Authentication Protocol: Optimization and verification of authentication protocols in the context of IoT for e Health

Cheikhou Oumar SOW* and Youssou FAYE*
*Department of Computer Sciences, UASZ University, Senegal

co.s1@zig.univ.sn, yfaye@univ-zig.sn

Abstract. Connected devices are increasingly deployed across diverse domains, including industrial monitoring and e-health. A persistent challenge common to all sensor-based applications lies in the inherent limitations of these devices, particularly their restricted hardware resources, with energy representing the most critical constraint. Within such Internet of Things (IoT) environments, notably in medical monitoring, sensors must undergo authentication prior to transmitting data to a gateway; consequently, secure authentication between communicating entities constitutes a fundamental requirement. As many IoT devices operate on battery power and transmit data periodically, it is imperative that they adopt lightweight authentication protocols to mitigate energy consumption. Although several approaches such as static and dynamic authentication have been proposed, they remain inadequate when confronted with the stringent energy and storage constraints of sensors. Continuous authentication has emerged as a promising alternative, complementing static methods by ensuring the legitimacy of a sender without reliance on computationally intensive cryptographic operations. In this paper, we enhance a continuous authentication protocol with the objective of optimising energy expenditure associated with security mechanisms. Our proposed solution demonstrates reduced energy consumption and fewer computational operations through the adoption of a query-based model. Finally, the protocol is validated using the AVISPA formal verification tool.

Keywords: Sensor node; Internet of things, Cryptography, Data authentication, RFID, AVISPA

1 Introduction

Over the past few decades, the Internet has profoundly transformed our way of life. This transformation is characterised by the rise of the Internet of Things (IoT), a network interconnecting physical devices, sensors, and intelligent systems. By extending information and communication technologies (ICTs) to diverse sectors such as healthcare, industry, and home automation, the IoT enriches everyday life. However, as with any digital system, security remains a fundamental requirement, though in this case it is complicated by the energy constraints inherent to connected devices. To ensure reliable communications and safeguard data, it is crucial to implement lightweight authentication protocols that combine security with energy efficiency. In the medical domain, for instance, a biomedical sensor must authenticate itself before transmitting sensitive information to a gateway, thereby confirming its identity and preventing malicious intrusions. The major challenge lies in striking a balance between secure exchanges and energy efficiency, which necessitates minimising the cryptographic operations performed by sensors. To this end, various authentication solutions based on symmetric and asymmetric cryptography have been studied [1], [2], [3]; however, these

operations entail significant energy consumption. Moreover, although static authentication solutions have been proposed [4], [5], they still exhibit limitations in terms of energy saving. Consequently, continuous authentication was introduced for the first time in [6], [7–8] to address the issue of reducing energy consumption, proving to be lightweight in terms of computation. In [9], the authors propose a continuous authentication method that benefits from using only lightweight operations, notably hash-based message authentication codes (HMACs) [10] and the exclusive OR operation. As a future direction, the authors suggest initiating the process via the gateway. Building upon this optimisation approach, we propose a solution that enables communications to be initiated from the gateway, thereby supporting request-based applications while reducing the number of operations required at the sensor level. An analysis of cryptographic operations demonstrates that our solution is more cost-effective than the one described in [9]. The remainder of this article is structured as follows: Section 2 presents the state of the art in authentication protocols. Section 3 details the study of the protocol proposed by Yo-Hsuan Chuang et al., including its various phases as well as a security and performance analysis. Our new proposal is described in Section 4, followed by a performance evaluation in Section 5, and a formal analysis using AVISPA in Section 6. Finally, Section 7 concludes the study and outlines future research perspectives.

2 Relate work

IoT environments, which are often open or semi-open, expose sensors and gateways to multiple threats. Attackers can easily access deployed devices, making them vulnerable to various attacks. Authentication is therefore an essential mechanism for ensuring secure and reliable communication between devices. In this section, we present a review of the literature on static and continuous authentication protocols in IoT environments, with a focus on authentication between sensors and gateways. Static authentication protocols can be grouped into three categories: certificate based, encryption-based, and non-encryption-based. Certificate-based authentication refers to the use of a digital certificate to identify a user, machine or device before granting it access to a resource. This method is often deployed alongside other traditional methods such as username and password-based authentication. In 2013, Kothmayr et al. [2] proposed a DTLS-based system using RSA and X.509, but this solution remains computationally and storage-intensive. Porambage et al. [1] subsequently introduced a protocol in 2014 based on implicit certificates with ECC (ECQV and ECDH), which is lighter than RSA but still resource-intensive. More recently, in 2022, Kumar et al. [3] proposed a lightweight authentication scheme based on optimised ECC, suited to domestic sensors and reducing energy consumption. Encryption-based authentication relies on encrypted keys, stored in each router and client (computer) authorised to be on the network. Within this category of authentication, mechanisms based on AES and HMAC have been extensively studied. Khemissa and Tandjaoui [4] proposed a lightweight mutual authentication scheme using HMAC and AES in 2015, which was extended in 2016 to include remote users [5]. However, these approaches suffer from scalability and key management issues. Recent work, such as that by Alsahlani and Popa [6] in 2021, introduces LMAAS-IoT, a lightweight, multi-factor scheme designed for resource-constrained sensors and cloud gateways. For non-cryptographic authentication, the proposed approaches do not use any certification techniques or other cryptographic operations.

Some approaches rely solely on hash functions and XOR operations. Gope et al. [11] proposed a non-traceable protocol in 2015 that guarantees identity anonymity. In the same year, Kawamoto [12] presented a method based on the sensors' environmental information. More recently, Ebrahimpour and Babaie [7] in 2024 conducted a systematic review of non-cryptographic protocols, highlighting their relevance for ultra-constrained sensors, whilst warning against replay attacks. Thus, this approach appears to be the most suitable in the context of connected devices as it uses no cryptographic operations whilst overcoming security challenges with considerable scalability.

This subsection explores scholarly contributions concerning continuous authentication. The protocols are categorised into two principal models: user–device and device–device.

Within the user–device paradigm, a range of mechanisms has emerged in recent years [6, 7–13]. These approaches are designed to ensure the ongoing verification of the legitimate user, thereby mitigating risks of unauthorised access or identity fraud. Such frameworks predominantly employ biometric techniques as their foundation. Several biometric schemes have been proposed, such as Bailey et al. [16] in 2014 using keyboard/mouse interaction analysis, Kawamoto et al. [6] in 2015 using touch gestures, and Zhou et al. [14] in 2017 using brainwaves. However, these approaches are costly and intrusive. Recent work by MDPI, in 2021-2023, explores the use of artificial intelligence for continuous authentication based on sensor behaviour, reducing reliance on biometrics. The device–device framework, once largely disregarded, has now emerged as a pivotal element in IoT security. As early as 2015, Bamasag and Youcef-Toumi [15] underscored the imperative of adopting continuous authentication within interconnected environments. Subsequently, in 2018, Chang et al. [8] introduced a lightweight continuous authentication scheme tailored for resource-constrained IoT infrastructures. Their design incorporates the notion of a valid authentication interval, thereby sustaining robust protection whilst alleviating the computational burden imposed upon sensor nodes.

In summary, the proposal of an authentication scheme without the use of crypto graphic operations and the advantage of not storing personal information—unlike approaches using biometrics—motivated our choice of the protocol in [9]. In the context of monitoring patients with chronic conditions, such as diabetes or hypertension, connected devices can play a crucial role by enabling real-time monitoring of their vital signs. However, in the medical field, data security is an absolute priority. Medical information is extremely sensitive, and any unauthorised access can have serious consequences for patients. Furthermore, compliance with data protection regulations is essential. Thus, the major challenge lies in striking a balance between data security and user-friendliness for healthcare professionals and patients. It is essential to design a protocol that is both secure and energy-efficient, in order to maintain access to healthcare.

3 Protocol of Yo-Hsuan Chuang

3.1 Design Concept

This section analyses the authentication framework advanced by Yo-Hsuan Chuang et al. [8], outlining its conceptual design and the three constituent phases: initialisation, static authentication, and continuous authentication. In certain IoT environments, such as industrial monitoring or intelligent healthcare systems, sensor nodes frequently transmit considerable volumes of data to the gateway within exceedingly brief intervals. Owing to the brevity of these transmission cycles, the gateway must re-establish authentication with devices at the outset of each session. To expedite the verification of device legitimacy for every data packet exchanged during a valid session, the authors adopted a continuous authentication strategy, thereby minimising the time required for each transfer. The protocol described in [9] employs the residual battery capacity of sensor nodes as a dynamic parameter for authentication. The protocol is organised into two distinct phases: a static authentication phase and a continuous authentication phase, each governed by its respective mechanism. The static phase, comparable to conventional approaches, authenticates devices at the commencement of each period T. In contrast, the continuous phase is applied to every transmission within the same interval, thereby ensuring that the sensor and gateway mutually verify one another during each exchange. During the predefined interval T, the static process is first executed to generate an authenticated token, which is subsequently utilised throughout the session. Within this timeframe, the gateway can swiftly validate the legitimacy of the sensor node for each message or data packet. By leveraging this token, the continuous mechanism significantly reduces computational overhead compared with the static scheme. Importantly, the protocol outlined in [9] dispenses with cryptographic operations, thereby offering a lightweight authentication solution tailored to resource-constrained IoT environments.

3.2 Protocol phase

This subsection provides an in-depth examination of the protocol developed by Yo-Hsuan Chuang et al. [8], which is structured into three distinct phases: the initialisation phase, the static authentication phase, and the continuous authentication phase.

Initialisation phase

During the **initialisation stage**, a set of fundamental parameters must be established for both sensor nodes and their associated gateway. This procedure, executed only once between a sensor and the gateway, is considered to entail a negligible exposure to potential threats. Specifically, the sensor node conveys its identifier (IDSN) together with details concerning its battery status including lifetime and capacity to the gateway through a secure communication channel. Various strategies may be adopted to provision the secret values required by sensors and gateways; for instance, manufacturers may embed these values during the production process, while a gateway may retrieve them from a trusted third-party server.

Static Authentication Phase.

During the **static authentication stage**, the sensor node and the gateway engage in a process of mutual verification. This exchange facilitates the concurrent negotiation of

an initial token, denoted TK^{I}_{SN}, which is subsequently employed to support continuous authentication throughout the designated period T In parallel, the gateway computes an estimated threshold for the residual battery capacity, referred to as BCT_{SN} This value functions as a benchmark against which the plausibility of the sensor's reported battery capacity is assessed during communication. Each time data is received from the sensor, the gateway cross-checks the transmitted battery value against the threshold, thereby reinforcing both the reliability of energy reporting and the robustness of the authentication procedure.

Continuous Authentication Phase.

The continuous authentication phase commences during the transmission of data from the sensor node to the gateway, subsequent to a successful static authentication. In this configuration, the sensor retains the initial token TK^{I}_{SN}, whilst the gateway has already established the minimum energy threshold BCT_{SN} for the current authentication interval T. Upon receiving data, the gateway performs a series of verification procedures to confirm the authenticity of the sensor. It first ensures that the incoming message was generated within the prescribed authentication period, thereby mitigating threats such as replay or timing attacks. It then validates the integrity of the transmitted information through the parameter M5, and examines the residual battery capacity rb, which must remain within an acceptable range. Finally, the gateway confirms the legitimacy of the transmission by issuing an acknowledgement (ACK) to the sensor node.

4 Extension of the Protocol.

In this section, we first describe the improvements we have made to the protocol under study. These improvements are of two kinds. On the one hand, they involve optimising the energy consumption associated with additional operations at the sensor level. Secondly, we present our optimisation method, which not only reduces the size of the data to be encrypted but also decreases the number of operations performed by the sensors; we then analyse its performance in terms of operations; finally, we will carry out a simulation using the AVISPA tool to confirm the security properties. In their outlook, the authors had highlighted the initiation of an authentication request by the gateway. This would allow the gateway to query the sensors without waiting for an initialisation from them. Thus, they proposed a solution that modifies the static authentication phase, in which the gateway initiates a request. With a view to optimising this overall solution, we have improved the continuous authentication phase to adapt it to the needs of request-based applications. Our proposed solution, whilst retaining all the advantages of the authors' proposals, reduces the number of operations that impact energy consumption.

4.1 Flexibility of the Solution

The ability for the gateway to initiate communications makes the protocol more flexible and better suited to request-based applications, whilst maintaining security properties. Unlike the initial model, where sensors continuously sent authentication requests which burdened their operations and consumed their limited resources the proposed

solution reduces the number of authentications on the sensor side. It thus supports event-driven or request-based applications by allowing the gateway to trigger the initialisation, optimising the sensors' energy consumption and storage capacity.

4.2 Static Authentication Phase

The **static authentication phase** of the extended protocol, conceived to reinforce the initialisation procedure, unfolds through a structured sequence of operations. The gateway initiates the process by generating a random value n1, computes M0 = $HMAC_{SKSN}$ (ID_{SN} || n1), and transmits n1 and M0 to the sensor node. Upon reception, the sensor node validates the gateway's authenticity using its stored identifier. It retrieves the identity ID_{SN} established during initialisation, recomputes M0' = $HMAC_{SKSN}$ (ID_{SN} || n1), and compares this with the received value M0. If the two values match, authenticity is confirmed and the static authentication continues; otherwise, the session is immediately aborted. To reinforce mutual authentication, the sensor node initiates additional operations. It generates a random number v, retrieves its current residual energy capacity rb and secret key SK_{SN} from secure storage, and conceals the battery value using the XOR operator, yielding mb = rb ⊕ v. It then computes X = v ⊕ H(n1) and M1 = H ((rb || ID_{SN}) ⊕ H(SK_{SN})). Using the secret key, the sensor produces M2 = $HMAC_{SKSN}$ (ID_{SN}, X, M1, mb), ensuring message integrity and resistance to tampering. The set {ID_{SN}, X, M1, mb, M2} to the gateway. is then transmitted to the gateway. The gateway, upon receiving this set, retrieves the secret key SK_{SN} associated with ID_{SN} from its database and recomputes M2' = $HMAC_{SKSN}$ (IDSN, X, M1, mb). If M2' = M2, the authenticity of the message is validated; otherwise, the session is terminated. The gateway subsequently calculates v' = X ⊕ H(n1), derives rb' = mb ⊕ v', and generates M1' = H ((rb' || IDSN) ⊕ H(SKSN)). If M1' = M1, the integrity of the transmitted values v' and rb' is confirmed, if not, the protocol is interrupted. Once these verifications are successfully completed, the gateway proceeds without compromising security properties and transmits to the sensor the values M3, M4 and Y, these are required for subsequent validation steps prior to the transmission of collected data, in accordance with the defined timeframe.

4.3 Data Authentication – Continuous Phase

The **continuous authentication stage** is activated during the transmission of sensor-collected data to the gateway, subsequent to a validated static authentication within the current period T. As this phase follows a successful static exchange, the sensor node retains the initial token TK^{I}_{SN}, whilst the gateway has already determined the residual battery capacity threshold BCT_{SN} for the given interval. Upon receiving a message, the gateway undertakes a systematic series of verifications to ensure both authenticity and integrity. First, it confirms that the transmission falls within the ongoing period T, thereby mitigating replay and man-in-the-middle attacks. Next, it validates the integrity marker M5, and examines the residual battery capacity rb, to ensure that it lies within an acceptable operational range. Finally, the gateway issues an acknowledgement (ACK) to the sensor node, thereby confirming the legitimacy of the exchange. The extended protocol specifies the following operations. The sensor node generates a random number r2 and retrieves its current energy capacity rb. It then extracts the initial token TK^{I}_{SN} from secure storage and computes: mb = rb ⊕ H(TK^{I}_{SN} || (m ⊕ r2)) where m

denotes the random value generated during the static phase. The sensor masks the data by calculating ms = sd ⊕ H((TK$^{I}_{SN}$ ⊕ m) || r2). It then produces a control value M5 = HMAC$_{TK^{I}_{SN}}$ (ID$_{SN}$, ms, mb, r2), which safeguards against message tampering and ensures authenticity. The set {IDSN, M5, mb, ms, r2} is transmitted to the gateway. Upon reception, the gateway records the current timestamp tc and verifies that the message belongs to the active period T. If (tc - ts) ≥ T the transmission is deemed invalid and a new static authentication is required. The gateway then validates integrity by recomputing rb' = mb ⊕ H(TK$^{I}_{SN}$ || (m ⊕ r2)), sd' = ms ⊕ ((TK$^{I}_{SN}$ ⊕ m) || r2) and M5 = HMAC$_{TK^{I}_{SN}}$ (ID$_{SN}$, ms, mb, r2). If M5' = M5, the integrity of the message is confirmed. The gateway also ensures that the residual capacity satisfies (BCT$_{SN}$ ≤ rb' ≤ er). otherwise, the session is terminated. Once these checks are completed, the gateway updates er = rb', generates a random number n2, and computes: Y1 = n2 ⊕ H ((TK$^{I}_{SN}$ ⊕ r2) || m), ACK = H ((m ⊕ rb') || (n2 ⊕ r2) || (m ⊕ TK$^{I}_{SN}$)). It updates the value of m = n2 and transmits {Y1, ACK} to the sensor. Upon reception, the sensor verifies that the transmission occurred within a valid timeframe. It retrieves n2 by computing n2' = Y1 ⊕ H ((TK$^{I}_{SN}$ ⊕ r2) || m), and validates integrity by generating ACK' = H ((m ⊕ rb) || (n2' ⊕ r2) || (m ⊕ TK$^{I}_{SN}$)). If ACK' = ACK, authenticity and integrity are confirmed, thereby validating the success of continuous authentication. The sensor then updates m = n2' and prepares for subsequent continuous authentication in future transmissions. As in the static phase, the reception of ACK and Y1 authorises the sensor to continue transmitting data; otherwise, it remains idle until a new static authentication is initiated by the gateway.

5 Performance Evaluation

In the section devoted to security analysis, all fundamental properties have been preserved and no security service has been compromised. To evaluate the computational performance of our proposal, we define the following notations to express the time consumption of different operations: Tx where T represents the execution time and the subscript x denotes the security mechanism under consideration. Table 1 presents a comparison of the number of operations between the protocol of Khemissa et al. [4], the studied protocol [8], and our proposal. According to the findings presented in [17], the mean execution time of an AES encryption is approximately 2.76 ms, whilst a hash function requires around 1.5 ms, and an HMAC computation takes close to 3.54 ms. In addition, the generation of a pseudo-random number has been reported to demand roughly 0.65 ms [18]. Within the static authentication phase, our proposed protocol necessitates the following computational workload 4TRandom + 14THash + 6THMAC, by contrast, the studied protocol requires 4TRandom + 16THash + 4THMAC. This comparative analysis demonstrates that our design achieves a more balanced distribution of operations, thereby reducing the reliance on hash functions while strengthening message integrity through additional HMAC computations. Furthermore, in our solution, the lengths of the identifier IDSN the random numbers (n1, n2, v, w) and SKSN are all fixed at 128 bits. This choice ensures a robust level of security whilst simultaneously maintaining computational efficiency, thereby aligning with the requirements of lightweight IoT environments.

Table 1. Comparison of the Number of Operations

Phases	Khemissa et al.	Protocole Etudié	Extension du Protocole étudié
Authentification Statique	$2\,T_{Random} + 2\,T_{Hash} + 4\,T_{HMAC} + 2T_{AES}$	$4T_{Random} + 16T_{Hash} + 4T_{HMAC}$	$3T_{Random} + 14_{THash} + 6T_{HMAC}$
Authentification Continue	---	(1) : $2T_{Random} + 9T_{Hash} + 1T_{HMAC}$	(1) : $1T_{Random} + 2T_{Hash} + 1T_{HMAC}$
		(2) : $2T_{Random} + 8T_{Hash} + 2T_{HMAC}$	(2) : $2T_{Random} + 8T_{Hash} + 2T_{HMAC}$

6 Simulation

6.1 Tools presentation simulation

AVISPA (Automated Validation of Internet Security Protocols and Applications) is a tool for the formal verification of security protocols and applications, using the HLPSL (High Level Protocol Specification Language) to describe specifications which are then translated into the IF intermediate format using the hlpsl2if translator, before being analysed by one of its four complementary back-ends: OFMC, which explores transitions whilst supporting algebraic operators; CL-AtSe, which transforms specifications into logical constraints to search for attacks; SATMC, which encodes the system in propositional formulas to detect security violations; and TA4SP, which uses tree automata to approximate the intruder's knowledge and determine whether states are reachable, thereby enabling conclusions to be drawn regarding the absence or presence of attacks. An HLPSL specification consists of role definitions (basic and composition), a list of security properties expressed by keywords, and a main role triggered by the environment() command, which instantiates the roles and launches the verification.

6.2 Presentation of results obtained

Protocol specification.

The HLPSL script corresponding to our protocol is presented in the appendix. It defines two fundamental roles, namely gatewayeG and sensorS, which respectively describe the behaviour of the gateway and the sensor. The central principle of the protocol lies in the preservation of secret values (Ids, Hids, F, Ps, Pi) throughout the authentication phase between the gateway (GW) and the sensor (SN). Mutual authentication is formalised through the objectives witness and request, the details of which are provided in the appendix.

Verification results.

After executing our protocol specified in HLPSL, the results obtained are presented in **Table 2**. as indicated in this table, the AVISPA tool returns the state **SAFE** from its two main back-ends: the On-the-Fly Model Checker (OFMC) and the Constraint Logic-based Attack Searcher (CL-AtSe). In contrast, the SAT-based Model Checker (SATMC) and the Tree Automata-based Protocol Analyzer (TA4SP) produce an **INCONCLUSIVE** result, due to unsupported operations. This observation means that AVISPA did not detect any attack against our protocol in the tested scenarios, thereby confirming the robustness of the proposed solution.

Table 2: AVISPA validation results

Back-end AVISPA	Résultats
OFMC	SAFE
CL-AtSe	SAFE
SATMC	INCONCLUSIVE
TA4SP	INCONCLUSIVE

Figure 1: OFMC Back-end Output Results

Figure 2: CL-AtSe Back-end Output Results

7 Conclusion

In summary, this study has revealed the crucial importance of optimising authentication protocols for connected devices in the healthcare sector. Building on Chang's protocol, we have implemented significant changes that have led to a reduction in the size of the data sent and the number of operations required. These improvements result in a significant reduction in energy consumption and enhanced security. The results of our analysis and simulation confirm that our proposed extension maintains all the essential security properties while optimising performance. By allowing the gateway to initiate authentication requests, our solution provides greater flexibility and better support for request-based applications. Furthermore, the continuous authentication phase ensures that data transmissions remain secure and efficient throughout the communication period. The AVISPA simulation results validate the robustness of our protocol, showing that it is SAFE against potential attacks in the tested scenarios. Future research directions include extending this work to other IoT environments beyond healthcare, exploring additional lightweight cryptographic techniques, and integrating artificial intelligence methods to further enhance adaptability and resilience. Ultimately, the

proposed protocol contributes to the development of se cure, energy-efficient IoT systems that can reliably support sensitive applications such as e-health.

References

1. Porambage, P., Ylianttila, M., Schmitt, C., Kumar, P., Gurtov, A.: Certificate-based lightweight authentication for distributed IoT environments using ECC. *IEEE International Conference on Communications* (2014).
2. Kothmayr, T., Schmitt, C., Hu, W., Brunig, M., Weber, M.: DTLS based security and authentication for IoT. *IEEE International Conference on Computer Communications Workshops* (2013).
3. Kumar, R., Singh, A., Sharma, P.: Lightweight authentication scheme for smart home IoT devices. *Journal of Ambient Intelligence and Humanized Computing* 13(5), 2457–2469 (2022).
4. Khemissa, H., Tandjaoui, D.: A lightweight mutual authentication scheme for IoT. *International Conference on Advanced Networking Distributed Systems and Applications* (2015).
5. Khemissa, H., Tandjaoui, D.: Extended lightweight authentication for remote IoT users. *International Conference on Advanced Networking Distributed Systems and Applications* (2016).
6. Alsahlani, A., Popa, M.: LMAAS-IoT: Lightweight Multi-factor Authentication and Authorization Scheme for IoT. *Sensors* 21(12), 4123 (2021).
7. Ebrahimpour, A., Babaie, S.: Authentication in Internet of Things: Protocols, Attacks, and Open Issues. *Wireless Personal Communications* (Springer) (2024).
8. Chuang, Y.-H., Lo, N.-W., Yang, C.-Y., Tang, S.-W.: A Lightweight Continuous Authentication Protocol for the Internet of Things. *Sensors* 18(4), 1104 (2018).
9. Protocole étudié et modifié, basé sur Chuang et al. “Notre contribution, inspirée de [8]”
10. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. *Advances in Cryptology — CRYPTO’96*. Lecture Notes in Computer Science, vol. 1109, Springer, Berlin, Heidelberg (1996).
11. Gope, P., Hwang, T.: Untraceable lightweight authentication protocol for distributed IoT architecture. *Future Generation Computer Systems* 80, 128–137 (2018).
12. Kawamoto, Y.: Ambient information-based authentication in IoT environments. *International Conference on Ubiquitous Computing and Ambient Intelligence* (2015).
13. Liang, Y., Samtani, S., Guo, B., Yu, Z.: Behavioral Biometrics for Continuous Authentication in the Internet-of-Things Era: An Artificial Intelligence Perspective. *IEEE Internet of Things Journal* 7(9), 8953–8968 (2020).
14. Zhou, J., Cao, Z., Dong, X., Vasilakos, A.: Transparent authentication using brainwave patterns. *IEEE Transactions on Information Forensics and Security* 12(5), 1102–1113 (2017).
15. Bamasag, O., Youcef-Toumi, K.: Continuous lightweight authentication for IoT environments. *International Conference on Internet of Things and Applications* (2015).
16. Bailey, K., et al.: Continuous authentication using GUI interaction patterns. *International Conference on Human-Computer Interaction* (2014).
17. Pereira, G.C.C.F.; Alves, R.C.A.; da Silva, F.L.; Azevedo, R.M.; Albertini, B.C.; Margi, C.B. Performance Evaluation of Cryptographic Algorithms over IoT Platforms and Operating Systems. Secur. Commun. Netw. 2017, 2017, 2046735. [CrossRef]
18. Yeh, K.H.; Su, C.; Choo, K.R.; Chiu, W. A Novel Certificateless Signature Scheme for Smart Objects in the Internet-of-Things. Sensors 2017, 17, 1001. [CrossRef] [PubMed]