Computer Networks 258 (2025) 110938

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

ERASMIS: An ECC-based robust authentication protocol suitable for medical IoT systems

Mohammad Reza Servati [a], Masoumeh Safkhani [a,*], Amir Masoud Rahmani [b], Mehdi Hosseinzadeh [c,d,**]

[a] Department of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, 16788-15811, Iran
[b] Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan
[c] School of Computer Science, Duy Tan University, Da Nang, Viet Nam
[d] Jadara University Research Center, Jadara University, Irbid, Jordan

ARTICLE INFO

Keywords:
ECC
Medical wireless sensor network
Scyther
ProVerif
Authentication protocol

ABSTRACT

The "Internet of Things" (IoT) refers to the interconnection of physical devices that transmit and receive data over a network infrastructure. Wireless Sensor Networks (WSNs) are a key component of this infrastructure, facilitating data exchange through wireless channels. They are widely used in healthcare, transportation, smart home monitoring, and other applications. As IoT networks continue to evolve rapidly, security and privacy have become critical concerns. Security is essential within these systems, and privacy is especially important, as data transmitted over wireless channels can be intercepted, tracked, or tampered with.

In recent years, many authentication protocols have been proposed by researchers and experts. However, some of these protocols lack essential security features and fail to provide robust protection against various active and passive attacks. In this paper, we introduce ERASMIS, an authentication protocol based on elliptic curve cryptography (ECC) designed specifically for healthcare IoT systems. We explain how ERASMIS ensures security and present both informal and formal proofs of its security. Our formal security analyses, conducted with Real or Random (RoR) model and also tools such as Scyther and ProVerif demonstrate that the proposed protocol is resilient against numerous attacks while being more efficient than comparable schemes, with low computational and communication overhead. Furthermore, we have developed a Python implementation of the proposed protocol to evaluate its performance in real-world scenarios.

1. Introduction

The Internet of Things (IoT) infrastructure comprises a vast network of sensor nodes that operate on limited power and are spatially separated, forming an intelligent and autonomous ad hoc communication system. It can be deployed in any unfamiliar area to collect information from its surroundings. Its applications span remote settings such as military battle control, smart agriculture, smart cities, healthcare, wildlife tracking, smart homes, and traffic control [1]. A significant application of IoT is in healthcare, where it enhances patient care through smart surveillance and real-time data monitoring.

Smart healthcare technology represents a transformative approach to patient care, integrating real-time data monitoring, data dissemination, and automated systems to improve health outcomes. A smart medical surveillance system, a fundamental element of this technology, facilitates comprehensive data collection and sharing among healthcare institutions, crucial for precise patient monitoring and medical equipment oversight. By perpetually gathering information regarding a patient's vital signs, activity levels, and overall health status, these systems assist healthcare providers in making timely and accurate decisions. This data-driven methodology aids physicians in diagnosing ailments, modifying treatments, and executing individualized care plans. A key component of these smart healthcare systems is physiological monitoring, in which interconnected sensor networks track patients' vital signs and relay critical information to medical professionals [2]. In such systems, physiological sensor networks interconnected with communication networks have considerably improved patient monitoring in healthcare applications. Blood pressure, pulse rate, blood sugar level, breathing rates, ECG patterns, and blood oxygen levels are among the

* Corresponding author.
** Corresponding author at: School of Computer Science, Duy Tan University, Da Nang, Viet Nam.
E-mail addresses: Mohammadreza.Servati75@gmail.com (M.R. Servati), Safkhani@sru.ac.ir (M. Safkhani), Rahmania@yuntech.edu.tw (A.M. Rahmani), Mehdihosseinzadeh@duytan.edu.vn (M. Hosseinzadeh).

https://doi.org/10.1016/j.comnet.2024.110938
Received 2 September 2024; Received in revised form 11 November 2024; Accepted 18 November 2024
Available online 30 December 2024

Table 1
Common security requirements in authentication protocols.

Requirement	Description
Mutual Authentication	Ensures that only authorized users receive medical treatments. The authentication mechanism for WBANs must provide mutual authentication between the user and the service provider.
Anonymity	Protects the confidentiality of the user by preventing attackers from deducing the client's true identity from captured interactions.
Untraceability	Anonymity alone is insufficient to protect client privacy; location confidentiality is also necessary. Therefore, the authentication protocol must ensure non-traceability, meaning that no one, especially the application provider and network administrator, should be able to trace the client.
Session Key Agreement	To protect the privacy, security, and non-repudiation of medical data over WBANs, shared private keys must be established between the user and the application service.
Perfect Forward Secrecy	A cryptographic protocol exhibits Perfect Forward Secrecy if, for any two sessions S_1 and S_2 established at different times, the session key K_{S_1} used in session S_1 remains secure even if the long-term private keys of the participants are compromised after the session has concluded.
Attack Resistance	Due to the insecure communication channel, WBAN authentication is susceptible to various attacks, including all kinds of impersonation, replay, and man-in-the-middle attacks. To ensure security, the authentication scheme must withstand these types of attacks.

physiological parameters monitored by a vital sign surveillance system. The development of miniaturized portable sensors and communication systems has transformed healthcare surveillance into wireless medical sensor networks with potential advantages, including patient mobility and easy access to the patient's health data by doctors and other medical experts [3].

As smart healthcare systems rely on sensitive and critical patient data, ensuring the security and integrity of this information is paramount to prevent any misuse or misinterpretation. Protecting sensitive patient data gathered from body sensors is crucial, as these remote health monitoring networks are vulnerable to attacks such as manipulation, eavesdropping, and spoofing. Ensuring that critical information reaches the appropriate clinician is also essential. Medical professionals must trust the data obtained from the patient's body sensors. They must specifically ensure that the information comes from the correct person and was collected using the proper equipment. If this is not made clear, important healthcare decisions may be made without considering the patient or based on incorrect information. Therefore, the system must reduce the risk of associating data with the wrong patient. Furthermore, it is essential to prevent scenarios where fraudulent users send false data from legitimate devices or where device identifiers are cloned onto other devices. Additionally, other prevalent security requirements in this domain are explained in Table 1.

Therefore, for successful verification, it is important to associate the device ID with the user ID. Additionally, identity fraud is a common attack, making it essential to protect patients' identities from unauthorized users. It is also crucial to conceal the patient's identification from potential attackers, so only verified entities can determine the patient's true identity [4].

Sureshkumar et al. [5]'s proposed system provides an end-to-end authentication mechanism across multiple entities, including the patient or user, the gateway device, and the healthcare provider. This mechanism allows authorized healthcare providers to access body information from sensors. The sensors attached to the patient's body automatically send notifications to the gateway node, which uses wireless technologies like GSM (Global System for Mobile Communication) to serve as an interface between the patient and the medical server. In this design, a mobile phone functions as the gateway, providing a seamless connection to the medical center and enabling continuous care for the patient beyond clinical settings. The system is suitable for patient monitoring in clinical settings, homes, and hospitals.

In medical IoT systems, authentication schemes are essential to safeguard patient data and secure device interactions. Over the years, various protocols have been proposed, some of which have successfully provided adequate security against common attacks. To address the specific security requirements of medical IoT, this research introduces ERASMIS, an advanced ECC-based authentication protocol designed for IoT environments. The protocol employs a gateway node to authenticate all interacting entities, including users and wearable sensors. ERASMIS supports the creation of a shared session key for each communication session after authentication, allowing secure information sharing. Notably, it encrypts users' biometric features to maintain anonymity, a key privacy requirement. We employ the real-or-random (RoR) proof method to rigorously evaluate ERASMIS's security. This methodology objectively distinguishes between actual protocol executions and random simulations, providing a thorough analysis of the protocol's resilience to various security attacks.

1.1. Motivation and contributions

Data play an important and crucial role in smart environments such as healthcare systems, buildings, and military devices. Sensors transmit sensitive data to gateways via public or secure channels, which can contain information about a patient in a smart environment such as a hospital. It is necessary to secure this data against active and passive attacks. In this paper, we suggest an ECC-based authentication scheme to employ in IoT-based healthcare systems.

1. This paper proposes an ECC-based authentication scheme called ERASMIS: An ECC-based Robust Authentication Protocol for Medical IoT Systems, for IoT medical wireless sensor networks (MWSNs). ERASMIS offers several advantages, including providing adequate security against well-known passive and active attacks.
2. Our security analysis of ERASMIS involved both informal and formal techniques using tools such as Scyther and ProVerif. The results demonstrate that the proposed protocol ensures appropriate security against different types of attacks. It is worth noting that we utilized two standard and compromise versions of the Scyther tool, and we also use the RoR model, which is a suitable technique for manually proving security properties through logical deductions.
3. We also have implemented our proposed protocol using Python 3.9 to show how to use it practically.
4. An important step in validating a new cryptographic protocol is comparing its performance to existing solutions. We conducted a thorough performance evaluation of ERASMIS against numerous state-of-the-art methods addressing similar use cases. This allowed us to assess how well the proposed protocol fulfills its design goals. Our performance analysis results provided robust evidence that ERASMIS achieves its objectives of streamlined functionality suitable for resource-constrained modern applications.

1.2. Paper organization

The structure of this paper is as follows: Section 2 summarizes related work, while Section 3 discusses the architecture of the health

monitoring system along with its security requirements. A new ECC-based authentication protocol named ERASMIS is introduced in Section 4. An analysis of both informal and formal security evaluations of this protocol is presented in Section 5. Section 6 details the implementation of the proposed protocol in Python 3.9 and reports its results. The performance of this proposed scheme is assessed in Section 7 in comparison to other similar protocols. Finally, Section 8 concludes the paper by outlining the limitations of the proposed protocol and suggesting future research directions.

2. Related work

This section reviews various authentication schemes proposed in previous research, including those based on lightweight cryptography, elliptic curve cryptography (ECC), and radio-frequency identification (RFID) technologies. Table 2 provides a concise overview of the related works. Notably, Malasri and Wang [6] put forth an authentication scheme for healthcare systems but it was found susceptible to denial-of-service (DoS) attacks and security issues. Kumar et al. [7] proposed a protocol for wireless medical sensor networks claiming to satisfy security requirements; however, He et al. [8] identified vulnerabilities to insider attacks and offline password guessing along with a lack of user anonymity. He et al. [8] themselves modified the protocol, yet Li et al. [9] highlighted flaws such as vulnerability to de-synchronization attacks. Wu et al. [10] also demonstrated additional issues including susceptibility to impersonation, password guessing, and sensor node capture attacks. Similarly, Chandrakar and Om [11] proposed a scheme for telemedicine information management systems that was shown prone to impersonation and password guessing attacks.

Other scheme includes a cloud-assisted authentication and privacy preservation strategy for TMIS proposed by Li et al. [12], where the authors claimed their proposed scheme is secure from all known privacy and security attacks, however Kumar et al. [13] identifies its weaknesses, such as its vulnerability against impersonation attacks and patient anonymity contradiction attacks. Zheng et al. [14] suggests an innovative authentication scheme for use in smart campuses, which includes TMIS, but Safkhani and Vasilakos [15] demonstrates convincing impersonation and replay attacks on it. Xiang and Zheng [16] introduces a situation-aware protocol for device authentication in home automation systems with a smart grid, but Oh et al. [17] reveals its vulnerability to attacks, such as stolen smart devices, impersonation, and session key exposure, rendering it unable to provide a secure authentication mechanism. Finally, Shuai et al. [18] proposes an anonymous authentication technique based on ECC for smart home devices. Several lightweight authentication protocols for wearable sensor devices have been proposed in the literature. However, some of these protocols have been found to be insecure and vulnerable to various attacks. For instance, Gupta et al. [19] presented a lightweight authentication protocol for wearable sensor devices, but Hajian et al. [20] showed that it is insecure and vulnerable to privileged insiders, compromised sensing devices, and de-synchronization attacks. On the other hand, Xu et al. [21] proposed a lightweight authentication scheme that is claimed to be robust and secure against a wide range of attacks. Nevertheless, Alzahrani et al. [22] demonstrated that their scheme is not secure and is susceptible to key compromise, replay, and impersonation attacks. Wang et al. [23],Chander and Gopalakrishnan [24] also proposed ultra-lightweight and lightweight schemes, respectively. However, Servati et al. [25] showed that the protocol proposed by Wang et al. [23] is not secure against secret disclosure and de-synchronization attacks. Aghili et al. [26] suggested the SecLAP scheme based on lightweight rotation operations. Unfortunately, Safkhani et al. [27] demonstrated that the SecLap protocol is not immune and is vulnerable to traceability and secret value disclosure attacks. Gabsi et al. [28] proposed another ECC-based protocol for IoT applications that they showed to be secure. Nevertheless, Arslan and Bingöl [29] revealed that their protocol is not secure and is vulnerable to a wide range of attacks such as traceability, forward and backward secrecy contradictions, and tag anonymity contradiction attacks.

Another notable work is Truong et al. [43] proposal of an ECC-based scheme for mobile devices. However, He et al. [44] showed that this scheme lacks security and is also susceptible to impersonation attacks. In addition, Tseng et al. [31] put forth a self-certified public key authentication protocol utilizing hierarchy and dynamic elliptic curve cryptosystems for securing medical data. Nonetheless, their proposed framework was found to be inadequately robust and vulnerable to three common attacks, namely impersonation, replay, and key replication attacks. To elaborate, Amin et al. [33] proposed a solution in 2018. Subsequently Li et al. [45] showed through their work that the scheme

Table 2
A recap of recent related works.

References	Types	Weaknesses	Year
[12]	Lightweight	De-synchronization attack	2018
[11]	ECC-based	Offline-password guessing attack, Impersonation attack	2018
[12]	ECC-based	Violation of patient anonymity, Impersonation attack	2018
[14]	RFID	Replay attack, Impersonation attack	2018
[16]	Lightweight	Session key disclosure, Stolen smart card attack, Impersonation attacks	2020
[17]	Lightweight	–	2021
[30]	ECC-based	–	2021
[19]	Lightweight	Privileged-insider attack, Compromise sensing device attack, De-synchronization attack	2019
[21]	Lightweight	Key compromise attack, Replay attack, Impersonation attack	2019
[22]	Lightweight	–	2021
[23]	Ultra-lightweight	Secret disclosure attack, De-synchronization attack	2022
[26]	Lightweight	Traceability attack, Secret data disclosure attack	2019
[28]	ECC-based	Violation of tag anonymity, Traceability attack, Forward and backward secrecy	2021
[31]	ECC-based	Impersonation attack, Replay attack, Key replication attack	2015
[32]	ECC-based	Replay attack, DoS attack, Forgery attack	2018
[33]	ECC-based	DoS attack, Message-blocking attack	2018
[34]	ECC-based	Traceability attack, De-synchronization attack, Integrity contradiction attack	2019
[35]	Lightweight	Forged certificateless signature	2018
[36]	ECC-based	–	2022
[37]	ECC-based	–	2022
[38]	ECC-based	–	2022
[39]	ECC-based	–	2023
[40]	Lightweight	–	2024
[41]	ECC-based	–	2024
[42]	ECC-based	–	2024

presented by Amin et al. is not robust and can be compromised via DoS and message tampering attacks. Furthermore, Abbasinezhad-Mood and Nikooghadam [37,46] proposed two schemes based on ECC for medical IoT systems and smart grids, which have demonstrated appropriate performance.

Saeed et al. [35] introduced a lightweight scheme for WBANs called L-OOCLS. They claimed that their protocol is secure and exhibits satisfactory performance. However, subsequent research conducted by Shim [47] revealed that their protocol is susceptible to forged certificateless signatures. On the other hand, Vijayakumar et al. [48] proposed a lightweight scheme specifically designed for WBANs. This scheme offers location privacy while maintaining efficient computational and communication costs. Furthermore, Yang et al. [38] have also proposed an efficient ECC-based scheme for this infrastructure, however, it does not guarantee the user anonymity, because an adversary can trace the user.

In 2016, Qian et al. [49] proposed an ECC-based protocol. However, Wei et al. [50] have presented that this protocol is weak against impersonation attacks. Additionally, Challa et al. [32] suggested a three-factor mutual authentication scheme for wireless healthcare sensor networks based on ECC. Nonetheless, Ali et al. [51] showed that this protocol lacks resistance against a plethora of attacks, including replay, denial-of-service (DoS), and forgery attacks. Sureshkumar et al. [34] also proposed an ECC-based authentication protocol for medical Internet of Things (IoT) systems with improved computational and communication performance. However, Servati and Safkhani [39] demonstrated that the envisaged system remains susceptible to numerous attacks, such as de-synchronization, integrity contradiction, and traceability attacks.

In addition, other notable authentication protocol proposals include Jia et al. [36],Jegadeesan et al. [52] schemes, which put forth lightweight and elliptic curve cryptography (ECC)-based authentication frameworks, respectively, for healthcare systems. To elaborate, Jia et al. [36],Jegadeesan et al. [52] independently presented authentication schemes that aimed to address resource constraints and provide security. Specifically, Jegadeesan et al. [52] focused on a lightweight design, while Jia et al. [36] proposed a blockchain-based scheme for healthcare applications. However, these schemes also do not achieve the desired level of security. For instance, in the proposed scheme by Jegadeesan et al. [52], the user could be traced, and in Jia et al. [36]'s scheme, a secure parameter (PID_u) could be extracted. In 2022, two new schemes were introduced. The first scheme, named RC2PAS, was developed by Wang and Liu [53]. The second scheme, proposed by Pu et al. [54], is a lightweight protocol specifically designed for wireless body area networks. This protocol offers a high level of security and efficiency, effectively countering a wide range of attacks. Two other authentication protocols, Shariq et al. [55],Khan et al. [56] proposed authentication schemes suitable for RFID and limited systems however, Hosseinzadeh et al. [40] demonstrated that their proposed protocols are not only secure and strong but also susceptible to secret disclosure attacks. Hosseinzadeh et al. [40] also proposed a new authentication protocol that is efficient and immune against well-known attacks.

Huang [41] proposed an ECC-based three-factor authentication technique specifically designed for wireless sensor networks (WSNs), aimed at enhancing both security and resource efficiency in constrained environments. Additionally, Kumar et al. [57] developed 2F-MASK-VSS, a two-factor authentication mechanism for secure video surveillance, addressing the specific security needs of real-time monitoring systems. Another notable work by Rani and Tripathi [58] introduced a blockchain-based protocol for secure health data sharing among hospitals, leveraging blockchain's decentralized nature to enhance data integrity and access control.

Building on these advancements, Chatterjee et al. [59] focused on improving authentication and key management for IoT-based WSNs using ECC, successfully addressing efficiency challenges in resource-limited scenarios. Meanwhile, Wang et al. [42] proposed a dual-layered approach combining ECC and blockchain for secure IoT device identification, providing both scalability and resilience against security breaches. In the healthcare context, Khan et al. [60] introduced a secure and efficient authentication scheme tailored for e-healthcare systems, aiming to ensure trustworthy data handling and privacy in healthcare data exchanges.

3. System model

This section provides essential context for understanding the typical architecture and operation of a medical wireless sensor network (MWSN). Smart body sensors, illustrated in Fig. 1, continuously collect critical health data, such as temperature and heart rate, and transmit it wirelessly using short-range technologies like Bluetooth or infrared. An MWSN gateway functions as an intermediary, receiving real-time updates on patients' vital signs from mobile sensors. This setup enables continuous patient monitoring across various settings—clinic, home, or while mobile. The gateway grants authorized healthcare providers access to dynamic medical telemetry.

Direct wireless connections between sensors and gateways lack sufficient access restrictions, making them vulnerable to passive and active attacks, which poses significant security and privacy risks. Since personally identifiable medical data is frequently transmitted and used for clinical monitoring, secure authentication of both sensor nodes and gateways is crucial. Additionally, these systems often store collected health data on cloud servers for extended periods, introducing new vulnerabilities during data transmission and storage processes. Consequently, a robust authentication method that addresses the multi-layer interactions between sensors, gateways, and cloud components is vital for ensuring the reliability and security of MWSN networks that manage sensitive patient data in both local and long-term off-site storage.

4. Proposed authentication protocol: ERASMIS

The proposed protocol consists of different phases which are explained below:

4.1. Initialization phase

In this step, the security or system administrator (SA) individually configures each object on the server, with each object having its own credential stored on the server. This scheme utilizes elliptic curve cryptography (ECC) for application in smart devices, defined as $E(F_q) = \langle p, q, a, b, n, G(P)\rangle$ and utilizing a persistent secret key $SA_{Se} \in F_q$. It is worth noting that the notations used in this paper are explained in Table 3, which defines the parameters of the elliptic curve and authentication scheme. To elaborate, ECC with parameter set $E(F_q)$ is employed to secure communications with smart devices, where each device shares a secret SA_{Se} with the security administrator (SA). The SA independently assigns credentials to all objects during the initialization phase.

4.2. Gateway and sensor node enrollment phase

The following steps show how GW_j and SN_k nodes can be registered by an SA.

(1) SA opts a $GWID_j$ identity for GW_j, then calculates and saves the value $S_g = h(SA_{Se} \parallel GWID_j)$ in its storage. Besides, SA saves $\langle S_g, GWID_j\rangle$ in the storage of the GW_j.

(2) The system administrator provides an identity $SNID_K$ for the Kth sensor node and calculates $SSN_K = h(SA_{Se} \parallel SNID_k)$ then keeps $\langle SNID_k, SSN_k, GWID_j\rangle$ and also stores $\langle SSN_k, GWID_j, P\rangle$ in SN_k and $\langle SNID_k, SSN_k, P\rangle$ in GW_j nodes respectively.

It should be noted that shared secret credentials are used to authenticate the gateway (GW) and sensor nodes (SN) during the users

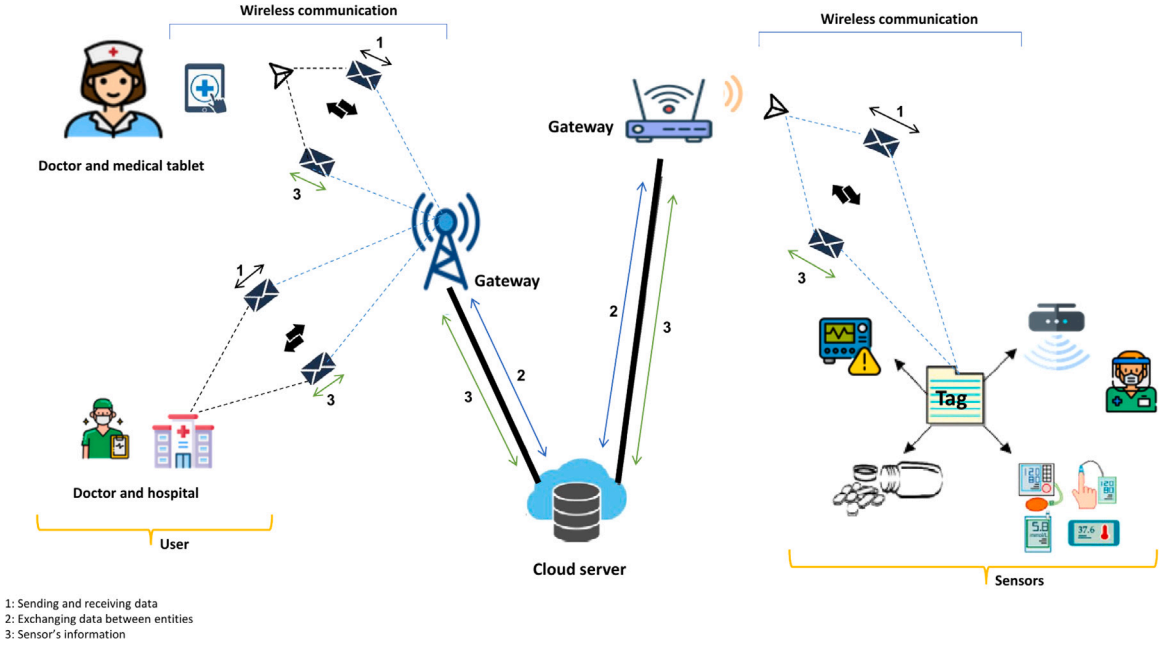


Fig. 1. A standard infrastructure for Medical Wireless Sensor Networks (MWSN).

Table 3
Notations.

Symbol	Description
SA	System administrator
SA_{Se}	Secret value of the system administrator (SA)
S_g	Secret value of the gateway
PK_G	Public key of the gateway, defined as $PK_G = S_g \cdot P$
U_i	User
u_i	Random number generated by the user
SN_K	Kth sensor node
GW_j	jth gateway node
SC	Smart card
$SNID_K$	Identifier of the Kth sensor node (SN_K)
U_i	ith user
ID_i	Identifier of the user U_i
PW_i	Password of the user U_i
$GWID_j$	Identifier of the jth gateway node (GW_j)
SSN_k	Secret value of the Kth sensor node (SN_K)
ECC	Elliptic curve cryptography
SS_A	Secret key of the system administrator (SA)
m	Number of sensors
P	Base point in elliptic curve cryptography (ECC)
n	Defined as $n = p \cdot q$
r_u, r_s	Random numbers
M	Message
F_q	A finite field consisting of q elements $\{0, 1, \ldots, q-1\}$, where q is a prime number
ΔT	Time latency
F_q^*	The set of natural numbers $\{1, \ldots, q-1\}$, i.e. $F_q^*/0$
h	Hash function
$A \stackrel{?}{=} B$	Equality check between A and B
$\oplus$	Exclusive-OR (XOR) operation
$\|\|$	Concatenation operation
Sk_u, Sk_s	Session keys on the user side and server side that must match each other

login and authentication process. These credentials allow the gateway and sensor nodes to verify each other's identities securely. Finally, the system administrator publishes the identities of registered gateway nodes so that users can access them. However, to preserve the privacy of sensor nodes, their private information is kept confidential and not disclosed. The goal is to authenticate the communication between the gateways, sensor nodes, and users while protecting the sensitive data and identities of the sensor nodes from being exposed publicly. This approach aims to provide a secure authentication method without revealing the nodes' internal details or compromising their anonymity.

4.3. User enrollment phase

Upon a successful authentication process, a trustworthy user can access the identified data. This issue occurs when the GW_j reads the sensor's observed data. The user enrollment process in the proposed protocol is accomplished as below:

1. U_i opts ID_i, and PW_i and calculates its bio hashing $b_i = H(B_i)$ with its identity ID_i and password PW_i. After that, U_i computes $HPID_i = h(ID_i \| b_i)$ and $HPW_i = h(PW_i \| b_i)$. Then it transmits $\langle HPW_i, HPID_i, GWID_j \rangle$ to SA.
2. After getting the message, the SA computes $B_1 = h(HPW_i \| HPID_i)$ then makes the $SC = \langle B_1, h(.), P \rangle$ and transmits it to the U_i through a private channel.
3. Upon receiving SC from SA, U_i selects u_i and calculates $B_1^* = h(HPW_i \| HPID_i \| u_i)$. Then U_i makes the $SC = \langle B_1, h(.), P \rangle$.

4.4. Login phase

In this step, we show how the user U_i can access data via the GW_j, but first we show how the user can login to the system. The steps are as follows:

1. U_i inserts the SC into the reader or terminal and enters ID_i, u_i, and a password using biometric data B_i.
2. The smart card computes: $b_i = h(B_i)$, $HPW_i = h(PW_i \| b_i)$, $HPID_i = h(ID_i \| HPW_i)$ and $A_1 = h(HPW_i \| HPID_i \| u_i)$.
3. The SC checks whether $A_1 \stackrel{?}{=} B_1$. When this equivalence is incorrect, the continues of scheme is halted; apart from that, SC opts a random value such a $r_u \in F_q$ and calculates: $A_2 = r_u \cdot P$ and $A_3 = r_u \cdot PK_G$, $A_4 = A_3 \oplus SNID_K$, and $A_5 = h(A_2 \| A_3 \| A_4 \| T_1)$ where T_1 is the current timestamp.

4. The SC transmits login messages $M_1 = \langle A_2, A_4, A_5, T_1 \rangle$ to the GW_j.

4.5. Authentication phase

This stage aims to authenticate the protocol entities and facilitate generation of a shared secret key among the user (U_i), and sensor node (SN_k). The following steps detail the procedure to accomplish this:

1. $U_i \rightarrow GW_j$ When the gateway node receives a login message request, it calculates the time delay $\Delta T = T_2 - T_1$ using the gateway node's time stamp T_2, and if the time latency ΔT is not satisfactory, the login and authentication procedure fails. If the latency is reasonable, GW_j computes $A_3^* = A_2 \cdot S_g$, and computes $A_5^* = h(A_2 \parallel A_3^* \parallel A_4 \parallel T_1)$. Then GW_j checks whether $A_5^* \stackrel{?}{=} A_5$ is or not. If equality is correct, the protocol will be continued; otherwise, it will be halted. If so, GW_j computes $SNID_K = A_4 \oplus A_3^*$ and $A_6 = h(A_2 \parallel SSN_K \parallel GWID_j \parallel T_2)$. Therefore, GW_j sends $\langle A_2, A_6, T_2 \rangle$ to the SN_K.
2. $GW_j \rightarrow SN_k$ Whenever the SN_k gets a message from the GW_j, it uses its own time stamp to verify the time delay of $\Delta T = T_3 - T_2$, and the procedure is halted if this time delay is inaccurate. Otherwise, the SN_k calculates $A_6^* = h(A_2, SSN_K, GWID_j, T_2)$, and checks $A_6^* \stackrel{?}{=} A_6$, if equality is correct, the protocol will be continued; otherwise, it will be halted. SN_k opts its own random value, i.e., r_s, and calculates $A_7 = r_s \cdot .P$, $Sk_s = r_s \cdot A_2$, and $A_8 = h(A_6 \parallel A_2 \parallel SSN_K \parallel GWID_j)$. Finally, it transmits $\langle A_8, A_7 \rangle$ to the GW_j.
3. $SN_K \rightarrow GW_j$
   Following receipt of M_3 from the SN_k, the GW_j confirms the time delay using its computed round trip time, and if this time latency ΔT is not reasonable, the procedure is rejected. Otherwise, it calculates
   $A_8^* = h(A_6 \parallel A_2 \parallel SSN_K \parallel GWID_j)$, and checks whether $A_8 \stackrel{?}{=} A_8^*$ is met or not. If it is so, the GW_j calculates $A_9 = h(A_7 \parallel SNID_K)$. Then GW_j transmits $\langle A_7, A_9 \rangle$ to the U_i.
4. $GW_j \rightarrow U_i$ Following the receipt of messages from the GW_j, the user examines the time delay using its round trip time and computes $A_9^* = h(A_7 \parallel SNID_K)$ and determines whether $A_9 \stackrel{?}{=} A_9^*$ is or not. If it is not, the session key would not be established between entities. If so, the U_i computes $Sk_u = A_7 \cdot r_u$ as its secret key with the SN_K.

The ERASMIS login and authentication phase is also shown in Fig. 2.

4.6. Update and change password phase

Passwords should be modified on a regular basis to improve protection. The ERASMIS protocol takes the following steps into account for this reason:

1. U_i inserts his/her smart card into card reader and imputs his ID_i, PW_i, u_i and B_i.
2. SC computes $b_i = h(B_i)$, $HPW_i = h(PW_i \parallel b_i)$, $HPID_i = h(ID_i \parallel HPW_i)$ and SC also calculates $A_1 = h(HPW_i \parallel HPID_i \parallel u_i)$ and checks whether $B_1 \stackrel{?}{=} A_1$. If it is not so, the SC halts the session; otherwise, the SC lets U_i to enter a new password. U_i's new password, which he or she chooses and enters into the smart card, is represented by PW_i^{new}.
3. The smart card calculates $HPW_i^{new} = h(PW_i^{new} \parallel b_i)$, $HPID_i^{new} = h(ID_i \parallel HPW_i^{new})$, and
   $B_1^{new} = h(HPW_i^{new} \parallel HPID_i^{new} \parallel u_i)$.

Finally, SC replaces B_1, with B_1^{new}. In addition, the described procedure in Algorithm 1 illustrates how the password is updated in the ERASMIS's protocol.

4.7. Sensor node addition

In situations where the SN_K is cracked or hacked by an intruder, or SN_K loses its ability to store energy, it should be replaced. The steps for adding a new sensor node are explained below.

1. SA opts a new identity $SNID_K^{new}$ for the SN_K.
2. After that, SA computes a secret value i.e. $SSN_K = h(SA_{Se} \parallel SNID_K^{new})$.
3. Finally, SA saves $\langle SNID_K, SSN_K \rangle$ into memory of the GW_j and $\langle SSN_K \rangle$ in to memory of SN_K.

Algorithm 1 Algorithm of updating the password in ERASMIS protocol.

Data: Personality information such as $\langle ID_i, PW_i, B_i \rangle$
Result: Update password

1. U_i inserts his SC and puts ID_i, PW_i, u_i and his/her bio-metric B_i.
2. SC calculates $b_i = h(B_i)$, $HPW_i = h(PW_i \| b_i)$ and $HPID_i = h(ID_i \| HPW_i)$.
3. SC also calculates $A_1 = h(HPW_i \| HPID_i \| u_i)$.
4. **if**$(A_1 == B_1)$ **then**
   (a) U_i enters his new password PW_i^{new} and u_i into SC.
   (b) SC calculates $HPW_i^{new} = h(PW_i^{new} \| b_i)$, $HPID_i^{new} = h(ID_i \| HPW_i^{new})$, $B_1^{new} = h(HPW_i^{new} \| HPID_i^{new} \| u_i)$.
   (c) SC substitutes old B_1 with B_1^{new}.
5. **else**
6. SC terminates the session.
7. **end if**

5. ERASMIS security evaluation

5.1. Informal security assessment

5.1.1. Man in the middle attack

In a MITM attack, the attackers position themselves between the two parties involved in the communication. This allows them to intercept and manipulate the messages being exchanged. When the attacker disrupts a specific message, such as M_1 from the initiator, they prevent it from reaching the intended recipient. Given an intruder has access to messages exchanged in the proposed protocol, such as $\langle A_2, A_4, A_5, T_1 \rangle$ which are computed respectively as: $A_2 = r_u \cdot P$, $A_3 = r_u \cdot PK_G$, and $A_4 = A_3 \oplus SNID_k$. Since all transmitted protocol messages are protected using ECC and their integrity also preserved using hash function, the adversary cannot retrieve secret values such as $SNID_k$ and even secret random numbers such as r_u. Moreover, such $\langle A_2, A_4, A_5, T_1 \rangle$ messages cannot be produced by the adversary. Consequently, the suggested authentication scheme is secure and invulnerable to MITM attacks.

5.1.2. Anonymity

In the enrollment phase, the U_i provides his or her ID_i to the SA after using the hash function to mask it with his or her bio metric data, b_i, preventing an insider attacker from obtaining the U_i' identity. If a malicious U_i attempts to deduce another user's ID_i from the transitional messages $HPID_i$, s/he will fail. Because this is obscured by additional unknown secrets such as HPW_i, verifying the correctness of the deduced ID_i is nearly impossible. Therefore, the attacker is unable to discover or guess the user's identity ID_i. As a result, the suggested protocol guarantees the user anonymity.

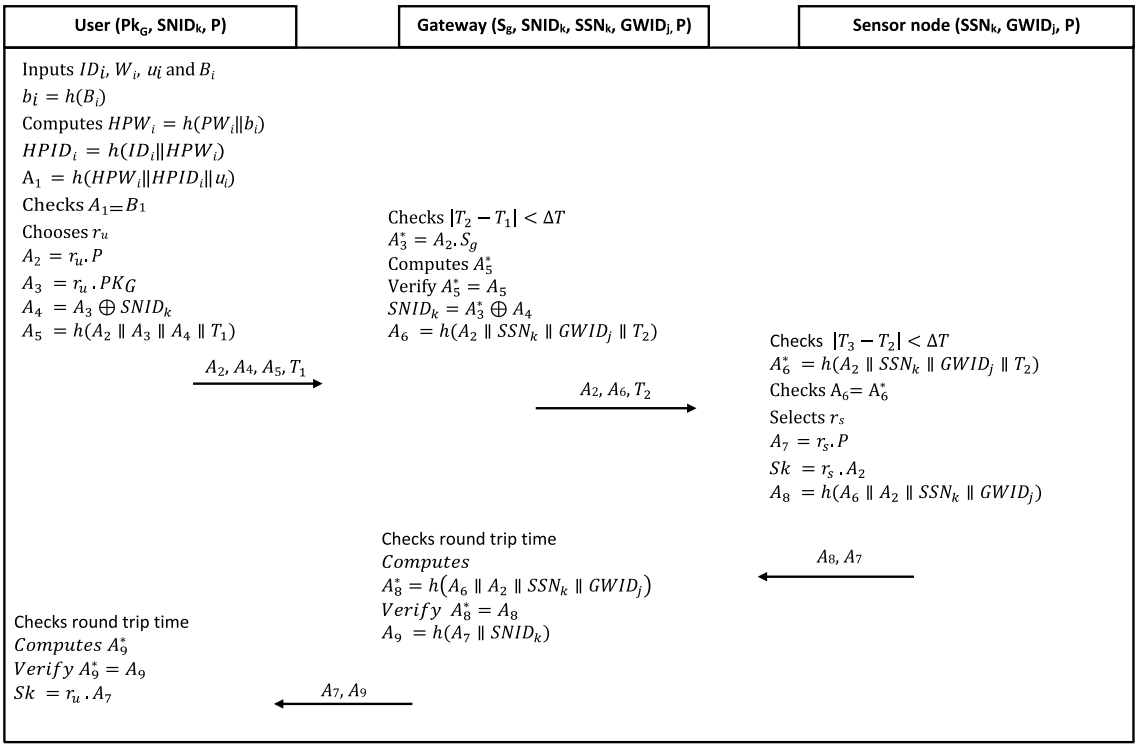


Fig. 2. Proposed authentication scheme: ERASMIS.

5.1.3. Traceability attack

In a traceability analysis attack, the intruder typically collects at least two login authentication messages from different sessions to find correlations and deduce the sender's identity. However, in the proposed scheme, the adversary cannot trace any single user by analyzing one or more public messages due to the use of time-varying parameters. The inclusion of timestamps and random numbers ensures the login messages differ across sessions. Even if an attacker obtains a message like $\langle A_2, A_4, A_5, T_1 \rangle$, values such as the random number r_u or sensor identifier, i.e., $SNID_k$ remain confidential due to their fresh, ephemeral nature. Additionally, the random numbers regenerated independently per session prevent constant entity attributes from being derived. As a result, the proposed authentication mechanism effectively thwarts various traceability attacks since the adversary cannot compute confidential user-specific information across multiple intercepted messages.

5.1.4. De-synchronization attack

If the attack can lead common values in different protocol's parties, will update to different values, it will be interpreted as a de-synchronization attack. Since most of the exchanged messages have been protected by using hash functions, any changes in them will be detected by the receiver and the attack will be prevented. Therefore, ERASMIS is robust against all kinds of de-synchronization attacks.

5.1.5. Privileged insider attack

As a privileged insider, an intruder, i.e., A, can obtain information about the U_i from the SA side. Despite having all of the user enrollment details, like HID_i, HPW_i, and $GWID_j$ the attacker cannot guess the user's identity, ID_i. Because it is secured by the user's b_i and is unique for every person, an attacker cannot guess it. So, the proposed scheme has enough security against privileged insider attacks.

5.1.6. Replay attack

Assuming an attacker obtains a message, s/he wishes to re send it in the later times, but s/he is unable to do so because every message communicated on the public channel is using timestamps like T_1, and T_2. Also, the latency in the timestamp is checked when the message is delivered. If the timestamp ΔT has a delay greater than the allowed delay, the message is rejected. Moreover, the response messages are calculated using the parameters exsit in the request, so the messages of one session cannot be used for another session resposes. Therefore, ERASMIS is secure against all kinds of replay attacks because the message's freshness and timestamp is checked before admitting it.

5.1.7. Mutual authentication

The ultimate objective of the suggested authentication protocol is to establish a session key that enables secure communication among the participating entities. The session key ($Sk_u = Sk_s$) generated in the protocol is used for mutual authentication between the entities, including SN_k, and U_i.

5.1.8. Denial of service (DoS) attack

While denial-of-service (DoS) attacks are still possible at many network layers, the proposed authentication protocol architecture includes elements that effectively mitigate this threat. ERASMIS employs a challenge-response exchange, which necessitates sensor nodes responding to clients with either approval or rejection communications. This assures that any answer obtained is genuine, rather than an attempt by an attacker to overwhelm the target with superfluous traffic. The protocol architecture makes DoS attacks impossible to perform by requiring proper confirmations or dismissals from sensors. An intruder cannot impose extra processing load by launching many false authentication attempts. Legitimate clients and sensors are guaranteed to receive feedback on requests, preventing attempts to deny availability by sending deceptive or unnecessary messages. As a result, ERASMIS is highly resilient to all types of DoS assaults attempting to exhaust target resources or disrupt normal execution flow. The use of challenge-response semantics during authentication ensures the integrity of the response, ensuring dependable system functioning even in the face of denial-of-service attacks. This protocol design choice effectively tackles a significant network security flaw.

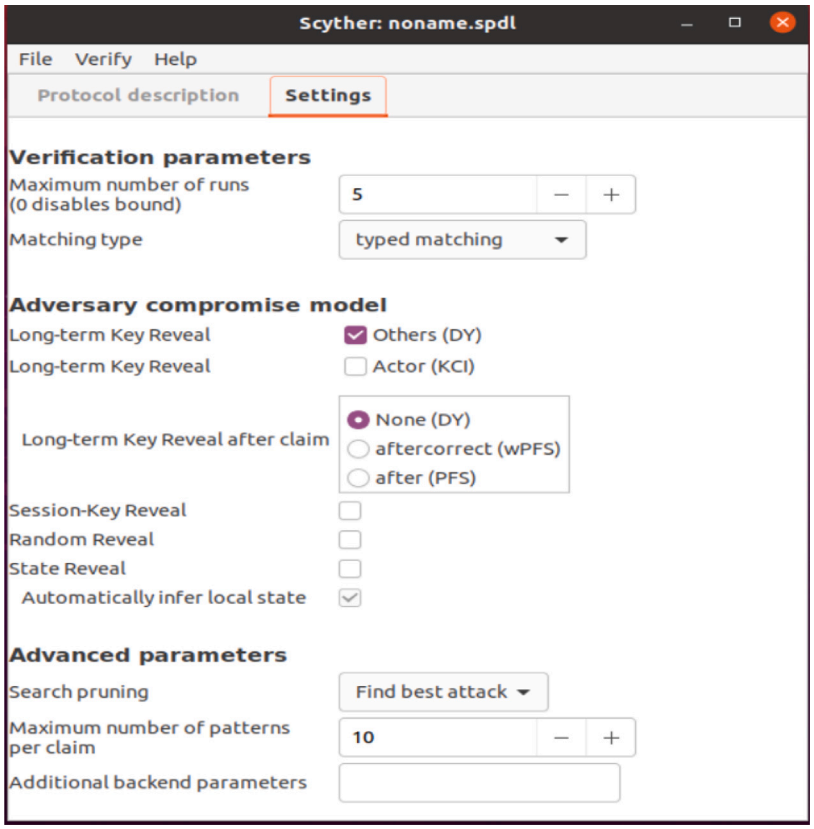


Fig. 3. The setting screen in the Compromise-0.9.2 version of Scyther tool.

5.1.9. Forward /backward secrecy

The proposed key agreement scheme incorporates forward and backward secrecy features. Even if an adversary obtains all current long-term secrets, they will be unable to compromise past or future session keys. This stems from the session key formula $Sk = (r_u \cdot r_s).P$, which utilizes two independently generated random nonces (r_u, r_s) by the user, and sensor respectively. Since each session employs different fresh random values, an attacker who gains access to long-term keys cannot accurately predict prior or subsequent nonces used in the key agreement calculation. As a result, a malicious party acquiring present secret information is highly unlikely to infer prior or future session keys agreed upon between legitimate participants. The incorporation of multiple transient random components in the key derivation process ensures ERASMIS achieves appropriate levels of forward and backward secrecy properties.

5.2. Formal security assessment

Several formal methods have been developed for evaluating the security of cryptographic protocols. The Real or Random (RoR) model, GNY logic and BAN logic are examples of manual processes, while Scyther and ProVerif are examples of automated methods. With respect to our research objectives, Real or Random (RoR) model, Scyther and ProVeirf were chosen to formally proof, simulate and validate the security of the proposed ERASMIS scheme. Our analysis of the proposed approach was conducted using two versions of Scyther, namely the standard version and the compromise version. Furthermore, RoR model, which is one of the most prevalent manual methods, was also used for formal validation.

5.2.1. Through scyther

Formal security analysis of cryptographic protocols is an important step in validating their effectiveness. Scyther is a widely used automated tool that supports rigorous formal analysis based on the [61] threat model. It allows exploring protocol executions and security properties using a formal semantics. Scyther uses the Security Protocol Description Language (SPDL) to specify roles, messages, and security claims. This tool then analyzes all possible executions to verify compliance with defined security goals. It can detect attacks by generating graphs that demonstrate how adversaries can violate intended claims. Security claims include notions like “secrecy” of data, “authentication” of participants, and “integrity” of messages. The Scyther tool operates under standard assumptions of black-box cryptography and message integrity. Protocols are modeled in terms of communicating roles that exchange messages according to predefined functions. This abstraction allows thorough symbolic analysis. The Scyther was used to evaluate the ERASMIS protocol proposed in this study. Three roles — gateway, user, and sensor — were defined to capture the essential interactions. The analysis verified that all security claims such as data confidentiality and entity authentication were upheld against active attacks. Two variants of Scyther were employed—the standard and “compromise” versions which introduce extended adversary capabilities. Results demonstrated ERASMIS’s resilience even under stronger threat models. Figs. 3 and 4 respectively show the setting screen in the compromise and standard versions. As can be seen, the difference between the compromise and standard versions is several adversarial model types that were introduced in the compromise version. The security verification result of the proposed scheme in the Scyther Compromise-0.9.2 tool “after PFC” and “Session Key Reveal” are depicted in Fig. 5 and Fig. 6, respectively and the security verification through the standard version of Scyther tool is depicted in Fig. 7. The implementation code of the ERASMIS protocol can be seen in Fig. 8.

5.2.2. Through ProVerif

We used the ProVerif automated security protocol analysis tool to validate the security properties of the proposed protocol. ProVerif was selected as it supports symbolic modeling and verification of a rich

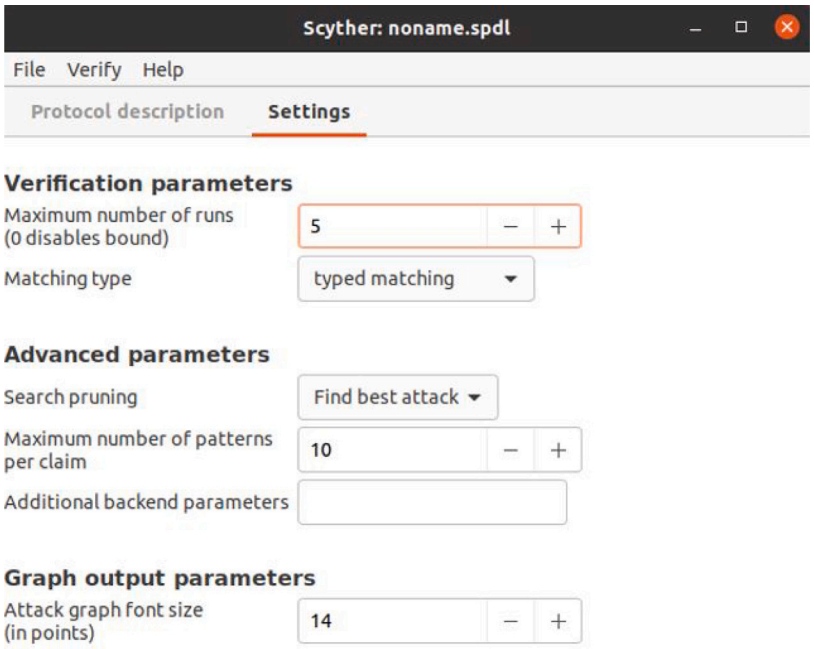


Fig. 4. The setting screen in the standard version of Scyther tool.

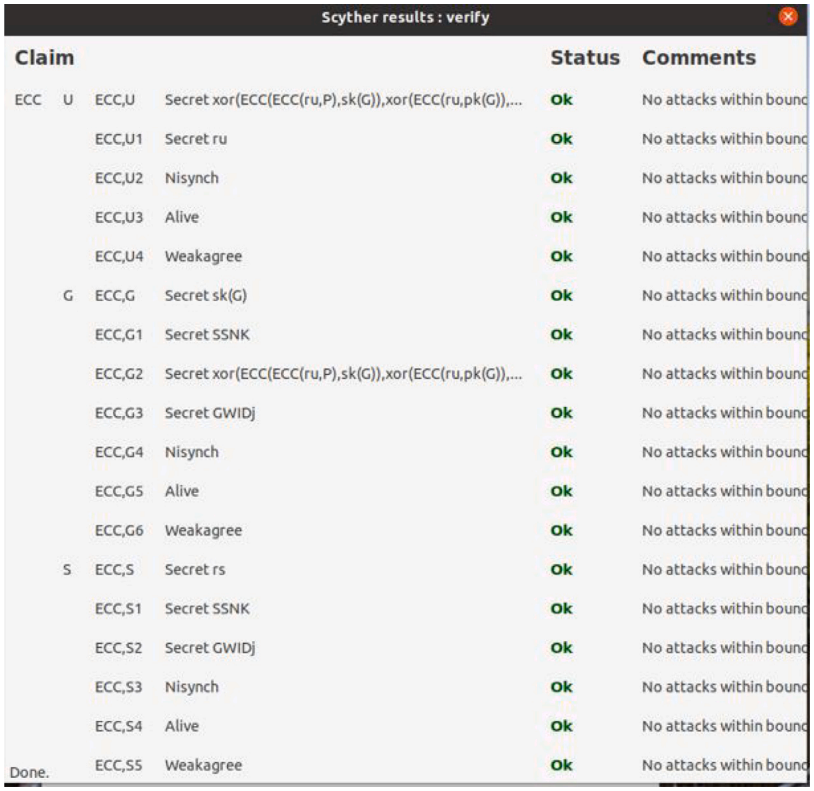
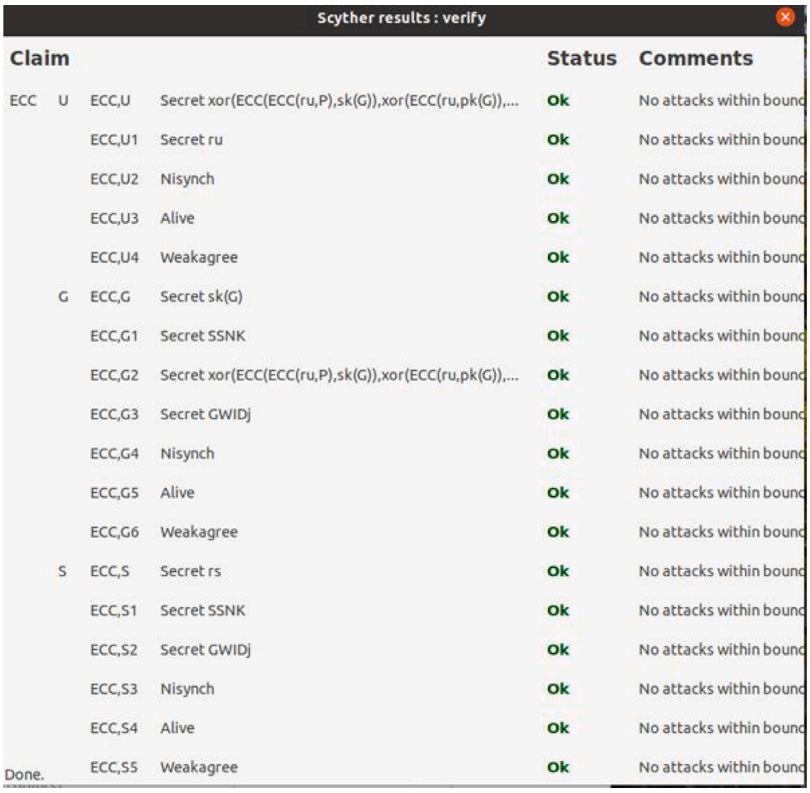


Fig. 5. Security assessment of the ERASMIS authentication protocol utilizing the Scyther Compromise- 0.9.2 tool (opting "after PFC").

variety of cryptographic primitives. Examples include one-way hash functions, public/private key encryption and decryption. Properties such as mutual authentication between entities, confidentiality of session keys, and resistance to key compromise were formally analyzed. During the enrollment phase, secure channels were modeled to reflect the establishment of long-term secrets. However, the mutual authentication and key agreement stages utilize public channels with typical assumptions. ProVerif is well-suited for this task as it can verify authentication, confidentiality, and equivalence properties via an automatic prover. The ProVerif tool simulates an active network attacker with full control over public communication links. Initial testing focused on the authentication and key exchange portions involving public channels. More extensive analysis validated the entire protocol specification comprising all phases. Proverif's symbolic foundations and automated

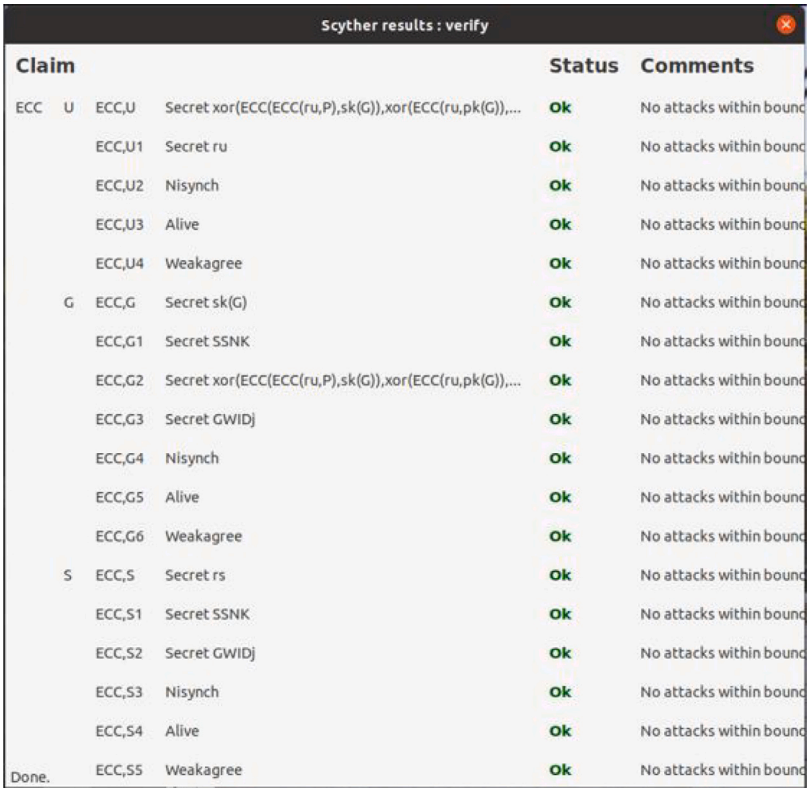


Scyther results : verify

Claim				Status	Comments
ECC	U	ECC,U	Secret xor(ECC(ECC(ru,P),sk(G)),xor(ECC(ru,pk(G)),...	Ok	No attacks within bound
		ECC,U1	Secret ru	Ok	No attacks within bound
		ECC,U2	Nisynch	Ok	No attacks within bound
		ECC,U3	Alive	Ok	No attacks within bound
		ECC,U4	Weakagree	Ok	No attacks within bound
	G	ECC,G	Secret sk(G)	Ok	No attacks within bound
		ECC,G1	Secret SSNK	Ok	No attacks within bound
		ECC,G2	Secret xor(ECC(ECC(ru,P),sk(G)),xor(ECC(ru,pk(G)),...	Ok	No attacks within bound
		ECC,G3	Secret GWIDj	Ok	No attacks within bound
		ECC,G4	Nisynch	Ok	No attacks within bound
		ECC,G5	Alive	Ok	No attacks within bound
		ECC,G6	Weakagree	Ok	No attacks within bound
	S	ECC,S	Secret rs	Ok	No attacks within bound
		ECC,S1	Secret SSNK	Ok	No attacks within bound
		ECC,S2	Secret GWIDj	Ok	No attacks within bound
		ECC,S3	Nisynch	Ok	No attacks within bound
		ECC,S4	Alive	Ok	No attacks within bound
		ECC,S5	Weakagree	Ok	No attacks within bound

Done.

Fig. 6. Security assessment of the ERASMIS authentication protocol utilizing the Scyther Compromise- 0.9.2 tool (opting "Session Key Reveal)".



Scyther results : verify

Claim				Status	Comments
ECC	U	ECC,U	Secret xor(ECC(ECC(ru,P),sk(G)),xor(ECC(ru,pk(G)),...	Ok	No attacks within bound
		ECC,U1	Secret ru	Ok	No attacks within bound
		ECC,U2	Nisynch	Ok	No attacks within bound
		ECC,U3	Alive	Ok	No attacks within bound
		ECC,U4	Weakagree	Ok	No attacks within bound
	G	ECC,G	Secret sk(G)	Ok	No attacks within bound
		ECC,G1	Secret SSNK	Ok	No attacks within bound
		ECC,G2	Secret xor(ECC(ECC(ru,P),sk(G)),xor(ECC(ru,pk(G)),...	Ok	No attacks within bound
		ECC,G3	Secret GWIDj	Ok	No attacks within bound
		ECC,G4	Nisynch	Ok	No attacks within bound
		ECC,G5	Alive	Ok	No attacks within bound
		ECC,G6	Weakagree	Ok	No attacks within bound
	S	ECC,S	Secret rs	Ok	No attacks within bound
		ECC,S1	Secret SSNK	Ok	No attacks within bound
		ECC,S2	Secret GWIDj	Ok	No attacks within bound
		ECC,S3	Nisynch	Ok	No attacks within bound
		ECC,S4	Alive	Ok	No attacks within bound
		ECC,S5	Weakagree	Ok	No attacks within bound

Done.

Fig. 7. ERASMIS protocol security verification through the standard version of Scyther tool.

reasoning capabilities make it a trusted method for rigorously assessing protocol security. In summary, employing ProVerif's robust modeling and analysis features demonstrated the protocol satisfies vital security goals even under a powerful Dolev-Yao adversary model. This formal validation using an established automated protocol verification tool provides strong evidence of the design's security. To test the security

```
hashfunction H;
hashfunction ECC;
const xor : Function;
const con : Function;
const P;
usertype Timestamp;
usertype Ticket;
macro bi=H(Bi);
macro HPWi=H(con(PWi,bi));
macro HPIDi=H(con(IDi,HPWi));
macro A1=H(con(con(HPWi,HPIDi),ui));
macro A2=ECC(ru,P);
macro A3=ECC(ru,pk(G));
macro A4=xor(A3,SNIDk);
macro A5=H(con(con(con(A2,A3),A4),T1));
macro SKu=ECC(A7,ru);
macro A3star=ECC(A2,sk(G));
macro A5star=H(con(con(con(A2,A3star),A4),T1));
macro SNIDk=xor(A3star,A4);
macro A6=H(con(con(con(A2,SSNK),GWIDj),T2));
macro A6star=H(con(con(con(A2,SSNK),GWIDj),T2));
macro A7=ECC(rs,P);
macro A8=H(con(con(con(A6,A2),SSNK),GWIDj));
macro A8star=H(con(con(con(A6,A2),SSNK),GWIDj));
macro A9=H(con(A7,SNIDk));
macro A9star=H(con(A7,SNIDk));
macro Sks=ECC(rs,A2);
protocol @oracle (X){
role Y {
   var X, Y:Agent;
   const P;
    var R;
   recv_!X1(X, Y,  ECC(R,pk(Y)));
   send_!X2( Y,X , ECC(sk(Y),ECC(R,P)) );
     }
}
protocol ECC(U,G,S){
role U{
     fresh ru: Nonce;
     var rs: Nonce;
     fresh T1: Timestamp;
     secret IDi;
     secret PWi;
     secret Bi:
     secret SNIDk;
     secret ui;
     const P;
     send_1(U,G,A2,A4,A5,T1);
     recv_4(G,U,A7,A9);
     match(A9star,A9);
     claim_U (U, Secret,SNIDk );
     claim_U (U, Secret, ru);
     claim_U (U, Nisynch);
     claim_U (U, Alive);
     claim_U (U, Weakagree);
}
role G{
     fresh T2: Timestamp;
     var ru: Nonce;
     var rs: Nonce;
     var T1: Timestamp;
     secret SSNK;
     secret SNIDk;
     secret GWIDj;
     const P;
     recv_1 (U,G,A2,A4,A5,T1);
     match(A5star,A5);
     send_2 (G,S,A2,A6,T2);
     recv_3(S,G,A7,A8);
     match(A8star,A8);
     send_4 (G,U,A7,A9);
     claim_G (G, Secret, sk(G));
     claim_G (G, Secret, SSNK);
     claim_G (G, Secret, SNIDk);
     claim_G (G, Secret, GWIDj);
     claim_G (G, Nisynch);
     claim_G (G, Alive);
     claim_G(G, Weakagree);
}
role S{
     var T2: Timestamp;
     fresh rs:Nonce;
     var ru: Nonce;
     secret SSNK;
     secret GWIDj;
     const P;
     recv_2 (G,S,A2,A6,T2);
     match (A6star,A6);
     send_3(S,G,A7,A8);
     claim_S (S, Secret,rs);
     claim_S (S, Secret, SSNK);
     claim_S (S, Secret, GWIDj);
     claim_S (S, Nisynch);
     claim_S (S, Alive);
     claim_S(S, Weakagree);
}
}
```

Fig. 8. ERASMIS implementation code in the Scyther tool.

of the proposed scheme, Fig. 9 illustrates the security validation results of ERASMIS through Proverif.

The ProVerif verification summary confirms that the ERASMIS protocol meets key security requirements:

1. **Weak Secret Verification:** The protocol protects weak secrets, such as ID_i (user ID), SN_{ID_k} (sensor node ID), SSN_K (session key), and GW_{ID_j} (gateway ID), ensuring they cannot be compromised by an attacker. This indicates that user identities, session keys, and gateway identifiers remain private and secure during protocol execution.
2. **Attacker Queries:** The results for 'not attacker(uru[!1 = v])' and 'not attacker($srs[T2_1 = v, A6_1 = v_1, A2_2 = v_2, !1 = v_3]$) show that the protocol prevents unauthorized access to critical values, mitigating potential vulnerabilities and protecting against interception or manipulation by adversaries.
3. **Event Injection Queries:** The findings for event injection queries, such as 'inj-event(endUserA) ==> inj-event(beginUserA)', validate the integrity of user, gateway, and sensor node interactions. Each session initiates and completes securely, preventing session hijacking or replay attacks.

These results indicate that the ERASMIS protocol effectively protects sensitive identifiers, ensures session integrity, and defends against unauthorized access and various potential attacks.

5.2.3. Through real or random model (RoR)

This study aims to conduct a formal security analysis of the proposed authentication protocol. To objectively evaluate the protocol's resistance against attacks, the authors leverage the well-established real-or-random (RoR) model. The RoR model is a rigorous cryptanalytic technique used to gauge a protocol's security. It works by simulating

Verification summary:

- Weak secret IDi is **true**.
- Weak secret SNIDk is **true**.
- Weak secret SSNK is **true**.
- Weak secret GWIDj is **true**.
- Query not attacker(uru[!1 = v]) is **true**.
- Query not attacker(srs[T2_1 = v,A6_1 = v_1,A2_2 = v_2,!1 = v_3]) is **true**.
- Query inj-event(endUserA) ==> inj-event(beginUserA) is **true**.
- Query inj-event(endGatewayNodeA) ==> inj-event(beginGatewayNodeA) is **true**.
- Query inj-event(endSensorNodeA) ==> inj-event(beginSensorNodeA) is **true**.

Fig. 9. Security evaluation results of ERASMIS protocol using the ProVerif tool.

multiple rounds of a "game" between an attacker and a challenger. In each round, the attacker is presented with either: A real execution transcript of the protocol and a completely random string. The attacker's task is to determine which of the two options it received. This process is repeated over several rounds to calculate the likelihood (advantage) of the attacker successfully distinguishing the real session key Sk from a random value. By applying the RoR model, one can systematically and quantitatively assess the protocol's security under different threat scenarios. More specifically, it allows for demonstrating resilience against session key extraction attacks.

5.2.4. RoR model

Our scheme is made up of three entities, such as U_i, GW_j, and SN_K. We utilize $\mathrm{Pi}^a_{u_i}$, $\mathrm{Pi}^b_{GW_j}$, and $\mathrm{Pi}^c_{SN_K}$ to represent the a-th $U-i$, b-th GW_j, and cth SN_K, respectively; in this way, R={$\mathrm{Pi}^a_{u_i}$, $\mathrm{Pi}^b_{GW_j}$, and $\mathrm{Pi}^c_{SN_K}$}. Assume that intruder $\mathcal{A}$ is capable of running the following queries:

- Execute(R) : In this context, the adversary A possesses the capability to monitor the messages transmitted over the insecure channel by entities U_i, GW_j, and SN_K.

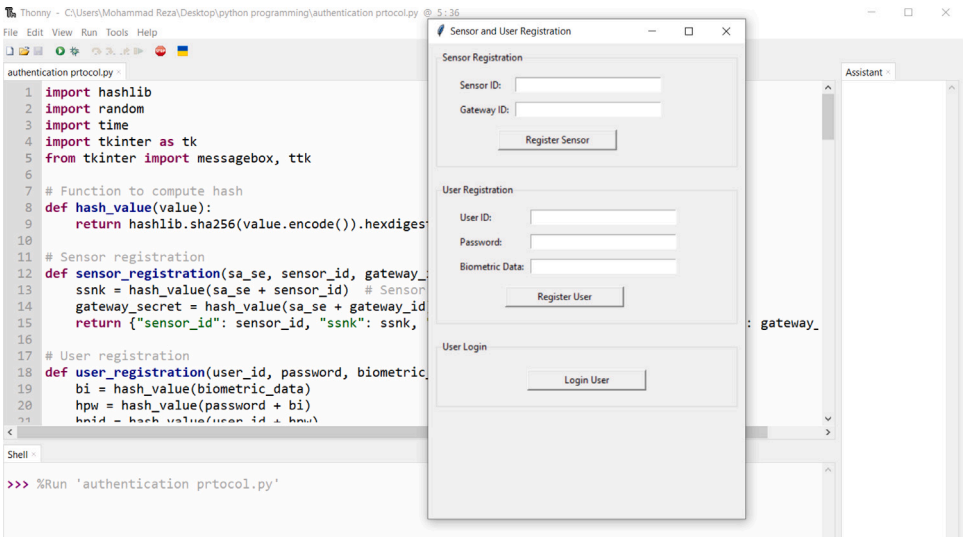


Fig. 10. Overview of our Thonny IDE environment, showcasing a user-friendly GUI for registering users, sensors, and gateways, along with the login phase.

- Send(R, M) : The adversary A has the ability to initiate the transmission of message M to R and subsequently receive the response message from R when executing this query.
- Hash(String) : In this context, the adversary A can retrieve the hash value of a string with a specific length by inserting it into this procedure.
- Corrupt(R) : The adversary A retrieves the secret data of an entity, including a long-term key, temporary information, or parameters saved in a smart card, by running this query.
- Test(R) : Suppose A runs this query and determines the security of the session key by flipping a coin C. A acquires the right session key if C = 1. Alternatively, A will be given a random string.

Theorem 1. *We define $Adv_{\mathcal{A}}^{P}$, in the RoR model as an indicator of the attacker's capability to break the protocol via query activities. $Adv_{\mathcal{A}}^{P}$, in particular, indicates the likelihood that the attacker A would gain the session key and is restricted by the equation $Adv_{\mathcal{A}}^{P} \leq \frac{q_h^2}{|H|} + \frac{q_s}{2^{t-1}|D|}$. The variables q_h and q_s in this expression represent the number of instances in which the attacker can run the hash and transmit queries accordingly. $|H|$ and $|D|$ reflect the hash operation's space scope and dictionary size, respectively, while t is the bit number of biological data contained in the protocol.*

5.3. Security proof

We have participated in the four phases of a game known as $GM_i(i = 0, 1, 2, 3)$, where $Succ_{\mathcal{A}}^{GM_i}$ represents the probability of attacker A succeeding in each phase of the game. Here are the details of the game:

GM_0: In the initial phase (GM_0), the adversary A only needs to identify a bit value, and no query operations are performed. Consequently, we can determine the probability of the adversary A prevailing in GM_0 as:

$$Adv_{\mathcal{A}}^{P} = |2Pr[Succ_{\mathcal{A}}^{GM_0}] - 1|$$

GM_1 : In GM_1 which follows GM_0, an intercept procedure takes place. In this phase, the adversary A is restricted to capturing messages transmitted through specific channels, namely $A_2, A_4, A_5, T_1, A_6, T_2, A_7, A_8$ and A_9. However, during the interaction, the adversary A is unable to perform test queries to retrieve the session key $Sk = (ru \cdot rs).P$ because the random values r_u and r_s cannot be gained solely from the information available in the public channels. Consequently, the probability of the adversary A winning the game following an Execute query remains the same as in GM_0.

$$Pr[Succ_{\mathcal{A}}^{GM_0}] = Pr[Succ_{\mathcal{A}}^{GM_1}]$$

GM_2 : GM_2 is the third phase of the game, which follows the hash query and send operation in GM_1. During GM_2, forging is not possible due to using ECC functions for A_7 and A_2. Additionally, the session key's crucial properties, r_u, and r_s, are random in every session. The concept of the birthday paradox comes into play in this phase. The birthday paradox refers to the phenomenon where the probability of two or more people sharing the same birthday becomes surprisingly high in a relatively small group. In the context of GM_2, the birthday paradox helps us derive certain conclusions or probabilities.

$$Pr[Succ_{\mathcal{A}}^{GM_2}] - Pr[Succ_{\mathcal{A}}^{GM_1}] \leq \frac{q_h^2}{|2H|}$$

GM_3 : The Corrupt query is conducted in this phase, and the adversary A can access the secret value of an entity such as $SNID_k, GWID_j, SSN_k$ and A_2. Furthermore, the adversary A tries to figure out ID_i and PW_i; but, even if the adversary A correctly guesses ID_i and PW_i at the same time, he or she cannot get the random number u_i. The adversary A is also unable to get the biological eigenvalue Bio $b_i = H(B_i)$, $HPID_i = h(ID_i \parallel b_i)$ and $HPW_i = h(PW_i \parallel b_i)$, so the chance of the biometric being calculated is $1/2^t$. We know that the adversary A is allowed to type the code for a certain number of times.

$$Pr[Succ_{\mathcal{A}}^{GM_3}] - Pr[Succ_{\mathcal{A}}^{GM_2}] \leq \frac{q_s}{2^t|D|}$$

The adversary A is able to win the game if the proper bit b is recognized.

$$Pr[Succ_{\mathcal{A}}^{GM_3}] = 1/2$$

Utilizing the given formulas, we get

$$1/2Adv_{\mathcal{A}}^{P} = |Pr[Succ_{\mathcal{A}}^{GM_0}] - 1/2| =$$
$$Pr[Succ_{\mathcal{A}}^{GM_1}] - Pr[Succ_{\mathcal{A}}^{GM_3}] \leq Pr[Succ_{\mathcal{A}}^{GM_2}] - Pr[Succ_{\mathcal{A}}^{GM_1}]$$
$$+ Pr[Succ_{\mathcal{A}}^{GM_3}] - Pr[Succ_{\mathcal{A}}^{GM_2}] = \frac{q_s}{2^t|D|} + \frac{q_h^2}{|2H|} \quad (1)$$

$$1/2Adv_{\mathcal{A}}^{P} \leq \frac{q_s}{2^t|D|} + \frac{q_h^2}{|2H|}$$

$$Adv_{\mathcal{A}}^{P} \leq \frac{q_s}{2^{t-1}|D|} + \frac{q_h^2}{|H|}$$

6. Implementation of ERASMIS

This section provides a comprehensive overview of the ERASMIS scheme's implementation, detailing its deployment across multiple

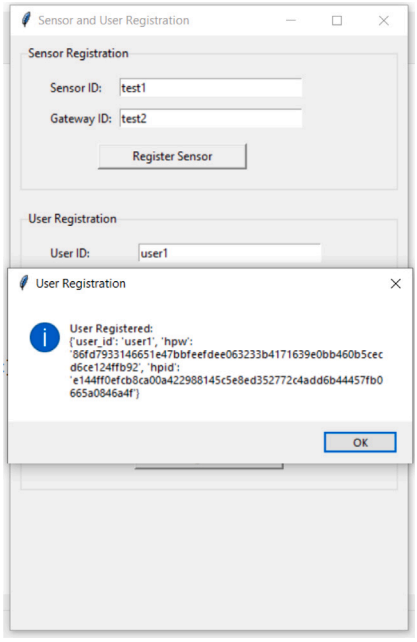


Fig. 11. User registration phase GUI in our implemented protocol.

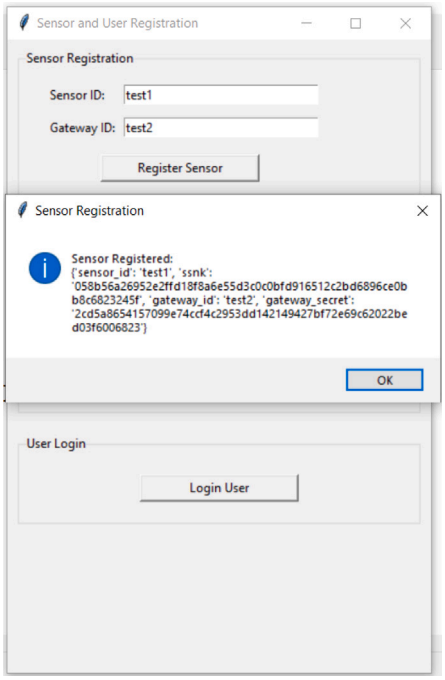


Fig. 12. Sensor and Gateway registration phase GUI in our implemented protocol.

configurations. We evaluated the scheme in three setups: a computer with a temperature sensor, a standard computer, and a gateway connected to temperature-sensing devices. The protocol was implemented in Python 3.9 to ensure compatibility and efficiency across these varied environments.

Fig. 10 shows the Thonny IDE development environment, where the graphical user interface (GUI) manages user, sensor, and gateway registration, as well as the login process. Fig. 11 illustrates the user registration phase, where users input and register their credentials within the system. Fig. 12 depicts the sensor and gateway registration and configuring these devices for secure network interactions. Fig. 13 highlights a successful login and authentication, confirming verified access to the system. Fig. 14 captures the session key calculation phase, in which the user, gateway, and IoT devices establish a shared secret session key following authentication. Finally, Fig. 15 displays a temperature reading retrieved from authenticated IoT devices after successful login and authentication (see Table 4).

7. Assessment and comparison

This section compares the ERASMIS protocol with other recent similar protocols, evaluating various aspects such as security, communication, computational efficiency, and storage costs.

7.1. Security comparison of ERASMIS with similar authentication schemes

We compare ERASMIS with other authentication protocols in this section, including [34,45,62–65]. Our considerations demonstrate that our authentication protocol is robust against replay, privilege, overcoming the session key, traceability, forward secrecy contradiction, and de-synchronization attacks. While the other mentioned schemes are not secure against the explained attacks. Table 5 shows a comparison between ERASMIS and other similar recent authentication protocols.

7.2. Communication and computational costs comparisons

According to [66] $T_{hf}, T_{en/d}$, and T_{mu} are 3 ms, 3.7 ms, and 21 ms, respectively. The used notation for computational cost is listed in Table 6. Our proposed authentication protocol has a computation cost of $11T_{hf} + T_{bh} + 6T_{mu}$. A comparison of ERASMIS with other similar protocols concerning computational cost is presented in Table 7 and Fig. 16.

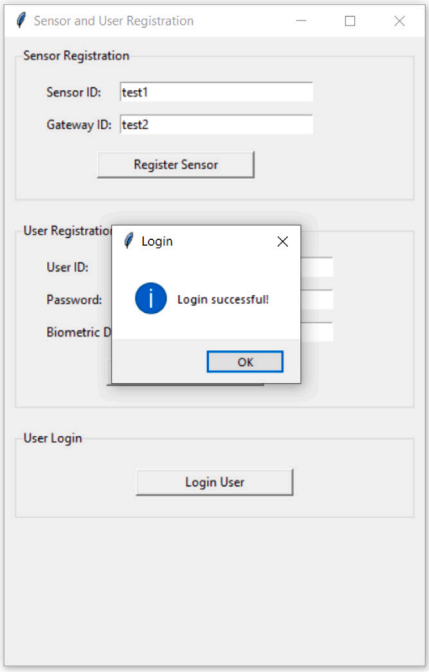


Fig. 13. The successful Login and Authentication phase GUI in our implemented protocol.

During the login and authentication processes, we compare the ERASMIS protocol communication costs, which are the number of bits that are exchanged through the protocol run. Table 6 indicates that the communication overhead includes 128 bits for transmitting identity, 32 bits for time delay, 256 bits for ECC multiplication, 128 bits for

Table 4
Details utilized in the execution of the implemented ERASMIS protocol.

Parameter	Hexadecimal Value
ID_i	0×75736572313233 (hex for "user123")
PW_i	$0 \times 70617373776f7264313233$ (hex for "password123")
u_i	$0 \times 756e697175655f76616c7565$ (hex for "unique_value")
B_i	$0 \times 62696f6d65747269635f64617461$ (hex for "biometric_data")
b_i	$0 \times 86bfa4acdb4202f80bd3d696aea78bbf50c8244fe6eb560563ca739b28a8be3a$
HPW_i	$0 \times 2ab4fe861355c074b4ec0fdf8b412ca161ee542b989555d8987143fdb60e2f9d$
$HPID_i$	$0 \times 64195421603bb1f244f56e477b07258b2426c2eceb544aea049f8092d4b1bf9f$
A_1	0xa8f93b0469d60166d7d94a855daff247e70d30ee218eed17285d45e1e74b67d7
r_u	0×05
P	0×03
PkG	0×07
$SNID_k$	$0 \times 6e6574776f726b5f69645f313233$
T_1	$0 \times 74696d657374616d705f313233$
A_2	$0 \times 0f$
A_3	0×23
A_4	$0 \times 6e6574776f726b5f69645f313210$
A_5	$0 \times 8e6a7dbfb026d0c89320d89d0b2233ddd32b2a9aeb30d5d39fe492888b2236f7$
	True (indicates $\|T_2 - T_1\| < \Delta T$ check passed)
A_3^*	$0 \times 3c$
A_5^*	$0 \times 8e6a7dbfb026d0c89320d89d0b2233ddd32b2a9aeb30d5d39fe492888b2236f7$
A5	Verification? True (indicates $A_5^* = A_5$)
$SNID_k^{new}$	$0 \times 6e6574776f726b5f69645f31322c$
T_2	$0 \times 74696d657374616d705f313234$
A_6	0xca9924d372586418c91849273b88b2905cd25c343e43132b6db4870d894b55aa
A_6^*	0xca9924d372586418c91849273b88b2905cd25c343e43132b6db4870d894b55aa
A6	Verification ? True (indicates $A6^* = A6$)
A_7	0×18
Sk	0×78
A_8	$0 \times 3aa754bfcd10a53bdcecebd8e6bbea855c1a8203ca872062b67869ab1c752767$
A_9^*	0xbabdee60cda4a7da7ca534b5c836d18cfed6c130271ce2615d595c5d30c2af9b
Sk^*	0×78 Verification ? True (indicates $Sk^* = Sk$)

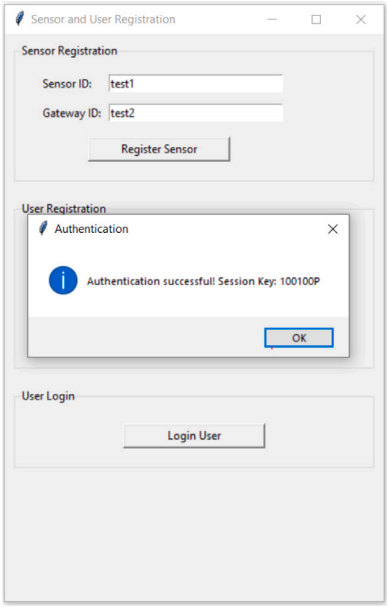


Fig. 14. Illustration depicting the user, gateway, and IoT devices, all utilizing a common session key within the implemented protocol.

Table 5
ERASMIS security compared to other recent similar protocols.

Protocols	A_1	A_2	A_3	A_4	A_5	A_6	A_7
[34]	✓	✓	✗	✗	✓	✓	✗
[62]	✓	✗	✓	✓	✗	✓	✓
[45]	✓	✗	✓	✗	✓	✓	✓
[63]	✓	✓	✗	✗	✓	✓	✗
[64]	✗	✓	✓	✓	✓	✓	✗
[65]	✓	✗	✓	✓	✓	✗	✗
ERASMIS	✓	✓	✓	✓	✓	✓	✓

A_1: Resilience to replay attacks; A_2: Resilience to privilege insider invasion;
A_3: Reveal the session key ; A_4: Untraceability;
A_5: Resilience to spoofing attacks; A_6: Forward secrecy;
A_7: Resilience to de-synchronization attack;
✓: Robustness ✗ : weakness.

Table 6
Notations used for computational and communication cost comparisons [66].

Symbol	Description	Execution time	Communication cost
T_h	Hash function execution time	3 ms	256 bits
$T_{en/d}$	Encryption and decryption execution time	3.7 ms	128 bits
T_{mu}	ECC Scalar multiplication execution time	21 ms	256 bits
T_{ts}	Timestamp length	–	32 bits
T_{bh}	Execution time of bio-hashing	5 ms	256 bits

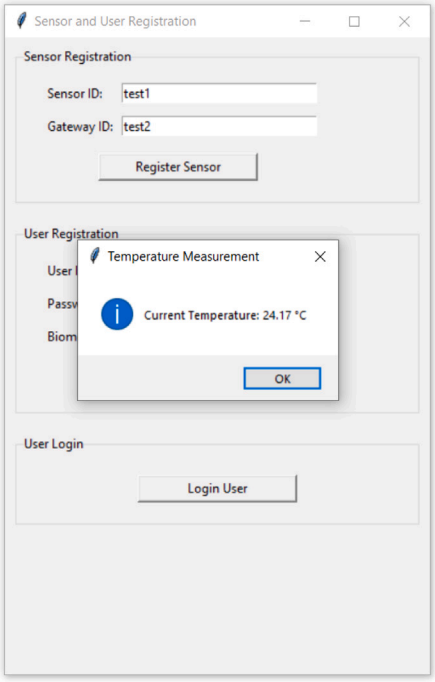


Fig. 15. Display of the sensor's temperature following successful login and authentication within the implemented protocol.

Table 7
Computational cost comparison of ERASMIS with recently introduced authentication protocols (in milliseconds)

Protocols	Overall computational cost for GW_j, SN_k, and U_i in ms
[62]	$(27+3\ m)T_h + 1T_{bh} = 176$ where $m = 10$ is the number of sensors
[34]	$17T_h + 17T_p + T_{bh} = 392$
[45]	$22T_h + 4T_{mu} + T_{E/D} = 157.4$
[63]	$32T_h = 102$
[64]	$22T_h = 66$ ms
[65]	$8T_{mu} + 15T_h + 2T_{E/D} = 220.4$
[41]	$12T_{mu} + 48T_h = 396$
ERASMIS	$11T_h + T_{bh} + 6T_{mu} = 164$

encryption/decryption, and 256 bits for the random values and 256 bits for hash function output, respectively. As demonstrated in Table 8 and Fig. 17, the communication cost comparison results reveal that ERASMIS generates a reasonable communication cost compared to other authentication schemes.

7.3. Storage cost comparison

Sensor nodes in IoT systems have significantly less storage capacity than gateway nodes due to their limited computing and memory resources. It is crucial to minimize the storage overhead on sensor nodes. The proposed ERASMIS authentication scheme encrypts data into 128-bit ciphertexts. The hash function outputs and random numbers used in the scheme are 256 bits in length. Meanwhile, user passwords and identities require 128 bits each for storage. The storage costs of ERASMIS and other similar authentication protocols are analyzed and compared, and the results are shown in Table 9. Table 9 provides numeric values for the different types of data each protocol requires nodes to store, such as encrypted values, random numbers, hash values and etc. It is worth noting that Fig. 18 presents the storage comparison graphically. Although the storage cost of the proposed protocol is higher than the rest of the compared protocols, but instead it has established full security while the other compared protocols are not secure.

Table 8
Communication cost comparison of ERASMIS with recently introduced authentication protocols.

Protocols	Year	Overall communication cost for GW_j, SN_K, and U_i (in bits)
[62]	2019	3456
[34]	2019	3168
[45]	2019	3328
[63]	2020	3200
[64]	2022	2944
[65]	2022	3200
[41]	2024	5664
Our	–	2912

Table 9
ERASMIS storage expenses compared to recent comparable protocols.

Protocols	Storage cost (in bits)
[62]	384
[34]	512
[45]	512
[63]	1024
[64]	640
[65]	384
[41]	384
ERASMIS	768

8. Conclusion

This paper introduces the ERASMIS authentication protocol, aimed at securing communications within medical IoT systems. ERASMIS enables mutual authentication between users and IoT devices through a gateway that creates a secure session key for data access. We validated the protocol's security using the RoR model. To automatically evaluate the resilience of ERASMIS against known threats, we utilized ProVerif and Scyther simulation tools. The security assessments indicate that ERASMIS effectively counters various threats and surpasses existing protocols in terms of security, communication efficiency, and cost-effectiveness. However, ERASMIS currently encounters challenges related to scalability and integration with legacy systems. We also created a Python implementation of ERASMIS to support its practical application and further testing. Future research will focus on exploring the integration of the protocol with emerging technologies such as artificial intelligence, blockchain, or advancements in IoT to enhance both security and efficiency. Additionally, we aim to implement the protocol in real-world settings, such as hospitals or clinics, to assess its performance under actual conditions and investigate its impact on user experience and usability, with the goal of further improvements.

CRediT authorship contribution statement

Mohammad Reza Servati: Writing – original draft, Methodology. **Masoumeh Safkhani:** Writing – original draft, Project administration. **Amir Masoud Rahmani:** Writing – review & editing, Resources. **Mehdi Hosseinzadeh:** Writing – review & editing, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Masoumeh Safkhani was supported by Shahid Rajaee Teacher Training University under grant number 5973.20.

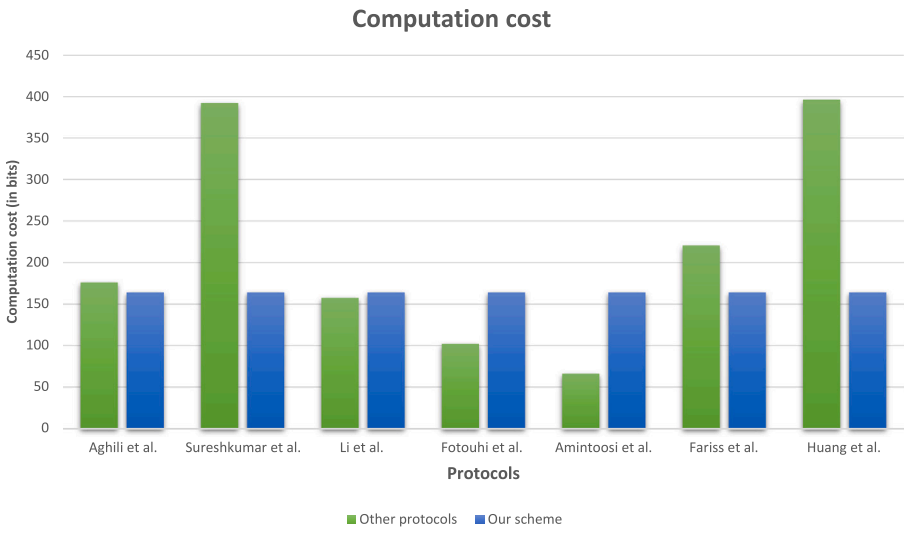


Fig. 16. A comparison of the whole computational overhead of ERASMIS with similar recent authentication schemes.

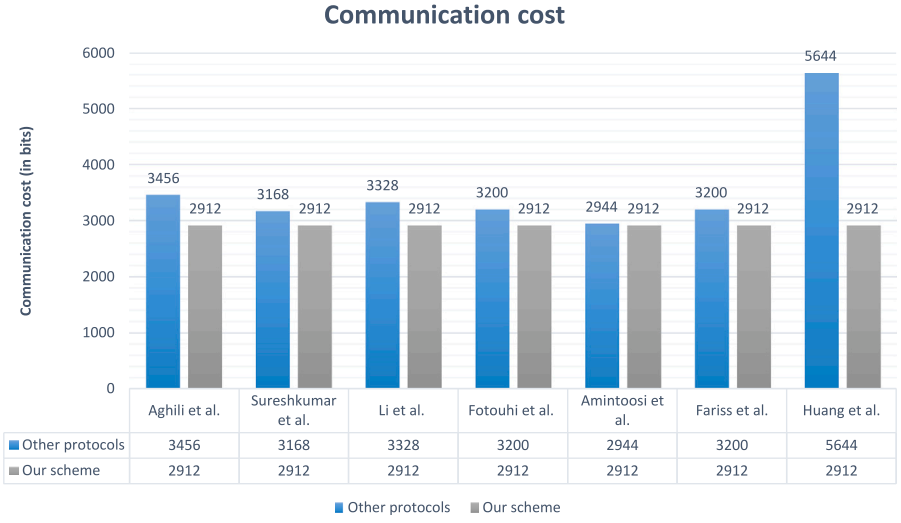


Fig. 17. A comparison of the whole communication overhead of ERASMIS with similar recent schemes.

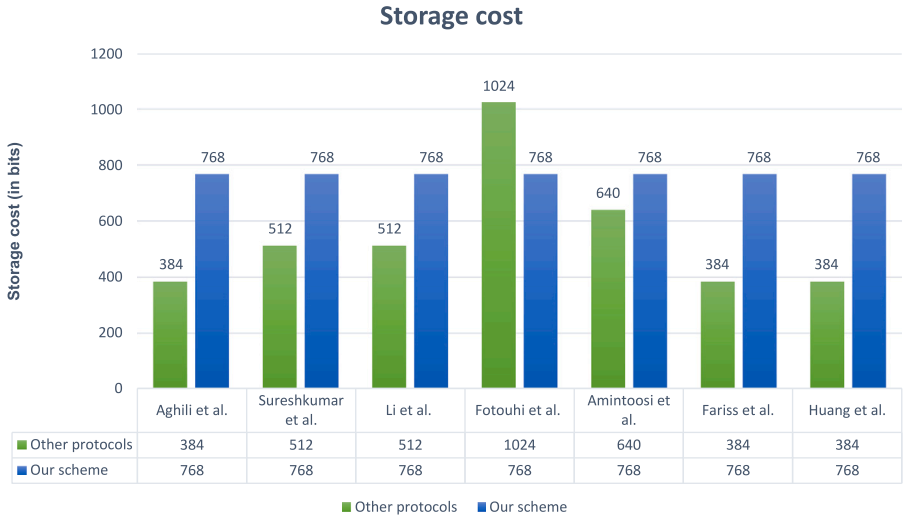


Fig. 18. A comparison of the storage cost of ERASMIS with similar recent authentication schemes.

Data availability

No data was used for the research described in the article.

References

[1] D. Sadhukhan, S. Ray, G. Biswas, M.K. Khan, M. Dasgupta, A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography, J. Supercomput. 77 (2) (2021) 1114–1151.

[2] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen, Energy efficient dynamic offloading in mobile edge computing for internet of things, IEEE Trans. Cloud Comput. 9 (3) (2019) 1050–1060.

[3] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, Y. Yang, An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks, J. Netw. Comput. Appl. 76 (2016) 37–48.

[4] S. Suganthi, R. Anitha, V. Sureshkumar, S. Harish, S. Agalya, End to end light weight mutual authentication scheme in IoT-based healthcare environment, J. Reliab. Intell. Environ. 6 (1) (2020) 3–13.

[5] V. Sureshkumar, R. Amin, V. Vijaykumar, S.R. Sekar, Robust secure communication protocol for smart healthcare system with FPGA implementation, Future Gener. Comput. Syst. 100 (2019) 938–951, http://dx.doi.org/10.1016/j.future.2019.05.058, URL: https://www.sciencedirect.com/science/article/pii/S0167739X18332448.

[6] K. Malasri, L. Wang, Design and implementation of a secure wireless mote-based medical sensor network, Sensors 9 (8) (2009) 6273–6297.

[7] P. Kumar, S.G. Lee, H.J. Lee, E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks, Sensors 12 (2) (2012) 1625–1647.

[8] D. He, N. Kumar, J. Chen, C.C. Lee, N. Chilamkurti, S.S. Yeo, Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks, Multimedia Syst. 21 (1) (2015) 49–60.

[9] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, M.K. Khan, A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity, Secur. Commun. Netw. 9 (15) (2016) 2643–2655.

[10] F. Wu, L. Xu, S. Kumari, X. Li, An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks, Multimedia Syst. 23 (2) (2017) 195–205.

[11] P. Chandrakar, H. Om, An extended ECC-based anonymity-preserving 3-factor remote authentication scheme usable in TMIS, Int. J. Commun. Syst. 31 (8) (2018) e3540.

[12] C.T. Li, D.H. Shih, C.C. Wang, Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems, Comput. Methods Programs Biomed. 157 (2018) 191–203.

[13] V. Kumar, M. Ahmad, A. Kumari, A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS, Telemat. Inform. 38 (2019) 100–117.

[14] L. Zheng, C. Song, N. Cao, Z. Li, W. Zhou, J. Chen, L. Meng, A new mutual authentication protocol in mobile RFID for smart campus, IEEE Access 6 (2018) 60996–61005.

[15] M. Safkhani, A. Vasilakos, A new secure authentication protocol for telecare medicine information system and smart campus, IEEE Access 7 (2019) 23514–23526.

[16] A. Xiang, J. Zheng, A situation-aware scheme for efficient device authentication in smart grid-enabled home area networks, Electronics 9 (6) (2020) 989.

[17] J. Oh, S. Yu, J. Lee, S. Son, M. Kim, Y. Park, A secure and lightweight authentication protocol for IoT-based smart homes, Sensors 21 (4) (2021) 1488.

[18] M. Shuai, N. Yu, H. Wang, L. Xiong, Anonymous authentication scheme for smart home environment with provable security, Comput. Secur. 86 (2019) 132–146.

[19] A. Gupta, M. Tripathi, T.J. Shaikh, A. Sharma, A lightweight anonymous user authentication and key establishment scheme for wearable devices, Comput. Netw. 149 (2019) 29–42.

[20] R. Hajian, S. ZakeriKia, S.H. Erfani, M. Mirabi, SHAPARAK: Scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement, Comput. Netw. 183 (2020) 107567.

[21] Z. Xu, C. Xu, H. Chen, F. Yang, A lightweight anonymous mutual authentication and key agreement scheme for WBAN, Concurr. Comput. Pract. Exp. 31 (14) (2019) e5295.

[22] B.A. Alzahrani, A. Irshad, A. Albeshri, K. Alsubhi, A provably secure and lightweight patient-healthcare authentication protocol in wireless body area networks, Wirel. Pers. Commun. 117 (1) (2021) 47–69.

[23] X. Wang, K. Fan, K. Yang, X. Cheng, Q. Dong, H. Li, Y. Yang, A new RFID ultra-lightweight authentication protocol for medical privacy protection in smart living, Comput. Commun. 186 (2022) 121–132.

[24] B. Chander, K. Gopalakrishnan, A secured and lightweight RFID-tag based authentication protocol with privacy-preserving in telecare medicine information system, Comput. Commun. 191 (2022) 425–437.

[25] M.R. Servati, M. Safkhani, S. Ali, M.H. Malik, O.H. Ahmed, M. Hosseinzadeh, A.H. Mosavi, Cryptanalysis of two recent ultra-lightweight authentication protocols, Mathematics 10 (23) (2022) 4611.

[26] S.F. Aghili, H. Mala, P. Kaliyar, M. Conti, SecLAP: Secure and lightweight RFID authentication protocol for medical IoT, Future Gener. Comput. Syst. 101 (2019) 621–634.

[27] M. Safkhani, S. Rostampour, Y. Bendavid, N. Bagheri, IoT in medical & pharmaceutical: Designing lightweight RFID security protocols for ensuring supply chain integrity, Comput. Netw. 181 (2020) 107558.

[28] S. Gabsi, Y. Kortli, V. Beroulle, Y. Kieffer, A. Alasiry, B. Hamdi, Novel ECC-based RFID mutual authentication protocol for emerging IoT applications, IEEE Access 9 (2021) 130895–130913.

[29] A. Arslan, M.A. Bingöl, Security and privacy analysis of recently proposed ECC-based RFID authentication schemes, Cryptol. ePrint Arch. (2022).

[30] D. Sadhukhan, S. Ray, M.S. Obaidat, M. Dasgupta, A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography, J. Syst. Archit. 114 (2021) 101938, http://dx.doi.org/10.1016/j.sysarc.2020.101938, URL: https://www.sciencedirect.com/science/article/pii/S1383762120301958.

[31] C.H. Tseng, S.H. Wang, W.J. Tsaur, Hierarchical and dynamic elliptic curve cryptosystem based self-certified public key scheme for medical data protection, IEEE Trans. Reliab. 64 (3) (2015) 1078–1085.

[32] S. Challa, A.K. Das, V. Odelu, N. Kumar, S. Kumari, M.K. Khan, A.V. Vasilakos, An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Comput. Electr. Eng. 69 (2018) 534–554.

[33] R. Amin, S.H. Islam, G. Biswas, M.K. Khan, N. Kumar, A robust and anonymous patient monitoring system using wireless medical sensor networks, Future Gener. Comput. Syst. 80 (2018) 483–495.

[34] V. Sureshkumar, R. Amin, V. Vijaykumar, S.R. Sekar, Robust secure communication protocol for smart healthcare system with FPGA implementation, Future Gener. Comput. Syst. 100 (2019) 938–951.

[35] M.E.S. Saeed, Q.Y. Liu, G. Tian, B. Gao, F. Li, Remote authentication schemes for Wireless Body Area networks based on the internet of things, IEEE Internet Things J. 5 (6) (2018) 4926–4944, http://dx.doi.org/10.1109/JIOT.2018.2876133.

[36] X. Jia, M. Luo, H. Wang, J. Shen, D. He, A blockchain-assisted privacy-aware authentication scheme for Internet of Medical Things, IEEE Internet Things J. 9 (21) (2022) 21838–21850.

[37] D. Abbasinezhad-Mood, M. Nikooghadam, Efficient design of a novel ECC-based public key scheme for medical data protection by utilization of NanoPi fire, IEEE Trans. Reliab. 67 (3) (2018) 1328–1339, http://dx.doi.org/10.1109/TR.2018.2850966.

[38] X. Yang, X. Yi, S. Nepal, I. Khalil, X. Huang, J. Shen, Efficient and anonymous authentication for healthcare service with cloud based WBANs, IEEE Trans. Serv. Comput. 15 (5) (2022) 2728–2741, http://dx.doi.org/10.1109/TSC.2021.3059856.

[39] M.R. Servati, M. Safkhani, ECCbAS: An ECC based authentication scheme for healthcare IoT systems, Pervasive Mob. Comput. (2023) 101753, http://dx.doi.org/10.1016/j.pmcj.2023.101753, URL: https://www.sciencedirect.com/science/article/pii/S1574119223000111.

[40] M. Hosseinzadeh, M.R. Servati, A.M. Rahmani, M. Safkhani, J. Lansky, R. Janoscova, O.H. Ahmed, J. Tanveer, S.W. Lee, An enhanced authentication protocol suitable for constrained RFID systems, IEEE Access (2024).

[41] W. Huang, ECC-based three-factor authentication and key agreement scheme for wireless sensor networks, Sci. Rep. 14 (1) (2024) 1787.

[42] W. Wang, B. Yan, B. Chai, R. Shen, A. Dong, J. Yu, EBIAS: ECC-enabled blockchain-based identity authentication scheme for IoT device, High-Confid. Comput. (2024) 100240.

[43] T.T. Truong, M.T. Tran, A.D. Duong, Improvement of the more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on ECC, in: 2012 26th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2012, pp. 698–703.

[44] D. He, S. Zeadally, N. Kumar, J.H. Lee, Anonymous authentication for wireless body area networks with provable security, IEEE Syst. J. 11 (4) (2016) 2590–2601.

[45] X. Li, J. Peng, M.S. Obaidat, F. Wu, M.K. Khan, C. Chen, A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems, IEEE Syst. J. 14 (1) (2019) 39–50.

[46] D. Abbasinezhad-Mood, M. Nikooghadam, An anonymous ECC-based self-certified key distribution scheme for the smart grid, IEEE Trans. Ind. Electron. 65 (10) (2018) 7996–8004, http://dx.doi.org/10.1109/TIE.2018.2807383.

[47] K.A. Shim, Universal forgery attacks on remote authentication schemes for wireless body area networks based on internet of things, IEEE Internet Things J. 6 (5) (2019) 9211–9212, http://dx.doi.org/10.1109/JIOT.2019.2922701.

[48] P. Vijayakumar, M.S. Obaidat, M. Azees, S.H. Islam, N. Kumar, Efficient and secure anonymous authentication with location privacy for IoT-based WBANs, IEEE Trans. Ind. Inform. 16 (4) (2020) 2603–2611, http://dx.doi.org/10.1109/TII.2019.2925071.

[49] Q. Qian, Y.L. Jia, R. Zhang, A lightweight RFID security protocol based on elliptic curve crytography, Int. J. Netw. Secur. 18 (2) (2016) 354–361.

[50] G.-h. Wei, Y.-l. Qin, W. Fu, An improved security authentication protocol for lightweight RFID based on ECC, J. Sens. 2022 (2022).

[51] Z. Ali, A. Ghani, I. Khan, S.A. Chaudhry, S.H. Islam, D. Giri, A robust authentication and access control protocol for securing wireless healthcare sensor networks, J. Inf. Secur. Appl. 52 (2020) 102502.
[52] S. Jegadeesan, M. Azees, A.S. Rajasekaran, F. Al-Turjman, Lightweight privacy and confidentiality preserving anonymous authentication scheme for WBANs, IEEE Trans. Ind. Inf. 18 (5) (2022) 3484–3491, http://dx.doi.org/10.1109/TII.2021.3097759.
[53] Y. Wang, Y. Liu, RC2PAS: Revocable certificateless conditional privacy-preserving authentication scheme in WBANs, IEEE Syst. J. 16 (4) (2022) 5675–5685.
[54] C. Pu, H. Zerkle, A. Wall, S. Lim, K.K.R. Choo, I. Ahmed, A lightweight and anonymous authentication and key agreement protocol for wireless body area networks, IEEE Internet Things J. 9 (21) (2022) 21136–21146.
[55] M. Shariq, K. Singh, P.K. Maurya, A. Ahmadian, M.R.K. Ariffin, URASP: An ultralightweight RFID authentication scheme using permutation operation, Peer-to-Peer Netw. Appl. 14 (2021) 3737–3757.
[56] M.A. Khan, S. Ullah, T. Ahmad, K. Jawad, A. Buriro, Enhancing security and privacy in healthcare systems using a lightweight RFID protocol, Sensors 23 (12) (2023) 5518.
[57] P. Kumar, A.K. Pal, S.H. Islam, 2F-MASK-VSS: Two-factor mutual authentication and session key agreement scheme for video surveillance system, J. Syst. Archit. (2024) 103196.
[58] D. Rani, S. Tripathi, Design of blockchain-based authentication and key agreement protocol for health data sharing in cooperative hospital network, J. Supercomput. 80 (2) (2024) 2681–2717.
[59] U. Chatterjee, S. Ray, S. Adhikari, M.K. Khan, M. Dasgupta, An improved authentication and key management scheme in context of IoT-based wireless sensor network using ECC, Comput. Commun. 209 (2023) 47–62.
[60] N. Khan, J. Zhang, G.A. Mallah, S.A. Chaudhry, A secure and efficient information authentication scheme for E-healthcare system, Comput. Mater. Continua 76 (3) (2023).
[61] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.
[62] S.F. Aghili, H. Mala, M. Shojafar, P. Peris-Lopez, LACO: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT, Future Gener. Comput. Syst. 96 (2019) 410–424.
[63] M. Fotouhi, M. Bayat, A.K. Das, H.A.N. Far, S.M. Pournaghi, M.A. Doostari, A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT, Comput. Netw. 177 (2020) 107333.
[64] H. Amintoosi, M. Nikooghadam, M. Shojafar, S. Kumari, M. Alazab, Slight: A lightweight authentication scheme for smart healthcare services, Comput. Electr. Eng. 99 (2022) 107803.
[65] M. Fariss, H. El Gafif, A. Toumanari, A lightweight ECC-based three-factor mutual authentication and key agreement protocol for WSNs in IoT, Int. J. Adv. Comput. Sci. Appl. 13 (6) (2022).
[66] M. Safkhani, S. Kumari, M. Shojafar, S. Kumar, An authentication and key agreement scheme for smart grid, Peer-to-Peer Netw. Appl. 15 (3) (2022) 1595–1616.

Mohammad Reza Servati is a highly skilled computer engineer and researcher with a remarkable focus on security and privacy in resourceconstrained environments. He graduated with an impressive A grade for his Master's degree in Computer Engineering. During his master's program, Servati dedicated his research to evaluating a wide range of protocols, from basic and lightweight to fully functional and ultra-lightweight. He conducted thorough security analyses of these protocols using both manual techniques, such as BAN logic and the Random Oracle Model, as well as formal verification methods like ProVerif, AVISPA, and Scyther. Servati's investigations uncovered vulnerabilities in several of the studied protocols, which motivated him to design a robust and secure alternative that could withstand both passive and active attacks while maintaining appropriate computational and communication costs. Servati's contributions to the fields of privacy and security have been recognized through the publication of his research in prestigious journals and conference proceedings. He has a keen interest in exploring privacy and security in the context of machine learning and constrained environments.

Masoumeh Safkhani received the Ph.D. degree in electrical engineering from the Iran University of Science and Technology, in 2012, with the security analysis of RFID protocols as her major field. She is currently an Assistant Professor with the Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. Her current research interests include the security analysis of lightweight and ultra-lightweight protocols, targeting constrained environments, such as RFID, the IoT, VANET, and WSN. She is the author/co-author of over 50 technical articles in information security and cryptology in major international journals and conferences.

Amir Masoud Rahmani received his BS in Computer Engineering from Amir Kabir University, Tehran, in 1996, the MS from Sharif University of Technology, Tehran, in 1998, and the Ph.D. degree in computer engineering from IAU University Tehran, in 2005. Currently, he is a professor in the department of computer engineering. He is the author/co-author of more than 350 publications in technical journals and conferences. His research interests are in distributed systems, the Internet of things, and evolutionary computing.

Mehdi Hosseinzadeh is Director of the DTU AI & Data Science Hub (DAIDASH) at Duy Tan University, Vietnam. He received his B.S. degree in Computer Hardware Engineering from Azad University, Dezful Branch, Iran, in 2003, and his M.Sc. and Ph.D. degrees in Computer System Architecture from Azad University, Science and Research Branch, Tehran, Iran, in 2005 and 2008, respectively. With a prolific research career, Mehdi has authored over 400 peer-reviewed publications and supervised more than 120 Master's and Ph.D. students. His research interests include applied artificial intelligence, deep learning, health informatics, data analysis, Internet of Things (IoT), and social network analysis.