* **2021**, 65–84, doi: 10.1109/JIOT.2020.3020076.

[176] Liang, Y.; Samtani, S.; Guo, B.; Yu, Z., Behavioral Biometrics for Continuous Authentication in the Internet-of-Things Era: An Artificial Intelligence Perspective. *IEEE Internet of Things Journal, vol. 7, no. 9* **2020**, 9128–9143, doi: 10.1109/JIOT.2020.3004077.

[177] Duckham, M.; Kulik, L., A Formal Model of Obfuscation and Negotiation for Location Privacy. *Pervasive Computing. Pervasive 2005. Lecture Notes in Computer Science, vol. 3468* **2005**, 152–170, doi:10.1007/11428572_10.

[178] Fang, H.; Wang, X.; Tomasin, S., Machine Learning for Intelligent Authentication in 5G and Beyond Wireless Networks. *IEEE Wireless Communications, vol. 26, no. 5* **2019**, 55–61,doi:10.1109/MWC.001.1900054.

[179] Perez, B.; Musolesi, M.; Stringhini, G., You are your metadata: Identification and obfuscation of social media users using metadata information. *Proceedings of the International AAAI Conference on Web and Social Media, vol. 12, no. 1* **2018**.

[180] Varadi, S.; Varkonyi, G. G.; Kertesz, A., Law and IoT: How to see things clearly in the Fog. *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)* **2018**, 233–238, doi:10.1109/FMEC.2018.8364070.

[181] Politou, E.; Michota, A.; Alepis, E.; Pocs, M.; Patsakis, C., Backups and the right to be forgotten in the GDPR: An uneasy relationship. *Computer Law & Security Review, vol. 34, no. 6* **2018**, 1247–1257, doi:10.1016/j.clsr.2018.08.006.

[182] Mohammadi, M.; Al-Fuqaha, A.; Guizani, M.; Oh, J. S., Semi-supervised Deep Reinforcement Learning in Support of IoT and Smart City Services. *IEEE Internet of Things Journal* **2017**, 1–12, doi:10.1109/JIOT.2017.2712560.

[183] Mohammadi, M.; Al-Fuqaha, A., Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges. *IEEE Communications Magazine, vol. 56, no. 2* **2018**, 94–101, doi:10.1109/MCOM.2018.1700298.

[184] Bhatt, R., Fuzzy-Rough Approaches for Pattern Classification: Hybrid measures, Mathematical analysis, Feature selection algorithms, Decision tree algorithms, Neural learning, and Applications. *Decision Tree Algorithms, Neural Learning, and Applications* **2017**, Amazon Books.

[185] Rohra, J. G.; Perumal, B.; Narayanan, S. J.; Thakur, P.; Bhatt, R. B., User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks. *Proceedings of Sixth International Conference on Soft Computing for Problem Solving* **2017**, 286–295, doi:10.1007/978-981-10-3322-3_27.

[186] Carpineti, C.; Lomonaco, V.; Bedogni, L.; Di Felice, M.; Bononi, L., Custom Dual Transportation Mode Detection by Smartphone Devices Exploiting Sensor Diversity. *Proceedings of the 14th Workshop on Context and Activity Modeling and Recognition (IEEE COMOREA 2018), Athens, Greece, March 19-23, 2018* **2018**.

[187] Aşuroğlu, T.; Açici, K.; Erdaş, Ç. B.; Oğul, H., Texture of Activities: Exploiting Local Binary Patterns for Accelerometer Data Analysis. *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* **2016** 135–138, doi:10.1109/SITIS.2016.29.

[188] Erdaş, Ç. B.; Atasoy, I.; Açıcı, K.; Oğul, H., Integrating features for accelerometer-based activity recognition. *Procedia Computer Science* **2016**, 522–527, doi:10.1016/j.procs.2016.09.070.

[189] Güney, S.; Erdaş, Ç. B., A Deep LSTM Approach for Activity Recognition. *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* **2019**, 294–297, doi:10.1109/TSP.2019.8768815.

[190] WEKA, Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Available online: `http://old-www.cms.waikato.ac.nz/~ml/weka/`. (accessed on 10 May 2021).

Appendix A

HLPSL modeling of LAKD

role sensor(G,S : agent, KPij : symmetric_key, AIDi: text, B0i, B1i: text, HASHF: hash_func, SND, RCV : channel (dy))

played_by S def=

local

State: nat,

Kst: symmetric_key,

T1i, T2i, T3i, T4i: text,

R1i, R2i: text,

M1, M2, M3: text,

M7: text,

Idxi: text

init

State := 0

transition

1. State = 0 /\RCV(start) = |>

State'= 2 /\T1i' := new() /\R1i' := new()

/\SND(T1i'.AIDi.xor(HASHF(AIDi.B1i.T1i'),R1i').HASHF(R1i'.T1i'.B0i))

2. State = 2 /\RCV(T2i'.xor(HASHF(R1i.T2i'),R2i').xor(Idxi'.R1i.R2i',HASHF(B1i.R2i')).

HASHF(Idxi'.R2i'.B0i.R1i)) =|>

State'= 4 /\T3i' := new() /\SND(T3i'.HASHF(B1i.R1i.T3i'.R2i'.KPij))

3. State = 4 /\RCV(T4i'.HASHF(B0i.R1i.T4i'.R2i.KPij)) =|>

State' := 6 /\Kst' := HASHF(R1i.R2i.AIDi.KPij) /\request(S,G,sensor_gateway_kst,Kst') end role

role gateway(G,S : agent, KPij : symmetric_key, AIDi: text, B0i, B1i: text, HASHF: hash_func, SND, RCV : channel (dy))

played_by G def=

local

State: nat,

Kst: symmetric_key,

T1i, T2i, T3i, T4i: text,

R1i, R2i: text,

M4, M5, M6: text,

M8, M9: text,

Idxi: text

init

State := 1

transition

1. State = 1

/\RCV(T1i'.AIDi.xor(HASHF(AIDi.B1i.T1i'),R1i').HASHF(R1i'.T1i'.B0i)) =|>

State' := 3

/\T2i' := new() /\R2i' := new() /\Idxi' := new()

/\SND(T2i'.xor(HASHF(R1i'.T2i'),R2i').xor(Idxi'.R1i'.R2i',HASHF(B1i.R2i')).

HASHF(Idxi'.R2i'.B0i.R1i'))

2. State = 3 /\RCV(T3i'.HASHF(B1i.R1i.T3i'.R2i.KPij)) =|>

State':=5 /\T4i' := new() /\SND(T4i'.HASHF(B0i.R1i.T4i'.R2i.KPij))

/\Kst' := HASHF(R1i.R2i.AIDi.KPij)

/\witness(G,S,sensor_gateway_kst,Kst')

/\secret(Kst',kst,{G,S})

end role

role session(

G,S : agent,

KPij : symmetric_key,

AIDi: text,

B0i, B1i: text,

```
HASHF: hash_func)
def=
local SG, RG, SS, RS: channel (dy)
composition
gateway (G, S, KPij, AIDi, B0i, B1i, HASHF, SG, RG)
/\sensor (G, S, KPij, AIDi, B0i, B1i, HASHF, SS, RS)
end role

role environment()
def=
const
sensor_gateway_kst, kst : protocol_id,
kpij, kpijintg: symmetric_key,
aidi, aidiint: text,
b0i, b1i, b0iintg, b1iintg: text,
g, s : agent,
h : hash_func

intruder_knowledge = {g, s, h, kpijintg, aidi, aidiint, b0iintg, b1iintg}
composition
session(g, s, kpij, aidi, b0i, b1i, h)
/\session(g, s, kpij, aidi, b0i, b1i, h)
/\session(g, i, kpijintg, aidiint, b0iintg, b1iintg, h)
end role

goal
secrecy_of kst
authentication_on sensor_gateway_kst
end goal
environment()
```

Appendix B

HLPSL modeling of TLAP

role entityA(A,B : agent, IDb: text, Puba, Pubb: public_key, HASHF: hash_func, SND, RCV : channel (dy))
played_by A def=
local
State: nat,
F: hash_func, % Addition in ECC
IDa: text,
T1, T2, T3: text,
R1, R2: text

init
State := 0
transition
1. State = 0 /\RCV(start) =|>
State'= 2 /\T1' := new() /\R1' := new() /\IDa' := new()
/\SND(T1'.{F(R1')}_Pubb.{IDa'}_F(R1').HASHF(F(R1').IDa'.IDb.A.T1'))
/\secret(F(R1'),f_r1,{A,B})

2. State = 2 /\RCV(T2'.{F(R2')}_Puba.HASHF(F(R2').F(R1).IDb.IDa.B.A.T2')) =|>
State'= 4 /\T3' := new() /\SND(T3'.HASHF(F(R1).F(R2').IDa.IDb.A.B.T3'))
/\request(A,B,a_b_f_r1,F(R1))
/\witness(A,B,a_b_f_r2,F(R2'))
end role

role entityB(A,B : agent, IDb: text, Puba, Pubb: public_key, HASHF: hash_func, SND, RCV : channel (dy))
played_by B def=
local

```
State: nat,
F: hash_func, % Addition in ECC
IDa: text,
T1, T2, T3: text,
R1, R2: text

init
State := 1
transition
1. State = 1/\RCV(T1'.{F(R1')}_Pubb.{IDa'}_F(R1').HASHF(F(R1').IDa'.IDb.A.T1')) =|>
State' := 3 /\T2' := new()/\R2' := new()
/\SND(T2'.{F(R2')}_Puba.HASHF(F(R2').F(R1').IDb.IDa'.B.A.T2'))
/\secret(F(R2'),f_r2,{A,B})
/\witness(B,A,a_b_f_r1,F(R1'))

2. State = 3/\RCV(T3'.HASHF(F(R1).F(R2).IDa.IDb.A.B.T3')) =|>
State':=5
/\request(B,A,a_b_f_r2,F(R2))
end role

role session( A,B : agent, IDb: text, Puba, Pubb: public_key, HASHF: hash_func)
def=
local SA, RA, SB, RB: channel (dy)
composition
entityA (A, B, IDb, Puba, Pubb, HASHF, SA, RA)
/\entityB (A, B, IDb, Puba, Pubb, HASHF, SB, RB)
end role

role environment()
def=
const
a, b : agent,
idb, idi: text,
puba, pubb, pubi: public_key,
```

h : hash_func,
a_b_f_r1, a_b_f_r2, f_r1, f_r2 : protocol_id

intruder_knowledge = {a, b, h, idb, idi, puba, pubb, pubi, inv(pubi)}
composition
session(a, b, idb, puba, pubb, h)
/\session(a, b, idb, puba, pubb, h)
/\session(a, i, idi, puba, pubi, h)
% /\session(i, b, idb, pubi, pubb, h)
end role

goal
secrecy_of f_r1, f_r2
authentication_on a_b_f_r1, a_b_f_r2
end goal
environment()