


Article

SLMAS: A Secure and Light Weight Mutual Authentication Scheme for the Smart Wheelchair

Abdulwahab Ali Almazroi ¹, Misbah Liaqat ¹, Rana Liaqat Ali ^{2,*} and Abdullah Gani ³¹ Department of Information Technology, College of Computing and Information Technology at Khulais, University of Jeddah, Jeddah 21959, Saudi Arabia² Department of Physics, COMSATS University Islamabad, Islamabad 45550, Pakistan³ Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia* Correspondence: liaqat_ali@comsats.edu.pk

Abstract: The modern innovation called the Internet of Things (IoT) empowers individuals to connect to anybody and anything at any point, wherever. The application of the IoT in smart cities concerning smart healthcare management can improve patient welfare, user acceptance, the standard of living, and accurate illness monitoring. Powered wheelchairs (PW) with sensors, computers, and other connected assistive technologies are called smart wheelchairs. Smart wheelchairs with sensing abilities are intended to offer universal connectivity using cloud and edge computing technology. Numerous outstanding people were impacted by paralyzing phenomena, including Stephen Hawking and Max Brito. The issue of legitimacy is one of the most important difficulties in e-health applications, because of how sensitive the technology is, and this needs to be appropriately handled. To safeguard the data transport, usage, and interchange between sensor nodes/smart wheelchairs and servers, e-health applications require an authentication method. As all conversations use wireless channels, e-health apps are exposed to various vulnerabilities. Additionally, the IoT has limited computational and power capacity limitations. To combat the various security risks, the present research offers a user authentication technique that is efficient and ensures anonymity. The suggested method creates a safe connection for the authorized entity and forbids unauthorized entities from accessing the Internet of Things sensor nodes. The suggested approach has lower communication and computation overheads than the traditional techniques, making it more effective. In addition, the security verification of the presented protocol is scrutinized through AVISPA. The proposed scheme, on average, requires only 12.4% more computation cost to execute. Compared to the existing approaches, the suggested protocol's extra computational cost can be compensated for by its enhanced security, while the suggested method's communication cost is 46.3% smaller.

Keywords: AVISPA; smart city; smart wheelchair; edge and cloud computing; Internet of Things (IoT); sensors; authentication; protocols; security



Citation: Almazroi, A.A.; Liaqat, M.; Ali, L.; Gani, A. SLMAS: A Secure and Lightweight Mutual Authentication Scheme for a Smart Wheelchair. *Appl. Sci.* **2023**, *13*, 6564. <https://doi.org/10.3390/app13116564>

Academic Editor: Christos Bouras

Received: 13 March 2023

Revised: 23 April 2023

Accepted: 30 April 2023

Published: 28 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Human life is marvelous. By creating safer equipment with intelligent technologies, technology and science are crucial for ensuring people's security. In the last few years, technology has made significant progress. Rather than being called by their names, many products in our homes and everyday life are now prefixed with the word "smart" [1]. For example, the terminology used to describe modern "Smart Wheelchairs" and the hardware and software needed to make traditional wheelchairs, smart homes, smart TVs, and smartphones have influenced this. The very first motorized wheelchair was created by George Klein fifteen years ago [2,3]; and since then, there have been several initiatives in this area, leading to entirely robotic wheelchairs and intelligent wheelchairs [4]. Various advancements, especially artificial intelligence (AI) [5], Internet of Things (IoT) [6,7], and edge and cloud computing technology [8] have indeed been successfully applied to smart

wheelchairs, to assist users in getting around and moving safely without assistance. This research presents another effort to provide safety to mobility impaired humans during the Hajj and Umrah services [9].

The term “Internet of Things” (IoT) refers to enormous networks that combine the Internet with a variety of sensing technologies, to achieve the connection of people, machines, and things at whichever time and at any location, which has become crucial in the age of information. Wireless sensor systems are managed by connecting small nodes [10] or devices through Zigbee, Bluetooth, or WiFi. The IoT in smart cities can have various futuristic applications, especially in healthcare and assistive technology. One such application is the development of intelligent wheelchairs that utilize WSNs for tracking and monitoring the location and movements of the wheelchair user. This can aid in providing better assistance and care for people with mobility impairments, as presented in Figure 1. Smart wheelchairs with sensing abilities are intended to offer universal connectivity using cloud and edge computing technology.

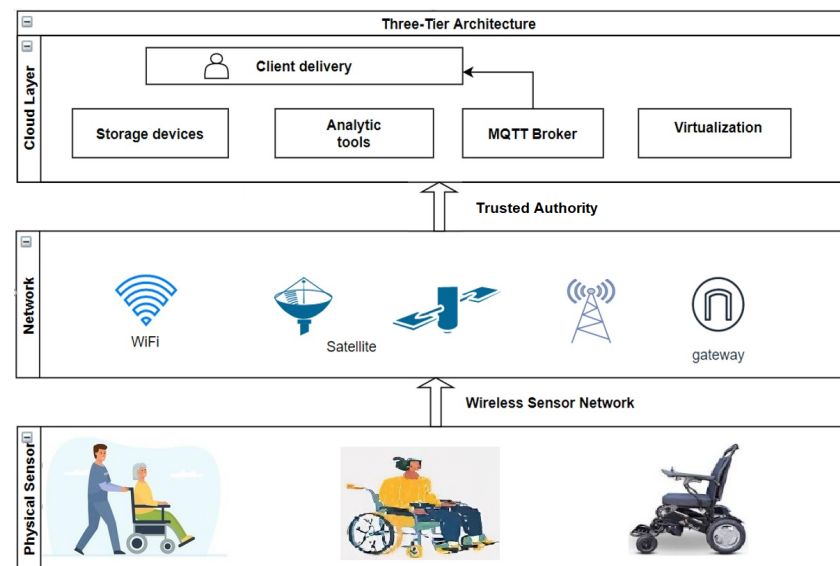


Figure 1. System Architecture.

Another application can be using WSNs for asset tracking in industry, such as tracking the movement and location of goods and products within a warehouse or during transportation. This can improve inventory management and logistics. WSNs can also be used in agriculture for tracking the movement and location of livestock, crops, and equipment. This can aid in improving the overall efficiency and productivity of farms and enable better management of resources. In smart cities, WSNs can be used for traffic management, monitoring air quality, and detecting environmental hazards. This can make cities more sustainable, efficient, and safer for citizens.

Overall, the potential applications of WSN-based tracking technology are vast and may significantly impact various fields and industries in the future. Moreover, wireless sensor networks (WSN) have additional security concerns over traditional networks, since the data gathered by nodes is relayed across open channels; these nodes are frequently installed in hostile or unsupervised areas, where they are easy targets for destruction or capture [11]. Undoubtedly, a security compromise might have severe and far-reaching repercussions if private data such as user names or crucial node information were to be revealed. It is crucial to create a secure authentication mechanism, to protect the integrity of the data communicated in WSN and the legitimacy of each entity. Additionally, perhaps a scheme might utilize numerous security measures, such as mutual authentication, user anonymity, unlinkability, password updates, two-factor security, secure session key agreement, perfect forward secrecy, and known session key security, and it should resist other well-known

attacks [12]. Moreover, the subsequent section provides definitions and explanations for key terms and concepts used throughout this document.

1.1. Key Terms Explanation

This section is designed to help readers better understand the language and terminology used in this article and to ensure that everyone uses the same definitions for essential concepts.

1.1.1. XOR Operation

The XOR (exclusive or) operation is a fundamental building block for encryption and decryption techniques in cryptography. A logical operation called XOR produces a binary output from two binary inputs. This is how the XOR operation is described: If both input bits are the same, the output bit is 0. The output value is 1 if the input bits are different.

1.1.2. Cryptographic Hash Mechanism

A mathematical operation called a cryptographic hash function converts arbitrary sized input data into a fixed-size output presented as a hash value or a message digest. The hash value is a singular presentation of the input data; hence, any modification to the input data will result in a non-identical hash value. Cryptographic hash functions are frequently utilized for many tasks in cryptography, such as digital signatures, message authentication codes (MACs), and password storage.

1.1.3. Elliptic Curve Cryptography (ECC)

The algebraic behavior of elliptic curves over finite fields provides the basis for the public-key encryption known as elliptic curve cryptography (ECC). ECC is a relatively new and potent cryptographic approach, with various benefits compared to more established public-key encryption technologies such as RSA and Diffie–Hellman.

1.1.4. Symmetric Cryptography

Data may be encrypted and decrypted employing a single secret key with symmetric cryptography. Therefore, this means the sending party and the person who receives the encrypted data must have access to the private key. Other terms used for symmetric cryptography include shared-secret and secret-key.

1.1.5. Asymmetric Cryptography

Information is encrypted via a public key and decrypted via a private key in asymmetric cryptography. This makes it possible for two people to communicate securely without revealing the secret key. Asymmetric cryptography is frequently used in various applications, such as secure email, online banking, and e-commerce, as well as for digital signatures, key exchange, and encryption.

1.1.6. Hajj and Umrah Services

Islam's two most significant pilgrimages, the Hajj and Umrah, both require visits to the Saudi Arabian holy city of Mecca. Every able-bodied Muslim with the financial means must perform the Hajj, one of Islam's five pillars, whereas Umrah is an optional trip that can be made any time of the year.

1.2. Adversarial Model

This paper considers the widely recognized Dolev–Yao (DY) adversarial model [13], which was applied in [14–16]. In an adversarial model, the adversary (\mathcal{A}) is assumed to possess the following capabilities:

1. The communication between two parties occurs via a public channel, and neither endpoint is deemed trustworthy.
2. The \mathcal{A} possesses complete owning authority over the public communication channel.

3. The \mathcal{A} can improve or edit the message being transmitted through the public channel and create a fraudulent message.
4. It is impossible to compromise the secret/private key of the trusted authority (TA)/central authority (CA).

1.3. Motivations and Contributions

SLMAS aims to address the security and privacy issues that smart wheelchairs encounter. These challenges include unlawful access, data manipulation, and privacy violations, which may jeopardize the wheelchair user's security and privacy. With sensors and communication tools, smart wheelchairs can connect with other hardware and software, including smartphones, Internet of Things (IoT) devices, and cloud services. Nevertheless, because these products are susceptible to cyberattacks such as eavesdropping, impersonation, and data theft, their connection also brings new security dangers. By lowering the computational and communication overheads, SLMAS aims to provide a safe and effective communication system for intelligent wheelchairs, which secures them against these security threats. SLMAS mutual authentication (MA) mechanism will ensure that only permitted devices can communicate with the wheelchair and that all communications are secured against tampering or unauthorized access.

The rest of the article is designed as follows: a review of the literature examines the prior literature and research on the subject of the planned project in Section 2. The proposed scheme and the methodology are provided in Section 3. Section 4 offers a rigorous security assessment of the suggested research project and highlights potential weaknesses. A computation, communication, and feature comparison is given in Section 5. The article's main conclusions are outlined in the last section, along with the article's contributions and potential directions for further study.

2. Literature Review

Numerous authentication mechanisms have been researched in the literature since Lamport [17] introduced the first authentication mechanism in 1981. Das et al. [18] presented an efficient two-factor authentication system for WSN in 2009. Afterwards, Khan et al. [19] and Chen et al. [20] observed that Das et al.'s technique is prone to impersonation, offline password-guessing, and insider attacks. Later, they also suggested a different plan to address the security challenges with the Das et al. scheme.

Suh et al. [21]'s physical unclonable functions (PUF) design uses logic circuits' built-in latency to verify the authenticity of integrated circuits. A PUF circuit that provides privacy-protected verification among a server and limited devices was created by Aysu et al. [22]. On the verification side, Majzoobi et al. suggested a technique replicating a PUF circuit [23,24]. Regarding passive devices such as radio-frequency identification (RFID), this study combines the authentication procedures presented in [25–32]. Although the suggested techniques offer physical-level protection, they are frequently challenging to apply in an IoT environment, because an authentication server stores many challenge-response pairs (CRPs).

The technique proposed by Challa et al. [33] that used an ECC-based user authentication scheme was deemed insecure against impersonation assaults by Jia et al. [34]. Additionally, [33] has large communication and computational overheads. An approach centered on IoT-based cloud systems was presented by Zhou et al. [35]. This method is, unfortunately, susceptible to man-in-the-middle (MITM), impersonation, privileged insider, and replay attacks [36].

User authentication and key agreement mechanisms for various WSN and IoT networks were proposed by Farash et al. [37]. Amin et al. [38] found some flaws with this method and exposure to impersonation attacks and offline password-guessing attacks. In the meantime, Sharma and Kalra used a lightweight user authentication scheme, whereas Canetti and Krawczyk [39] demonstrated this as unsafe against privileged insider attacks. In addition, another novel technique for user authentication and key agreement was sug-

gested by Turkanović et al. [40]. This technique was discovered to be susceptible to various attacks, including offline password guessing, user impersonation, and attacks on sensor nodes [41]. A minimalist key management authentication technique ideal for Internet of Things deployment was developed by Wazid et al. [42]. This method offers faster and more efficient connection using the xor operation and a one-way cryptographic hash mechanism.

Different authentication methods were studied by Hussain et al. [43] with two different classifications. Additionally, they investigated the various authentication strategies and outlined the benefits, drawbacks, obstacles, efficiency evaluations, and resilience versus various security attacks. The security assessment and performance assessment were unfortunately incomplete. Regarding the security component, the researchers only highlighted a few of the discovered threats. Only a portion of the computational costs was covered in the efficiency section, and no discussion of communication, energy, and storage costs was included.

The existing security solutions are exposed to numerous attacks, such as impersonation, MITM, and replay, as shown by the relevant schemes covered in this literature survey; most of the solutions offered failed to solve these anonymity and untraceability problems. Therefore, the current schemes are inappropriate for resource-constrained deployments of IoT-based smart wheelchairs, due to poor security and pricey features; these drawbacks are displayed in Table 1.

Table 1. An overview of the shortcomings/drawbacks of earlier user authentication methods for wireless sensor networks.

| Scheme | Year | Drawbacks |
|--------------------|------|---|
| Shreya et al. [44] | 2022 | Using IoMT devices entails new security and privacy problems, such as unwanted access to private medical information or the danger of data breaches because of device or communication channel flaws. |
| Masud et al. [36] | 2021 | It is prone to session key leakage, offline password guessing, and traceability attacks. |
| Zhou et al. [35] | 2019 | It is open to man-in-the-middle, privileged insider, impersonation and replay attacks. Moreover, its computation cost is very high |
| Sharma et al. [45] | 2019 | It is open to privileged insider and password guessing attacks |
| Wazid et al. [42] | 2019 | It is open to impersonation and lacks anonymity property attacks |
| Chang et al. [46] | 2017 | Its disadvantage is that the user ID and OTP are not secured throughout the login and authentication process |
| Wu et al. [47] | 2018 | It is not secure against user impersonation attacks and can also not provide user anonymity |

3. Proposed Scheme

User authentication for a smart wheelchair is highlighted in this section. The sensor node (SN) gathers real-time data and transmits it to the server (Ss). In this case, SN and Ss registration is performed by a trusted authority (TA). Before the data is sent, the validity of SN and Ss is checked. After mutual authentication, the parties engaged in communication create a shared session key for secure communication. Moreover, the different notations employed in this study are provided in Table 2. Furthermore, a block diagram is given in Figure 2, which represents the system and process and shows the proposed scheme's major component interrelationships.

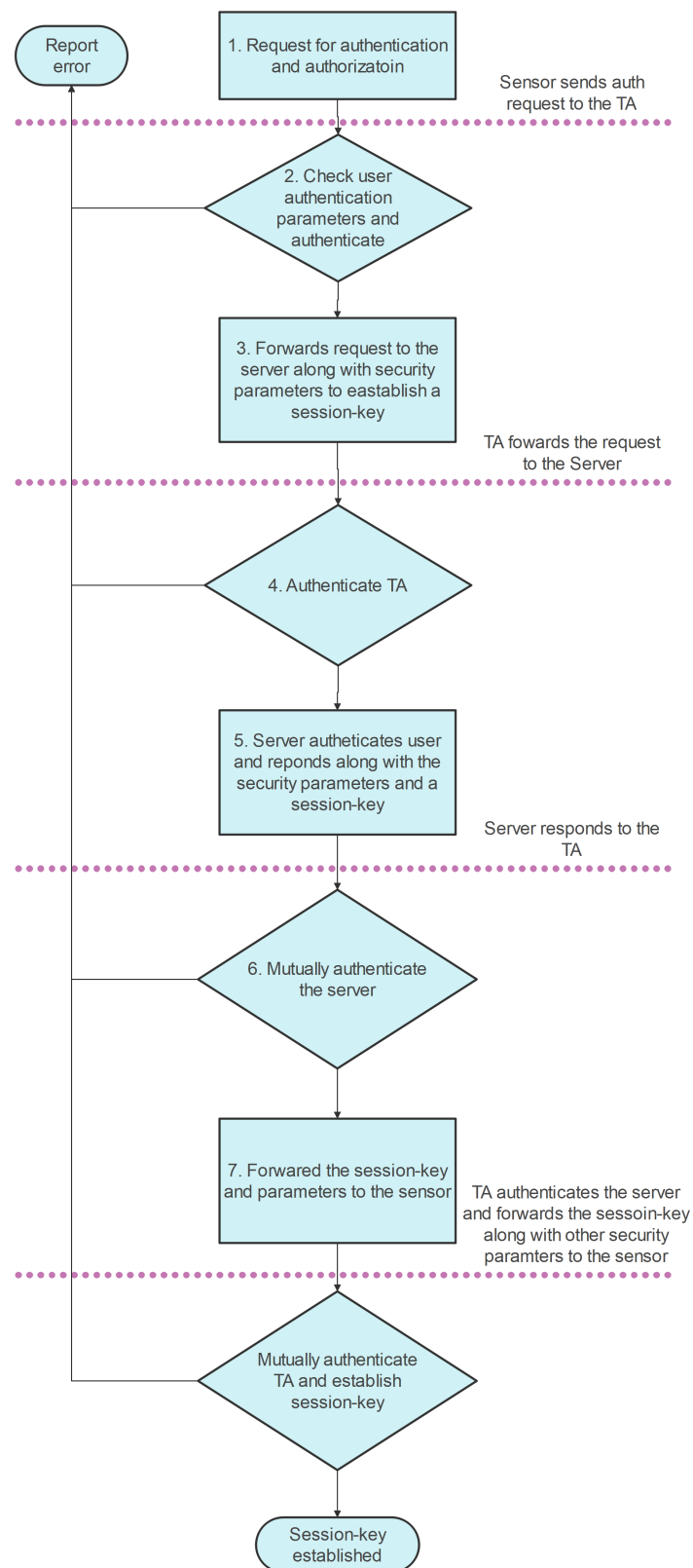


Figure 2. Block diagram.

Table 2. Notation Guide.

| Symbols | Representations |
|--|---|
| $SN_n, SN_{ID_n}, PR_{ID_n}$ | n th sensor node, its personal identity, pseudo-random identity |
| S_s, S_{ID_s}, PR_{ID_s} | s th server, its personal identity, pseudo-random identity |
| TA, MSK | Trusted Authority and its secret master-key |
| $\delta T, TC$ | Maximum admissible transmission-delay and present-time |
| $SK_{ns}(= SK_{sn})$ | Shared session key between SN_n and S_s |
| $SKEY_{TA,n}$ | Shared-secret-key among the TA and SN |
| $SKEY_{TA,s}$ | Shared-secret-key among the TA and server |
| T_{TA}, T_n, T_s | Current timestamps of TA, SN_n , and S_s |
| RND_i | i th random value of 160 bits |
| $i \stackrel{?}{=} j$ | Verify if i equals to j |
| $H(.)$ | Cryptographic one way hash function |
| $\mathcal{A}, \mathcal{U}_{\mathcal{A}}$ | An adversary and privileged insider |
| $\oplus, $ | Bitwise exclusive or and concatenation-operators |

3.1. Initialization Process

In this procedure, the trusted authority (TA) chooses a publicly available one-way hash function $\{H(.)\}$ and $MSK \in \mathcal{Z}_p$ a private master key.

3.2. Server Registration Process

This phase covers the process of enrolling the servers, with the TA as depicted in Table 3:

1. Server (S_s) picks an identity S_{ID_s} , and sends it through a protected link to the (TA).
2. TA obtains the registration request from S_s , opts for a random number $RND_1 \in \mathcal{Z}_p^*$, a temporary identity $PR_{ID_s} \in \mathcal{Z}_p^*$, and computes $X_s = H(S_{ID_s} || RND_1)$, $SKEY_{TA,s} = H(X_s || MSK)$. TA sends the message that contains $\{PR_{ID_s}, SKEY_{TA,s}\}$ to S_s over the trusted channel. TA further save the parameter $\{PR_{ID_s}, ENC_{MSK}[S_{ID_s}, X_s]\}$ in the database.
3. Upon receiving the response from TA , S_s saves the parameters $\{PR_{ID_s}, S_{ID_s}, SKEY_{TA,s}\}$.

Table 3. Proposed server registration process.

| Server (S_s) | Trusted Authority (TA) |
|--|---|
| Select an identity S_{ID_s} | |
| $\xrightarrow[(S_s \rightarrow TA)]{S_{ID_s}}$ | Picks arbitrary number $RND_1 \in \mathcal{Z}_p$ and Pseudo-random identity PR_{ID_s} COMPUTE: $X_s = H(S_{ID_s} RND_1)$ $SKEY_{TA,s} = H(X_s MSK)$ Store $\{PR_{ID_s}, ENC_{MSK}[S_{ID_s}, X_s]\}$ tuple in the database $\xleftarrow[(S_s \leftarrow TA)]{\langle PR_{ID_s}, SKEY_{TA,s} \rangle}$ |
| Save $\{PR_{ID_s}, S_{ID_s}, SKEY_{TA,s}\}$ | |

3.3. Sensor Node Registration Process

The steps taken to enrol a sensor node with the system presented in Table 4 include the following:

1. Sensor Node (SN_n) catches an identity SN_{ID_n} , and sends it through a protected route to the (TA).
2. TA is contacted by the registration request from SN_n , decides a random values $RND_2 \in \mathcal{Z}_p^*$, a temporary identity $PR_{ID_n} \in \mathcal{Z}_p^*$, and computes $X_n = H(SN_{ID_n} || RND_2)$,

Table 4. Proposed SN registration process.

| Sensor Node (SN) | Trusted Authority (TA) |
|--|---|
| Select an identity SN_{ID_n} | $\xrightarrow[\text{(SN} \rightarrow \text{TA)}]{SN_{ID_n}}$ Picks arbitrary number $RND_2 \in \mathbb{Z}_p$ and Pseudo-random identity PR_{ID_n} COMPUTE: $X_n = H(SN_{ID_n} RND_2)$ $SKEY_{TA,n} = H(X_n MSK)$ Store $\{PR_{ID_n}, ENC_{MSK}[SN_{ID_n}, X_n]\}$ tuple in the database $\xleftarrow[\text{(SN} \leftarrow \text{TA)}]{\langle PR_{ID_n}, SKEY_{TA,n} \rangle}$ |
| Save $\{PR_{ID_n}, SN_{ID_n}, SKEY_{TA,n}\}$ | |

$SKEY_{TA,n} = H(X_n || MSK)$. TA communicates the message that contains $\{PR_{ID_n}, SKEY_{TA,n}\}$ to SN_n over the protected medium. TA also saves the value $\{PR_{ID_n}, ENC_{MSK}[SN_{ID_n}, X_n]\}$ in the database.

- Upon receiving the response from TA, SN_n saves the parameters $\{PR_{ID_n}, SN_{ID_n}, SKEY_{TA,n}\}$, as presented in Table 7.

3.4. Authentication and Key-Agreement Process

Below are the steps that are performed by SN_n and S_s to build a session key with the assistance of a TA, to secure the communication and mutually authenticate each other:

- SN_n selects an arbitrary number RND_3 and timestamp T_{SN} and computes $V_{temp1} = H(SKEY_{TA,n} || SN_{ID_n})$, $RND'_3 = RND_3 \oplus V_{temp1}$, $AUTH_n = H(R_3 || V_{temp1} || T_{SN})$. Finally, SN_n transmits the message to the TA containing $\langle RND'_3, AUTH_n, PR_{ID_n}, T_{SN} \rangle$ over the public channel.
- Upon obtaining the message from SN_n , TA first verifies the message freshness by examining the condition $|TC - T_{SN}| \leq \delta T$ and check if PR_{ID_n} exists in DB.
- If true, TA further computes $[SN_{ID_n}, X_n] = DEC_{MSK}[SN_{ID_n}, X_n]$, $SKEY_{TA,n} = H(X_n || MSK)$, $V_{temp2} = H(SKEY_{TA,n} || SN_{ID_n})$, $RND_3 = RND'_3 \oplus V_{temp2}$. TA verifies the authenticity of the SN_n by examining the condition $AUTH_n \stackrel{?}{=} H(RND_3 || V_{temp2} || T_{SN})$.
- If true, TA further computes $[S_{ID_s}, X_s] = DEC_{MSK}[X_s]$, $SKEY_{TA,s} = H(X_s || MSK)$, $V_{temp3} = H(SKEY_{TA,s} || S_{ID_s} || T_{TA})$. TA picks two arbitrary numbers RND_4, RND_5 and further computes $AUTH_{TA,s} = H(RND_4 || V_{temp3} || T_{TA})$, $RND'_4 = RND_4 \oplus V_{temp3}$, $Y_{TA,s} = H(RND_5 || V_{temp3})$. Finally, TA sends the message containing $\langle PR_{ID_s}, RND'_4, AUTH_{TA,s}, Z_{TA,s}, T_{TA} \rangle$ to the S_s over the insecure channel.
- Upon obtaining the message from TA, S_s first verifies the messages freshness by examining the condition $|TC - T_{TA}| \leq \delta T$. Next S_s computes $V_{temp4} = H(SKEY_{TA,s} || S_{ID_s} || T_{TA})$, $RND_4 = RND'_4 \oplus V_{temp4}$ and confirms the legitimacy of the TA by assessing the scenario $AUTH_{TA,s} \stackrel{?}{=} H(RND_4 || V_{temp4} || T_{TA})$.
- If true, S_s picks two arbitrary numbers RND_6, RND_7 , and presents timestamp T_s , and further computes $SK_{s,n} = H(RND_6 || Z_{TA,s} \oplus V_{temp4} || T_s)$, $RND'_6 = RND_6 \oplus V_{temp4}$, $RND'_7 = RND_7 \oplus V_{temp4}$, $AUTH_s = H(RND_7 || V_{temp4} || T_s)$. Finally, S_s sends the message containing $\langle SK_{s,n}, RND'_6, RND'_7, AUTH_s, T_s \rangle$ to the TA via an open channel.
- When the message arrives from S_s to TA, TA first assesses the message freshness by validating the condition $|TC - T_s| \leq \delta T$. If true, TA further computes pick $RND_6 = RND'_6 \oplus V_{temp3}$, $RND_7 = RND'_7 \oplus V_{temp3}$, and corroborates the validity of the S_s by checking the condition $AUTH_s \stackrel{?}{=} H(RND_7 || V_{temp3} || T_s)$.

8. If true TA , picks a current timestamp T_{TA}^+ and further computes $Z_{TA,n} = Z_{TA,s} \oplus V_{temp2} \oplus V_{temp3}$, $AUTH_{TA,n} = H(RND_3 || V_{temp2} || T_{TA}^+)$, $RND'_6 = RND_6 \oplus V_{temp2} \oplus V_{temp3}$. TA finally transmits the message containing $\langle RND'_6, Z_{TA,n}, AUTH_{TA,n}, T_s, T_{TA}^+ \rangle$ to the SN_n via an open channel.
9. Upon the arrival of messages from the TA , SN_n firstly checks the message's timeliness by inspecting the condition $|TC - T_{TA}^+| \leq \delta T$.
10. If true, SN further checks the condition $AUTH_{TA,n} \stackrel{?}{=} H(RND_3 || V_{temp1} || T_{TA}^+)$ to authenticate the TA .
11. If the condition is validated successfully SN_n will compute $Z_{TA,s} = Z_{TA,n} \oplus V_{temp1}$, $SK_{ns} \stackrel{?}{=} H(RND_6 || Z_{TA,s} \oplus T_s)$. If $SK_{ns} = SK_{sn}$, this key safeguards communication among the SN_n and S_s .

4. Security Evaluation of the Proposed Methodology

4.1. Informal Analysis

The resilience of the proposed method is examined in this section using the adversarial model described in Section 1.2. This analysis identified potential attacks that could be carried out against the protocol and possible countermeasures that could be implemented to mitigate these risks. The informal analysis section describes the protocol's security and identifies areas where further improvements or modifications may be needed, to enhance its security properties. In the subsections that follows, it is demonstrated that our system is safe against well-known threats.

4.1.1. Mutual Authentication (MA)

We can justify the domination that \mathcal{A} has over a legitimate login response, and reply authentication response is significant. Thus, by checking the validity of the sent communications, SN_n and S_s may authenticate one another using TA . Consequently, mutual authentication may be accomplished using the suggested approach.

4.1.2. Untraceability

Randomized nonces (RND_1, \dots, RND_7) and the current time-stamp are selected arbitrarily throughout the authentication process, to ensure each candidate's messages (MSG_1, \dots, MSG_4) are unique. An attacker cannot identify any connections between the messages delivered by SN and cannot identify the source. Additionally, genuine identities or pseudonyms are carried out in a protected one-way collision-resistant hash function, rather than being used openly in communications. The suggested approach can thereby attain untraceability.

4.1.3. Anonymity

In our proposed system, $V_{temp1} = H(SKEY_{TA,n} || SN_{ID_n})$, $RND'_3 = RND_3 \oplus V_{temp1}$, $AUTH_n = H(R_3 || V_{temp1} || T_{SN})$ the sensor's node identification SN_{ID_n} is communicated in a masked form rather than directly in plain text. Additionally, SN_{ID_n} is included in $M1 = \langle RND'_3, AUTH_n, PR_{ID_n}, T_{SN} \rangle$. Due to the difficulty of predicting a 160-bit random integer, it is simply not possible for the attacker \mathcal{A} to determine the true identity of the SN without knowing the mask key MSK . The suggested technique can consequently ensure anonymity.

4.1.4. Session Key Agreement

SN authenticates TA by evaluating the authenticity of $AUTH_{TA,n}$, TA authenticates S_s by evaluating the authenticity of $AUTH_{TA,s}$, and TA authenticates the SN by evaluating the authenticity of $AUTH_n$; thus, SN , TA , and S_s ensure they are entitled to random nonce $RND_3, RND_4, RND_5, RND_6$ and RND_7 . To generate the session key, $SK = SK_{ns} = SK_{s,n} = H(RND_6 || Z_{TA,s} \oplus V_{temp4} || T_s)$ and employ the session key when communicating. The presented approach can thus offer a robust session key agreement, as shown in Table 5.

4.1.5. Sensor Node Impersonation Attack (IA)

Suppose an attacker \mathcal{A} appears to be trying to portray a communication on behalf of a SN to a trusted authority TA . \mathcal{A} attains $\{PR_{ID_n}, SN_{ID_n}, SKEY_{TA,n}\}$ from the sensor node memory and $\langle RND'_3, AUTH_n, PR_{ID_n}, T_{SN} \rangle$ while communicating. At this instant, \mathcal{A} attempts to create a response but cannot, since it is unaware of these variables V_{temp1} , R_3 , and RND_3 ; therefore, it is difficult for the adversary to manufacture them. Similarly, trusted authority and server impersonation are impossible due to the secret parameters.

4.1.6. Smart Node Capture Attack

Assume \mathcal{A} has successfully seized a smart node and has obtained its saved and additional data: $\{PR_{ID_n}, SN_{ID_n}, SKEY_{TA,n}\}$. The proper master key MSK and cover-up key X_n cannot be calculated by \mathcal{A} , irrespective of whether it receives information, because the master key MSK and mask key X_n are encoded to be resilient to collisions with the one-way hash function. \mathcal{A} cannot construct a further communication session, as the session key is required for further sessions, and the session key is made up of random numbers and pseudonyms. As a result, the proposed approach can withstand attempts to capture smart nodes.

4.1.7. Replay Attack (RA)

The three entities use the random integers RND_3 , RND_4 , RND_5 , and RND_6 , together with the timestamps T_{SN} , T_{TA} , and T_S , to construct the login messages MSG_1 and MSG_2 , as well as the response messages MSG_3 and MSG_4 . Owing to their recentness, SN , TA , and Ss can distinguish between the acquired and replayed communication, owing to the validity of random nonces and timestamps. As a result, the suggested method can thwart a RA.

4.1.8. Man-in-the-Middle Attack (MITM)

According to Section 3.4, the three participants authenticate each other. As a result, everyone involved can verify one another. As a result, the suggested system can withstand a MITM attack.

4.1.9. Known Session Key Attack

Attacker \mathcal{A} is aware of the SK for an individual session. As is well known, the hash value of the session key SK is created from random numbers and pseudonyms by the parties involved. The robust, resilient to collisions one-way hash function prevents \mathcal{A} from deriving the random integers from SK . However, lacking knowledge of the most recent random values, \mathcal{A} cannot determine the correct SK for all the other sessions. The suggested approach can thus defend against known session key attacks.

4.2. Automated Security Analysis Performed Formally with the AVISPA Tool

AVISPA virtual environment program from [48] is used in this section to formally verify the suggested technique and test its resistance to RA and MITM attacks. The following are the AVISPA simulation phases: (1) The role framework for the role-oriented execution of the protocol processes in a high-level language is provided by the HLPSP (high-level protocol specification language), after which it is interpreted into intermediate format (IF) by its converter HLPSP2IF. (2) The security check is subsequently carried out by the OF (output format) utilizing the translated IF.

Table 5. Authentication and key-agreement process.

| Sensor Node (SN) | Trusted Authority (TA) | Server (S _s) |
|--|---|--|
| Select RND_3 and T_{SN} COMPUTE: $V_{temp1} = H(SKEY_{TA,n} SN_{ID_n})$ $RND_3' = RND_3 \oplus V_{temp1}$ $AUTH_n = H(R_3 V_{temp1} T_{SN})$ $MSG_1 = (RND_3', AUTH_n, PR_{ID_n}, T_{SN})$ $(SN \rightarrow TA)$ | $ TC - T_{SN} \leq \delta T $ and check if PR_{ID_n} exists in DB. IF TRUE: $[SN_{ID_n}, X_n] = DEC_{MSK}[SN_{ID_n}, X_n]$ $SKEY_{TA,n} = H(X_n MSK)$ $V_{temp2} = H(SKEY_{TA,n} SN_{ID_n})$ $RND_3 = RND_3' \oplus V_{temp2}$ $AUTH_n \stackrel{?}{=} H(RND_3 V_{temp2} T_{SN})$ IF TRUE: $[S_{ID_s}, X_s] = DEC_{MSK}[X_s]$ $SKEY_{TA,s} = H(X_s MSK)$ $V_{temp3} = H(SKEY_{TA,s} S_{ID_s} T_{TA})$ Pick RND_4, RND_5 and T_{TA} $AUTH_{TA,s} = H(RND_4 V_{temp3} T_{TA})$ $RND_4' = RND_4 \oplus V_{temp3}$ $Y_{TA,s} = H(RND_5 V_{temp2})$ $Z_{TA,s} = H(SKEY_{TA,s} RND_3 RND_5 Y_{TA,s} MSK) \oplus V_{temp3}$ $MSG_2 = (PR_{ID_s}, RND_4', AUTH_{TA,s}, Z_{TA,s}, T_{TA})$ $(TA \rightarrow S_s)$ | $ TC - T_{TA} \leq \delta T $ and IF TRUE: $V_{temp4} = H(SKEY_{TA,s} S_{ID_s} T_{TA})$ $RND_4 = RND_4' \oplus V_{temp4}$ $AUTH_{TA,s} \stackrel{?}{=} H(RND_4 V_{temp4} T_{TA})$ IF TRUE: Pick RND_6, RND_7 , and T_s $SK_{s,n} = H(RND_6 Z_{TA,s} \oplus V_{temp4} T_s)$ $RND_6' = RND_6 \oplus V_{temp4}$ $RND_7 = RND_7 \oplus V_{temp4}$ $AUTH_s = H(RND_7 V_{temp4} T_s)$ $MSG_3 = (SK_{s,n}, RND_6', RND_7, AUTH_s, T_s)$ $(TA \leftarrow S_n)$ |
| $ TC - T_{TA}^+ \leq \delta T $ IF TRUE: $AUTH_{TA,n} \stackrel{?}{=} H(RND_3 V_{temp1} T_{TA}^+)$ $RND_6 = RND_6' \oplus V_{temp1}$ $Z_{TA,s} = Z_{TA,n} \oplus V_{temp1}$ $SK_{ns} \stackrel{?}{=} H(RND_6 Z_{TA,s} \oplus T_s)$ | $ TC - T_s \leq \delta T $ IF TRUE: Pick T_{TA}^+ $RND_6 = RND_6' \oplus V_{temp3}$ $RND_7 = RND_7 \oplus V_{temp3}$ $AUTH_s \stackrel{?}{=} H(RND_7 V_{temp3} T_s)$ $Z_{TA,n} = Z_{TA,s} \oplus V_{temp2} \oplus V_{temp3}$ $AUTH_{TA,n} = H(RND_3 V_{temp2} T_{TA}^+)$ $RND_6' = RND_6 \oplus V_{temp2} \oplus V_{temp3}$ $MSG_4 = (RND_6', Z_{TA,n}, AUTH_{TA,n}, T_s, T_{TA}^+)$ $(SN \leftarrow TA)$ | |
| $SK_{ns} (= SK_{sn})$ $(SN_n \text{ \& } S_s \text{ both save the same session-key})$ | | |

The role specifications for the sensor node (SN), server (S), trusted authority (TA), goal, environment, and session are depicted in Figure 3 and 4, accordingly. The AVISPA results, as depicted in Figure 5a,b, demonstrate the presented architecture's resilience against RA and MITM attacks. While the CL-AtSe back end analyzed 244 states in under 0.51 s, the OFMC back end evaluated 4816 nodes with a search time of 58.27 s and a heap depth of 9.

| | |
|---|---|
| <pre> role role_SN(SN, TA, SERVER:agent, MSK,SKntas:symmetric_key, H:hash_func,SND RCV:channel(dy)) played_by SN def= local State:nat, SKEYtan, PRIDn, SIDs, SNIDn, RND1, RND2, RND3, RND6, RND6, Vtemp1, RND6enc, AUTHn, Tsn, Tta, Ts:text init State := 0 transition 1. State=0 /\ RCV(start) => State':=1 /\ SNIDn':=new() /\ secret(SNIDn',sec_6,{TA,SN}) XXX /\ SND({SNIDn'},SKntas) XXX 2. State=1 /\ RCV({PRIDn',H(H(SNIDn'.RND2'))}.MSK)}.SKntas) => State':=2 XXX /\ SKEYtan':=H(H(SNIDn'.RND2')).MSK) /\ Tsn':=new() /\ RND3':=new() /\ Vtemp1':=H(SKEYtan'.SNIDn) /\ RND6enc':=xor(RND3'.Vtemp1'.Tsn') /\ AUTHn':=H(RND3'.Vtemp1'.Tsn') /\ secret(MSK,sec_1,{TA}) /\ secret(RND3',sec_8,{SN,TA}) /\ witness(SN,TA,auth_15,RND3') XXX /\ SND(RND6enc',AUTHn',PRIDn',Tsn') XXX end role </pre> | <pre> role role_SERVER(SERVER, SN,TA:agent,MSK,SKntas:symmetric_ key,H:hash_func,SND,RCV:channel(dy)) played_by SERVER def= local State:nat,SKEYYtan,SKEYtan,PRIDs,SNIDn,SIDs,SKsn,Ytas ,Ztas,Vtemp4,RND1,RND2,RND3,RND4,RND5,RND6,RND7,RND6enc, RND7enc,Vtemp2,Vtemp3,AUTHs,Ts,Tta:text init State := 0 transition 3. State=0 /\ RCV(start)=> State':=1 /\ SIDs':=new() /\ secret(SIDs',sec_7,{TA,SERVER}) /\ SND({SIDs'},SKntas) 4. State=1 /\ RCV({PRIDs'.H(H(SIDs'.RND1')).MSK)}.SKntas)=>State':=2 6. State=2 /\ RCV(PRIDs.xor(RND4',H(H(SIDs'.RND1').MSK).SIDs.Tta')) .H(RND4'.H(H(H(SIDs'.RND1').MSK).SIDs.Tta')).Tta')).H(H(H(SIDs .RND1).MSK).RND3'.RND6'.H(RND6'.H(H(SNIDn'.RND2')).MSK). SNIDn').MSK).Tta') => State':=3 /\ secret(MSK,sec_1,{TA}) /\ Ts':=new() /\ RND6':=new() /\ RND7':=new() /\ SKEYtas':=H(H(SIDs'.RND1').MSK) /\ SKEYtan':=H(H(SNIDn'.RND2')).MSK /\ Vtemp2':=H(SKEYtan '.SNIDn') /\ Vtemp3':=H(SKEYtas'.SIDs.Tta') /\ Vtemp4':=H (SKEYtas'.SIDs.Tta) /\ Ytas':=H(RND5'.Vtemp2') /\ Ztas':=xor(H(SKEYtas'.RND3'.RND5'.Ytas'.MSK),Vtemp3') /\ SKsn':=H(RND6'.xor(Ztas',Vtemp4')).Ts') /\ RND6enc':= xor(RND6',Vtemp4') /\ RND7enc':=xor(RND7',Vtemp4') /\ AUTHs':=H(RND7'.Vtemp4'.Ts') /\ witness(SERVER,TA,auth_18, Ts') /\ secret(RND7',sec_12,{TA,SERVER}) /\ witness(SERVER ,TA,auth_19,RND7') /\ secret(RND6',sec_13,{SN,TA,SERVER}) /\ witness(SERVER,TA,auth_20,RND6') XXX /\ SND(SKsn'.RND6enc'.RND7enc'.AUTHs'.Ts') end role </pre> |
|---|---|

Figure 3. Role specification for sensor node (SN_n) and server (S_s).

| | |
|--|---|
| <pre> role role_TA(TA, SN, SERVER:agent,MSK,SKntas:symmetric_key,H:hash_func, SND,RCV:channel(dy)) played_by TA def= local State:nat,SKEYYtan,SKEYtan,PRIDn,AUTHtas,Vtemp2,Vtemp3,Xn,Xs,Ytas,SIDs, SNIDn,PRIDs,Ztas,Ttan,AUTHtan,RND1,RND2,RND3,RND4,RND5,RND6,RND7, RND4enc,RND6enc,Tsn,Tta,Ts:text init State := 0 transition 1. State=0 XXX /\ RCV({SNIDn'},SKntas) => State':=1 XXX /\ RND2':=new() /\ PRIDn':=new() /\ Xn':=H(SNIDn'.RND2') /\ SKEYtan':=H(Xn'.MSK) /\ secret(SKEYtan',sec_4,{TA,SN}) XXX /\ SND({PRIDn',SKEYtan'},SKntas) XXX 3. State=1 XXX /\ RCV({SIDs'},SKntas) => State':=2 XXX /\ RND1':=new() /\ PRIDs':=new() /\ Xs':=H(SIDs'.RND1') /\ SKEYtas':=H(Xs'.MSK) /\ secret(SKEYtas',sec_5,{TA,SERVER}) XXX /\ SND({PRIDs',SKEYtas'},SKntas) XXX 5. State=2 XXX /\ RCV(H(xor(RND3'.H(H(H(SNIDn'.RND2').MSK).SNIDn))).H(RND3'.H(H(H SNIDn'.RND2').MSK).SNIDn).Tsn')).PRIDn.Tsn') => State':=3 XXX /\ secret(MSK,sec_1,{TA}) /\ Tta':=new() /\ RND4':=new() /\ RND5':=new() /\ witness(TA,SERVER,auth_16,Tta') /\ Vtemp3':=H(SKEYtas.SIDs.Tta') /\ AUTHtas':=H(RND4'.Vtemp3'.Tta') /\ Vtemp2':=H(SKEYtan.SNIDn) /\ RND4enc':=xor(RND4',Vtemp3') /\ Ytas':=H(RND5'.Vtemp2') /\ Ztas':=xor(H(SKEYtas.RND3'.RND5'.Ytas'.MSK),Vtemp3') /\ secret(RND5',sec_10,{TA}) /\ secret(RND4',sec_9,{TA,SERVER}) /\ witness(TA,SERVER,auth_17,RND4') XXX /\ SND(PRIDs.RND4enc'.AUTHtas'.Ztas'.Tta') XXX 7. State=3 XXX /\ RCV(H(RND6'.xor(xor(H(H(H(SIDs'.RND1').MSK).RND3.RND5.H(H(H(H SNIDn'.RND2').MSK).SNIDn)).MSK).H(H(H(SIDs'.RND1').MSK).SIDs.Tta')),H(H(H (SIDs'.RND1).MSK).SIDs.Tta')).Ts')).xor(RND6'.H(H(H(SIDs'.RND1).MSK). SIDs.Tta')).xor(RND7'.H(H(H(SIDs'.RND1).MSK).SIDs.Tta')).H(RND7'.H(H(H (SIDs'.RND1).MSK).SIDs.Tta')).Ts')) => State':=4 XXX /\ witness(TA,SN,auth_23,Ts') /\ secret(MSK,sec_1,{TA}) /\ witness(TA,SN,auth_21,RND6') /\ Tta':=new() /\ RND6enc':=xor(RND6',xor(Vtemp2,Vtemp3)) /\ Ztan':=xor(Ztas,xor (Vtemp2,Vtemp3)) /\ AUTHtan':=H(RND3.Vtemp2.Tta') XXX /\ SND(RND6enc'.Ztan'.AUTHtan'.Ts'.Tta') XXX end role </pre> | <pre> role session(SERVER,SN,TA:agent,MSK,SKntas:symmetric_key,H:hash_func) def= local SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy) composition role_SERVER(SERVER,SN,TA,MSK,SKntas,H,SND3,RCV3) /\ role_TA(TA,SN,SERVER,MSK,SKntas,H,SND2,RCV2) /\ role_SN(SN,TA,SERVER,MSK,SKntas,H,SND1,RCV1) end role role environment() def= const ta, server,sensor:agent, hashes:hash_func, tn,tta,ts:text, msk,skntas:symmetric_key, sec_1,sec_4,sec_5,sec_6,sec_7,sec_8,sec_9,sec_10,sec_12,sec_13, auth_15,auth_16,auth_17,auth_18,auth_19,auth_20,auth_21,auth_23: protocol_id intruder_knowledge = {hashes,tn,tta,ts} composition session(server,sensor,ta,msk,skntas,hashes) /\ session(i,sensor,ta,msk,skntas,hashes) /\ session(server,sensor,i,msk,skntas,hashes) /\ session(server,i,ta,msk,skntas,hashes) end role goal secrecy_of sec_1 secrecy_of sec_4 secrecy_of sec_5 secrecy_of sec_6 secrecy_of sec_7 secrecy_of sec_8 secrecy_of sec_9 secrecy_of sec_10 secrecy_of sec_12 secrecy_of sec_13 authentication_on auth_15 authentication_on auth_16 authentication_on auth_17 authentication_on auth_18 authentication_on auth_19 authentication_on auth_20 authentication_on auth_21 authentication_on auth_23 end goal environment() </pre> |
|--|---|

(a) TA

(b) Environment, goal, and session

Figure 4. Role specification for TA and Environment, goal, and session.

| | |
|---|--|
| <pre> % OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/avispa_hlpel.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 58.27s visitedNodes: 4816 nodes depth: 9 plies </pre> | <pre> SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/avispa_hlpel.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 244 states Reachable : 48 states Translation: 0.51 seconds Computation: 0.00 seconds </pre> |
| (a) OFMC | (b) CL-AtSe |

Figure 5. The findings of the study performed utilizing OFMC and the CL-AtSe back end.

5. Comparative Analysis

This section comprehensively compares different security protocols that can address security requirements and challenges. The analysis compares the schemes with respect to their security and performance characteristics. This section also compares the potential risks of each protocol, such as susceptibility to attacks or other security vulnerabilities. A comparative analysis is an important part of any security document, as it provides decision-makers with a clear understanding of the advantages and disadvantages of different security protocols and helps them make informed decisions about which protocol best suits their needs. Comparisons between the new SLMAS and previously established protocols [49–52] are provided in this study.

5.1. Functionality Comparison

Table 6 shows a functional comparison of the introduced and comparable protocols. Table 6 makes it clear that the new protocol provides higher security in comparison to the existing relevant protocols and also provides more advanced security features. Here, ✓ indicates whether a certain feature is present or if a protocol can withstand an attack, and ✗ indicates whether a specific feature is absent or whether a protocol cannot withstand an attack.

Table 6. Functionality characteristic comparison.

| | [49] | [50] | [51] | [52] | Our |
|--|------|------|------|------|-----|
| Sensor node anonymity | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ephemeral Secret Leakage (ESL) | ✓ | ✗ | ✓ | ✗ | ✓ |
| Protection against RA | ✓ | ✓ | ✓ | ✓ | ✓ |
| Efficient protocol design | ✗ | ✓ | ✓ | ✗ | ✓ |
| Stolen verifier attack | ✗ | ✓ | ✗ | ✗ | ✓ |
| Stolen SN attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Untraceability | ✓ | ✓ | ✓ | ✓ | ✓ |
| Parallel SK attack | ✗ | ✓ | ✓ | ✓ | ✓ |
| Reply Attack | ✓ | ✗ | ✗ | ✓ | ✓ |
| Sensor nodes IA | ✗ | ✓ | ✓ | ✓ | ✓ |
| Server IA | ✓ | ✓ | ✓ | ✓ | ✓ |
| MITM attack | ✓ | ✓ | ✗ | ✓ | ✓ |
| Insider attack | ✓ | ✓ | ✗ | ✗ | ✓ |
| Mutual Authentication | ✓ | ✓ | ✗ | ✓ | ✓ |
| Formal automated security verification | ✓ | ✓ | ✓ | ✗ | ✓ |

5.2. Communication Analysis

The estimated cost of communication is shown in Table 7. For comparison, the size of the identities is assumed to be 16 bytes long, timestamps to be 4 bytes long, SHA-1 hash outputs to be 20 bytes long [53], the cost for an ECC point is $(20 + 20) = 40$ bytes, random numbers to be 20 bytes long, and the symmetric encryption/decryption block size is 16 bytes [54,55]. Figure 6 also displays the communication costs of the various protocols.

The communication cost of $MSG_1 = \langle RND'_3, AUTH_n, PR_{ID_n}, T_{SN} \rangle$ is $\langle 20 + 20 + 16 + 4 = 60 \rangle$ bytes, $MSG_2 = \langle PR_{ID_s}, RND'_4, AUTH_{TA,s}, Z_{TA,s}, T_{TA} \rangle$ is $\langle 16 + 20 + 20 + 20 + 4 = 80 \rangle$ bytes, $MSG_3 = \langle SK_{sn}, RND'_6, RND'_7, AUTH_s, T_s \rangle$ is $\langle 20 + 20 + 20 + 20 + 4 = 84 \rangle$ bytes, and $MSG_4 = \langle RND'_6, Z_{TA,n}, AUTH_{TA,n}, T_s, T_{TA}^+ \rangle$ is $\langle 20 + 20 + 20 + 4 + 4 = 68 \rangle$ bytes, respectively.

Table 7. Cost comparison for communication.

| Protocols | # of Messages | # of Bytes |
|-------------------------|---------------|------------------------------|
| Banerjee et al. [49] | 4 | $(68 + 40 + 56 + 72) = 236$ |
| Fakroon et al. [50] | 4 | $(100 + 52 + 52 + 84) = 288$ |
| Nikooghadam et al. [51] | 4 | $(132 + 64 + 40 + 68) = 304$ |
| Moghadam et al. [52] | 4 | $(60 + 64 + 44 + 40) = 208$ |
| Our | 4 | $(60 + 80 + 84 + 68) = 292$ |

Adding together all these results, 292 bytes is the overall communication expense of the newly implemented protocol during the login and authentication process. Table 7 denotes that the presented protocol's communication cost is minimized as compared to the protocol in [51], and somewhat greater than that in [49,50,52], but this is acceptable because the presented protocol offers greater security than all previously compared protocols, as shown in the Table 6. Figure 6 also illustrates the new protocol's communication costs.

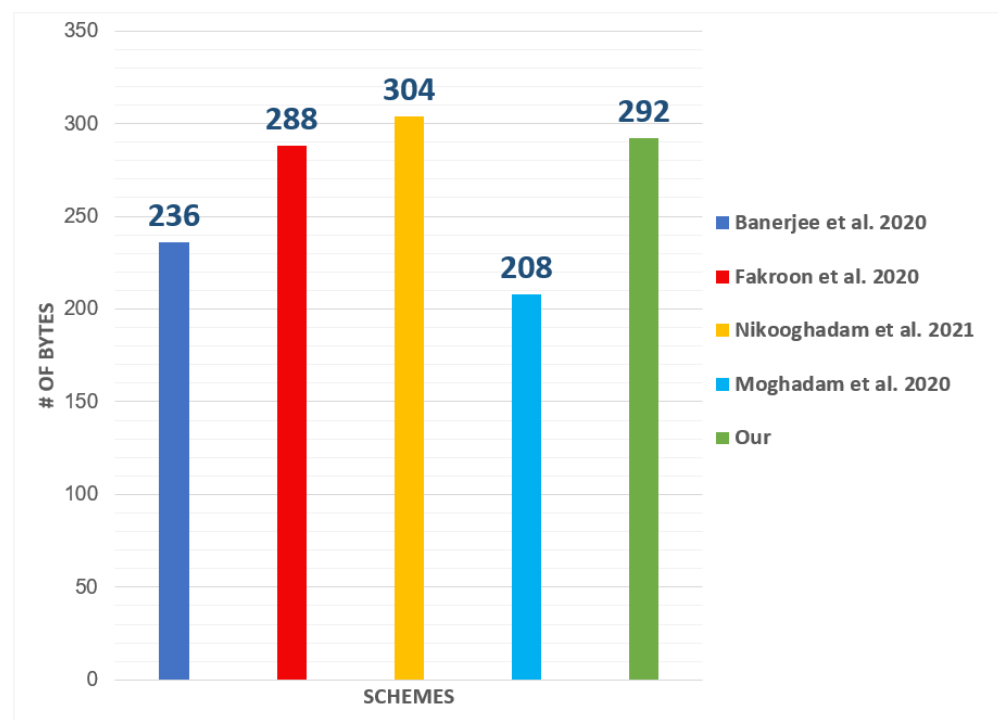


Figure 6. Communication cost comparison of [49], [50], [51], and [52].

5.3. Computation Analysis

This part evaluates the cost of computation for the various schemes. The computation times for the ECC multiplication, symmetric encryption/decryption, fuzzy extractor, bilinear pairing, and hash are 13.405, 1.657, 13.405, 32.713, and 0.056 ms, respectively, where $T_{fe} \approx T_{ecm}$ as explained in [49]. Table 8 also demonstrates the estimated time frames required for specific cryptographic operations and associated notations.

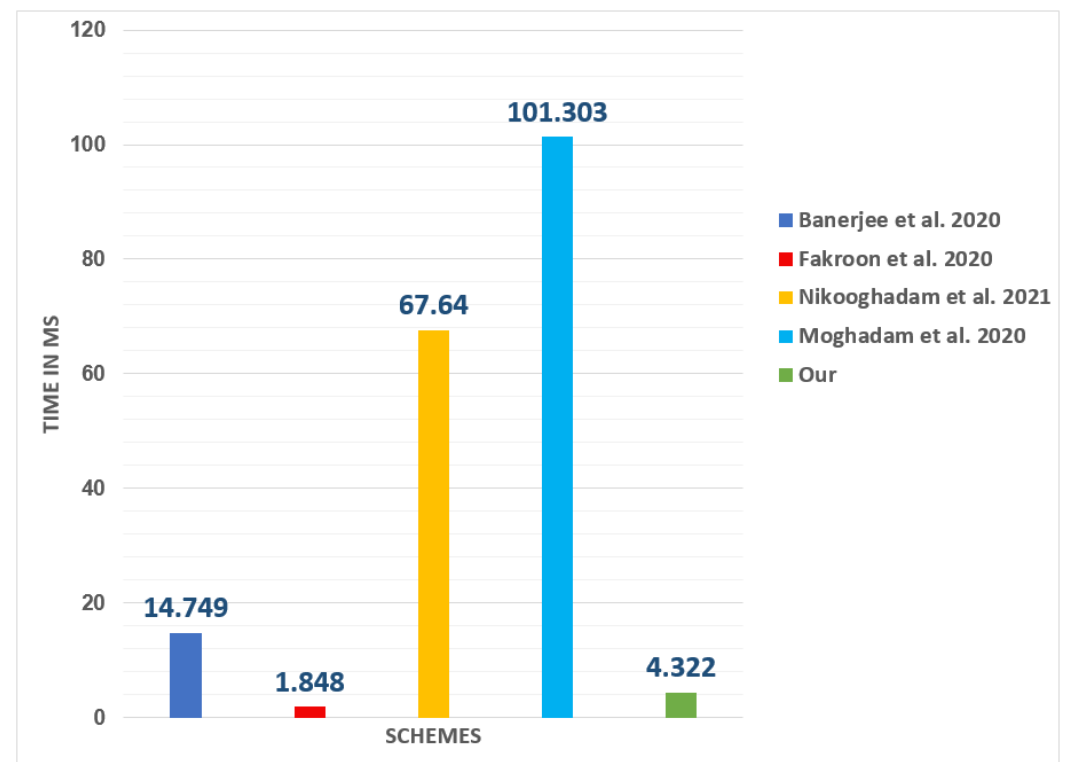
The computing cost of the presented work is a little higher than [50], as seen in Table 9. However, the fact that these studies lack some security features in comparison to the security offered by the new scheme, as indicated in Table 6, justifies this difference. The computing cost of the newly implemented protocol is also shown in Figure 7.

Table 8. Approximated computation costs for different procedures.

| Notations | Explanation | ≈ Computation Time |
|-----------|--------------------------|--------------------|
| T_H | Hash function | 0.056 ms |
| T_{SM} | Scalar multiplication | 13.405 ms |
| T_{SED} | Symmetric enc/dec | 1.657 ms |
| T_{FE} | Fuzzy extractor function | 13.405 ms |
| T_{BP} | Bilinear pairing | 32.713 ms |

Table 9. Computation cost comparison.

| Protocol | User/Mobile Device | TA/RA/Server | Gateway | SD/SN | Total Cost |
|-------------------------|---|---------------------------------|---|--------------------------------|-------------|
| Banerjee et al. [49] | $10T_H + 1T_{FE}$ ≈13.965 ms | — | $10T_h$ ≈0.56 ms | $4T_h$ ≈0.224 ms | ≈14.749 ms |
| Fakroon et al. [50] | $4T_h$ ≈0.224 ms | — | $5T_h$ ≈0.28 ms | $24T_H$ ≈1.344 ms | ≈1.848 ms |
| Nikooghadam et al. [51] | $9T_h + 2T_{SM}$ ≈27.314 ms | $2T_H + 1T_{SM}$ ≈13.517 ms | — | $3T_H + 2T_{SM}$ ≈26.81 ms | ≈67.64 ms |
| Moghadam et al. [52] | $5T_H + 3T_{SM} + 2T_{SED}$ ≈43.809 ms | — | $5T_H + 3T_{SM} + 2T_{SED}$ ≈43.809 ms | $3T_H + 1T_{SM}$ ≈13.685 ms | ≈101.303 ms |
| Our | — | $14T_H + 2T_{SEC}$ ≈4.098 ms | — | $4T_H$ ≈0.224 ms | ≈4.322 ms |

**Figure 7.** Computation cost comparison [49], [50], [51], and [52].

6. Limitations and Challenges

This section presents the limitations and challenges encountered while developing this authentication protocol for WSNs. It critically assesses the proposed approach and identifies issues or challenges that may limit its effectiveness.

1. **Limited Resources:** WSNs have resource constraints regarding memory, processing power, and battery life. Therefore, security protocols must be designed with these limitations in mind, to ensure they do not consume too much energy or memory.

2. **Limited Physical Security:** Sensor nodes are often deployed in unattended environments vulnerable to physical attacks. Security protocols should be designed to withstand physical attacks, such as tampering, destruction, or theft of the nodes.
3. **Communication Overhead:** Security protocols often introduce additional communication overheads, leading to increased latency, energy consumption, and reduced network performance. Therefore, it is essential to design security protocols that minimize communication overheads, while providing sufficient security.

7. Conclusions

This article reviewed the current state-of-the-art user authentication mechanisms for WSNs and briefly discussed their benefits and drawbacks. We presented a user authentication scheme for an intelligent wheelchair, which protects wheelchair-transmitted data privacy, while enhancing data protection and effectiveness. Since symmetric keys are used, the suggested approach has minimal communication and computational overheads. Using the well-known AVISPA simulation platform, it was also demonstrated that the offered scheme is resilient against passive and active attacks. The suggested approach is suitable for various intelligent wheelchair scenarios, such as Hajj and Umrah pilgrims [9], since it has low communication and computation running costs and provides robust security. The communication cost of the proposed scheme is 23.97%, 1.39%, and 40.41% higher than the Banerjee et al. [49], Fakroon et al. [50], and Moghadam et al. [52], respectively; and 4.12% less than Nikooghadam et al. [51]. Similarly, the computation cost is 120.67%, 1366.57%, and 2142.97% less than Banerjee et al. [49], Nikooghadam et al. [51], and Moghadam et al. [52], and 85.35% higher than Fakroon et al. [50], respectively.

Author Contributions: R.L.A.: conceptualization, methodology, A.A.A.: literature review; A.G.: supervision; M.L.: computation cost; R.L.A.: formal security analysis; M.L.: draft formatting and draft preparation, A.A.A.: project administration. All authors have read and agreed to the published version of the manuscript.

Funding: The work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant number (UJ-21-ICI-3). The authors, therefore, acknowledge with thanks the University of Jeddah for technical and financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interests.

References

1. Oueida, S.; Kotb, Y.; Aloqaily, M.; Jararweh, Y.; Baker, T. An edge computing based smart healthcare framework for resource management. *Sensors* **2018**, *18*, 4307. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Al Shabibi, M.A.K.; Kesavan, S.M. Iot based smart wheelchair for disabled people. In Proceedings of the 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 30–31 July 2021; pp. 1–6.
3. Bourgeois-Doyle, R.I. *George J. Klein: The Great Inventor*; Number 2; NRC Research Press: Ottawa, ON, Canada, 2004.
4. Khan, N.A.; Jhanjhi, N.; Brohi, S.N.; Almazroi, A.A.; Almazroi, A.A. A secure communication protocol for unmanned aerial vehicles. *Comput. Mater. Contin.* **2022**, *70*, 601–618.
5. Rahimunnisa, K.; Atchaya, M.; Arunachalam, B.; Divyaa, V. AI-based smart and intelligent wheelchair. *J. Appl. Res. Technol.* **2020**, *18*, 362–367. [\[CrossRef\]](#)
6. Haseeb-ur Rehman, R.M.A.; Liaqat, M.; Aman, A.H.M.; Almazroi, A.A.; Hasan, M.K.; Ali, Z.; Ali, R.L. LR-AKAP: A Lightweight and Robust Security Protocol for Smart Home Environments. *Sensors* **2022**, *22*, 6902. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Kumar, D.; Maurya, A.K.; Baranwal, G. IoT services in healthcare industry with fog/edge and cloud computing. In *IoT-Based Data Analytics for the Healthcare Industry*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 81–103.
8. Udaya, R.V.; Poojasree, S. An IOT Driven Eyeball And Gesture-Controlled Smart Wheelchair System for Disabled Person. In Proceedings of the 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 25–26 March 2022; Volume 1, pp. 1287–1291.
9. Mohamed, M.N. *Hajj & Umrah from A to Z*; IslamKotob: Riyadh, Saudi Arabia, 1996.

10. Liaqat, M.; Gani, A.; Anisi, M.H.; Ab Hamid, S.H.; Akhunzada, A.; Khan, M.K.; Ali, R.L. Distance-based and low energy adaptive clustering protocol for wireless sensor networks. *PLoS ONE* **2016**, *11*, e0161340. [\[CrossRef\]](#)
11. Haseeb-Ur-Rehman, R.M.A.; Liaqat, M.; Aman, A.H.M.; Ab Hamid, S.H.; Ali, R.L.; Shuja, J.; Khan, M.K. Sensor cloud frameworks: State-of-the-art, taxonomy, and research issues. *IEEE Sens. J.* **2021**, *21*, 22347–22370. [\[CrossRef\]](#)
12. Ali, Z.; Hussain, S.; Rehman, R.H.U.; Munshi, A.; Liaqat, M.; Kumar, N.; Chaudhry, S.A. ITSSAKA-MS: An improved three-factor symmetric-key based secure AKA scheme for multi-server environments. *IEEE Access* **2020**, *8*, 107993–108003. [\[CrossRef\]](#)
13. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [\[CrossRef\]](#)
14. Chaudhry, S.A.; Shon, T.; Al-Turjman, F.; Alsharif, M.H. Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems. *Comput. Commun.* **2020**, *153*, 527–537. [\[CrossRef\]](#)
15. Ali, Z.; Ghani, A.; Khan, I.; Chaudhry, S.A.; Islam, S.H.; Giri, D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *J. Inf. Secur. Appl.* **2020**, *52*, 102502. [\[CrossRef\]](#)
16. Ghani, A.; Mansoor, K.; Mehmood, S.; Chaudhry, S.A.; Rahman, A.U.; Saqib, M.N. Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key. *Int. J. Commun. Syst.* **2019**, *32*, e4139. [\[CrossRef\]](#)
17. Lamport, L. Password authentication with insecure communication. *Commun. ACM* **1981**, *24*, 770–772. [\[CrossRef\]](#)
18. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090. [\[CrossRef\]](#)
19. Khan, M.K.; Alghathbar, K. Cryptanalysis and Security Improvements of ‘Two-Factor User Authentication in Wireless Sensor Networks’. *Sensors* **2010**, *10*, 2450–2459. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Chen, T.H.; Shih, W.K. A Robust Mutual Authentication Protocol for Wireless Sensor Networks. *ETRI J.* **2010**, *32*, 704–712. [\[CrossRef\]](#)
21. Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
22. Aysu, A.; Gulcan, E.; Moriyama, D.; Schaumont, P.; Yung, M. End-to-end design of a PUF-based privacy preserving authentication protocol. In *Cryptographic Hardware and Embedded Systems—CHES 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 556–576.
23. Majzoobi, M.; Elnably, A.; Koushanfar, F. FPGA Time-Bounded Unclonable Authentication. In *Information Hiding*; Böhme, R., Fong, P.W.L., Safavi-Naini, R., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–16. [\[CrossRef\]](#)
24. Rührmair, U. SIMPL systems: On a public key variant of physical unclonable functions. In *SOFSEM 2011: Theory and Practice of Computer Science*; SOFSEM 2011; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 6543.
25. Bassil, R.; El-Beaino, W.; Kayssi, A.; Chehab, A. A PUF-based ultra-lightweight mutual-authentication RFID protocol. In Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions, Abu Dhabi, United Arab Emirates, 11–14 December 2011; pp. 495–499.
26. Kulseng, L.; Yu, Z.; Wei, Y.; Guan, Y. Lightweight mutual authentication and ownership transfer for RFID systems. In Proceedings of the 2010 IEEE Infocom, San Diego, CA, USA, 14–19 March 2010; pp. 1–5.
27. Zhang, X.; Huang, W.; Xu, H.; Wang, Y. The lightweight ownership transfer protocol using physically unclonable function. *Int. J. Secur. Its Appl.* **2016**, *10*, 115–128. [\[CrossRef\]](#)
28. Jung, S.W.; Jung, S. HRP: A HMAC-based RFID mutual authentication protocol using PUF. In Proceedings of the The International Conference on Information Networking 2013 (ICOIN), Bangkok, Thailand, 28–30 January 2013; pp. 578–582.
29. Lee, Y.S.; Lee, H.J.; Alasaarela, E. Mutual authentication in wireless body sensor networks (WBSN) based on physical unclonable function (PUF). In Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013; pp. 1314–1318.
30. Cortese, P.F.; Gemmiti, F.; Palazzi, B.; Pizzonia, M.; Rimondini, M. Efficient and practical authentication of PUF-based RFID tags in supply chains. In Proceedings of the 2010 IEEE International Conference on RFID-Technology and Applications, Guangzhou, China, 17–19 June 2010; pp. 182–188.
31. Wallrabenstein, J.R. Practical and secure IoT device authentication using physical unclonable functions. In Proceedings of the 2016 IEEE 4th international conference on future internet of things and cloud (FiCloud), Vienna, Austria, 22–24 August 2016; pp. 99–106.
32. Sutar, S.; Raha, A.; Raghunathan, V. D-PUF: An intrinsically reconfigurable DRAM PUF for device authentication in embedded systems. In Proceedings of the 2016 International Conference on Compilers, Architectures, and Sythesis of Embedded Systems (CASES), Pittsburgh, PA, USA, 2–7 October 2016; pp. 1–10. [\[CrossRef\]](#)
33. Challa, S.; Wazid, M.; Das, A.K.; Kumar, N.; Reddy, A.G.; Yoon, E.J.; Yoo, K.Y. Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access* **2017**, *5*, 3028–3043. [\[CrossRef\]](#)
34. Jia, X.; He, D.; Li, L.; Choo, K.K.R. Signature-based three-factor authenticated key exchange for internet of things applications. *Multimed. Tools Appl.* **2018**, *77*, 18355–18382. [\[CrossRef\]](#)
35. 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**, *91*, 244–251. [\[CrossRef\]](#)
36. Masud, M.; Gaba, G.S.; Alqahtani, S.; Muhammad, G.; Gupta, B.B.; Kumar, P.; Ghoneim, A. A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care. *IEEE Internet Things J.* **2020**, *8*, 15694–15703. [\[CrossRef\]](#)

37. Farash, M.S.; Turkanović, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176. [[CrossRef](#)]
38. Amin, R.; Islam, S.H.; Biswas, G.; Khan, M.K.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62. [[CrossRef](#)]
39. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In *Advances in Cryptology—EUROCRYPT 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
40. 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 Netw.* **2014**, *20*, 96–112. [[CrossRef](#)]
41. Amin, R.; Biswas, G. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80. [[CrossRef](#)]
42. Wazid, M.; Das, A.K.; Shetty, S.; JPC Rodrigues, J.; Park, Y. LDKM-LoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment. *Sensors* **2019**, *19*, 5539. [[CrossRef](#)]
43. Hussain, S.; Ullah, S.S.; Uddin, M.; Iqbal, J.; Chen, C.L. A comprehensive survey on signcryption security mechanisms in wireless body area networks. *Sensors* **2022**, *22*, 1072. [[CrossRef](#)] [[PubMed](#)]
44. Shreya, S.; Chatterjee, K.; Singh, A. A smart secure healthcare monitoring system with Internet of Medical Things. *Comput. Electr. Eng.* **2022**, *101*, 107969. [[CrossRef](#)]
45. Sharma, G.; Kalra, S. A lightweight user authentication scheme for cloud-IoT based healthcare services. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2019**, *43*, 619–636. [[CrossRef](#)]
46. Chang, C.C.; Lee, J.S.; Lo, Y.Y.; Liu, Y. A secure authentication scheme for telecare medical information systems. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing, Proceeding of the Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Taiwan, 21–23 November 2016*; Springer: Cham, Switzerland, 2017; Volume 1, pp. 303–312.
47. Wu, F.; Xu, L.; Kumari, S.; Li, X. An improved and provably secure three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2018**, *11*, 1–20. [[CrossRef](#)]
48. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuellar, J.; Drielsma, P.H.; Heám, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285. [[CrossRef](#)]
49. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Park, Y. An Efficient, Anonymous and Robust Authentication Scheme for Smart Home Environments. *Sensors* **2020**, *20*, 1215. [[CrossRef](#)]
50. Fakroon, M.; Alshahrani, M.; Gebali, F.; Traore, I. Secure remote anonymous user authentication scheme for smart home environment. *Internet Things* **2020**, *9*, 100158. [[CrossRef](#)]
51. Nikooghadam, M.; Amintoosi, H.; Islam, S.H.; Moghadam, M.F. A provably secure and lightweight authentication scheme for Internet of Drones for smart city surveillance. *J. Syst. Archit.* **2021**, *115*, 101955. [[CrossRef](#)]
52. Moghadam, M.F.; Nikooghadam, M.; Jabban, M.A.B.A.; Alishahi, M.; Mortazavi, L.; Mohajerzadeh, A. An Efficient Authentication and Key Agreement Scheme Based on ECDH for Wireless Sensor Network. *IEEE Access* **2020**, *8*, 73182–73192. [[CrossRef](#)]
53. Eastlake, D., 3rd; Jones, P. US Secure Hash Algorithm 1 (SHA1). *RFC* **2001**, 3174, 1–22.
54. Alotaibi, M. An Enhanced Symmetric Cryptosystem and Biometric-Based Anonymous User Authentication and Session Key Establishment Scheme for WSN. *IEEE Access* **2018**, *6*, 70072–70087. [[CrossRef](#)]
55. Ali, Z.; Alzahrani, B.A.; Barnawi, A.; Al-Barakati, A.; Vijayakumar, P.; Chaudhry, S.A. TC-PSLAP: Temporal Credential-Based Provably Secure and Lightweight Authentication Protocol for IoT-Enabled Drone Environments. *Secur. Commun. Netw.* **2021**, *2021*, 9919460. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.