Hindawi
Security and Communication Networks
Volume 2022, Article ID 1203691, 11 pages
https://doi.org/10.1155/2022/1203691

Research Article

Lightweight and Anonymous Mutual Authentication Protocol for Edge IoT Nodes with Physical Unclonable Function

Hongyuan Wang,[1] Jin Meng,[2] Xilong Du,[1] Tengfei Cao,[2] and Yong Xie[2]

[1]*Qinghai Province Yindajihuang Project Construction and Operation Bureau, Xining, China*
[2]*Department of Computer Technology and Application, Qinghai University, Xining, China*

Correspondence should be addressed to Yong Xie; mark.y.xie@qq.com

Received 10 September 2021; Accepted 23 October 2021; Published 4 January 2022

Academic Editor: Jie Cui

Internet of Things (IoT) has been widely used in many fields, bringing great convenience to people's traditional work and life. IoT generates tremendous amounts of data at the edge of network. However, the security of data transmission is facing severe challenges. In particular, edge IoT nodes cannot run complex encryption operations due to their limited computing and storage resources. Therefore, edge IoT nodes are more susceptible to various security attacks. To this end, a lightweight mutual authentication and key agreement protocol is proposed to achieve the security of IoT nodes' communication. The protocol uses the reverse fuzzy extractor to acclimatize to the noisy environment and introduces the supplementary subprotocol to enhance resistance to the desynchronization attack. It uses only lightweight cryptographic operations, such as hash function, XORs, and PUF. It only stores one pseudo-identity. The protocol is proven to be secure by rigid security analysis based on improved BAN logic. Performance analysis shows the proposed protocol has more comprehensive functions and incurs lower computation and communication cost when compared with similar protocols.

1. Introduction

With the rapid development of new network technologies such as cloud computing and artificial intelligence, Internet of Things (IoT) has been more and more widely used. It has continuously brought great convenience to people's lives and work [1]. IoT devices play an important role in the power generation, transmission, and distribution of smart grids and can monitor power transmission conditions in a more timely manner [2]. A system called iERS can monitor and notify the availability of parking spaces near the smart community through the IoT infrastructure and help users find suitable parking spaces [3]. Baker et al. [4] created a general model that can be used in most similar healthcare systems using end-to-end IoT. Therefore, diverse technologies based on the IoT make users' comfortable and convenient life possible.

According to the predictions of relevant agencies, IoT devices are expected to grow exponentially in the next few years, followed by the explosive growth of IoT data [5]. In some low-latency IoT applications, the design idea of combining the computing functions of the edge cloud to complete the reception and management of massive data has become a way to improve the efficiency of IoT. Edge cloud helps edge IoT nodes process data nearby, reducing the heavy computing tasks of cloud data centers.

However, due to the openness of channels and data sensitivity, data security and user privacy issues have attracted more and more attention. Data security issues are also one of the biggest obstacles restricting the widespread deployment and application of Internet of Things [6]. Due to IoT characteristics, the specific challenges faced by data security are as follows: (1) IoT device resources are generally limited. Internet of Things consists of many heterogeneous and resource-constrained devices, which often have a single function and limited computing and storage resources [7]; (2) massive data: the number of IoT devices and users is huge, and massive amounts of data are generated in real

time, which brings great workload to security authentication; (3) interactive dynamics: in the environment of Internet of Things, nodes and users are often in constant movement, which makes real-time requirements for secure access and authentication; and (4) strong data privacy: the advent of the big data era puts forward higher requirements for the protection of personal privacy information, and both visitors and IoT nodes must be protected [8].

In order to solve the above-mentioned IoT data security issues, many researchers have proposed various security authentication and key agreement protocols to solve the IoT data security issues [9]. However, as we all know, Internet of Things has many remote nodes. In this scenario, an attacker can extract stored authentication information and keys from the IoT device and then can perform security attacks according to their own needs. At present, most studies have not considered this aspect of security issues. Therefore, the communication protocol designed for the IoT system should ensure that the entire system remains secure, even if the equipment or sensors are damaged. Fortunately, physical unclonable functions (PUF) provide a viable option to achieve this goal. Recently, some PUF-based authentication protocols have been proposed to protect sensor security and data security.

To solve the above issues, we propose a lightweight and anonymous mutual authentication protocol for edge IoT nodes with physical unclonable function. The proposed protocol only needs some lightweight cryptographic operations and stores one pseudo-identity. It is very suitable for data security protection scenarios of IoT nodes in a wide range of deployment scenarios. To sum it up, the main contributions of the proposed protocol are as follows:

(i) The proposed protocol realizes secure, lightweight mutual authentication for edge IoT nodes. More importantly, in addition to the noise of the nonideal 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.

(ii) The proposed protocol only store one pseudo-identity to prevent physical security attack such as side-channel security attacks and memory data theft while ensuring anonymity.

(iii) We introduced a supplementary subprotocol for desynchronization attacks to overcome the shortcomings in [10]. It also improves efficiency by querying the relevant subset in the database based on the registration time instead of traversing the entire subset.

(iv) We present rigid security proof based on improved BAN logic [11] to demonstrate the proposed protocol is against all of secure attacks.

The paper's organization is as follows: Section 2 shows the related works on the authentication protocols for the IoT system. Section 3 and Section 4 introduce, respectively, related preliminaries and system model and security requirements. Section 5 presents the proposed scheme with its supplementary subprotocol in detail. Section 6 and Section 7 show the security and performance analysis. Finally, the conclusion and future work are described in Section 8.

2. Related Works

As IoT has gained steam in recent decades, its security issues have also attracted widespread attention. In 2014, a study by Hewlett Packard suggested that about seventy percent of IoT devices suffer from acute vulnerability, which cannot be ignored [12]. Therefore, considerable authentication protocols for Internet of Things sprang up.

Most of the incipient authentication protocols are based on 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. [13] 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. [14] involved the elliptic curve cryptography belonging to the public key system to achieve the implicit certificate-based protocol. Besides, Amin et al. [15] 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 physical attack but also it contributes to terribly large computation costs.

Then, the study on protocols with symmetric cryptography is generally extensive. Das et al. [16] 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 Internet of Things. However, it is similar that the scheme, which is not tamper-proof, cannot avoid physical attacks. Turkanovi and Holbl [17] designed another protocol for HWSN, which pointed out the flaws in [16] and eliminated its redundant components, taking advantage of the symmetric encryption or decryption. Nevertheless, even if 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 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 research. Aman et al. [18] showed the scheme where the response generated by PUF encrypted the data and verified the source. Chatterjee et al. [19] proposed the scheme which used the response value to construct the session key. What is more, there is no need to explicitly store the challenge-response pair. However, the protocols mentioned in [18, 19] fail to guarantee anonymity. In addition, the challenge-response pair is not updated and replaced every round, even when the protocol introduced by Feikken et al. [20] avoids conveying the identity in plain text. Consequently,

considering the device anonymity, Gope and Sikdar [10] presented a scheme with plentiful alternative pseudonyms and challenge-response pairs. Instead of direct identity, it completes communication with the help of 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. [21] resolved the above two weaknesses, but its overhead increases due to asymmetric cryptography. Additionally, the protocol in [22] performs better than that in [10] in terms of resistance to desynchronization attack. On the contrary, the majority of protocols such as [18] merely consider the ideal PUF. Since noisy factors are inescapable in daily life, it is required to take appropriate measures against them. Significantly, the fuzzy extractor is regarded as a widely used and practical tool for error correction. In the part of noisy PUF in [22], the fuzzy extractor emerges to convert the error response values. Besides, the protocol in [20] 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 [10, 21, 23, 24] 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 [25], 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; thereby, the PUF is increasingly shining in the security domain.

Additionally, the definition in [26] 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 shown in the following equation, C is the challenge inputted and R is the response outputted:

$$R = \mathrm{PUF}(C). \tag{1}$$

In ideal circumstances, 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 [27].

A fuzzy extractor (m, l, t, ε) comprises two algorithms, which are $\mathrm{Gen}(\cdot)$ and $\mathrm{Rec}(\cdot)$, according to [20,27]. As a probabilistic algorithm, $\mathrm{Gen}(\cdot)$ 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, with (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 $R\prime$ is at most t, $\mathrm{Rec}(\cdot)$ can utilize hd and $R\prime$ to reproduce k, according to (3):

$$(k, hd) = \mathrm{Gen}(R), \tag{2}$$

$$k = \mathrm{Rec}(R', hd). \tag{3}$$

Generally, the reconstruction function $\mathrm{Rec}(\cdot)$ is deployed on the device with a PUF, while the key generation function $\mathrm{Gen}(\cdot)$ is placed in the server. However, it is a critical defect that the reconstruction algorithm is performed on the device end with limited memory and computing resources as a consequence of numerous gates and time costs when correcting errors [28]. Therefore, the reverse fuzzy extractor, which sets $\mathrm{Gen}(\cdot)$ on the PUF-equipped device and $\mathrm{Rec}(\cdot)$ on the server, is applied to resolve the problem.

3.3. Symbols and Descriptions. The symbols and descriptions involved in the protocol are presented in Table 1.

4. System Model and Security Requirements

4.1. System Model. Figure 1 shows two roles in the system model: a series of IoT devices and a server situated in the data center. Moreover, the communication between devices and the server is through Internet in the IoT system.

(i) 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.

(ii) 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.

4.2. Adversary Model. In matters of the adversary model, we refer to the well-known Dolev–Yao attack model in [29], with an assumption that an adversary A boasts a series of capabilities as described below:

(i) 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.

Table 1: Symbols and descriptions.

Symbols	Descriptions
Di	The identity of the IoT device
TDi	The one-time temporary identity of IoT device
RTi	The registration time
Ti	The current timestamp
(C_i, R_i)	The challenge-response pair
(Ni)	The nonce generated by the IoT device
(Ns)	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
$h(\cdot)$	The secure one-way hash function
$\|$	The concatenation operation
$\oplus$	The XOR operation

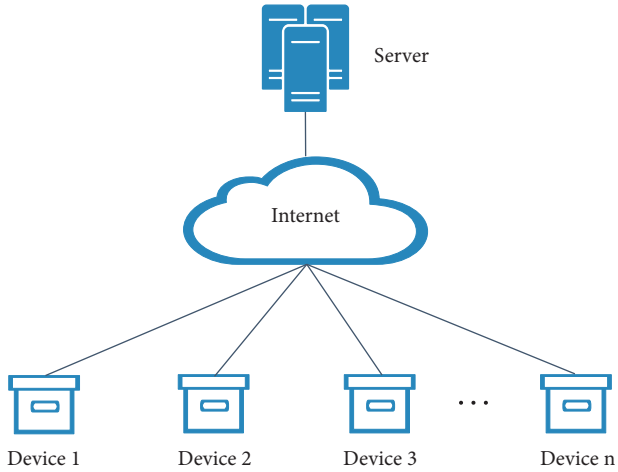

Figure 1: The system model.

(ii) 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 adversary model, we take account of the related security requirements for the proposed two-party authentication protocol:

(i) 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.

(ii) 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.

(iii) Anonymity: It is indispensable to use one-time aliases so that the adversary cannot know the true identity of the device.

(iv) 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 edge IoT nodes 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: $(0,1)^* \longrightarrow \{0,1\}^l$ is selected to achieve 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 shown in Figure 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 $\mathrm{FR}_i = \mathrm{PUF}(D_i \| \mathrm{RT}_i)$ in order to prepare for the supplementary subprotocol against the desynchronization attack. Then, the device randomly chooses a one-time temporary alias $\mathrm{TD}_i \in \{0,1\}^l$ and 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: $\{D_i, \mathrm{TD}_i, (C_i, R_i), \mathrm{FR}_i, \mathrm{RT}\}_i$ 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 this stage, the device and the server in the IoT system conduct mutual authentication where a few pseudo-identities and shared 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 TD_i of this round to the server S. On receiving the alias, the server searches for it in the database. If found 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, Msg_1: $\{C_i, N_S^*, h_S\}$ is given to the IoT device.

(2) Upon receiving Msg_1, the IoT device calculates $R_i' = \mathrm{PUF}(C_i)$, $(k_i', hd_i') = \mathrm{Gen}(R_i')$, $N_S' = h(D_i \| C_i) \oplus N_S^*$, 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

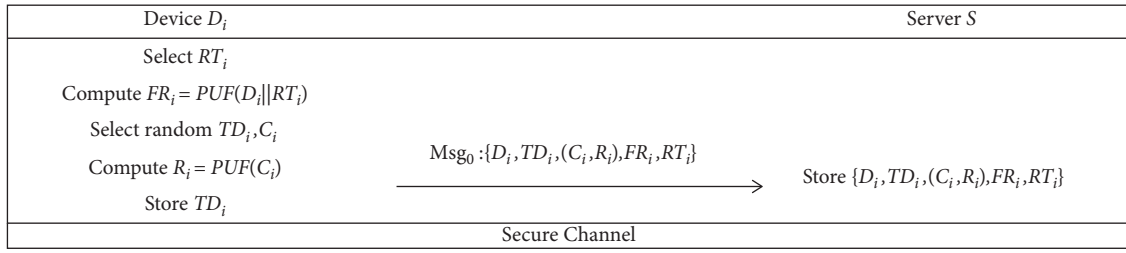

Figure 2: The registration phase.

the next round, the corresponding response $R_i^n = \text{PUF}(C_i^n)$, 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 $\text{TD}_i^n = h(\text{TD}_i \| k_i')$ 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 computes 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 $\text{TD}_i^{n'} = h(\text{TD}_i \| k_i)$ for the following round. Eventually, $\{\text{TD}_i^{n'}, (C_i^{n'}, R_i^{n'})\}$ 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 Figure 3.

5.4. The Supplementary Subprotocol. 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 subprotocol against the attack for the sake of the normal continuation of our authentication.

In the registration phase, the IoT device has calculated $\text{FR}_i = \text{PUF}(D_i \| \text{RT}_i)$ and sent it to the server for storage. In the subprotocol phase shown in Figure 4, with the current timestamp T_i, the device computes $\text{FR}_i' = \text{PUF}(D_i \| \text{RT}_i)$, $Fk_i'' = h(D_i \| \text{RT}_i \| T_i) \oplus Fk_i^*$, and $Fk_i^* = h(D_i \| \text{RT}_i \| T_i) \oplus Fk_i'$ and then transmits $\text{Msg}_3 = \{Fk_i^*, \text{Fhd}_i^*, T_i, \text{RT}_i\}$ to the server end, which searches for the relevant data according to the registration time RT_i sent by the physical device and computes $Fk_i'' = h(D_i \| \text{RT}_i \| T_i) \oplus Fk_i^*$, $Fhd_i'' = h(D_i \| \text{RT}_i \| T_i) \oplus Fhd_i^*$ and $Fk_i = \text{Rec}(\text{FR}_i, Fhd_i'')$ to compare Fk_i with Fk_i'' after receiving the message. If both are the same, the resynchronization is completed and the authentication process can continue normally.

6. Security Analysis

The BAN logic, designed by Burrows, Abadi, and Needham [30], 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 [11]. 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 [11].

Some definitions are listed below:

$$P| \equiv X, \tag{4}$$

$$P \overset{K}{|\sim} M, \tag{5}$$

$$P \overset{K}{\triangleleft} M, \tag{6}$$

$$P \overset{K}{\longleftrightarrow} Q, \tag{7}$$

$$\#(N), \tag{8}$$

$$\sup(P), \tag{9}$$

$$P \triangleleft \| M. \tag{10}$$

Equation (4) denotes that principal P believes formula X to be true. Equation (5) shows that principal P says message M is encrypted with the key K. Equation (6) manifests that principal P sees message M is decrypted with key K. Equation (7) points out that K is considered as a good shared key between principals P and Q. Equation (8) suggests that message N is fresh that it has never appeared before the current protocol conducts. Equation (9) indicates that P is a super principal; namely, it is credible and legitimate. Equation (10) bespeaks that 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 [11] explained that the idealization of

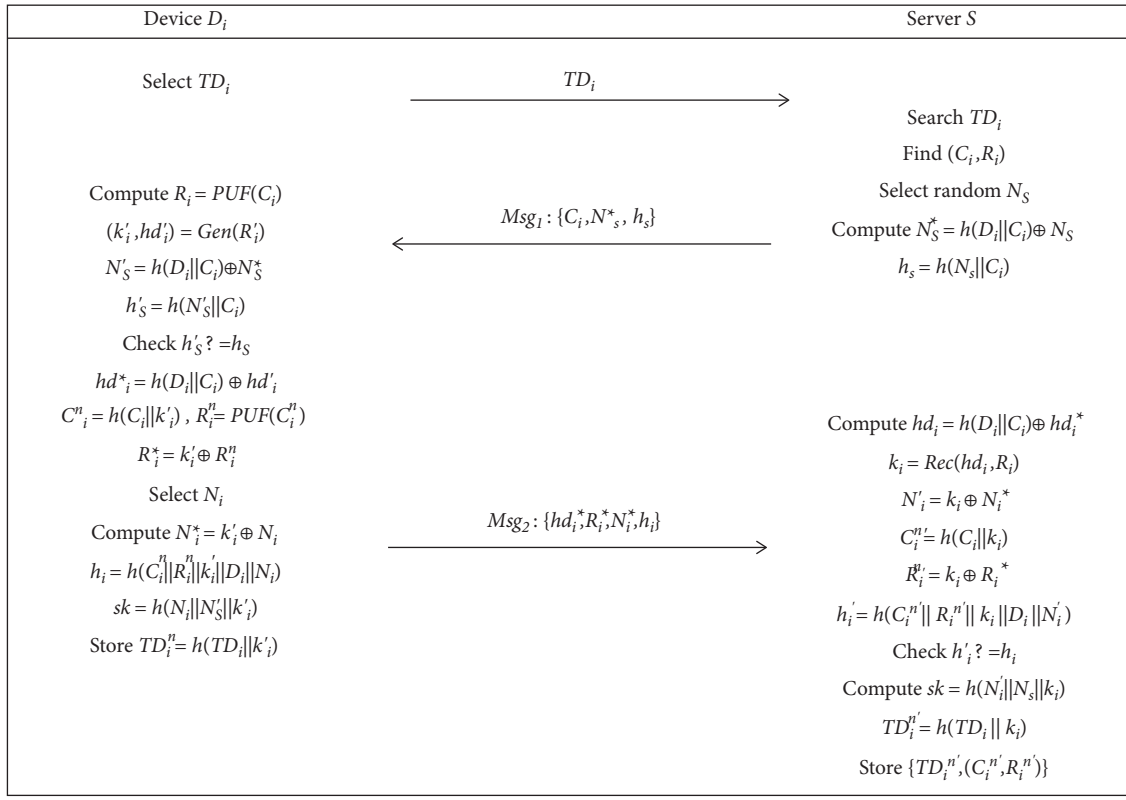

Figure 3: The authentication phase.

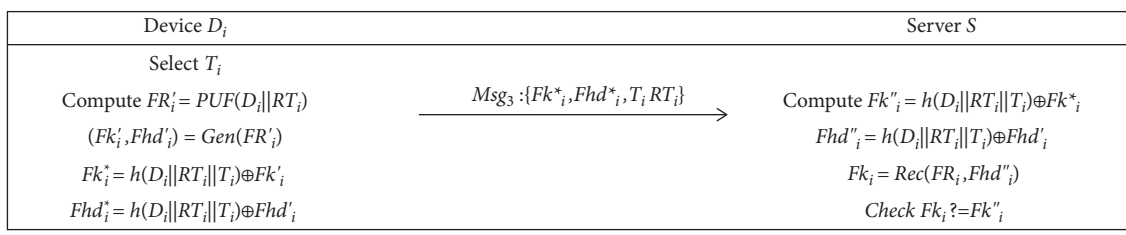

Figure 4: The supplementary subprotocol.

protocol messages converting the implicit contextual information to the explicit specification should be operated. There are some concepts of idealization regulations. On the one hand, there are five related concepts. The atomic message means a data unit with no symbols such as ",", "|", "$\mathfrak{R}$", "" or "", in a message, where "," is a combinator for a message and a principal, and "|" or "$\mathfrak{R}$" is a combinator for two messages. The challenge is an atomic message 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:

(i) All of the atomic messages considered as nonsenses are supposed to be erased.

(ii) If an atomic message plays both roles of the challenge and the response in a line, then it is regarded as a response.

(iii) The challenges separated by commas can be combined with the symbol " |", so do responses.

(iv) The challenge and its corresponding response can be combined with the symbol " $\mathfrak{R}$", whose form is "response $\mathfrak{R}$ replied challenge".

(v) The message and its timestamp can also be combined with " $\mathfrak{R}$", whose form is "message $\mathfrak{R}$ timestamp".

Moreover, according to [11], 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 symbol " $\wedge$" is a Boolean logic conjunction used to connect two formulas. For instance, if formula X and formula Y are true, then they can get the true formula Z, in the following form:

$$\frac{X \wedge Y}{Z}. \tag{11}$$

(vi) 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:

$$\frac{P| \equiv P \overset{K}{\longleftrightarrow} Q \wedge P \overset{K}{\triangleleft} M}{P| \equiv Q| \overset{K}{\sim} M}. \tag{12}$$

(vii) 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; and (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:

$$\frac{P| \equiv P \overset{K}{\longleftrightarrow} Q \wedge P| \equiv S^{C} \triangleleft \| M \wedge P| \overset{K}{\sim} M}{P| \equiv (S \cup \{Q\})^{C} \triangleleft \| M}. \tag{13}$$

(viii) 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 K is a good key between P and Q:

$$\frac{P| \equiv \#(M) \wedge P| \equiv Q| \overset{K}{\sim} M}{P| \equiv Q| \equiv P \overset{K}{\longleftrightarrow} Q}. \tag{14}$$

(ix) The superprincipal rule (15): if P believes that Q trusts X and Q is a legitimate server, P can believe X:

$$\frac{P| \equiv Q| \equiv X \wedge P| \equiv \sup(Q)}{P| \equiv X}. \tag{15}$$

(x) 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:

$$\frac{P| \equiv \#(M) \wedge P \triangleleft N \mathfrak{R} M}{P| \equiv \#(N)}. \tag{16}$$

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

$$\frac{P| \equiv \{P, Q\}^{C} \triangleleft \| K \wedge P| \equiv \#(K)}{P| \equiv P \overset{K}{\longleftrightarrow} Q}. \tag{17}$$

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

$$\frac{P \overset{K}{\triangleleft} M}{P \triangleleft M}. \tag{18}$$

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; and (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 for 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 Figure 5(a):

$$\begin{aligned}
&S| \equiv S \overset{D_i}{\longleftrightarrow} D\,(i),\\
&S| \equiv \#\left(N_S\right)(ii),\\
&S| \overset{D_i}{\sim} N_S\,(iii),\\
&S| \equiv \{S, D\}^{c} \triangleleft \| N_S\,(iv),\\
&S| \equiv \#\left(N_S\right)(v),\\
&S| \equiv S \overset{N_S}{\longleftrightarrow} D\,(vi).
\end{aligned} \tag{19}$$

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 for S; (xiii) means D considers the fact that S believes only D and itself can obtain the nonce N_s; and (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 superprincipal 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 Figure 5(b):

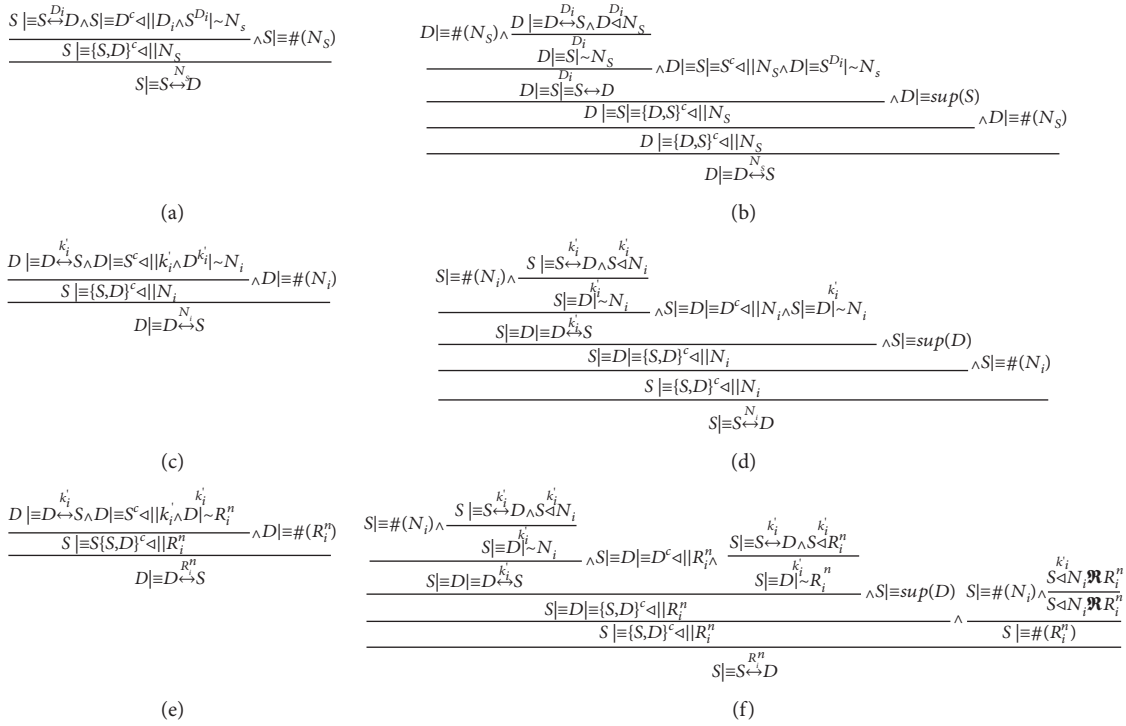

FIGURE 5: (a) The proof for "S believes that N_S is a good shared key between S and D". (b) The proof for "D believes that N_S is a good shared key between S and D". (c) "D believes that N_i is a good shared key between D and S". (d) "S believes that N_i is a good shared key between S and D". (e) "D 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".

$$
\begin{aligned}
&D| \equiv D \overset{D_i}{\longleftrightarrow} S\,(vii),\\
&D \overset{D_i}{\lhd} N_S\,(viii),\\
&D| \equiv S \overset{D_i}{\mid\sim} N_S\,(ix),\\
&D| \equiv \#\left(N_S\right)(x),\\
&D| \equiv S| \equiv S \overset{D_i}{\longleftrightarrow} D\,(xi),\\
&D| \equiv S| \equiv S^c \lhd \| N_S\,(xii),\\
&D| \equiv S| \equiv \{D,S\}^c \lhd \| N_S\,(xiii),\\
&D| \equiv \sup\left(S\right)(xiv),\\
&D| \equiv \{D,S\}^c \lhd \| N_S\,(xv),\\
&D| \equiv D \overset{N_S}{\longleftrightarrow} S\,(xvi).
\end{aligned}
\tag{20}
$$

Similarly, the proofs for "D believe 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" as, respectively, shown in Figures 5(c) and 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 for S"; and "D encrypts N_i with k_i'". These three conditions are involved in deducing 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 that "D believes N_i is fresh" into the good-key rule in order to obtain the final statement. Moreover, the latter is generated by "S believing that N_i is fresh" which is the result of "Sconvinced that D believes only S and D can know N_i"; "S believes that D is a legitimate principal" with the superprincipal 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 "Sbelieving 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 for 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 that "S believes N_i is a fresh nonce" and "S believes D can encrypt N_i with k_i'", which can be obtained by the combination of "S believing 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.

In Figures 5(e) and 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 Figure 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 for S"; and "R_i^n can be encrypted by D with k_i'" to conclude the statement of "S believes it is impossible that a third person can obtain R_i^n except for S and D", which is combined with the fact that "D believes R_i^n is fresh" to deduce the final belief of "D believes that R_i^n is a good shared key between D and S" with the good-

key rule. In Figure 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 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'. 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 as reliable enough 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 [10, 18, 21, 22] 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 is more, Y means achieved while N means not achieved.

In terms of resilience to desynchronization and the zero storage of the shared secrets, even when the scheme in [10] keeps a mass of alternate pseudonyms and keys, the desynchronization attack is still a problem. Although the protocol in [22] can prevent attacks to a certain degree, it still needs to store a large number of pseudo-identities and challenge-response pairs, which require 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 [18]. While the scheme in [22] 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, not only does it solve the noise problem, but it also takes reasonable advantage of resources. What is more, the protocol in [21] 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 [18] 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.

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
[18]	Y	Y	N	Y	Y	N	Y	N	Y
[10]	Y	N	Y	Y	Y	Y	N	Y	Y
[21]	Y	Y	Y	Y	Y	Y	Y	Y	N
[22]	Y	Y	Y	Y	Y	N	N	Y	Y
Our protocol	Y	Y	Y	Y	Y	Y	Y	Y	Y

7.2. Computation Costs Analysis. Considering the difference of the computation costs generated by various PUF-based protocols, we show the details in 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 [21] is based on the three-party authentication, we just conduct the comparative analysis of our protocol and those in [10, 18, 22]. In our protocol, $h(D_iC_i)$ in the IoT device is used twice. As a result, we only consider the time cost of calculating it once. According to Table 4, we can conclude that our protocol still has a slight advantage compared with the protocol in [18]. Although it uses fewer 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 [10], which is also a narrow margin. Furthermore, the computation costs of our PUFs and hash functions are similar to those of [22], 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 contrast the computation costs of relevant protocols in [10, 18, 22], as shown in Table 4, attributing to the fact that the protocol in [21] involves three parties and causes numerous costs with asymmetric encryption and decryption. In Table 4, Size means the size of messages and Times means the times of sending messages. It is apparent that the computation costs of the protocol in [18] are much more

Table 3: The analysis of computation costs (ms).

Protocols	[18]	[10]	[22]	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	$5T_H + 2T_S$	$7T_H + 1T_R$	$6T_H + 1T_G$	$6T_H + 1T_R$

Table 4: The analysis of communication costs (bits).

Protocols	[18]		[10]		[22]		Our protocol	
	Size	Times	Size	Times	Size	Times	Size	Times
The IoT device end	$35l$	2	$5l$	2	$4l$	2	$5l$	2
The server end	$26l$	1	$3l$	1	$5l$	2	$3l$	1
Total	$61l$	3	$8l$	3	$9l$	4	$8l$	3

Table 5: The summary comparisons of protocols.

Protocols	Security functions	Computation costs	Communication costs
[18]	Part	Highest	Highest
[10]	Part	Higher	Lowest
[21]	Part	—	—
[22]	Part	Higher	Higher
Our protocol	All	Lowest	Lowest

than any other protocol resulting from symmetric encryption and decryption. Additionally, the communication overhead of our protocol is as little as that in [10]. Besides, even though the communication costs of the IoT device in the protocol proposed by [22] are less than ours, regardless of the total size of messages or the total times of communications, the protocol in [22] 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 [10, 18, 21, 22] and this paper. Since the computation and communication costs of the protocol in [21] are not involved in the above comparisons, we ignore them in Table 5, 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 edge IoT nodes with physical unclonable functions. Instead of symmetric or asymmetric cryptography, the proposed protocol only uses lightweight operations, such as hash functions, PUFs, exclusive OR operations, and concatenation operations. On the one hand, we can solve the problem of a large number of pseudonyms in IoT devices due to anonymity and effectively resist physical security attacks from adversaries. On the other hand, we can consider PUF in nonideal environments and use fuzzy extractors to implement error correction to ensure the protocol's reliability. In addition, we present a strict formal security proof to show that the proposed protocol meets the expected security requirements. Performance comparison analysis shows it has better computing efficiency and communication performance when compared with similar protocols.

We use subprotocols to resist desynchronization attacks. Although it is simple to implement, it is still not a very effective method to solve the desynchronization attack in the lightweight anonymous security authentication protocol. Therefore, our next work will further find better solutions.

Data Availability

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

Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents a conflicts in connection with the work submitted.

Acknowledgments

The work was supported in part by the National Natural Science Foundation of China (61862052) and the Science and Technology Foundation of Qinghai Province (2020-ZJ-943Q).

References

[1] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (iot) authentication schemes," *Sensors*, vol. 19, no. 5, 2019.

[2] F. A. Turjman and M. Abujubbeh, "Iot-enabled smart grid via sm: an overview," *Future Generation Computer Systems*, vol. 96, pp. 579–590, 2019.

[3] V. Chauhan, M. Patel, S. Tanwar, S. Tyagi, and N. Kumar, "Iot enabled real-time urban transport management system," *Computers & Electrical Engineering*, vol. 86, Article ID 106746, 2020.

[4] B. S. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: technologies, challenges, and opportunities," *IEEE Access*, vol. 5, Article ID 26521, 2017.

[5] S. H Shah and I Yaqoob, "A survey: internet of things (iot) technologies, applications and challenges," in *Proceedings of the 2016 IEEE Smart Energy Grid Engineering (SEGE)*, pp. 381–385, IEEE, Oshawa, ON, Canada, August 2016.

[6] N. Bates, *Driverless Vehicle Security: Considering Potential Attacks and Countermeasures for Military Applications*, Department of Information Security, Egham, Surrey, 2020.

[7] J. Cui, F. Wang, Q. Zhang, Y. Xu, and H. Zhong, "An anonymous message authentication scheme for semi-trusted edge-enabled iiot," *IEEE Transactions on Industrial Electronics*, vol. 68, Article ID 12921, 2020.

[8] N. Tariq, A. Qamar, M. Asim, and F. A. Khan, "Blockchain and smart healthcare security: a survey," *Procedia Computer Science*, vol. 175, pp. 615–620, 2020.

[9] J. Cui, J. Lu, H. Zhong, Q. Zhang, C. Gu, and L. Liu, "Parallel key-insulated multi-user searchable encryption for industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2021.

[10] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[11] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *Proceedings of the Computer Security Foundations Workshop VI*, pp. 147–158, IEEE, Franconia, NH, USA, June 1993.

[12] I. Lee and K. Lee, "The internet of things (iot): applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[13] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. L. Rongxing, and X. S. S. Xuemin, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.

[14] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed iot applications," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2728–2733, Istanbul, Turkey, April 2014.

[15] R. Amin, S. K. Islam, M. K. Khan, A. Karati, D. Giri, and S. Kumari, "A Two-Factor Rsa-Based Robust Authentication System for Multiserver Environments," *Security and Communication Networks*, vol. 2017, Article ID 5989151, 15 pages, 2017.

[16] A. K. Das, P. Sharma, S. Chatterjee, J. K. Sing, and K. S. Jamuna, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1646–1656, 2012.

[17] M. Turkanovic and M. Holbl, "An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *Elektronika ir Elektrotechnika*, vol. 19, no. 6, pp. 109–116, 2013.

[18] M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual authentication in iot systems using physical unclonable functions," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1327–1340, 2017.

[19] U. Chatterjee, V. Govindan, R. Sadhukhan et al., "Building puf based authentication and key exchange protocol for iot without explicit crps in verifier database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, pp. 424–437, 2018.

[20] K. B. Frikken, M. Blanton, and M. J. Atallah, "Robust authentication using physically unclonable functions," in *Proceedings of the International Conference on Information Security*, pp. 262–277, Springer, Pisa, Italy, September 2009.

[21] Qi Jiang, X. Zhang, N. Zhang, Y. Tian, X. Ma, and J. Ma, "Two-factor authentication protocol using physical unclonable function for iov," in *Proceedings of the 2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 195–200, Changchun, China, October 2019.

[22] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for rfid systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[23] W. Feng, Y. Qin, S. Zhao, and D. Feng, "Aaot: lightweight attestation and authentication of low-resource things in iot and cps," *Computer Networks*, vol. 134, pp. 167–182, 2018.

[24] M. Mitev, M. H. Shekiba, A. Chorti, and M. Reed, "Multi-factor physical layer security authentication in short block-length communication," 2020, https://arxiv.org/abs/2010.14457.

[25] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*, Katholieke Universiteit Leuven, Leuven, Belgium, 2012.

[26] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 2007 44th ACM/IEEE Design Automation Conference*, pp. 9–14, San Diego, CA, USA, June 2007.

[27] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 523–540, Springer, Interlaken, Switzerland, May 2004.

[28] A. V. Herrewege, S. Katzenbeisser, R. Maes et al., "Reverse fuzzy extractors: enabling lightweight mutual authentication for puf-enabled rfids," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 374–389, Springer, Kralendijk, Bonaire, Sint Eustatius and Saba, March 2012.

[29] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[30] M. Burrows, M. Abadi, and R. N. Michael, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, pp. 233–271, 1989.