sustainability

Article

Toward Designing a Secure Authentication Protocol for IoT Environments

Mehdi Hosseinzadeh [1,2,3], Mazhar Hussain Malik [4], Masoumeh Safkhani [5,6], Nasour Bagheri [6,7], Quynh Hoang Le [1,2], Lilia Tightiz [8,*] and Amir H. Mosavi [9,10,*]

1 Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
2 School of Medicine and Pharmacy, Duy Tan University, Da Nang 550000, Vietnam
3 Computer Science, University of Human Development, Sulaymaniyah 0778-6, Iraq
4 School of Computing and Creative Technologies College of Arts, Technology and Environment (CATE) University of the West of England Frenchay Campus, Coldharbour Lane, Bristol BS16 1QY, UK
5 Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran P.O. Box 16788-15811, Iran
6 School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran P.O. Box 19395-5746, Iran
7 Faculty of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran P.O. Box 16788-15811, Iran
8 School of Computing, Gachon University, 1342 Seongnamdaero, Seongnam 13120, Republic of Korea
9 John von Neumann Faculty of Informatics, Obuda University, 1034 Budapest, Hungary
10 Institute of the Information Society, University of Public Service, 1083 Budapest, Hungary
* Correspondence: liliatightiz@gachon.ac.kr (L.T.); amirhosein.mosavi@stuba.sk (A.H.M.)

Abstract: Authentication protocol is a critical part of any application to manage the access control in many applications. A former research recently proposed a lightweight authentication scheme to transmit data in an IoT subsystem securely. Although the designers presented the first security analysis of the proposed protocol, that protocol has not been independently analyzed by third-party researchers, to the best of our knowledge. On the other hand, it is generally agreed that no cryptosystem should be used in a practical application unless its security has been verified through security analysis by third parties extensively, which is addressed in this paper. Although it is an efficient protocol by design compared to other related schemes, our security analysis identifies the non-ideal properties of this protocol. More specifically, we show that this protocol does not provide perfect forward secrecy. In addition, we show that it is vulnerable to an insider attacker, and an active insider adversary can successfully recover the shared keys between the protocol's entities. In addition, such an adversary can impersonate the remote server to the user and vice versa. Next, the adversary can trace the target user using the extracted information. Finally, we redesign the protocol such that the enhanced protocol can withstand all the aforementioned attacks. The overhead of the proposed protocol compared to its predecessor is only 15.5% in terms of computational cost.

Keywords: internet of things; security; authentication; key agreement; multi-factor; smart-card; hash function; insider attacker; key compromised impersonation; key recovery

MSC: 94A62

Citation: Hosseinzadeh, M.; Hussain Malik, M.; Safkhani, M.; Bagheri, N.; Le, Q.H.; Tightiz, L.; Mosavi, A.H. Toward Designing a Secure Authentication Protocol for IoT Environments. *Sustainability* **2023**, *15*, 5934. https://doi.org/10.3390/su15075934

Academic Editors: Martin Wynn and Kamal Bechkoum

Received: 19 November 2022
Revised: 15 January 2023
Accepted: 16 January 2023
Published: 29 March 2023

1. Introduction

The Internet of Things (IoT) improves daily life by providing a communication link between various items. This communication allows us to monitor those things in real-time and take the necessary actions to improve the process. An IoT reference model [1], including several levels, started from devices/sensors for different purposes and from different technologies such as RFID and Bluetooth Low Energy (BLE) ended at symbolizing empowered individuals and corporate processes that use IoT-enabled data to drive

action. However, information security risks can affect the data and communication in each conceptualization level of an IoT system as well as their connections. Hence, each device or subsystem must be secured.

Although IoT security requires a multi-tiered strategy [1], the bulk of data at the lower layer and also between this layer and the penultimate layer, which provides connectivity between the devices and the edge computing devices, is vulnerable to adversarial access. Various ways might be used to increase the security of the transmitted data. Among them, the authentication technique is a crucial method of differentiating between friends and foes. Various authentication schemes have been proposed by researchers and each of them has its own advantages and drawbacks. In recent research, Son et al. [2] independently analyzed the security of an authentication protocol that has been designed by Rajaram et al. [3]. Their investigation revealed that the examined scheme has some flaws and cannot provide the desired security for sensitive data, which is transferred in the edge layer of an IoT system. Furthermore, Rajaram et al.'s scheme employs bilinear pairing through protocol computations, which has a significant computational cost. To overcome that scheme's drawbacks, Son et al. have introduced a new one-way cryptographic hash function-based two-factor authentication protocol. The proposed protocol also benefits from an updating system to address user anonymity. As a result, this scheme is more efficient by design compared with the Rajaram et al.'s scheme. They also evaluated its security against common attacks, e.g., replay and privileged insider attacks, besides support for perfect secrecy, user anonymity, and user untraceability. It demonstrates that Son et al.'s protocol is a good solution for many applications, particularly those with few participants, such as passive RFID tags, assuming that these security assertions are also supported by independent third-party security research. Hence, we opted to analyze the security of this system in this work because there has been no previous such security study for it.

1.1. Our Contributions

Our main findings in this paper are highlighted below:

- We conduct the first independent security analysis of a recently proposed scheme [2], to the best of our knowledge;
- We demonstrate that assuming an adversary accesses long-term secrets and also monitors the messages transferred over the secure channel; it can retrieve the shared key at the end of the session.
- We demonstrate that an adversary with access to the user's smartcard and the publicly transferred data on n subsequent sessions can extract the session key of $n-2$ sessions and also trace the user.
- We efficiently redesign Son et al.'s protocol to overcome the mentioned security flaws. Our cost analysis shows that the overhead of the new protocol is just 15.5%.

1.2. Paper Organization

The required background, notations, a shallow survey of related works, and a brief background of cryptographic hash functions are described in Section 2. Next, we investigate the suggested protocol in a former study in Section 2.5. Then, a comprehensive security investigation of that protocol is given in Section 3. The improved protocol is included in Section 4. Finally, Section 6 provides concluding remarks.

2. Preliminaries

2.1. Notation

In this study, we employ the list of notations provided by Table 1.

Table 1. The list of the used notations.

Symbol	Description
U_X	The user X
RS	The remote server
ID_x	The unique identifier of U_X, of low entropy domain
PW_X	The secret password of U_X, of low entropy domain
r, t	The random numbers produced by U_X and RS, respectively
PWD_X	A parameter that computed as $PWD_X = H(PW_x \| r)$ by U_X
SC_X	A smartcard of U_X, issued by RS
TID_X	Temporary identifier of U_X
PID_X	Temporary secret identifier of U_X
$H(\cdot)$	A one-way cryptographic hash function
a_x, b_x	Fresh random numbers generated at each session, respectively by U_X and RS
s	RS's permanent secret key
SK	Shared key between U_X and RS

2.2. Related Works

The three most critical principles in information security are confidentiality, integrity, and authenticity. Authentication protocols are a crucial component of the majority of security mechanisms that are used to perform essential access control linked to authenticity or key agreement to ensure confidentiality. Although authentication techniques such as TLS and SSL are commonly used on the internet, they cannot be employed in IoT systems, owing to numerous resource constraints. As a result, many attempts have been made to develop a suitable authentication scheme for IoT devices.

A cryptographic protocol should adhere to the confusion and diffusion properties, just like any other cryptographic primitive, to offer adequate protection against attackers. Most of the proposed authentication protocols can be categorized as ultralightweight, lightweight, or non-lightweight protocols from a high-level perspective regarding the components used. The foundation of ultralightweight protocols is bit-level operations such as Exclusive-or (XOR), Rotation, AND, and OR, for instance, SASI [4], RAPP [5], R^{2}AP [6], RCIA[7], KMAP [8], SLAP [9], SecLAP [10], Eghdamian and Samsudin's protocol [11], David-Prasad ultralightweight authentication protocol [12], and UMAPSS [13]. However, due to a lack of sufficient confusion and diffusion, nearly all protocols in this class have been severely degraded up to this point [14–21]. Precisely, Avoin et al. [15] demonstrates that a long-term key, which is shared between a reader and a tag in Eghdamian and Samsudin's ultralightweight mutual authentication protocol [11], can be obtained by an adversary. In addition, Avoin et al. also in [14] offered guidelines to design a secure ultralightweight authentication protocol. A passive full secret disclosure attack on SASI was presented in [16]. Phan et al. proved in [17] that the SASI did not achieve one of its design goals, the non-traceability property. [18] presented a desynchronization attack and secret disclosure attack against the SASI. [20] provides powerful desynchronization, traceability and secret disclosure attack against RAPP. Barrero et al. [21] presented a Tango genetic attack that employs a genetic algorithm to facilitate the generation of automatic cryptanalysis of the proposed protocol in [12]. In particular, most of the ultralightweight protocols update the secret parameters to prevent traceability while shifting the expense of the session-dependent ephemeral keys to the server or reader side, for example, to lower the sensor side cost. However, Safkhani et al. demonstrated that all such protocols are vulnerable to a desynchronization attack, in which the adversary compels the server and the sensor to maintain inconsistent sharing data and prevents them from authenticating one another as a result [19]. Table 2 summarizes the ultralightweight authentication protocols and the security analysis reports that have been presented against them.

Table 2. Summary of ultralightweight security protocols and their cryptanalysis.

Protocol	Protocol Class	Reference of Its Security Analysis
[4]	ultralightweight	[16–19]
[5]	ultralightweight	[19,20]
[6]	ultralightweight	[19]
[7]	ultralightweight	[19]
[8]	ultralightweight	[19]
[9]	ultralightweight	[19]
[10]	ultralightweight	[19]
[11]	ultralightweight	[15]
[12]	ultralightweight	[21]
[13]	ultralightweight	[19]

On the other hand, lightweight protocols are using lightweight yet reliable cryptographic primitives, e.g., block cipher [22,23], stream cipher [24], hash function [25–28], and authenticated encryption [29], to achieve acceptable security. They are symmetric by nature; however, if they are also scalable, they might not offer complete anonymity. Furthermore, if the protocol's parties keep the shared parameters fixed, it will not guarantee perfect secrecy. It is important to note that backward secrecy and forward secrecy are two terms used in the field of security analysis. With forward secrecy, the adversary cannot obtain the session keys from earlier sessions even if the long-term secret values are disclosed.

It is intended to alleviate the shortcomings of lightweight protocols utilizing asymmetric components, such RSA [30,31], pairing [3,32,33] or ECC [34–39]. However, those primitives are time-consuming; therefore, they might not be the best option for devices with limited resources.

While most of the above-mentioned protocols rely on centralized servers for time-consuming computations and data storage, many other researchers recently target decentralized approaches, thanks to the recent advances in blockchain technology. Depending on the application, solutions are based on public blockchain [40], consortium blockchain [41] or private blockchain [42]. Each type of blockchain has its pros and cons and depending on the application it should be adopted. For instance, in a public blockchain, it should be possible for anyone to join the network to create blocks and read transactions. This could be a limitation in some applications with restrictions on the leaked data. In such applications, it may be better to use other types of blockchain.

The sensor nodes and edge devices in IoT systems are dispersed throughout the field and could be accessed by the adversary physically. Another class of protocols has been developed to include the device's fingerprint throughout the authentication process to prevent such attacks. Such a fingerprint may be produced by a physically unclonable function(PUF) [43–48]. The security of such protocols may seem promising if the PUF being used behaves in an ideal manner (i.e., behave fully reliable and random); however, the PUF response relies on the environment and is not entirely random. Consequently, certain protocols could be the target of modeling attacks [49–51]. In a human-assisted protocol, employing user name and password along with a smartcard is an option [52–55] or the user biometrics [56–59]. However, the disadvantage of the user name and password is that they have low entropy (because they must be memorized), and the disadvantage of biometrics is that they are noisy and require a fuzzy extractor, which takes time. Additionally, many IoT devices, particularly detecting sensors, operate through processes without requiring user input. However, such a solution is useful for many applications, such as mobile devices. Son et al.'s protocol belongs to the smartcard-based protocols and its security against various attacks is not clear, which we investigate in this study. Table 3 categorizes the protocols reviewed in this section according to their type.

Table 3. Classification of related work protocols based on their types.

Protocol	Protocol Class
[4–13]	Ultralightweight
[43–48]	PUF based
[52–55]	Smartcard based
[56–59]	Biometric based
[40–42]	Blockchain based

2.3. Hash Function

A hash function is a frequently used primitive that converts a message of any length into a message digest of a specific length (n), such as $H(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^n$. Most applications require $128 \neq n \neq 512$. NIST has standardized three well-known hash functions: SHA-1, SHA-2, and SHA-3. However, SHA-1 is no longer secure due to known attacks [60]. Aside from those hash functions, some hash functions for constrained environments have been developed, such as Quark [61], SPONGNET [62] and PHOTON [63].

Any secure cryptographic hash function should meet the following requirements:

- Collision Resistance: the computational complexity expected to find a pair (M, M') such that $M \neq M'$ and $H(M) = H(M')$ should be $2^{n/2}$.
- Preimage Resistance: given a message digest $Y \in \{0,1\}^n$, the expected computational complexity for finding a message M such that $H(M) = Y$ should be 2^n.
- Second Preimage Resistance: given a message $M \in \{0,1\}^*$, the expected computational complexity to find a message $M' \neq M$ such that $H(M) = H(M')$ should be 2^n.

In practice, hash functions use compression functions to compute the hash digest of an arbitrary-length message, such as Sponge [64] and Merkle-Damgård [65], are used to process a message of any length. As a security metric, such a hash function should be indistinguishable from a random oracle [66].

2.4. System Model

The used system model includes these entities: the user(s), the remote server, and the attacker. The secure channel is used for system setup and registration, whereas the public channel is used for user authentication. A secure channel is a method of transmitting data that is impervious to monitoring and manipulation. Symmetric keys are used between two parties to encrypt data from beginning to end. An insecure channel, in contrast to a secure channel, is not encrypted and is vulnerable to monitoring and tampering. If the information to be communicated is encrypted before being transmitted, secure communications are possible over an insecure channel.

Following the assumption of the former study [2] and similar to [67–69], Dolev-Yao (DY) [70], for an active adversary, and Canetti and Krawczyk (CK) [71] adversary models for a stronger attacker that has more capabilities than in the DY model. All attackers are active and capable of listening in, stopping, altering, or beginning message delivery.

In this study, we also consider an insider adversary's risk. This adversary could be the source of long-term secrets or privately transmitted data leakage. We consider perfect secrecy, for example, to assess the sustainability of the target protocol against the leakage of long-term secrets and its impact on the security of previous sessions. On the other hand, to assess the later risk, we consider the impact of an insider attack. As shown in Figure 1, such an adversary could access the exchanged messages during the registration phase, which is assumed to take place over a secure channel.

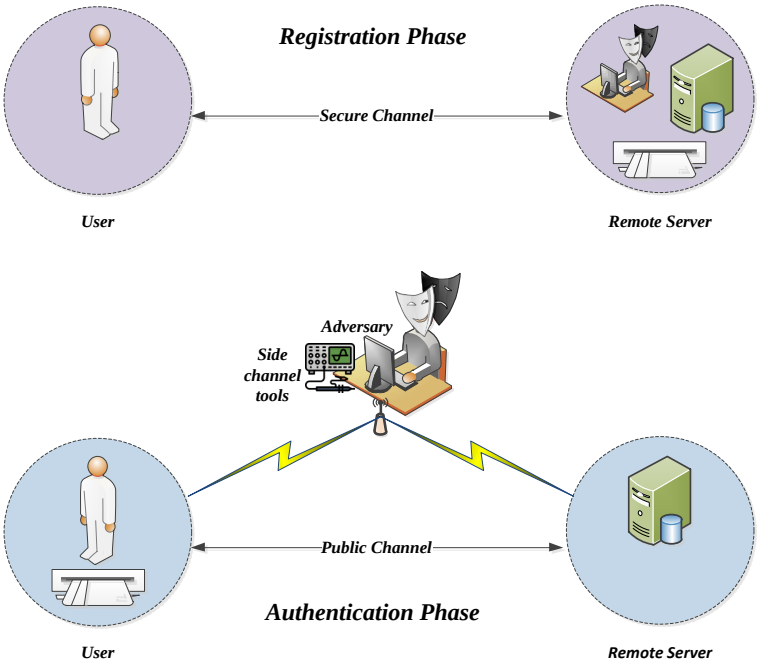

Figure 1. The used system model.

2.5. SPP Description

Son et al.'s scheme, which we call it SPP (the designers are Son, Park and Park), consists of five phases, i.e., initialization phase, registration phase, login phase, authentication phase, and password updating phase. For more details of these steps, we refer the interested reader to the original paper [2].

3. Security Analysis of SPP

SPP exceeds its predecessors regarding efficiency, although it is unclear how secure it is. As a result, a thorough security study can shed light on its particular security pros and downsides.

3.1. Insider Adversary

An insider attacker is a cyber-security danger that originates within a company. Access to the secret channel is a frequent advantage that an insider has over a regular adversary. If an insider attacker acquires a significant advantage in attacking a protocol as a result of this access, the target protocol becomes vulnerable to insider attack. In summary, an insider adversary is an authorized user in the system who can access the secure channels such as a registration channel.

An insider adversary has access to ID_X and $PWD_X = H(PW_x \| r)$ and it is also given the content of the smartcard, i.e., $(A_X, B_X, C_X, Auth_X)$, where:

$$\begin{aligned} A_X &= r \oplus H(ID_X \| PW_X) \\ B_X &= TID_X \oplus H(ID_X \| PW_X \| r) \\ C_X &= PID_X \oplus H(TID_X \| r) \\ Auth_X &= H(ID_X \| PW_X) \ mod \ l \\ TID_X &= H(ID_x \| t) \\ PID_X &= H(TID_x \| s) \end{aligned}$$

.

Apart from TID_X and timestamps T_1 and T_2, the following messages are transferred over the wireless channel:

$$\begin{aligned}
M_1 &= H(PID_X \| H(ID_X \| PWD_X)) \oplus a_X \\
M_2 &= H(TID_X \| PID_X \| a_X \| T_1) \\
M_3 &= H(PID_X \| H(ID_X \| PWD_X)) \oplus b_X \\
M_4 &= PID_X^{new} \oplus H(TID_X^{new} \| H(ID_X \| PWD_X) \| b_X) \\
M_5 &= H(SK \| PID_X^{new} \| T_2)
\end{aligned}$$

where,

$$\begin{aligned}
SK &= H(PID_X \| a_X \| b_X) \\
TID_X^{new} &= TID_X \oplus b_X \\
PID_X^{new} &= H(TID_X^{new} \| s)
\end{aligned}$$

Consider a naive opponent who has access to SC_X and the data exchanged over the public channel. Such an opponent can estimate both ID_X and PW_X simultaneously, and then use A_X to extract r, and the provided TID_X from the public channel checks the accuracy of the guessed ID_X and PW_X using $B_x \oplus TID_x = H(ID_X \| PW_X \| r)$. Assuming that the entropy of ID_X and PW_X is $\mathcal{H}_{ID}$ and $\mathcal{H}_{PW}$, respectively, the estimated complexity to drive ID_X and PW_X using the dictionary attack is $2^{\mathcal{H}_{ID}+\mathcal{H}_{PW}}$. An insider adversary with access to the sent data from U_X at the registration process, on the other hand, knows ID_X and $PWD_X = H(PW_X \| r)$. Given ID_X, the adversary guesses PW_X to determine $r = A_X \oplus H(ID_X \| PW_X)$ and uses either $PWD_X = H(PW_x \| r)$ or $B_x \oplus TID_x = H(ID_X \| PW_X \| r)$ to validate the guessed PW_X value. The complexity of this attack is $2^{\mathcal{H}_{PW}}$, which is a significant advantage over $2^{\mathcal{H}_{ID}+\mathcal{H}_{PW}}$ for a naive attacker. Consider the case when $\mathcal{H}_{ID} = \mathcal{H}_{PW} = 32$. The insider adversary is thus required to execute 2×2^{32} calculations to extract PW_X, which can be conducted in seconds on a typical personal computer, but a naive adversary is anticipated to do 2×2^{64} computations, which is presently not doable even for a medium-sized corporation [60]. As a result, SPP is vulnerable to a privileged opponent who has access to the protocol's registration step.

3.2. Key Recovery by an Insider Adversary

Consider an attacker with access to ID_X and $PWD_X = H(PW_X \| r)$. Moreover, let the attacker intercepts the transmitted messages in three subsequent sessions, namely i, $i+1$, and $i+2$. In addition to TID_X^i, T_1^i and T_2^i, the messages exchanged in j^{th} session, for $i \leq j \leq i+2$, are as follows:

$$\begin{aligned}
M_1^j &= H(PID_X^j \| H(ID_X \| PWD_X)) \oplus a_X^j \\
M_2^j &= H(TID_X^j \| PID_X^j \| a_X^j \| T_1^j) \\
M_3^j &= H(PID_X^j \| H(ID_X \| PWD_X)) \oplus b_X^j \\
M_4^j &= PID_X^{j+1} \oplus H(TID_X^{j+1} \| H(ID_X \| PWD_X) \| b_X^j) \\
M_5^j &= H(SK \| PID_X^{j+1} \| T_2^j)
\end{aligned}$$

where

$$\begin{aligned}
SK^j &= H(PID_X^j \| a_X^j \| b_X^j) \\
TID_X^{j+1} &= TID_X^j \oplus b_X^j \\
PID_X^{j+1} &= H(TID_X^{j+1} \| s)
\end{aligned}$$

Given this knowledge, the adversary does the following step-by-step computations for $i \leq j \leq i+1$:

$$\begin{aligned} b_X^j &= TID_X^j \oplus TID_X^{j+1} \\ H(PID_X^j \| H(ID_X \| PWD_X)) &= M_3^j \oplus b_X^j \\ a_X^j &= H(PID_X^j \| H(ID_X \| PWD_X)) \oplus M_1^j \\ PID_X^{j+1} &= M_4^j \oplus H(TID_X^{j+1} \| H(ID_X \| PWD_X) \| b_X^j) \\ M_2^j &= H(TID_X^j \| PID_X^j \| a_X^j \| T_1^j) \\ M_3^j &= H(PID_X^j \| H(ID_X \| PWD_X)) \oplus b_X^j \\ M_5^j &= H(SK \| PID_X^{j+1} \| T_2^j) \end{aligned}$$

Following these computations, the adversary has PID_X^{i+1}, a_X^{i+1} and b_X^{i+1}, which are sufficient to calculate $SK^{i+1} = H(PID_X^{i+1} \| a_X^{i+1} \| b_X^{i+1})$. If the adversary eavesdrops on $n \geq 3$ consequence sessions, it may identify the shared session key of $n-2$ sessions.

3.3. Impersonation by the Insider Adversary

Let us give the adversary access to the messages exchanged over the secure channel, i.e., ID_X and $PWD_X = H(PW_x \| r)$. Following the stated assault in the preceding section, such an adversary can also access PID_X and TID_X from the channel. Hence, the attacker can compute the necessary information to be authenticated as a valid user by RS. To be more explicit, given this information, the adversary constructs $a_X \in Z_P$ and extracts T_1 to compute $M_1 = H(PID_X \| H(ID_X \| PWD_X)) \oplus a_X$ and $M_2 = H(TID_X \| PID_X \| a_X \| T_1)$, and transmits (TID_X, M_1, M_2, T_1) to RS. Obviously, RS accepts this authentication message, and the attacker is authenticated as a real user.

Given ID_X and $PWD_X = H(PW_X \| r)$ from the registration phase, TID_X from the channel, and PID_X from the attack outlined in Section 3.2, you may impersonate the server. The attacker can spoof the U_X toward the RS. Next, once U_X computed $M_1 = H(PID_X \| H(ID_X \| PWD_X)) \oplus a_X$, $M_2 = H(TID_X \| PID_X \| a_X \| T_1)$ and sends (TID_X, M_1, M_2, T_1) to RS, the adversary extracts $a_X = H(PID_X \| H(ID_X \| PWD_X)) \oplus M_1$, generates $b_X \in Z_P$, computes $TID_X^{new} = TID_X \oplus b_X$, $PID_X^{new} = H(TID_X^{new} \| s)$, $M_3 = H(PID_X \| H(ID_X \| PWD_X)) \oplus b_X$, $M_4 = PID_X^{new} \oplus H(TID_X^{new} \| H(ID_X \| PWD_X) \| b_X)$, $SK = H(PID_X \| a_X \| b_X)$, and $M_5 = H(SK \| PID_X^{new} \| T_2)$. Finally, it sends (M_3, M_4, M_5, T_2), which is approved, and the attacker is identified as a valid server.

3.4. The Lack of Perfect Secrecy

Backward secrecy and forward secrecy are two concepts used in security analysis. Forward Secrecy prevents an attacker from recovering previous session keys once the long-term secret value has been revealed. According to backward secrecy, future session keys cannot be obtained by an adversary even if the long-term secret value is revealed. Perfect secure is a term used to describe a protocol that possesses both forward secrecy and backward secrecy capabilities.

Exposing a protocol participant's long-term secrets should have no effect on the security of the shared session keys in the past in order to provide forward secrecy [72]. In the registration process, RS keeps $(TID_X, H(ID_X \| PWD_X))$ for each user. While TID_X is changed after each successful protocol session, $H(ID_X \| PWD_X)$ is constant. As a result, it is vulnerable to long-term information leakage, which should not jeopardize the security of the previous session if provided to the adversary at any moment. However, if the adversary eavesdropped on $n \geq 3$ consequence sessions and is later supplied $H(ID_X \| PWD_X)$, it may use the proposed attack in Section 3.2 to recover $n-2$ shared session keys. As a result, SPP does not guarantee absolute forward secrecy.

Traceability and Anonymity

Given that a privileged insider has access to TID_X, PID_X and $H(ID_X\|PWD_X)$, based on the arguments supplied in prior sessions, it may simply trace the modified values of TID_X and PID_X from one session to the next, providing it is monitoring all sessions. However, if it fails to synchronize, it only takes three subsequent successful sessions to synchronize and retrace the target user. As a result, SPP protocol is vulnerable to Traceability by a privileged insider opponent.

4. Enhanced Protocol

The Enhanced Protocol, like its predecessor, SPP, has five pahes, which are discussed in this section.

4.1. Initialization Phase

In the initialization phase run by the remote server RS, a large prime q is selected, a secret key $s \in Z_P^*$ is chosen and a hash function $H(\cdot) : \{0,1\}^* \rightarrow Z_P$. RS keeps s securely and publishes $(q, H(\cdot))$ over the network.

4.2. Registration Phase

Any user U_X, which aims to participate in the communication network legitimately, should be registered to RS. To do so, U_X chooses the identity and password ID_X and PW_X, generates a random value $r \in Z_P^*$, computes $PWD_X = H(PW_x\|r)$ and $HID_X = H(ID_x\|r)$ and sends (HID_X, PWD_X) to the remote server. The pseudo identifier HID_x should be unique, otherwise, the registration will be rejected by RS. Assuming HID_X is unique, RS generates a random value $t \in Z_P^*$, computes $TID_X = H(HID_x\|t)$, $PID_X = H(TID_x\|s)$ and $(TID_X, H(s\|TID_x) \oplus H(HID_X\|PWD_X))$ in its secure memory and stores $(TID_X, PID_X, H(\cdot))$ in a smartcard SC_X and sends it to U_X. Once received SC_X, the user computes $A_X = r \oplus H(ID_X\|PW_X)$, $B_X = TID_X \oplus H(ID_X\|PW_X\|r)$, $C_X = PID_X \oplus H(TID_X\|r)$, and $Auth_X = H(ID_X\|PW_X\|r\|PID_X\|TID_X)$ and stores them in the received SC_X.

4.3. Login and Authentication Phases

To share a session key SK, as it is depicted in Figure 2, the user should login successfully using its smartcard S_X and also should be authenticated by the remote server. The required process is as follows:

1. U_X inputs ID_X and PW_X in SC_X. Then, SC_X computes $r = A_X \oplus H(ID_X\|PW_X)$, $TID_X = B_X \oplus H(ID_X\|PW_X\|r)$, $PID_X = C_X \oplus H(TID_X\|r)$, and checks $Auth_X \stackrel{?}{=} H(ID_X\|PW_X\|r\|PID_X\|TID_X)$. If they are equal, SC_X generates $a_X \in Z_P$ and extracts the current timestamp T_1, and computes $HID_X = H(ID_x\|r)$, $M_1 = H(PID_X\|H(HID_X\|PWD_X)) \oplus a_X$ and $M_2 = H(TID_X\|PID_X\|a_X\|T_1)$ and sends (TID_X, M_1, M_2, T_1) to RS.
2. When RS receives the authentication request message (TID_X, M_1, M_2, T_1), verifies timestamp T_1 based on the current timestamp T_2 and given TID_X retrieves $H(HID_X\|PWD_X)$ from the stored $(TID_X, H(s\|TID_x) \oplus H(HID_X\|PWD_X))$ in its memory and computes $PID_X = H(TID_X\|s)$ and $a_X = H(PID_X\|H(HID_X\|PWD_X)) \oplus M_1$ to verify whether $M_2 \stackrel{?}{=} H(TID_X\|PID_X\|a_X\|T_1)$. Assuming it is valid, it generates $b_X \in Z_P$, computes $TID_X^{new} = H(HID_X\|PWD_X) \oplus TID_X \oplus b_X$, $PID_X^{new} = H(TID_X^{new}\|s)$, $M_3 = H(H(HID_X\|PWD_X)\|PID_X) \oplus b_X$, $M_4 = PID_X^{new} \oplus H(TID_X^{new}\|H(ID_X\|PWD_X)\|b_X)$, $SK = H(PID_X\|a_X\|b_X)$, and $M_5 = H(SK\|PID_X^{new}\|T_2)$. Then it sends (M_3, M_4, M_5, T_2) to the user. The server also labelled $(TID_X, H(s\|TID_x) \oplus H(HID_X\|PWD_X))$ as old and stores $(TID_X^{new}, H(s\|TID_x^{new}) \oplus H(HID_X\|PWD_X))$ as the latest record for U_X.
3. U_X verifies the received T_2 to compute $b_X = H(H(HID_X\|PWD_X)\|PID_X) \oplus M_3$, $TID_X^{new} = H(HID_X\|PWD_X) \oplus TID_X \oplus b_X$, $PID_X^{new} = M_4 \oplus H(TID_X^{new}\|H(HID_X\|$

$PWD_X)\|b_X)$, and $SK = H(PID_X\|a_X\|b_X)$, and checks whether $M_5 \stackrel{?}{=} H(SK\|PID_X^{new}\|T_2)$. If they are equal, the session key is established. After that, U_X computes $B^{new} = TID^{new} \oplus H(HID_X\|PW_X\|r)$, $C^{new} = PID^{new} \oplus H(TID^{new}\|r)$, and $Auth^{new} = H(ID_X\|PW_X\|r\|PID_X^{new}\|TID_X^{new})$. Subsequently, U_X updates $(B_X, C_X, Auth_X)$ to $(B^{new}, C^{new}, Auth^{new})$ in SC_X.

U_X ($\{A_X, B_X, C_X, Auth_x\}$)		RS ($\{TID_X, H(HID_X\|PWD_X), s\}$)
Inserts ID_X and PW_X in SC_X. Then, SC_X calculates $r = A_X \oplus H(ID_X\|PW_X)$, $TID_X = B_X \oplus H(ID_X\|PW_X\|r)$, $PID_X = C_X \oplus H(TID_X\|r)$, and checks $Auth_X \stackrel{?}{=} H(ID_X\|PW_X\|r\|PID_X\|TID_X)$ to generate $a_X \in Z_P$, extract T_1, and calculate $HID_X = H(ID_x\|r)$, $M_1 = H(PID_X\|H(HID_X\|PWD_X)) \oplus a_X$ and $M_2 = H(TID_X\|PID_X\|a_X\|T_1)$	$\xrightarrow{(TID_X, M_1, M_2, T_1)}$	
		Checks T_1 to retrieve $H(HID_X\|PWD_X)$ given TID_X and the stored $(TID_X, H(s\|TID_x) \oplus H(HID_X\|PWD_X))$ to compute $PID_X = H(TID_X\|s)$ and $a_X = H(PID_X\|H(HID_X\|PWD_X)) \oplus M_1$ to verify whether $M_2 \stackrel{?}{=} H(TID_X\|PID_X\|a_X\|T_1)$ and generate $b_X \in Z_P$, compute $TID_X^{new} = H(HID_X\|PWD_X) \oplus TID_X \oplus b_X$, $PID_X^{new} = H(TID_X^{new}\|s)$, $M_3 = H(H(HID_X\|PWD_X)\|PID_X) \oplus b_X$, $M_4 = PID_X^{new} \oplus H(TID_X^{new}\|H(ID_X\|PWD_X)\|b_X)$, $SK = H(PID_X\|a_X\|b_X)$, and $M_5 = H(SK\|PID_X^{new}\|T_2)$, update memory
	$\xleftarrow{(M_3, M_4, M_5, T_2)}$	
Validates T_2, calculates $b_X = H(H(HID_X\|PWD_X)\|PID_X) \oplus M_3$, $TID_X^{new} = H(HID_X\|PWD_X) \oplus TID_X \oplus b_X$, $PID_X^{new} = M_4 \oplus H(TID_X^{new}\|H(HID_X\|PWD_X)\|b_X)$, and $SK = H(PID_X\|a_X\|b_X)$, and checks $M_5 = H(SK\|PID_X^{new}\|T_2)$ to establish SK and compute $B^{new} = TID^{new} \oplus H(HID_X\|PW_X\|r)$, $C^{new} = PID^{new} \oplus H(TID^{new}\|r)$, and $Auth^{new} = H(ID_X\|PW_X\|r\|PID_X^{new}\|TID_X^{new})$ and update $(B_X, C_X, Auth_X)$ in SC_X to $(B^{new}, C^{new}, Auth^{new})$		
Sets $SK = H(PID_X\|a_X\|b_X)$ as the session key		Sets $SK = H(PID_X\|a_X\|b_X)$ as the session key

Figure 2. The proposed mutual authentication phase between U_X and RS.

4.4. Password Change Phase

To change the current password, U_X generates a new password PW_X^{new} and a random number r^{new}, computes $PWD_X^{new} = H(PW_X^{new}\|r^{new})$, and sends a password change request message to RS including (ID_X, PWD_X^{new}). After that, RS updates $H(HID_X\|PWD_X)$ to $H(HID_X\|PWD_X^{new})$ and the password update is completed.

5. On the Security and Efficiency of the Enhanced Protocol

One method for avoiding dictionary attacks is to utilize a resource-intensive hash function to slow down the password search. As a result, specific hash algorithms for

password hashing, such as bcrypt [73], have been suggested in the literature. However, if we suppose the user is a resource-constrained device such as a smart meter, such an approach may not be viable for IoT systems. Furthermore, if we provide the opponent with the content of the smartcard, it may perform an offline dictionary attack on a strong server, implying that the password's hash would only slow down the genuine user and not the enemy. Hence, we assume that we will use a conventional hash function such as SHA2 [74] or a lightweight hash function, such as Quark [61] or PHOTON [63], which were designed for resource-constrained environments, but we will try to avoid the specific attack by involving salt in the computation and increasing the entropy space by the concatenation of $HID_X \| PWD_X$.

In the amended protocol, in the registration phase U_X computes $PWD_X = H(PW_x \| r)$ and $HID_X = H(ID_x \| r)$ and sends (HID_X, PWD_X) to RS. Since the insider has no access to r and r is selected randomly, its advantage due to the direct access to ID_X vanished in the enhanced protocol. On the other hand, the stored value on the remote server side is changed to $(TID_X, H(s \| TID_x) \oplus H(HID_X \| PWD_X))$ from $(TID_X, H(ID_X \| PWD_X))$. Since the insider has no access to the secret key of the server, it cannot compute $H(s \| TID_x)$ to retrieve $H(ID_X \| PWD_X)$. Hence, the proposed protocol provides security against insider adversaries.

Son et al.'s protocol had been improved in the way that the adversary could better compute the exchanged messages, from the security point of view, as follows:

$$\begin{aligned} M_1 &= H(PID_X \| H(HID_X \| PWD_X)) \oplus a_X \\ M_2 &= H(TID_X \| PID_X \| a_X \| T_1) \\ M_3 &= H(H(HID_X \| PWD_X) \| PID_X) \oplus b_X \\ M_4 &= PID_X^{new} \oplus H(TID_X^{new} \| H(ID_X \| PWD_X) \| b_X) \\ M_5 &= H(SK \| PID_X^{new} \| T_2) \end{aligned}$$

where

$$\begin{aligned} TID_X^{new} &= H(HID_X \| PWD_X) \oplus TID_X \oplus b_X \\ PID_X^{new} &= H(TID_X^{new} \| s) \\ SK &= H(PID_X \| a_X \| b_X) \end{aligned}$$

Compared to the SPP protocol, computation of M_3 and TID_X are modified; they were computed as $M_3 = H(PID_X \| H(ID_X \| PWD_X)) \oplus b_X$ and $TID_X^{new} = TID_X \oplus b_X$ in SPP protocol.

This is because extracting a_X or b_X in the enhanced protocol requires at least $H(HID_X \| PWD_X)$ and we already masked this value on the server side as $(TID_X, H(s \| TID_x) \oplus H(HID_X \| PWD_X))$. Hence, in the proposed protocol, even an insider adversary cannot retrieve the shared key. It should be noted, in the CK and DY adversary models, that the insider adversary has no access to the server's secret key.

The proposed protocol provides a better level of forward secrecy because the session key is computed as $SK = H(PID_X \| a_X \| b_X)$ and the adversary is not able to determine PID_X if it loses a session between the observed session and the compromising session.

For a key compromise impersonation (KCI) resistant protocol, in which a client is in communication with a server, the attacker should not be able to impersonate the server (resp. the client) toward the client (resp. the server) given all of the secret parameters of the client (resp. the server). Since the enhanced protocol is also symmetric by nature because it uses $H(\cdot)$ as the only source of diffusion and confusion, then this protocol also suffers from KCI. However, to do KCI against U_X or RS in the enhanced protocol, the adversary needs all the secret parameters of that party; however, in the SPP protocol, it is enough to access the RS memory.

SPP and the enhanced version use $H(\cdot)$ as the only nonlinear component and it is lightweight by nature, compared to asymmetric components such as the Elliptic Curve

Cryptography (ECC). Hence, these protocols belong to lightweight protocols, although the enhanced version does two extra calls to that function in each side of the protocol. Hence, the enhanced protocol is not efficient yet. Consider an Arduino UNO R3 board with an ATmega328P microcontroller as the user and an Intel Xeon CPU E5-2650V2 with a 2.60 GHz frequency as the server. For this setup, the computational time of SHA2 in the server and the user side is, respectively, 0.04 (ms) and 3 (ms), while the computational time of a point multiplication is, respectively, 2.5 (ms) and 21 (ms) [75]. A comparison of the computational time on the user and server side is given in Table 4 and illustrated in Figure 3 which confirms our claim on the efficiency of the proposed protocol because the computational overhead of the proposed protocol is only 15.5%.

Table 4. Details of computational cost comparison of the revised protocol vs. [GKK+, 2019] [76], [BKC+, 2022] [75] and [SPP, 2021] [2]; if the protocol includes more than one user in each session we just considered the cost of the first user to be fair.

Protocol	User	Server
[GKK+, 2019] [76]	$3T_{mn} + 4T_{hn} \approx 75$ ms	$6T_{\text{ms}} + 8T_{hs} \approx 15.345$ ms
[BKC+, 2022] [75]	$3T_{mn} + 6T_{hn} + 2T_{PUFn} \approx 87$ ms	$3T_{ms} + 8T_{hs} \approx 7.832$ ms
[SPP, 2021] [2]	$13T_{hn} \approx 39$ ms	$8T_{hs} \approx 0.32$ ms
Ours	$15T_{hn} \approx 45$ ms	$10T_{hs} \approx 0.4$ ms

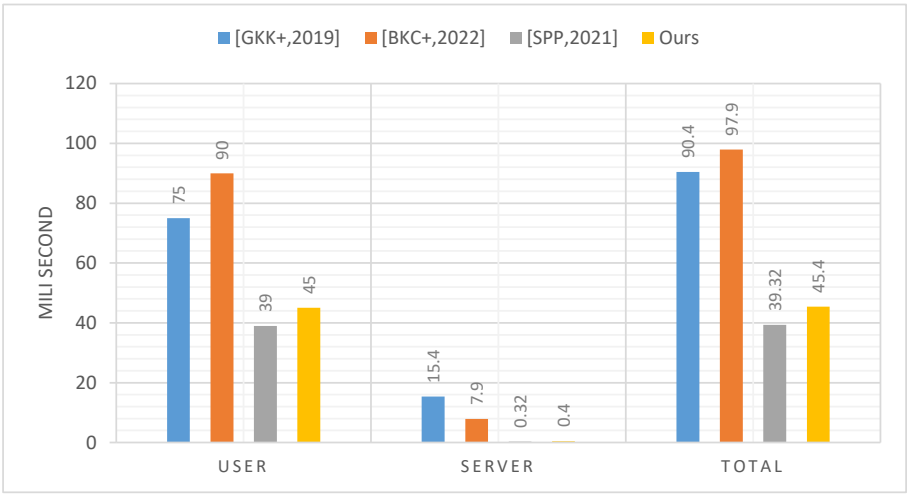

Figure 3. Computational cost comparison of the revised protocol vs. [GKK+, 2019] [76], [BKC+, 2022] [75] and [SPP, 2021] [2].

6. Conclusions and Future Works

In this paper, we presented the first third-party security analysis of a former study, which was a user authentication protocol for Internet of Things environments and applications. We highlighted its pros and cons and also we proposed an enhanced version of this protocol that is secure against various attacks.

One of the ways to grow and evolve the science of designing security protocols is to evaluate the security schemes provided by experts and researchers in this field. Hence, a suggestions for future work can be the analysis and evaluation of the security protocol proposed in this paper.

Author Contributions: M.H.M., A.H.M., Q.H.L. and L.T.: Conceptualization, Methodology, Validation, and Writing; M.S. and N.B.: Designing, Experimentation, Validation, and Writing—review and Editing, M.H.: Supervision, Review, Funding and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Abbreviations

In this paper, we use the following abbreviations:

IoT	Internet of Things
KCI	Key Compromised Impersonation Attack
ECC	Elliptic Curve Cryptography
RFID	Radio Frequency IDentification
TLS	Transport Layer Security
SSL	Secure Sockets Layer
PUF	Physically Unclonable Function
BLE	Bluetooth Low Energy

References

1. Bendavid, Y.; Bagheri, N.; Safkhani, M.; Rostampour, S. IoT Device Security: Challenging "A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function". *Sensors* **2018**, *18*, 4444. [CrossRef] [PubMed]
2. Son, S.; Park, Y.; Park, Y. A Secure, Lightweight, and Anonymous User Authentication Protocol for IoT Environments. *Sustainability* **2021**, *13*. [CrossRef]
3. Rajaram, S.; Maitra, T.; Vollala, S.; Ramasubramanian, N.; Amin, R. eUASBP: enhanced user authentication scheme based on bilinear pairing. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 2827–2840. [CrossRef]
4. Chien, H.Y. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Trans. Dependable Sec. Comput.* **2007**, *4*, 337–340. [CrossRef]
5. Tian, Y.; Chen, G.; Li, J. A New Ultralightweight RFID Authentication Protocol with Permutation. *IEEE Commun. Lett.* **2012**, *16*, 702–705. [CrossRef]
6. Zhuang, X.; Zhu, Y.; Chang, C. A New Ultralightweight RFID Protocol for Low-Cost Tags: R^2 AP. *Wirel. Pers. Commun.* **2014**, *79*, 1787–1802. [CrossRef]
7. Khokhar, U.M.; Najam-ul-Islam, M.; Shami, M.A. RCIA: A New Ultralightweight RFID Authentication Protocol Using Recursive Hash. *IJDSN* **2015**, *2015*, 642180:1–642180:8.
8. Khokhar, U.M.; Najam-ul-Islam, M.; Sarwar, S. A New Ultralightweight RFID Authentication Protocol for Passive Low Cost Tags: KMAP. *Wirel. Pers. Commun.* **2017**, *94*, 725–744.
9. Luo, H.; Wen, G.; Su, J.; Huang, Z. SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system. *Wirel. Networks* **2018**, *24*, 69–78. [CrossRef]
10. Aghili, S.F.; Mala, H.; Kaliyar, P.; Conti, M. SecLAP: Secure and lightweight RFID authentication protocol for Medical IoT. *Future Gener. Comput. Syst.* **2019**, *101*, 621–634. [CrossRef]
11. Eghdamian, A.; Samsudin, A. A secure protocol for ultralightweight radio frequency identification (RFID) tags. In *Proceedings of the International Conference on Informatics Engineering and Information Science*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 200–213.
12. David, M.; Prasad, N.R. Providing strong security and high privacy in low-cost RFID networks. In *Proceedings of the International conference on Security and Privacy in Mobile Information and Communication Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 172–179.
13. Liu, Y.; Ezerman, M.; Wang, H. Double verification protocol via secret sharing for low-cost RFID tags. *Future Gener. Comput. Syst.* **2019**, *90*, 118 – 128. [CrossRef]
14. Avoine, G.; Carpent, X.; Hernandez-Castro, J. Pitfalls in Ultralightweight Authentication Protocol Designs. *IEEE Trans. Mob. Comput.* **2016**, *15*, 2317–2332. [CrossRef]
15. Avoine, G.; Carpent, X. Yet Another Ultralightweight Authentication Protocol That Is Broken. In *Proceedings of the Radio Frequency Identification. Security and Privacy Issues—8th International Workshop, RFIDSec 2012, Nijmegen, The Netherlands, 2–3 July 2012*; Revised Selected Papers; Hoepman, J., Verbauwhede, I., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7739, pp. 20–30.
16. Avoine, G.; Carpent, X.; Martin, B. Privacy-friendly synchronized ultralightweight authentication protocols in the storm. *J. Netw. Comput. Appl.* **2012**, *35*, 826–843. [CrossRef]
17. Phan, R.C.W. Cryptanalysis of a New Ultralightweight RFID Authentication Protocol—SASI. *IEEE Trans. Dependable Secur. Comput.* **2009**, *6*, 316–320. [CrossRef]
18. D'Arco, P.; Santis, A.D. On Ultralightweight RFID Authentication Protocols. *IEEE Trans. Dependable Sec. Comput.* **2011**, *8*, 548–563. [CrossRef]

19. Safkhani, M.; Rostampour, S.; Bendavid, Y.; Sadeghi, S.; Bagheri, N. Improving RFID/IoT-based generalized ultra-lightweight mutual authentication protocols. *J. Inf. Secur. Appl.* **2022**, *67*, 103194. [CrossRef]
20. Bagheri, N.; Safkhani, M.; Peris-Lopez, P.; Tapiador, J.E. Weaknesses in a new ultralightweight RFID authentication protocol with permutation—RAPP. *Secur. Commun. Netw.* **2014**, *7*, 945–949. [CrossRef]
21. Barrero, D.F.; Castro, J.C.H.; Peris-Lopez, P.; Camacho, D.; Rodríguez-Moreno, M.D. A genetic tango attack against the David-Prasad RFID ultra-lightweight authentication protocol. *Expert Syst.* **2014**, *31*, 9–19. [CrossRef]
22. Trinh, C.; Huynh, B.; Lansky, J.; Mildeová, S.; Safkhani, M.; Bagheri, N.; Kumari, S.; Hosseinzadeh, M. A Novel Lightweight Block Cipher-Based Mutual Authentication Protocol for Constrained Environments. *IEEE Access* **2020**, *8*, 165536–165550. [CrossRef]
23. Hayajneh, T.; Ullah, S.; Mohd, B.J.; Balagani, K.S. An Enhanced WLAN Security System With FPGA Implementation for Multimedia Applications. *IEEE Syst. J.* **2017**, *11*, 2536–2545. [CrossRef]
24. Gao, L.; Lin, F.; Ma, M. Research on Ultra-Lightweight RFID Mutual Authentication Protocol Based on Stream Cipher. *IEICE Trans. Commun.* **2021**, *104-B*, 13–19. [CrossRef]
25. Vasudev, H.; Das, D. P^2-SHARP: Privacy Preserving Secure Hash based Authentication and Revelation Protocol in IoVs. *Comput. Netw.* **2021**, *191*, 107989. [CrossRef]
26. Paliwal, S. Hash-Based Conditional Privacy Preserving Authentication and Key Exchange Protocol Suitable for Industrial Internet of Things. *IEEE Access* **2019**, *7*, 136073–136093. [CrossRef]
27. Tanveer, M.; Alkhayyat, A.; Khan, A.U.; Kumar, N.; Alharbi, A.G. REAP-IIoT: Resource-Efficient Authentication Protocol for the Industrial Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 24453–24465. [CrossRef]
28. Rezazadeh Baee, M.A.; Simpson, L.; Boyen, X.; Foo, E.; Pieprzyk, J. ALI: Anonymous Lightweight Inter-Vehicle Broadcast Authentication with Encryption. *IEEE Trans. Dependable Secur. Comput.* **2022**, 1. [CrossRef]
29. Rostampour, S.; Bagheri, N.; Bendavid, Y.; Safkhani, M.; Kumari, S.; Rodrigues, J.J.P.C. An Authentication Protocol for Next Generation of Constrained IoT Systems. *IEEE Internet Things J.* **2022**, *9*, 21493–21504. [CrossRef]
30. Li, N.; Liu, D.; Nepal, S. Lightweight Mutual Authentication for IoT and Its Applications. *IEEE Trans. Sustain. Comput.* **2017**, *2*, 359–370. [CrossRef]
31. Bhattacharjya, A.; Zhong, X.; Li, X. A Lightweight and Efficient Secure Hybrid RSA (SHRSA) Messaging Scheme With Four-Layered Authentication Stack. *IEEE Access* **2019**, *7*, 30487–30506. [CrossRef]
32. He, D.; Chen, C.; Chan, S.; Bu, J. Secure and Efficient Handover Authentication Based on Bilinear Pairing Functions. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 48–53. [CrossRef]
33. Jiang, Y.; Zhang, K.; Qian, Y.; Zhou, L. Anonymous and Efficient Authentication Scheme for Privacy-Preserving Distributed Learning. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2227–2240. [CrossRef]
34. Patel, C.; Doshi, N. Secure Lightweight Key Exchange Using ECC for User-Gateway Paradigm. *IEEE Trans. Comput.* **2021**, *70*, 1789–1803. [CrossRef]
35. Ali, U.; Idris, M.Y.I.B.; Ayub, M.N.B.; Ullah, I.; Ali, I.; Nandy, T.; Yahuza, M.; Khan, N. RFID Authentication Scheme Based on Hyperelliptic Curve Signcryption. *IEEE Access* **2021**, *9*, 49942–49959. [CrossRef]
36. Yu, S.; Jho, N.; Park, Y. Lightweight Three-Factor-Based Privacy- Preserving Authentication Scheme for IoT-Enabled Smart Homes. *IEEE Access* **2021**, *9*, 126186–126197. [CrossRef]
37. Gabsi, S.; Kortli, Y.; Beroulle, V.; Kieffer, Y.; Alasiry, A.; Hamdi, B. Novel ECC-Based RFID Mutual Authentication Protocol for Emerging IoT Applications. *IEEE Access* **2021**, *9*, 130895–130913. [CrossRef]
38. Sharma, S.; Kaushik, B.; Rahmani, M.K.I.; Ahmed, M.E. Cryptographic Solution-Based Secure Elliptic Curve Cryptography Enabled Radio Frequency Identification Mutual Authentication Protocol for Internet of Vehicles. *IEEE Access* **2021**, *9*, 147114–147128. [CrossRef]
39. Abdaoui, A.; Erbad, A.; Al-Ali, A.K.; Mohamed, A.; Guizani, M. Fuzzy Elliptic Curve Cryptography for Authentication in Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 9987–9998. [CrossRef]
40. Khor, J.H.; Sidorov, M.; Ho, N.T.M.; Chia, T.H. Public Blockchain-based Lightweight Anonymous Authentication Platform Using Zk-SNARKs for Low-power IoT Devices. In Proceedings of the IEEE International Conference on Blockchain, Blockchain 2022, Espoo, Finland, 22–25 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 370–375. [CrossRef]
41. Zhang, R.; Xu, C.; Xie, M. Secure Decentralized IoT Service Platform Using Consortium Blockchain. *Sensors* **2022**, *22*, 8186. [CrossRef]
42. Chen, X.; Nguyen, K.; Sekiya, H. An experimental study on performance of private blockchain in IoT applications. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3075–3091. [CrossRef]
43. Adeli, M.; Bagheri, N.; Martín, H.; Peris-Lopez, P. Challenging the security of "A PUF-based hardware mutual authentication protocol". *J. Parallel Distrib. Comput.* **2022**, *169*, 199–210. [CrossRef]
44. Cao, J.; Li, S.; Ma, R.; Han, Y.; Zhang, Y.; Li, H. RPRIA: Reputation and PUF-Based Remote Identity Attestation Protocol for Massive IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 19174–19187. [CrossRef]
45. Aminian Modarres, A.M.; Sarbishaei, G. An Improved Lightweight Two-Factor Authentication Protocol for IoT Applications. *IEEE Trans. Ind. Inform.* **2022**, 1–11. [CrossRef]
46. Cho, Y.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y. A Secure and Anonymous User Authentication Scheme for IoT-Enabled Smart Home Environments Using PUF. *IEEE Access* **2022**, *10*, 101330–101346. [CrossRef]

47. Li, S.; Zhang, T.; Yu, B.; He, K. A Provably Secure and Practical PUF-Based End-to-End Mutual Authentication and Key Exchange Protocol for IoT. *IEEE Sensors J.* **2021**, *21*, 5487–5501. [CrossRef]
48. Lounis, K.; Zulkernine, M. T2T-MAP: A PUF-Based Thing-to-Thing Mutual Authentication Protocol for IoT. *IEEE Access* **2021**, *9*, 137384–137405. [CrossRef]
49. Xu, Y.; Lao, Y.; Liu, W.; Zhang, Z.; You, X.; Zhang, C. Mathematical Modeling Analysis of Strong Physical Unclonable Functions. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2020**, *39*, 4426–4438. [CrossRef]
50. Shi, J.; Lu, Y.; Zhang, J. Approximation Attacks on Strong PUFs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2020**, *39*, 2138–2151. [CrossRef]
51. Zhang, J.; Shen, C.; Guo, Z.; Wu, Q.; Chang, W. CT PUF: Configurable Tristate PUF Against Machine Learning Attacks for IoT Security. *IEEE Internet Things J.* **2022**, *9*, 14452–14462. [CrossRef]
52. Juang, W.S.; Chen, S.T.; Liaw, H.T. Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards. *IEEE Trans. Ind. Electron.* **2008**, *55*, 2551–2556. [CrossRef]
53. Tsai, J.L.; Lo, N.W.; Wu, T.C. Novel Anonymous Authentication Scheme Using Smart Cards. *IEEE Trans. Ind. Inform.* **2013**, *9*, 2004–2013. [CrossRef]
54. Shunmuganathan, S.; Saravanan, R.D.; Palanichamy, Y. Secure and Efficient Smart-Card-Based Remote User Authentication Scheme for Multiserver Environment. *Can. J. Electr. Comput. Eng.* **2015**, *38*, 20–30. [CrossRef]
55. Odelu, V.; Das, A.K.; Goswami, A. A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1953–1966. [CrossRef]
56. Badhib, A.; Alshehri, S.; Cherif, A. A Robust Device-to-Device Continuous Authentication Protocol for the Internet of Things. *IEEE Access* **2021**, *9*, 124768–124792. [CrossRef]
57. Zhang, R.; Xiao, Y.; Sun, S.; Ma, H. Efficient Multi-Factor Authenticated Key Exchange Scheme for Mobile Communications. *IEEE Trans. Dependable Secur. Comput.* **2019**, *16*, 625–634. [CrossRef]
58. Ryu, J.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y.; Park, Y. Secure ECC-Based Three-Factor Mutual Authentication Protocol for Telecare Medical Information System. *IEEE Access* **2022**, *10*, 11511–11526. [CrossRef]
59. Liu, Z.; Guo, C.; Wang, B. A Physically Secure, Lightweight Three-Factor and Anonymous User Authentication Protocol for IoT. *IEEE Access* **2020**, *8*, 195914–195928. [CrossRef]
60. Chattopadhyay, A.; Khairallah, M.; Leurent, G.; Najm, Z.; Peyrin, T.; Velichkov, V. On the Cost of ASIC Hardware Crackers: A SHA-1 Case Study. In Proceedings of the Topics in Cryptology—CT-RSA 2021—Cryptographers' Track at the RSA Conference 2021, Virtual Event, 17–20 May 2021; Proceedings; Paterson, K.G., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12704, pp. 657–681. [CrossRef]
61. Aumasson, J.; Henzen, L.; Meier, W.; Naya-Plasencia, M. Quark: A Lightweight Hash. *J. Cryptol.* **2013**, *26*, 313–339. [CrossRef]
62. Bogdanov, A.; Knezevic, M.; Leander, G.; Toz, D.; Varici, K.; Verbauwhede, I. SPONGENT: The Design Space of Lightweight Cryptographic Hashing. *IEEE Trans. Computers* **2013**, *62*, 2041–2053. [CrossRef]
63. Guo, J.; Peyrin, T.; Poschmann, A. The PHOTON Family of Lightweight Hash Functions. In Proceedings of the Advances in Cryptology—CRYPTO 2011—31st Annual Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2011; Proceedings; Rogaway, P., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6841, pp. 222–239. [CrossRef]
64. Bertoni, G.; Daemen, J.; Peeters, M.; Assche, G.V. On the Indifferentiability of the Sponge Construction. In Proceedings of the Advances in Cryptology—EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, 13–17 April 2008; Proceedings; Smart, N.P., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4965, pp. 181–197. [CrossRef]
65. Mironov, I. Hash Functions: From Merkle-Damgård to Shoup. In Proceedings of the Advances in Cryptology—EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, 6–10 May 2001; Proceeding; Pfitzmann, B., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2045, pp. 166–181. [CrossRef]
66. Bagheri, N.; Gauravaram, P.; Knudsen, L.R.; Zenner, E. The suffix-free-prefix-free hash function construction and its indifferentiability security analysis. *Int. J. Inf. Sec.* **2012**, *11*, 419–434. [CrossRef]
67. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Park, Y. An Efficient, Anonymous and Robust Authentication Scheme for Smart Home Environments. *Sensors* **2020**, *20*, 1215. [CrossRef]
68. Safkhani, M.; Bagheri, N.; Ali, S.; Hussain Malik, M.; Hassan Ahmed, O.; Hosseinzadeh, M.; Mosavi, A.H. Improvement and Cryptanalysis of a Physically Unclonable Functions Based Authentication Scheme for Smart Grids. *Mathematics* **2023**, *11*. [CrossRef]
69. Hosseinzadeh, M.; Ali Naqvi, R.; Safkhani, M.; Tightiz, L.; Majid Mehmood, R. Secure Authentication in the Smart Grid. *Mathematics* **2023**, *11*. [CrossRef]
70. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–207. [CrossRef]
71. Canetti, R.; Krawczyk, H. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In Proceedings of the Advances in Cryptology—EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, 6–10 May 2001; Proceeding; Pfitzmann, B., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2045, pp. 453–474. [CrossRef]

72. Lansky, J.; Rahmani, A.M.; Ali, S.; Bagheri, N.; Safkhani, M.; Hassan Ahmed, O.; Hosseinzadeh, M. BCmECC: A Lightweight Blockchain-Based Authentication and Key Agreement Protocol for Internet of Things. *Mathematics* **2021**, *9*, 3241. [CrossRef]
73. Provos, N.; Mazières, D. A Future-Adaptable Password Scheme. In Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference, Monterey, CA, USA, 6–11 June 1999; USENIX: Berkeley, CA, USA, 1999; pp. 81–91.
74. of Standards, N.I.; Technology. Federal Information Processing Standard (FIPS) 180-2, Secure Hash Standard; a Revision of FIPS 180-1, 2002. Available online: https://www.federalregister.gov/documents/2002/08/26/02-21599/announcing-approval-of-federal-information-processing-standard-fips-180-2-secure-hash-standard-a (accessed on 22 December 2022).
75. Bagheri, N.; Kumari, S.; Camara, C.; Peris-Lopez, P. Defending Industry 4.0: An Enhanced Authentication Scheme for IoT Devices. *IEEE Syst. J.* **2022**, *16*, 4501–4512. [CrossRef]
76. Garg, S.; Kaur, K.; Kaddoum, G.; Choo, K.K.R. Towards Secure and Provable Authentication for Internet of Things: Realizing Industry 4.0. *IEEE Internet Things J.* **2019**, *7*, 4598–4606. [CrossRef]