Lightweight and Anonymous Mutual Authentication Protocol for IoT Devices With Physical Unclonable Functions

Jin Meng

Qinghai University

Xufeng Zhang

Qinghai University

Tengfei Cao

Qinghai University

Yong Xie (✉ mark.y.xie@qq.com)

Qinghai University

Research Article

Keywords: IoT, Mutual authentication, Session key security, PUF, Reverse fuzzy extractor, Efficiency

Posted Date: March 22nd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-267155/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Lightweight and Anonymous Mutual Authentication Protocol for IoT Devices with Physical Unclonable Functions

Jin Meng · Xufeng Zhang · Tengfei Cao ·
Yong Xie

Received: date / Accepted: date

Abstract The past few years have seen the topic of Internet of Things (IoT) rush into the forefront of various industries, which is changing people's conventional production methods and lifestyles. Connected to the Internet, the physical devices could be as fluffy as kids' teddy bears or as balky as driverless cars. However, the security related to the IoT is faced with some serious challenges simultaneously. Confronted with these issues, we propose a mutual authentication protocol for devices in the IoT system. It is lightweight that just hash functions, XORs as well as PUFs are utilized and there is no need to store plenty of pseudo-identities. Furthermore, not only does it use the reverse fuzzy extractor to acclimatize to the noisy environment, but it also introduces the supplementary sub-protocol to enhance the resistance to the desynchronization attack. Besides, the security analysis based on the improved BAN logic by Mao and Boyd presents the higher security and reliability of the proposed protocol, and the performance analysis shows its more comprehensive functions as well as lower computation and communication overhead.

Keywords IoT · Mutual authentication · Session key security · PUF · Reverse fuzzy extractor · Efficiency

Jin Meng
Department of Computer Technology and Application, Qinghai University, Xining, China
E-mail: 793127280@qq.com

Xufeng Zhang
Department of Computer Technology and Application, Qinghai University, Xining, China
E-mail: 3426898872@qq.com

Tengfei Cao
Department of Computer Technology and Application, Qinghai University, Xining, China
E-mail: caotf@qhu.edu.cn

Yong Xie
Department of Computer Technology and Application, Qinghai University, Xining, China
E-mail: mark.y.xie@qq.com

1 Introduction

With a recent spurt of progress in the Internet of Things (IoT), a capacity of rendering devices approaches of identification and communication [12], an increasing number of IoT devices are applied in many fields such as the power grid, the transportation, the healthcare system, and the public infrastructure. For example, according to [1], the smart grid based on the IoT plays a significant part in power generation, transmission as well as distribution, and meanwhile, the condition of the power delivery to users can be timely monitored. Chauhan et al. [8] designed a system called iERS, which can supervise and notify the availability of parking spaces near the smart community and assist users in finding a proper parking spot, with an IoT-based fundamental framework. Baker et al. [4] confirmed every critical element of the healthcare system based on the end-to-end IoT, creating a universal model that can be utilized in most similar systems. Therefore, diverse technologies based on the IoT make users' comfortable and convenient life possible [25]. However, a cascade of security challenges has come in a continuous stream at the same time. Threatening the security of IoT devices and affecting the normal performance of users, manifold and various attacks are emerging endlessly. For instance, in terms of the smart power grid, if the information about the reduction of electricity consumption is disclosed, someone can steal into the void house to commit a burglary or worse [23]. And in a matter of transport, if a driverless car is maliciously controlled, a terrible accident will be caused [5]. Besides, in the realm of the smart healthcare, it is a potential danger that the data including patients' personal information and physical examination results is apt to be illegally obtained [28]. In this case, it is high time that current security mechanisms be constantly designed and improved. Therefore, it is of vital importance to propose a proper scheme.

1.1 Research Challenges

Despite the emergence of a large number of research results, some challenges cannot be ignored. Consequently, we analyze some problems of IoT and describe their feasible solutions.

First, unlike the Internet devices with unlimited power and storage assumed [2], IoT devices are small and easy to lose data, which need to operate specific tasks and have no constant power supply, so there are certain restrictions on the memory, processing capability, and power. Considering that, the paper introduces a lightweight mutual authentication, which eliminates the need to store the shared secret and a large number of pseudonyms, thereby reducing the resource burden of IoT devices. Second, IoT devices usually exist in the public locations, where it is prone to be accessible to adversaries. As a result, it is physical attacks and cloning attacks that are common threats. For instance, if having physical access to a device in the IoT system, an adversary can launch a physical attack or side-channel attack to acquire the data on it. In this regard, the Physically Unclonable Function (PUF) plays a critical role in addressing the problem as a substituted security primitive. Not only is the PUF equipped with the uniqueness attributed to the random as well as the uncontrollable influence during the process of producing integrated circuits, namely, it is less likely to be duplicated, but it also uses the

mechanism of the challenge and response whose outputs are easily evaluated but rarely predicted [26]. Additionally, in practice, it is inevitable for noisy PUF to generate error-several outputs. Hence, in order to adapt to the actual application scenarios, the protocol in the paper utilizes the fuzzy extractor to achieve error correction when using PUF. Finally, it is general that IoT devices and servers need to update their session keys to protect the forward secrecy in many schemes. When the message between the IoT device and the server is blocked, both of them cannot update their data synchronously and the normal communication cannot continue. Faced with the desynchronization attack, we proposed the supplementary sub-protocol in the paper to effectively prevent it.

1.2 Contribution

Above all, the scheme in the paper is a lightweight and anonymous mutual authentication protocol for IoT devices with Physically Unclonable Functions. Compared with other protocols, the main contributions are as follows:

- It is unnecessary for the IoT device to store the shared secret and numerous pseudonyms in the scheme. Therefore, the related resources of the IoT device are sufficient.
- The protocol realizes the mutual authentication with PUFs between the IoT devices and the server. More importantly, in addition to the noise of the non-ideal PUF, we also take the imbalance of resources between the device and the server into account, taking advantage of the reverse fuzzy extractor to reduce the cost.
- The protocol introduces the supplementary sub-protocol against the desynchronization attack to overcome the drawback in [17], which also enhances the efficiency by querying the relevant subset in the database according to the registration time, instead of traversing the whole of it.

1.3 Outline of the Paper

The organization of the rest of the paper is as follows. First, Sect. 2 shows the related works on the authentication protocols for the IoT system. Then, Sect. 3 and Sect. 4 introduce respectively related preliminaries and system model as well as security requirements. Hereafter, Sect. 5 presents the proposed scheme with its supplementary sub-protocol in detail. Next, Sect. 6 and Sect. 7 show the security and performance analysis. Finally, the conclusion and future work are described in Sect. 8, and Sect. 9 is some declarations.

2 Related Works

As the IoT has gained steam in recent decades, its security issues have also attracted wide-spread attention. As early as in 2014, a study by Hewlett Packard suggested that seventy percent or so of IoT devices suffer from the acute vulnerability which cannot be ignored [19]. Therefore, considerable authentication protocols for the Internet of Things sprang up.

Most of the incipient authentication protocols are based on the asymmetric cryptography, which cuts both ways in IoT: it boasts higher security, but bears inevitably the computational inefficiency and huge overhead. For instance, Fouda et al. [14] proposed a scheme that established the shared session key with Diffie-Hellman exchange protocol, whose needed computing resources put a certain burden on resource-constrained IoT devices. In addition, Porambage et al. [22] involved the elliptic curve cryptography belonging to the public key system to achieve the implicit certificate-based protocol. Besides, Amin et al. [3] utilized the smart card and the RSA algorithm. Therefore, not only does it have a major potential danger in tampering because it is vulnerable to the physical attack, but it also contributes to terribly large computation costs.

Then, the study on protocols with the symmetric cryptography is generally extensive. Das et al. [9] introduced a scheme with smart cards, which is a novel authentication protocol on the basis of passwords and symmetric cryptography for the hierarchical wireless sensor networks (HWSN), a branch of the Internet of Things. However, it is similar that the scheme, which is not tamper-proof, cannot avoid the attack in physical. Turkanovi and Holbl [29] designed another protocol for HWSN, which pointed out the flaws in [9] and eliminated its redundant components, taking advantage of the symmetric encryption or decryption. Nevertheless, even if the symmetric cryptography reduces the computational complexity and saves some resources with hash functions, XOR operations and concatenation operations, compared with the asymmetric one, the storage of secret keys still produces a large memory overhead in a matter of the IoT system connected with a substantial amount of devices.

The demand for more secure and efficient authentication protocols, recently, has prompted scholars to introduce the PUF, which makes up for the drawbacks of smart cards and is claimed as a hardware function with great promise, in recent researches. Aman et al. [2] showed the scheme that the response generated by PUF encrypted the data and verified the source. Chatterjee et al. [7] proposed the scheme which used the response value to construct the session key. What's more, there is no need to explicitly store the challenge-response pair. However, the protocols mentioned in [2] and [7] fail to guarantee the anonymity. In addition, the challenge-response pair is not updated and replaced every round, even when the protocol introduced by Feikken et al. [15] avoids conveying the identity in plain text. Consequently, considering the device anonymity, Gope and Sikdar [17] presented a scheme with plentiful alternative pseudonyms and challenge-response pairs. Instead of the direct identity, it completes the communication with the help of the pseudo-identity which together with the challenge-response pair is regenerated to prevent adversaries from the trail. However, it is more likely to encounter desynchronization attacks. The protocol proposed by Jiang et al. [18] resolves the above two weaknesses, but its overhead is increased resulting from the asymmetric cryptography. Additionally, the protocol in [16] performs better than that in [17] in terms of the resistance to the desynchronization attack. On the other hand, the majority of protocols like [2] merely consider the ideal PUF. Since noisy factors are inescapable in daily life, it is required to take appropriate measures against them. Especially, the fuzzy extractor is regarded as a widely-used and practical tool for error correction. In the part of noisy PUF in [16], the fuzzy extractor emerges to convert the error response values. Besides, the protocol in [15] also serves as an example to show the great role of the fuzzy extractor

in addressing noisy PUF issues. Furthermore, the fuzzy extractor in reverse is a feasible optimization method, which takes the resource difference between the device and the server in IoT system into full consideration and makes the resource utilization more reasonable. For instance, the protocols in [17], [18], [13] and [21] reverse the fuzzy extractor to arrange resources more evenly.

3 Preliminaries

3.1 Physical Unclonable Function

Described as “an expression of an inherent and unclonable instance-specific feature of a physical object” in [24], the PUF is considered a key factor in the physical uniqueness of a device. Thanks to the randomness and uncertainty during the fabrication of integrated circuits, it is less likely to produce a copy, and thereby the PUF is increasingly shining in the security domain.

Additionally, the definition in [27] that a PUF is deemed to be a special function that inputs a random challenge and generates the corresponding response relying on the complex physical character clarifies the PUF from another perspective. As is shown in the following Equation (1), C is the challenge inputted and R is the response outputted.

$$R = PUF(C) \quad (1)$$

In an ideal circumstance, there is a one-to-one correspondence between the challenge-response pair and the PUF, scilicet, if a challenge is assigned to the same PUF multiple times, the responses generated are identical, and if the same challenge is given to different PUFs, the responses obtained are distinct. However, due to the environmental and circuit noise, a PUF always outputs various responses with a few errors to a challenge value.

3.2 Reverse Fuzzy Extractor

Since the influence of noisy PUFs cannot be ignored, the fuzzy extractor is introduced to address the issue. Combined with the PUF, the fuzzy extractor with a secure sketch maps the responses with resemblance to the same result [10].

A fuzzy extractor (m, l, t, ε) comprises two algorithms, which are $Gen(\cdot)$ and $Rec(\cdot)$, according to [15] and [10]. As a probabilistic algorithm, $Gen(\cdot)$ generates a key string $k \in \{0, 1\}^l$ and a helper data hd with the input value R . In the phase, in terms of every R with min-entropy m , if Equation (2), the difference of statistics between (k, hd) and (U_l, k) is up to the threshold ε . U_l means a constellation of strings from $\{0, 1\}^l$, which are chosen in a random and uniform way. As a deterministic algorithm, if the hamming distance between R and R' is at most t , $Rec(\cdot)$ can utilize hd and R' to reproduce k , according to Equation (3).

$$(k, hd) = Gen(R) \quad (2)$$

$$k = Rec(R', hd) \quad (3)$$

Table 1 Symbols and Descriptions

Symbols	Descriptions
D_i	The identity of the IoT device
TD_i	The one-time temporary identity of IoT device
RT_i	The registration time
T_i	The current timestamp
(C_i, R_i)	The challenge-response pair
N_i	The nonce generated by the IoT device
N_S	The nonce generated by the server
sk	The session key
PUF	The physical unclonable function
$Gen(.)$	The key generation algorithm of the fuzzy extractor
$Rec(.)$	The reconstruction algorithm of the fuzzy extractor
$h(.)$	The secure one-way hash function
$ $	The concatenation operation
$\oplus$	The XOR operation

Generally, the reconstruction function $Rec(.)$ is deployed on the device with a PUF, while the key generation function $Gen(.)$ is placed in the server. However, it is a critical defect that the reconstruction algorithm is performed on the device end with the limited memory and computing resources as a consequence of numerous gates and time costs when correcting errors [30]. Therefore, the reverse fuzzy extractor, which sets $Gen(.)$ on the PUF-equipped device and $Rec(.)$ on the server, is applied to resolve the problem.

3.3 Symbols and Descriptions

In this section, the symbols and descriptions involved in the protocol are presented in the Table 1.

4 System Model and Security Requirements

4.1 System Model

As is shown in the following Fig. 1, there are two roles in the system model, which are a series of IoT devices and a server situated in the data center. Moreover, the communication between devices and the server is through the Internet in the IoT system.

- IoT devices: In the IoT system, every device possesses a PUF, in which any effort to manipulate the PUF will make it unavailable and any attempt to remove the PUF will comprise it. In addition, it is assumed that devices have finite resources.
- Server: The server is described as a secure, trusted and resource-unlimited entity, which can store the related information about IoT devices in the database to operate the mutual authentication.

Fig. 1 The System Model

4.2 Adversary Model

In matters of the adversary model, we refer to the well-known Dolev-Yao attack model in [11], with an assumption that an adversary A boasts a series of capabilities as described below:

- According to the Dolev-Yao model, the adversary A has complete control over the open channel, who can grasp total information on the insecure channel between the IoT device D_i and the server S and thereby intercept, tamper or cancel it.
- Besides the threats mentioned above, aiming at acquiring the essential data, the adversary can also launch physical attacks, cloning attacks, counterfeit attacks, desynchronization attacks and so forth.

4.3 Security Requirements

After the analysis of the attack model, we take account of the related security requirements for the proposed two-party authentication protocol:

- Mutual authentication: The genesis of the fact that it is crucial to achieve the mutual authentication between the IoT device and the server before the formal communication lurks in the issue that an attacker may disguise as a trusted device sending malicious information to others with the impersonation attack.
- Reliable session key generation: The problem that an adversary is more likely to obtain the messages transmitted through the open channel serves as an explanation of the requirement that both the device end and the server end ensure the same session key is held during communication.
- Anonymity: It is indispensable to use one-time aliases so that the adversary cannot know the true identity of the device.

Fig. 2 The Registration Phase

- Defense against the known attacks: The designed protocol is supposed to resist the known attacks, such as physical attacks, cloning attacks, impersonation attacks and especially desynchronization attacks.

5 The Proposed Scheme

In this section, we propose a lightweight and anonymous mutual authentication protocol for IoT devices with physical unclonable functions, which features the zero storage of shared secrets and a large number of pseudonyms. In total, the protocol is composed of three phases: the setup phase, the registration phase and the authentication phase.

5.1 Setup Phase

In this stage, a reliable one-way hash function $h : \{0, 1\}^* \rightarrow \{0, 1\}^l$ is selected to achieve the mutual authentication, where l is a secure parameter chosen by the server.

5.2 Registration Phase

In this stage, the IoT device sends its relevant messages to the server through the secure channel as is shown in the Fig. 2. The IoT device selects a registration time RT_i (a time slot such as three days or five days), which together with the identity D_i is utilized to calculate $FR_i = PUF(D_i || RT_i)$ in order to prepare for the supplementary sub-protocol against the desynchronization attack. Then the device randomly chooses a one-time temporary alias $TD_i \in \{0, 1\}^l$ as well as a challenge value $C_i \in \{0, 1\}^l$, and obtains the response R_i from the PUF. The device stores the TD_i needed in this round temporarily, while the registration time RT_i is also stored in a secure environment. Next, $Msg_0 : \{D_i, TD_i, (C_i, R_i), FR_i, RT_i\}$ is sent to the server through the ideal channel. After receiving Msg_0 , the server stores it in the database.

5.3 Authentication Phase

In the stage, the device and the server in the IoT system conduct the mutual authentication where few pseudo-identities and share secrets are stored by the

device end. The final generation of the same session key on the device and the server means the achievement of their mutual authentication.

(1) The IoT device transmits the TD_i of this round to the server S . On receiving the alias, the server searches for it in the database. If finding successfully, S gets the corresponding challenge-response pair (C_i, R_i) and selects a nonce N_S . Then the server computes $N_S^* = h(D_i||C_i) \oplus N_S$ and $h_S = h(N_S||C_i)$. Finally, the $Msg_1 : \{C_i, N_S^*, h_S\}$ is given to the IoT device.

(2) Upon receiving Msg_1 , the IoT device calculates $R_i' = PUF(C_i)$, $(k_i', hd_i') = Gen(R_i')$, $N_S' = h(D_i||C_i) \oplus N_S^*$ and $h_S' = h(N_S'||C_i)$, and then verifies whether h_S' is equal to h_S . If successful, the device computes $hd_i^* = h(D_i||C_i) \oplus hd_i'$, the challenge $C_i^n = h(C_i||k_i')$ in the next round, the corresponding response $R_i^n = PUF(C_i^n)$ and $R_i^* = k_i' \oplus R_i^n$. Then the device selects a nonce N_i , which is used to generate $N_i^* = k_i' \oplus N_i$, $h_i = h(C_i^n||R_i^n||k_i'||D_i||N_i)$ and the session key $sk = h(N_i||N_S'||k_i')$. Next, the device stores $TD_i^n = h(TD_i||k_i')$ for the next round and sends $Msg_2 : \{hd_i^*, R_i^*, N_i^*, h_i\}$ to the server.

(3) After acquiring Msg_2 , the server needs to compute the helper data $hd_i = h(D_i||C_i) \oplus hd_i^*$, the nonce $N_i' = k_i \oplus N_i^*$, the challenge $C_i^{n'} = h(C_i||k_i)$, and its response $R_i^{n'} = k_i \oplus R_i^*$. Then $h_i' = h(C_i^{n'}||R_i^{n'}||k_i||D_i||N_i')$ is computed to verify the identity of h_i' and h_i . If the verification is passed, the server generates the session key $sk = h(N_i'||N_S||k_i)$ and the temporary pseudo-identity $TD_i^{n'} = h(TD_i||k_i)$ for the following round. Eventually, $\{TD_i^{n'}, (C_i^{n'}, R_i^{n'})\}$ is kept in the database.

In summary, the procedure for an agreement of the session key between the physical device and the server in the IoT system is accomplished. The details are presented in the Fig. 3.

5.4 The Supplementary Sub-protocol

If a desynchronization attack is launched when Msg_2 is sent to the server, the one-time temporary alias of the IoT device on the server end cannot be updated in time, which causes the messages of the IoT device and the server to be out of synchronization. In this regard, it is of vital necessity to introduce the supplementary sub-protocol against the attack for the sake of the normal continuation of our authentication.

In the registration phase, the IoT device has calculated $FR_i = PUF(D_i||RT_i)$ and sent it to the server for storage. In the sub-protocol phase shown in the Fig. 4 with the current timestamp T_i , the device computes $FR_i' = PUF(D_i||RT_i)$, $Fk_i'' = h(D_i||RT_i||T_i) \oplus Fk_i^*$ and $Fk_i^* = h(D_i||RT_i||T_i) \oplus Fk_i'$, and then transmits $Msg_3 = \{Fk_i^*, Fhd_i^*, T_i, RT_i\}$ to the server end, which searches for the relevant data according to the registration time RT_i sent by the physical device and computes $Fk_i' = h(D_i||RT_i||T_i) \oplus Fk_i^*$, $Fhd_i' = h(D_i||RT_i||T_i) \oplus Fhd_i^*$ and $Fk_i = Rec(FR_i, Fhd_i')$ to compare Fk_i with Fk_i'' after receiving the message. If both of them are same, the resynchronization is completed and the authentication process can continue normally.

Fig. 3 The Authentication Phase

Fig. 4 The Supplementary Sub-protocol

6 Security Analysis

The BAN logic, designed by Burrows, Abadi and Needham [6], features its simplicity and practicality, resulting in the general application to the formal security analysis of identity verification protocols. However, even though it pioneered the formal analysis, its pitfalls were pointed out by Mao and Boyd [20]. Hence, we attempt to prove our proposed protocol to meet a series of requirements for the authentication between the IoT device and the server with the Mao and Boyd logic, namely the improved BAN logic, in this section.

6.1 Basic Definitions

For the sake of eliminating negative features caused by the type mismatch, Mao and Boyd logic constructed three groups of type-specific objects, including principals, messages and formulas, so we employ letters P and Q to describe principals,

K , M and N to represent messages, while X , Y and Z symbolize formulas for the clarity and convenience [20].

Some definitions are listed below:

$$P \models X \quad (4)$$

$$P \stackrel{K}{|} \sim M \quad (5)$$

$$P \stackrel{K}{\triangleleft} M \quad (6)$$

$$P \stackrel{K}{\leftrightarrow} Q \quad (7)$$

$$\#(N) \quad (8)$$

$$\text{sup}(P) \quad (9)$$

$$P \triangleleft || M \quad (10)$$

(4) denotes that the principal P believes the formula X to be true. (5) shows that the principal P says the message M is encrypted with the key K . (6) manifests that the principal P sees the message M is decrypted with the key K . (7) points out that K is considered as a good shared key between the principals P and Q . (8) suggests that the message N is fresh that it has never appeared before the current protocol conducts. (9) indicates that P is a super principal, namely, it is credible and legitimate. (10) bespeaks that the principal P cannot see the message M .

Considering the issue that the syntax is context-free while the relationship between messages is context based, Mao and Boyd [20] explained that the idealization of protocol messages converting the implicit contextual information to the explicit specification should be operated. There are some concepts of idealization regulations. On one hand, there are five related concepts. The atomic message means a data unit with no symbols such as “,” “|”, “ $\Re$ ”, “()” or “{}”, in a message, where “,” is a combinator for a message and a principal, and “|” or “ $\Re$ ” is a combinator for two messages. The challenge is an atomic message that is sent and received in two different lines by its originator, namely, a principal. In the meantime, the atomic message is not a timestamp. The replied challenge is a challenge existing in the message on the way to its originator. The response also belongs to the set of atomic messages excluding timestamps, which is sent with a replied challenge by its sender. If an atomic message is not a challenge, a response or a timestamp, it is called nonsense. On the other hand, there are several idealization rules of messages in the protocol in the following:

- All of atomic messages considered as nonsenses are supposed to be erased.
- If an atomic message plays both the roles of the challenge and the response in a line, then it is regarded as a response.
- The challenges separated by commas can be combined with the symbol “|”, so do responses.
- The challenge and its corresponding response can be combined with the symbol “ $\Re$ ”, whose form is “response $\Re$ replied challenge”.
- The message and its timestamp can be also combined with “ $\Re$ ”, whose form is “message $\Re$ timestamp”.

Moreover, according to [20], there are some inference rules which are created to achieve the intuitive formal analysis on the scheme of authentication and confidentiality in actual applications, where the symbol " $\wedge$ " is a boolean logic conjunction used to connect two formulas. For instance, if the formula X and the formula Y are true, then they can get the true formula Z , in the form of (11).

$$\frac{X \wedge Y}{Z} \quad (11)$$

- The authentication rule (12): If P believes that K is a good shared key between P and Q and P sees M with K , P can believe Q encrypts M with K .

$$\frac{P \models P \overset{K}{\leftrightarrow} Q \wedge P \overset{K}{\triangleleft} M}{P \models Q \mid \sim M} \quad (12)$$

- The confidentiality rule (13): There are three conditions: (1) P believes that K is a good key between P and Q . (2) P believes that M cannot be obtained by anyone else. (3) P can use K to encrypt the message M . If they are met, P can believe that only M can be available to P and Q .

$$\frac{P \models P \overset{K}{\leftrightarrow} Q \wedge P \models S^C \triangleleft \mid M \wedge P \overset{K}{\mid} \sim M}{P \mid \equiv (S \cup \{Q\})^C \triangleleft \mid M} \quad (13)$$

- The nonce-verification rule (14): If P believes that M is fresh and that Q encrypts M with K , then P can believe that Q thinks that K is a good key between P and Q .

$$\frac{P \models \#(M) \wedge P \models Q \overset{K}{\mid} \sim M}{P \models Q \mid \equiv P \overset{K}{\leftrightarrow} Q} \quad (14)$$

- The super-principal rule (15): If P believes that Q trusts X and Q is a legitimate server, P can believe X .

$$\frac{P \models Q \mid \equiv X \wedge P \models \text{sup}(Q)}{P \models X} \quad (15)$$

- The fresh rule (16): If P believes that M is fresh and P receives the message combined with N and M , P can believe that N is fresh.

$$\frac{P \models \#(M) \wedge P \triangleleft N \Re M}{P \models \#(N)} \quad (16)$$

- The good-key rule (17): If P believes that K is not available to any other principal than P as well as Q and K is fresh, P can believe that K is a good key between P and Q .

$$\begin{array}{c}
\text{(a)} \quad \frac{S \models S \leftrightarrow D \wedge S \models D' \triangleleft \| D_i \wedge S \models N_S \wedge S \models \#(N_S)}{S \models \{S, D'\} \triangleleft \| N_S} \quad \frac{D \models \#(N_S) \wedge \frac{D \models D \leftrightarrow S \wedge D \triangleleft N_S}{D \models S \models \sim N_S} \wedge D \models S \models S' \triangleleft \| N_S \wedge D \models S \models \sim N_S}{D \models S \models S \leftrightarrow D} \quad \frac{D \models S \models \{D, S'\} \triangleleft \| N_S}{D \models \{D, S'\} \triangleleft \| N_S} \wedge D \models \sup(S) \wedge D \models \#(N_S)}{D \models D \leftrightarrow S} \\
\text{(b)} \quad \frac{S \models \#(N_i) \wedge \frac{S \models S \leftrightarrow D \wedge S \triangleleft N_i}{S \models D \models \sim N_i} \wedge S \models D \models D' \triangleleft N_i \wedge S \models D \models \sim N_i}{S \models D \models D \leftrightarrow S} \quad \frac{D \models \#(N_i) \wedge \frac{S \models S \leftrightarrow D \wedge S \triangleleft N_i}{S \models D \models \{S, D'\} \triangleleft N_i} \wedge S \models \sup(D) \wedge S \models \#(N_i)}{S \models S \leftrightarrow D} \\
\text{(c)} \quad \frac{D \models D \leftrightarrow S \wedge D \models S' \triangleleft \| k'_i \wedge D \models N_i \wedge D \models \#(N_i)}{D \models \{S, D'\} \triangleleft \| N_i} \quad \frac{D \models \{S, D'\} \triangleleft \| N_i}{D \models D \leftrightarrow S} \\
\text{(d)} \quad \frac{S \models \#(N_i) \wedge \frac{S \models S \leftrightarrow D \wedge S \triangleleft N_i}{S \models D \models \sim N_i} \wedge S \models D \models D' \triangleleft R_i^* \wedge S \models S \leftrightarrow D \wedge S \triangleleft R_i^*}{S \models D \models D \leftrightarrow S} \quad \frac{S \models D \models \{S, D'\} \triangleleft R_i^*}{S \models \{S, D'\} \triangleleft R_i^*} \wedge S \models \sup(D) \wedge S \models \#(N_i) \wedge \frac{S \triangleleft N_i \wedge R_i^*}{S \triangleleft N_i \wedge R_i^*} \\
\text{(e)} \quad \frac{D \models D \leftrightarrow S \wedge D \models S' \triangleleft \| k'_i \wedge D \models R_i^* \wedge D \models \#(R_i^*)}{S \models \{S, D'\} \triangleleft \| R_i^*} \quad \frac{S \models \{S, D'\} \triangleleft \| R_i^*}{D \models D \leftrightarrow S} \\
\text{(f)} \quad \frac{S \models \#(R_i^*) \wedge \frac{S \models S \leftrightarrow D}{S \models S \leftrightarrow D}}{S \models S \leftrightarrow D}
\end{array}$$

Fig. 5 (a): The Proof for “ S believes that N_S is a good shared key between S and D ” (b): The Proof for “ D believes that N_S is a good shared key between S and D ” (c): “ D believes that N_i is a good shared key between D and S ” (d): “ S believes that N_i is a good shared key between S and D ” (e): “ D believes that R_i^n is a good shared key between D and S ” (f): “ S believes that R_i^n is a good shared key between S and D ”

$$\frac{P \models \{P, Q\}^C \triangleleft \| K \wedge P \models \#(K)}{P \models P \xleftrightarrow{K} Q} \quad (17)$$

- The intuitive rule (18): It is a rule ignored usually that if P decrypts M with K , then P can see M .

$$\frac{P \xleftrightarrow{K} M}{P \triangleleft M} \quad (18)$$

6.2 Formal Security Analysis on Proposed Protocol

According to the above inference rules, we propose some initial beliefs and assumptions for our protocol between the device and the server in the IoT system, which then are used to construct the security proofs.

Regarding the IoT device as D and the server as S , first we try to prove the proposition (vi), which is “ S believes that N_S is a good shared key between S and D ”. As is shown in the following, (i) shows that S believes D_i is a good key between S and D , because it is the real identity of the IoT device stored in the server. (ii) shows that S believes D_i cannot be known by any other one except D . (iii) shows that S can encrypt N_S with D_i . (v) shows that S believes N_S is fresh because S generates the nonce N_S . In the light of the confidentiality rule, we use (i), (ii) and (iii) to obtain the statement “ S believes that no one else knows N_S except S and D ”, which is (iv). Then (iv) and (v) are applied in the good-key

rule to get the final statement (vi). The detailed proof process is shown in the Fig. 5(a).

$$S \equiv S \stackrel{D_i}{\leftrightarrow} D \quad (\text{i})$$

$$S \equiv D^c \triangleleft || D_i \quad (\text{ii})$$

$$S \mid \stackrel{D_i}{\sim} N_S \quad (\text{iii})$$

$$S \equiv \{S, D\}^c \triangleleft || N_S \quad (\text{iv})$$

$$S \equiv \#(N_S) \quad (\text{v})$$

$$S \equiv S \stackrel{N_S}{\leftrightarrow} D \quad (\text{vi})$$

Then we attempt to prove the proposition (xvi), which is “ D believes that N_S is a good shared key between S and D ”. In the following, (vii) means D believes that D_i is a good shared key between D and S . (viii) means that D can decrypt N_S with D_i . (ix) means D believes that S encrypts N_S with D_i . (x) means D believes that N_S is fresh. (xi) means D believes that S holds the belief that D_i is a good shared key between S and D . (xii) means that D believes that S takes the belief that N_S cannot be known by others except S . (xiii) means D considers the fact that S believes only D and itself can obtain the nonce N_S . (xiv) means that D believes that S is a credible principal. Therefore, we can use these beliefs and assumptions to deduce the final conclusion. With the authentication rule, (vii) can be combined with (viii) to draw (ix). Additionally, (xi) can be derived from the combination between (ix) and (x) with the nonce-verification rule. With the three conditions (ix), (xi) and (xii) substituted into a variant of the confidentiality rule, we can reason out (xiii), which thereby together with (xiv) can be used in the super-principal rule to obtain (xv). Then (xv) and (x) are utilized to generate the final conclusion (xvi) with the good-key rule. The proof process is vividly shown in the Fig. 5(b).

$$D \equiv D \stackrel{D_i}{\leftrightarrow} S \quad (\text{vii})$$

$$D \stackrel{D_i}{\triangleleft} N_S \quad (\text{viii})$$

$$D \equiv S \mid \stackrel{D_i}{\sim} N_S \quad (\text{ix})$$

$$D \equiv \#(N_S) \quad (\text{x})$$

$$D \models S \models S \overset{D_i}{\leftrightarrow} D \quad (\text{xi})$$

$$D \models S \models S^c \triangleleft ||N_S \quad (\text{xii})$$

$$D \models S \models \{D, S\}^c \triangleleft ||N_S \quad (\text{xiii})$$

$$D \models \text{sup}(S) \quad (\text{xiv})$$

$$D \models \{D, S\}^c \triangleleft ||N_S \quad (\text{xv})$$

$$D \models D \overset{N_S}{\leftrightarrow} S \quad (\text{xvi})$$

Similarly, the proofs for “ D believes that N_i is a good shared key between D and S ” and “ S believes that N_i is a good shared key between S and D ” are respectively shown in the Fig. 5(c) and Fig. 5(d). In the matters of the former, according to the confidentiality rule, “ D believes that k'_i is a good shared key between itself and S ”, “ D believes that no one can obtain k'_i except S ” and “ D encrypts N_i with k'_i ”, these three conditions are involved to deduce a statement, which is “ S holds the view that N_i can merely be known by S and D ”. In the light of the conclusion, we can introduce it with the belief “ D believes that N_i is fresh” into the good-key rule in order to obtain the final statement. Moreover, the latter is generated by “ S believes that N_i is fresh” which is the result of “ S is convinced that D believes only S and D can know N_i ” and “ S believes that D is a legitimate principal” with the super-principal rule, and “ S believes that only S and D can obtain N_i ” with the good-share key rule. Obtained with the developed confidentiality rule, the statement “ S is convinced that D believes only S and D can know N_i ” is the result of “ S believes that D holds the belief that k'_i is a good shared key between D and S ”, “ S is convinced that D believes that it is less likely for N_i to be attached by others except D ” and “ S believes that N_i is encrypted by D with k'_i ”. In terms of the conclusion “ S believes that D trusts k'_i as a good shared key between D and S ”, it can be deduced with the nonce-verification rule by “ S believes that N_i is a fresh nonce” and “ S believes that D can encrypt N_i with k'_i ”, which can be obtained by the combination of “ S believes that k'_i is a good shared key between S and D ” and “ N_i can be decrypted by S with k'_i ” with the authentication rule.

Respectively shown in the Fig. 5(e) and Fig. 5(f), the similar manner of the proofs for “ D believes that R_i^n is a good shared key between D and S ” and “ S believes that R_i^n is a good shared key between S and D ” is described in the specific process. In the Fig. 5(e), with the confidentiality rule, we utilize three conditions “ D believes that k'_i is a good shared key between D and S ”, “ D believes that no one can obtain k'_i except S ” and “ R_i^n can be encrypted by D with k'_i ” to conclude the statement “ S believes it is impossible that a third person can obtain R_i^n except S and D ”, which is combined with the fact “ D believes that R_i^n is fresh” to deduce the final belief “ D believes that R_i^n is a good shared key between D and S ” with the good-key rule. In the Fig. 5(f), what calls for special attention is that, with the fresh rule, the statement “ S trusts R_i^n as fresh” is generated by “ S believes

Table 2 The Analysis of Security Functions

Protocols	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
[2]	Y	Y	N	Y	Y	N	Y	N	Y
[17]	Y	N	Y	Y	Y	Y	N	Y	Y
[18]	Y	Y	Y	Y	Y	Y	Y	Y	N
[16]	Y	Y	Y	Y	Y	N	N	Y	Y
Our Protocol	Y	Y	Y	Y	Y	Y	Y	Y	Y

that N_i is a fresh nonce” and “ S can obtain N_i and R_i^n ”, which is concluded from “ S can decrypt N_i and R_i^n with k_i' ”, according to the intuitive rule.

In conclusion, generally, D_i is rarely known by others excluding D and S , so an adversary cannot obtain the secrets involved in the formal security proofs, which are N_S , N_i , R_i^n and k_i . And some attacks like impersonation attacks are even less likely to be operated. Additionally, thanks to the feature of the PUF, they cannot get valid challenge-response pairs from it even when adversaries control an IoT device. Consequently, our protocol is regarded enough reliable against some common security attacks.

7 Performance Analysis

In this section, we analyze the performance of the proposed scheme in three respects: security functions, computation costs and communication costs, whose comparison results with the protocols in [2], [17], [18] and [16] are introduced in the following.

7.1 Security Function Analysis

Aiming to present the strengths of the scheme proposed in the paper, we first compare it with four other PUF-based mutual authentication protocols on their security functions in Table 2, where F_1 , F_2 , F_3 , F_4 , F_5 , F_6 , F_7 , F_8 and F_9 respectively represent the mutual authentication, the resilience to desynchronization, the impersonation attack, the session key security, the physical security, the reverse fuzzy extractor, the zero storage of shared secrets, the anonymity and the lightweight feature. What’s more, Y means achieved while N means not achieved.

In terms of the resilience to desynchronization and the zero storage of the shared secrets, even when the scheme in [17] keeps a mass of alternate pseudonyms and keys, the desynchronization attack is still a problem. Although the protocol in [16] can prevent attacks to a certain degree, it still needs to store a large number of pseudo-identities and challenge-response pairs, which requires a lot of storage space. According to the solution proposed in the paper, it is unnecessary for the IoT device and server to store those. When they are subjected to the desynchronization attack, they merely need to search for a subset in the database in the light of the registration time and finish the resynchronization. Moreover, the issue that it is more likely for noise to lead to some errors in the output is neglected by the scheme in [2]. While the scheme in [16] involves the fuzzy extractor, it does not reverse it to consider the resource imbalance between the device and server. Our scheme takes these factors into full consideration, and with the reverse fuzzy extractor,

Table 3 The Analysis of Computation Costs(ms)

Protocols	[2]	[17]	[16]	Our Protocol
The IoT Device End	$2T_P + 5T_H + 2T_S$	$2T_P + 7T_H + 1T_G$	$2T_P + 6T_H + 1T_R$	$2T_P + 6T_H + 1T_G$
The Server End	$5T_H + 2T_S$	$7T_H + 1T_R$	$6T_H + 1T_G$	$6T_H + 1T_R$

not only does it solve the noise problem, but it also takes reasonable advantage of resources. What's more, the protocol in [18] addresses the above issues, but it contains the public key cryptography, resulting in a surge of costs. Instead of it, our protocol is characterized by a series of lightweight functions, such as PUFs, hash functions and XORs. Additionally, since the protocol in [2] directly uses the original identity of the device rather than its pseudo-identity, the anonymity is not achieved. Our resolve in the paper that uses the one-time temporary alias updated in each round of communication protects the privacy of the physical device in the IoT system.

7.2 Computation Costs Analysis

Considering the difference of the computation costs generated by various PUF-based protocols, we show the details in the Table 3, where T_P , T_H , T_G , T_R and T_S respectively symbolize the time costs of PUFs, hash functions (including the MAC), the key generation function of the fuzzy extractor, the reconstruction function of the fuzzy extractor and symmetric encryption or decryption. Generally, we think that various time costs roughly meet the following magnitude relationships: $T_S > T_P \approx T_H$ and $T_R > T_G$.

Since the protocol in the [18] is based on the three-party authentication, we just conduct the comparative analysis of our protocol and those in [2], [17] and [16]. In our protocol, $h(D_i||C_i)$ in the IoT device is used twice, as a result, we only consider the time cost of calculating it once. According to the table, we can conclude that our protocol still has a slight advantage, compared with the protocol in [2]. Although it uses less hash functions, the time costs caused by the symmetric encryption and decryption with the response value bring our protocol the latest edge through a small victory. In addition, our protocol is one hash function less than that of [17], which is also a narrow margin. Furthermore, the computation costs of our PUFs and hash functions are similar to those of [16], but the device end equipped with the key generation function of the reverse fuzzy extractor costs fewer resources and less time.

7.3 Communication Costs Analysis

By analyzing the communication costs, we can still demonstrate some advantages of our proposed protocol. Since we regard l as a security parameter, utilizing the hash function to convert a bit string of arbitrary length into that of l -bit length, we define the length of nonces, identities, challenge values and response values as l bits, and the l -bit data is changed to $8l$ -bit one after the symmetric encryption.

We just contrast the computation costs of relevant protocols in [2], [17] and [16], as shown in the Table 4, attributing to the face that the protocol in [18] involves three parties and causes numerous costs with asymmetric encryption and

Table 4 The Analysis of Communication Costs(bits)

Protocols	[2]		[17]		[16]		Our Protocol	
	Size	Times	Size	Times	Size	Times	Size	Times
The IoT Device End	35 <i>l</i>	2	5 <i>l</i>	2	4 <i>l</i>	2	5 <i>l</i>	2
The Server End	26 <i>l</i>	1	3 <i>l</i>	1	5 <i>l</i>	2	3 <i>l</i>	1
Total	61 <i>l</i>	3	8 <i>l</i>	3	9 <i>l</i>	4	8 <i>l</i>	3

Table 5 The Summary Comparisons of Protocols

Protocols	Security Functions	Computation Costs	Communication Costs
[2]	Part	Highest	Highest
[17]	Part	Higher	Lowest
[18]	Part	-	-
[16]	Part	Higher	Higher
Our Protocol	All	Lowest	Lowest

decryption. In the table, *Size* means the size of messages and *Times* means the times of sending messages. And it is apparent that the computation costs of the protocol in [2] are much more than any other protocol resulting from symmetric encryption and decryption. Additionally, the communication overhead of our protocol is as little as that in [17]. Besides, even though the communication costs of the IoT device in the protocol proposed by [16] are less than ours, regardless of the total size of messages or the total times of communications, the protocol in [16] is slightly more than ours. Therefore, our protocol in this paper can be treated low-overhead.

Above all, our protocol fully demonstrates its advantages in terms of security functions, computing costs and communication overhead. Table 5 shows the summary comparisons among the protocols in [2], [17], [18], [16] and this paper. Since the computation and communication costs of the protocol in [18] are not involved in the above comparisons, we ignore them in the following table, in which we can know that, not only does our protocol meet all the security functions mentioned, but its computation and communication overhead is also the lowest.

8 Conclusion and Future Work

In this paper, we propose a lightweight and anonymous mutual authentication protocol for IoT devices with physical unclonable functions, which, though not a universal panacea for the security of the current IoT system, would be a meaningful step in the right direction. Instead of symmetric or asymmetric cryptography, the hash functions, PUFs, exclusive OR operations and concatenation operations are at play in the lightweight. For one thing, we can resolve the issue that a large number of pseudonyms exist in the IoT device because of the anonymity. We can also resist the desynchronization attacks effectively by designing the supplementary sub-protocol. For another, we can implement the error correction with the fuzzy extractor considering the PUF in a non-ideal environment. We can also ensure that the resources of devices and the server in the IoT system are reasonably used by reversing the fuzzy extractor. Moreover, both the formal security proof and the performance analysis sufficiently illustrate the security, efficiency and cost rationality of our proposed protocol.

However, in extreme cases, if the real identity D_i of the IoT device is obtained by an adversary, it is more likely for the nonce N_S generated by the server to be

put on the street, which is an inadequacy needed to be improved in our future work.

9 Declarations

9.1 Funding

The work was supported in part by the National Natural Science Foundation of China (61862052), and the Science and Technology Foundation of Qinghai Province (2019-ZJ-7065).

9.2 Conflicts of interest/Competing interests

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

9.3 Availability of data and material

The data used to support the findings of this study are available from the corresponding author upon request.

9.4 Code availability

Not applicable.

9.5 Authors' contributions

Methodology, Jin Meng, Xufeng Zhang, Tengfei Cao and Yong Xie; Supervision, Yong Xie; Writing – original draft, Jin Meng; Writing – review and editing, Xufeng Zhang and Tengfei Cao.

9.6 Ethics approval

Not applicable.

9.7 Consent to participate

Not applicable.

9.8 Consent for publication

Not applicable.

References

1. Al-Turjman, F., AbuJubbeh, M.: Iot-enabled smart grid via sm: An overview. *Future Generation Computer Systems* **96**, 579–590 (2019)
2. Aman, M.N., Chua, K.C., Sikdar, B.: Mutual authentication in iot systems using physical unclonable functions. *IEEE Internet of Things Journal* **4**(5), 1327–1340 (2017)
3. Amin, R., Islam, S., Khan, M.K., Karati, A., Giri, D., Kumari, S.: A two-factor rsa-based robust authentication system for multiserver environments. *Security and Communication Networks* **2017** (2017)
4. Baker, S.B., Xiang, W., Atkinson, I.: Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access* **5**, 26521–26544 (2017)
5. Bates, N.: Driverless vehicle security: Considering potential attacks and countermeasures for military applications (2020)
6. Burrows, M., Abadi, M., Needham, R.M.: A logic of authentication. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* **426**(1871), 233–271 (1989)
7. Chatterjee, U., Govindan, V., Sadhukhan, R., Mukhopadhyay, D., Chakraborty, R.S., Mahata, D., Prabhu, M.M.: Building puf based authentication and key exchange protocol for iot without explicit crps in verifier database. *IEEE Transactions on Dependable and Secure Computing* pp. 1–1 (2018)
8. Chauhan, V., Patel, M., Tanwar, S., Tyagi, S., Kumar, N.: Iot enabled real-time urban transport management system. *Computers & Electrical Engineering* **86**, 106746 (2020)
9. Das, A.K., Sharma, P., Chatterjee, S., Sing, J.K.: A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *Journal of Network and Computer Applications* **35**(5), 1646–1656 (2012)
10. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: *International conference on the theory and applications of cryptographic techniques*, pp. 523–540. Springer (2004)
11. Dolev, D., Yao, A.: On the security of public key protocols. *IEEE Transactions on information theory* **29**(2), 198–208 (1983)
12. El-hajj, M., Fadlallah, A., Chamoun, M., Serhrouchni, A.: A survey of internet of things (iot) authentication schemes. *Sensors* **19**(5), 1141 (2019)
13. Feng, W., Qin, Y., Zhao, S., Feng, D.: Aaot: Lightweight attestation and authentication of low-resource things in iot and cps. *Computer Networks* **134**(APR.7), 167–182 (2018)
14. Fouda, M.M., Fadlullah, Z.M., Kato, N., Lu, R., Shen, X.S.: A lightweight message authentication scheme for smart grid communications. *IEEE Transactions on Smart grid* **2**(4), 675–685 (2011)
15. Frikken, K.B., Blanton, M., Atallah, M.J.: Robust authentication using physically unclonable functions. In: *International Conference on Information Security*, pp. 262–277. Springer (2009)
16. Gope, P., Lee, J., Quek, T.Q.: Lightweight and practical anonymous authentication protocol for rfid systems using physically unclonable functions. *IEEE Transactions on Information Forensics and Security* **13**(11), 2831–2843 (2018)
17. Gope, P., Sikdar, B.: Lightweight and privacy-preserving two-factor authentication scheme for iot devices. *IEEE Internet of Things Journal* **6**(1), 580–589 (2018)
18. Jiang, Q., Zhang, X., Zhang, N., Tian, Y., Ma, X., Ma, J.: Two-factor authentication protocol using physical unclonable function for iov. In: *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 195–200. IEEE (2019)
19. Lee, I., Lee, K.: The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons* **58**(4), 431–440 (2015)
20. Mao, W., Boyd, C.: Towards formal analysis of security protocols. In: *[1993] Proceedings Computer Security Foundations Workshop VI*, pp. 147–158. IEEE (1993)
21. Mitev, M., Shekiba-Herfeh, M., Chorti, A., Reed, M.: Multi-factor physical layer security authentication in short blocklength communication. *arXiv preprint arXiv:2010.14457* (2020)
22. Porambage, P., Schmitt, C., Kumar, P., Gurtov, A., Ylianttila, M.: Two-phase authentication protocol for wireless sensor networks in distributed iot applications. In: *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2728–2733. Ieee (2014)
23. River, W.: Security in the internet of things. Wind River Systems, Tech. Rep (2015)

24. Roel, M.: Physically unclonable functions: Constructions, properties and applications. Katholieke Universiteit Leuven, Belgium (2012)
25. Shah, S.H., Yaqoob, I.: A survey: Internet of things (iot) technologies, applications and challenges. In: 2016 IEEE Smart Energy Grid Engineering (SEGE), pp. 381–385. IEEE (2016)
26. Shamsoshoara, A., Korenda, A., Afghah, F., Zeadally, S.: A survey on physical unclonable function (puf)-based security solutions for internet of things. *Computer Networks* p. 107593 (2020)
27. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE Design Automation Conference, pp. 9–14. IEEE (2007)
28. Tariq, N., Qamar, A., Asim, M., Khan, F.A.: Blockchain and smart healthcare security: A survey. *Procedia Computer Science* **175**, 615–620 (2020)
29. Turkanovic, M., Holbl, M.: An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *Elektronika ir Elektrotechnika* **19**(6), 109–116 (2013)
30. Van Herrewege, A., Katzenbeisser, S., Maes, R., Peeters, R., Sadeghi, A.R., Verbauwhede, I., Wachsmann, C.: Reverse fuzzy extractors: Enabling lightweight mutual authentication for puf-enabled rfids. In: International Conference on Financial Cryptography and Data Security, pp. 374–389. Springer (2012)

Figures

Figure 1

The System Model

Device D_i	Server S
Select RT_i	
Compute $FR_i = PUF(D_i RT_i)$	
Select random TD_i, C_i	
Compute $R_i = PUF(C_i)$	
Store TD_i	
$Msg_0 : \{D_i, TD_i, (C_i, R_i), FR_i, RT_i\}$	
	Store $\{D_i, TD_i, (C_i, R_i), FR_i, RT_i\}$
Secure Channel	

Figure 2

The Registration Phase

Figure 3

The Authentication Phase

Figure 4

The Supplementary Sub-protocol

$$\begin{array}{c}
\frac{S \models S \leftrightarrow D \wedge S \models D^c \triangleleft \| D_i \wedge S \mid \sim N_k \wedge S \models \#(N_s)}{S \models \{S, D\}^c \triangleleft \| N_s} \quad \frac{D \models \#(N_s) \wedge \frac{D \models D \leftrightarrow S \wedge D \triangleleft N_s}{D \models S \mid \sim N_s} \wedge D \models S \models S^c \triangleleft \| N_s \wedge D \models S \mid \sim N_s}{D \models S \models S \leftrightarrow D} \wedge D \models \sup(S) \\
\hline
\frac{S \models \{S, D\}^c \triangleleft \| N_s}{S \models S \leftrightarrow D} \quad \frac{D \models \{D, S\}^c \triangleleft \| N_s}{D \models D \leftrightarrow S} \\
\text{(a)} \qquad \text{(b)}
\end{array}$$

$$\begin{array}{c}
\frac{S \models \#(N_i) \wedge \frac{S \models S \leftrightarrow D \wedge S \triangleleft N_i}{S \models D \mid \sim N_i} \wedge S \models D \models D^c \triangleleft \| N_i \wedge S \models D \mid \sim N_i}{S \models D \models D \leftrightarrow S} \wedge S \models \sup(D) \\
\hline
\frac{D \models D \leftrightarrow S \wedge D \models S^c \triangleleft \| k'_i \wedge D \mid \sim N_i \wedge D \models \#(N_i)}{D \models D \leftrightarrow S} \quad \frac{S \models \{S, D\}^c \triangleleft \| N_i}{S \models S \leftrightarrow D} \\
\text{(c)} \qquad \text{(d)}
\end{array}$$

$$\begin{array}{c}
\frac{S \models \#(N_i) \wedge \frac{S \models S \leftrightarrow D \wedge S \triangleleft N_i}{S \models D \mid \sim N_i} \wedge S \models D \models D^c \triangleleft \| R_i^* \wedge \frac{S \models S \leftrightarrow D \wedge S \triangleleft R_i^*}{S \models D \mid \sim R_i^*} \wedge S \models \sup(D)}{S \models \{S, D\}^c \triangleleft \| R_i^*} \wedge S \models \#(N_i) \wedge \frac{S \triangleleft N_i \mathfrak{R} R_i^*}{S \triangleleft N_i \mathfrak{R} R_i^*} \\
\hline
\frac{D \models D \leftrightarrow S \wedge D \models S^c \triangleleft \| k'_i \wedge D \mid \sim R_i^* \wedge D \models \#(R_i^*)}{D \models D \leftrightarrow S} \quad \frac{S \models \{S, D\}^c \triangleleft \| R_i^*}{S \models S \leftrightarrow D} \\
\text{(e)} \qquad \text{(f)}
\end{array}$$

Figure 5

Please see the manuscript file for the full caption.