TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/13 pp625–635
DOI: 10.26599/TST.2019.9010072
Volume 25, Number 5, October 2020

ePUF: A Lightweight Double Identity Verification in IoT

Bo Zhao, Pengyuan Zhao*, and Peiru Fan

Abstract: Remote authentication is a safe and verifiable mechanism. In the Internet of Things (IoT), remote hosts need to verify the legitimacy of identity of terminal devices. However, embedded devices can hardly afford sufficient resources for the necessary trusted hardware components. Software authentication with no hardware guarantee is generally vulnerable to various network attacks. In this paper, we propose a lightweight remote verification protocol. The protocol utilizes the unique response returned by Physical Unclonable Function (PUF) as legitimate identity basis of the terminal devices and uses quadratic residues to encrypt the PUF authentication process to perform a double identity verification scheme. Our scheme is secure against middleman attacks on the attestation response by preventing conspiracy attacks from forgery authentication.

Key words: Internet of Things (IoT); Identity-Based Encryption (IBE); Physically Unclonable Functions (PUFs)

1 Introduction

In recent years, the Internet of Things (IoT) has been the most popular concept and product of developments in information technology. By 2020, more than 50 billion terminal devices are expected to be connected to the internet[1]. One key challenge in the safety of IoT is the vulnerability of ending devices in defending numerous attacks or intrusions. In IoT, sensor devices are often exposed in the environment that are out of control. Thus, their functioning is easily influenced by various factors such as environmental issues and malicious attacks. Compared with traditional concepts of security, IoT broadens and amplifies the traditional attack surface. One of the most common intrusions in IoT is impersonation attack[2]. An attacker can pretend to be a legal device to steal or tamper with sensitive data. Such an attack will greatly cut down the accuracy of the data on the IoT. Therefore, how to correctly identify sensor devices is a major concern for the security of IoT[3].

Physical Unclonable Functions (PUFs) have provided an effective and low-cost mechanism to solve the aforementioned problem. Accordingly, some authentication protocols based on such technology have been proposed in several studies[4–6]. However, these PUFs are vulnerable to modeling attacks as shown in many actual scenes[7–9]. Especially, the corresponding communication protocols are not immune to Man-In-The-Middle (MITM) attacks. In IoT, an adversary can easily intercept the challenge sent by a sensor node (prover) and pretend to be the legal ending device. Thus, authenticating the identity of an ending device with PUF is a primary task of ensuring secure communication in IoT.

Traditional authentication protocols using PUF described have inherent limitations[10]. In enrollment of the sensor stage, the Trusted Third Party (TTP) stores a large number of Challenge-Response Pairs (CRPs) in its database. To avoid an MITM replay attack, once the CRP is used, it is removed from the database. Although this protocol provides a good solution for using a PUF as primitive authentication evidence, an effective key-exchange mechanism is absent if the former protocol is used independently, thereby creating an obstacle. Another major reason for why it is unsuitable for the IoT environment is that this protocol does not scale well

• Bo Zhao, Pengyuan Zhao, and Peiru Fan are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China. E-mail: zhaobo@whu.edu.cn; zhaopengyuan@whu.edu.cn; fanpeiru@whu.edu.cn.
• Pengyuan Zhao is also with the School of Cyber Security and Computer, Hebei University, Baoding 071002, China.
* To whom correspondence should be addressed.
Manuscript received: 2019-11-14; accepted: 2019-11-19

with the huge number of sensor devices in IoT, because the database is required to store a very large number of CRPs[11]. The maintenance cost is always on a high point because when a device's CRP in the database is depleted, a new enrollment is necessary for the particular device.

Furthermore, using the traditional key exchange mechanism to protect the PUF authentication process is not suitable for IoT. Diffie-Hellman[12] method cannot distinguish identity, so it cannot prevent MITM attacks; it is also prone to CRP leakage at the sensor node. To resolve this, many communication protocols use the way of storing certificates in sensor devices[13, 14]. Although this method can ensure the integrity of the keys, the actual deployment is difficult. The reason is that in the enrollment phase, it is almost impossible of ensuring that the certificate embedded in the Non-Volatile memory (NV) of the sensor is exactly the server's certificate when sensor and server (verifier) are deployed in IoT; thus, a highly complicated certificate issuing mechanism is needed. Another main reason for not applying it in IoT is the limited computing and storage capability of sensor nodes. Furthermore, the certificates stored in NV are vulnerable to attacks and can be easily replaced by attackers because of poor self-protection capability of the sensor devices.

In this paper, we propose a new terminal authentication scheme based on PUF in the IoT environment. Encrypted Physically Unclonable Function (ePUF) is a protocol that utilizes a double identity verification using device feature information Meta Data of Sensor node (MDS) and PUF. We adapt an identity-based encryption scheme that utilizes quadratic residues to encrypt the PUF authentication process. The scheme that we propose is simple, lightweight, and flexible. This scheme has the following characteristics:

• **Suitable for IoT:** The authentication protocol proposed in this paper does not need to store certificates in sensor nodes, nor does it need to store large amounts of CRP on the server. Instead, only a small amount of public information is stored in the sensor node, and the server only needs to store one CRP for each terminal device. Sensor devices can be flexibly connected to any server, which is in accordance with the characteristics of easy access and flexibility of IoT.

• **Secure in key exchange:** When a server needs to create a session key between the sensor node and itself, a "seed" of the session key can be issued from the server to the sensor node. According to the algorithm, the key will not be exposed to the adversary.

• **Trustful in authentication:** In our authentication scheme, the server sends a challenge of PUF that is encrypted to the sensor node. The CRP is never revealed to the outside. Therefore, it is trustworthy in considering that the response from the sensor device can represent its real identity. The identity legitimacy of IoT terminals is guaranteed.

The rest of this paper is organized as follows: Section 2 reviews the related work on using PUF as authentication and typical session key exchange scheme. Section 3 introduces the principle of PUF and identity-based encryption scheme based on quadratic residues. Section 4 describes the process of our key exchange protocol and improvement of the referenced algorithm in detail. Section 5 analyzes the protocol to prove the security and robustness. Section 6 presents the evaluation design and experimental results. We conclude the paper with a discussion of limitations and potential future work in Section 7.

2 Related Work

Since the advent of PUF technology, a great deal of improvement has been observed in PUF and the design of security communication framework by leveraging it. Rührmair[15] discussed a recent cryptographic primitive termed SIMulation Possible, but Laborious (SIMPL), which is a type of strong PUF coming with a publicly known, individual numeric description that allows slow simulation and output prediction. Another strong PUF-based authentication protocol has been proposed in the past, such as controlled PUF[16], which reduces the number of CRPs used to protect against modeling attacks. To avoid Denial Of Service attack (DOS), Özttextürk et al.[17] presented a noisy PUF that introduced a random parameter called time stamp, through which the server can keep track of the sessions running between two communicating and Katzenbeisser et al.[18] presented a reconfigurable PUF protocol that implemented a pseudo-random number generator[19, 20] to counteract impersonation and also used the timestamp to prevent the protocol.

Kocabaş et al.[21] proposed a converse PUF-based authentication that is secure against passive and active attacks by providing an extensive security analysis against passive adversaries and a concrete instantiation using controlled arbiter PUF. Moriyama et al.[22] proposed such a PUF-based authentication protocol under complete memory leakage. Their protocol was

further adapted by Aysu et al.[23] by reversing generating and copying programs to improve the adaptability to lightweight devices at the cost of introducing pre-secrecy secrets.

However, the existing PUF security protocols have certain limitations. Some studies have reviewed eight prominent proposals in chronological order: from the original strong PUF proposal to the more complicated converse and slender PUF proposals[24]. Other studies have surveyed different ultralightweight authentication protocols designed for Radio Frequency Identification (RFID) authentication[25]. Zhang et al.[26] proposed a lightweight PUF-based and cloud-assisted authentication mechanism for multi-hop body area networks.

3 Background Knowledge

3.1 Physical unclonable function

A PUF is a noisy function that is embedded into a physical object, such as an integrated circuit. The concept of PUF was first introduced by Pappu et al.[5] Creating a copy of the PUF in a physical way is believed to be an impossible mission so that an attacker cannot forge an identical challenge-response behavior. The structure in products is unclonable, even by the same manufacturer. Gassend et al.[27] introduced silicon PUFs, which aim to exploit the uncontrollable manufacturing variations that are inherent properties of the Integrated Circuits (IC) fabrication process. Today, several security products are based on PUF in the market, such as PUF-enabled RFID chips and proposals for Internet Protocol (IP) protection and anti-counterfeiting solutions[28]. PUFs' physical basis is noise (e.g., thermal noise), so when they are queried with the same challenge, their responses are typically slightly different. These output changes can be eliminated by using fuzzy extractors[29], and the methods can be efficiently implemented on resource-constrained devices. Based on the assumption that PUF and PUF' are different, PUFs can be considered to have the following properties[30]:

• **Independence:** When the two PUFs are queried with the same challenge x, PUF and PUF$'$ will definitely return two different responses y and y'.

• **Robustness:** When a PUF is queried many times by the same challenge x, a close to 100% likelihood is that it returns the same response$y$.

• **Pseudo-randomness:** Physically distinguishing a PUF by any pseudo-random function is infeasible.

• **Tamper-evidence:** Any attempt to tamper with the physical features of PUF is futile, i.e., PUF can no longer be evaluated but is turned into a random PUF$'$, PUF$\neq$PUF$'$.

3.2 Identity-based encryption scheme based on quadratic residues

An identity-based encryption system includes a function that uses one unique identity (for example, a person's email address) to create the encryption key, thereby avoiding the complex scheme of creating a separate public key. The possibility of such a scheme was first proposed by Shamir[31], but it has proved difficult to implement the above assumption because of absence of practicality and safety. Thereafter, Boneh and Franklin[32] proposed an implementation based on elliptic curves, by which we can discuss the possibility of applying this mechanism to the actual environment. In this study, we adopt another identity-based cryptosystem that uses quadratic residues modulo—a large composite integer[33].

The basic element of the system is a product M of two big primes P and Q, which are held privately by an authority; P and Q are both congruent to 3 mod 4[31]. However, M is a universally available public modulus. By contrast, P and Q are not public and only possessed by an authority. Also, the system turns a universally available secure hash function into an identity-mapping calculation.

Our innovation is based on the theory that although some basic parameters are public in the entire key exchange process, they do not affect the security of the session key. The attacker cannot calculate the key through the existing captured parameters. Therefore, no public key is required to protect the session keys.

The key exchange system is simply described as follows: In the network, an entity's unique identity represents an encryption "seed" by some types of unidirectional hash functions. The authority is responsible for creating the "seed", which is called value a , and the a modulo the general public modulus M satisfies the character in which the Jacobi symbol $\left(\frac{a}{M}\right)$ is $+1$. We can conclude that $\left(\frac{a}{P}\right) = \left(\frac{a}{Q}\right) = +1$, which has the same result as $\left(\frac{a}{M}\right)$, so a is a square modulo of both P and Q , thus a is a square modulo M[33].

At the same time, a is issued to the authenticator, and

the authority also creates a value v, which must satisfy the following:

$$v = a^{\frac{M+5-(P+Q)}{8}} \mod M \tag{1}$$

It is noted that both M and a can be public, on the contrary, v is a private value and only the authority can possess it, since only the authority itself can calculate the square root modulo of M. Such v will satisfy $v^2 \equiv a \mod M$.

In key exchange phase, if an entity wants to communicate to another securely, it must generate a session key first to encrypt the data through symmetric encryption. From the original design, the entity sends to the recipient each bit of the session key in turn as follows: We consider k_i as a single bit of the session key, which $i = 1, 2, \ldots, n$, coded as 1 or 0 corresponding to the Jacobi symbol $+1$ or -1. For each k_i, the transmitter selects a value t at random modulo M, such that the Jacobi symbol $\left(\frac{t}{M}\right)$ equals k_i. Then the transmitter sends $w = \left(t + \frac{a}{t}\right) \mod M$ to the receiver[33].

In order to ensure the secure communication between the two parties, the authority (also as the recipient of w) issues the value v secretly. After accepting w, the authority calculates to recover the bit k_i according to Formula (2):

$$w + 2v = t\left(1 + \frac{v}{t}\right) \times \left(1 + \frac{v}{t}\right) \mod M \tag{2}$$

It follows the Jacobi symbol:

$$\left(\frac{w + 2v}{M}\right) = \left(\frac{t}{M}\right) = k_i \tag{3}$$

As only the recipient knows the value v, he can calculate the Jacobi symbol $\left(\frac{w + 2v}{M}\right)$ and hence recover each k_i.

The largest feature of this mechanism is that no public key is used, and the generation and distribution of keys are relatively simple. We consider this mechanism to be highly suitable for IoT in regarding sensor nodes as the transmitter and servers as the receiver.

4 Protocol Description

The typical infrastructure of protocol consists of several sensor nodes and a server node. Sensor nodes are "tentacles" of IoT because they are responsible for collecting data from the environment and data sources. The server node is responsible for verifying the trustworthiness of data submitted by the sensor nodes. We can simply assume that a certain number of sensor nodes is directly connected to a server node and any single sensor node only submits its data to a specific server. Each sensor node and server node embeds a PUF instance in its device, which is used to authenticate the device identity. We will describe our scheme with respect to the sensor and server nodes. For simplicity, we focus on the identity authentication and secure communication between the sensor and server. However, the scheme can also extend to the superstructure of IoT. The phases of our proposed scheme are described in Fig. 1. For convenience, we call the sensor node the prover and the server node the verifier, respectively.

4.1 Enrollment phase

Initially, an enrollment phase is executed. In this phase, when each sensor node (prover) is in the manufacturing stage, a PUF instance is embedded in it, and one Challenge and Response (CR) pair is created and stored in the TTP's CRP database, which is severely protected. Also, some device-inherent properties during the manufacturing process are also enrolled in the database, such as device ID, type, and production data. We call them MDS. These data can be used as assistant data in the PUF verification phase. To avoid revealing sensitive information, the database also stores a hash code of these messages, which are also stored in the prover. In our scheme, these messages can be read by each prover accordingly.

4.2 MDS verification phase

When a prover needs to communicate with the verifier, it must show its identity first. The detailed steps are described as follows: A prover sends its hash code of identity messages to the verifier, and the verifier goes through the database to compare the code. If the code is correctly compared, it shows that the MDS is enrolled in the previous phase correctly and has a legal device identity. Then, the verifier fetches the MDS string and produces value a modulo M such that the Jacobi symbol $\left(\frac{a}{M}\right)$ is $+1$, which applies the hash function to the MDS string. Typically, this process can be implemented with multiple structured applications of the hash function to produce a set of candidate values for a , until the value $\left(\frac{a}{M}\right) = +1$ is found. Then, the verifier sends a to the prover through a public or unencrypted manner, that is, the value a can be exposed to the outside. We do not need to worry about the integrity of a. The specific security analysis will be discussed in the next section.

Besides sending the value a to the prover, the verifier also produces a square root modulo M, which is called

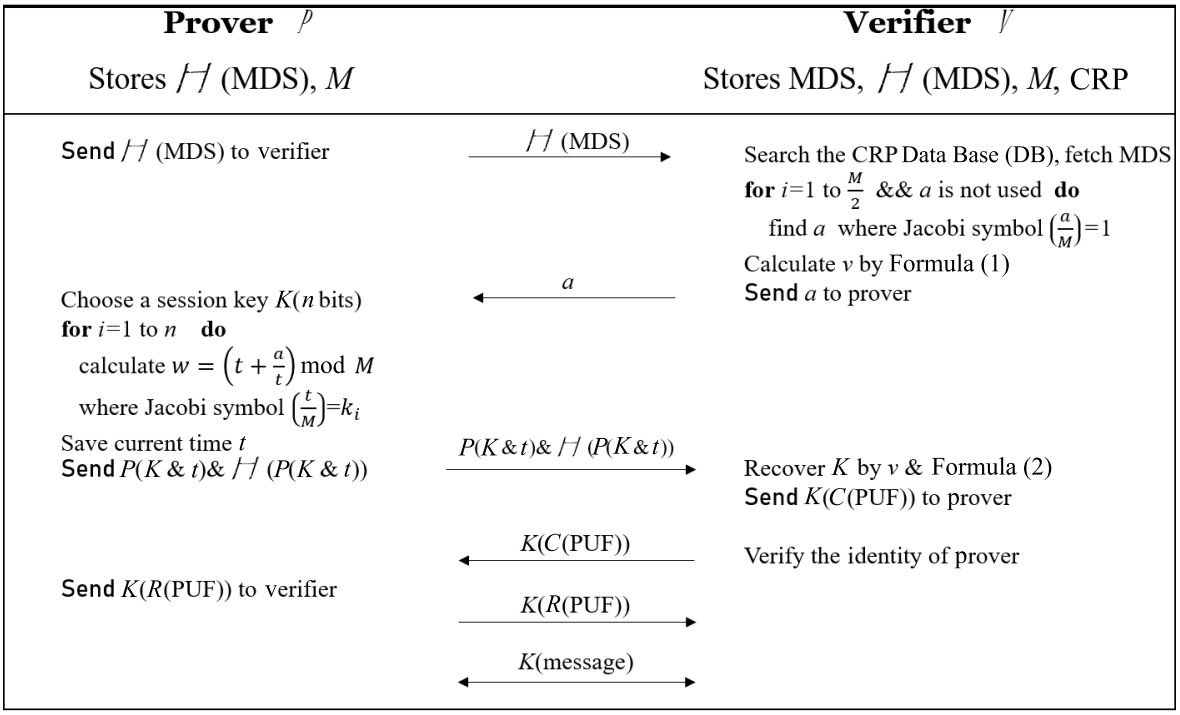


Fig. 1 Remote attestation based on physical functions.

the value v. The method of producing v follows Formula (1). Such v will satisfy $v^2 \equiv a \bmod M$. It is noted that only the verifier can calculate the value v because only the verifier holds the prime factors P and Q, v is severely protected by the verifier and only the verifier can access and use it.

4.3 Session key creation phase

After receiving the value a, a prover needs to create a session key to encrypt the next communication process and send it to the verifier safely. In the original design[33], the prover sends to the verifier each bit of the transport key in turn. However, we consider it as error prone and inefficient. Thus, we adjusted the bit stream and replaced it with a structured byte package. The prover creates a session key, let x be a single bit of it and coded as $+1$ or -1. Then the prover selects a value t at random modulo M, such that the Jacobi symbol $\left(\frac{t}{M}\right)$ equals x and the prover calculates $w = \left(\frac{t+a}{t}\right) \bmod M$ for every x. After finishing the calculation to every bit of the session key, the prover packs all of the x with the following format, as shown in Fig. 2. Each number w is allocated 4 byte of a fixed length. At the end of the last w, 4-byte-long data with all 1 bits are attached, because in our calculation scheme, any meaningful number should be a positive integer. A timestamp follows so that the verifier can identify the exact time that the package was created. At the end is a hash summary value for all the bytes ahead to guarantee the integrity of the package. After packing all the bytes, the prover sends the package to the verifier.

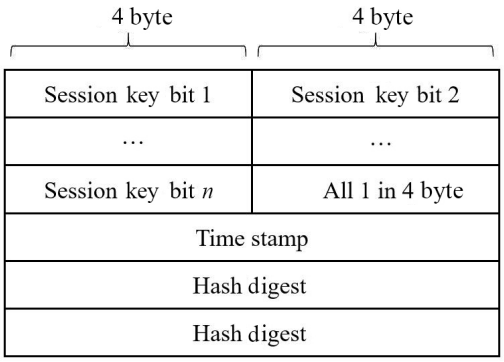


Fig. 2 Structure of session key package.

4.4 PUF verification phase

When the verifier receives the session key package, it decodes each bit of the key using the value v, which is only processed by itself. Formulae (2) and (3) indicate that the verifier can calculate the Jacobi symbol $\left(\frac{w+2v}{M}\right) = k_i$, and therefore recovers x. Then, the verifier fetches the PUF challenge from the CRP database and sends it to the prover encrypted by the recovered session key. If the returned response is

matched, then the prover is a legitimate device and its data can be trusted. Otherwise, the submitted data should be viewed as suspicious or untrusted. The goal of this phase is to authenticate the unique identity of every enrolled device. As in the entire process, the CRP is always encrypted, so we can conclude that the CRP can represent the legal identity of every sensor device in IoT.

The preceding phases have the following two purposes:

(1) Effectively validating the identity of all sensor devices connected to IoT. To achieve this goal, we design a dual mechanism using MDS information and PUF authentication. As the MDS and seed a are not encrypted in the transmission, we utilize MDS information to generate a session key and use the latter to protect CRP in the PUF verification phase. According to the algorithm in Ref. [23], although the MDS and the seed a are public, the attacker cannot calculate the session key, as is discussed in the next sector.

(2) Generating and exchanging session keys for secure communication. Besides encrypting the CRP, the session key can also be utilized in future data transmission from the sensor node to the server. The duration of session keys can be freely determined by the IoT environment according to the actual circumstances. In the most rigorous conditions, the session key can be regenerated every time. It can also be used for a longer period to sacrifice part of security in exchange for efficiency.

5 Security Analysis

In this section, we provide a security analysis of the session key exchange protocol that is described in the previous section. To prove the security of the protocol, we describe some attack models that probably appear in the scene assumed in this paper. We consider two different attack models as described in the following:

• **Malicious PUF model.** In this model, we assume that after the enrollment phase, the attacker forges a PUF by embedding a PUF simulator in the integrated circuit or tampers with the PUF of a senor node by replacing it with malicious or untrusted PUFs. The purpose of these actions is to impersonate legitimate PUFs to obtain encrypted keys communicating with server nodes.

• **Session key security model.** In this scene, we assume that all of the sensor nodes (prover) and servers (verifier) are trusted without enduring any attack. The attacker either eavesdrops on the communication link without tampering with the messages (e.g., packet sniffing) or has full control over the link and can modify the messages (e.g., packet injection or re-routing attack).

Theorem 1. If the enrollment phase in our scheme is operated in a secure environment as we have assumed previously (that is, the CRPs cannot be leaked to the adversary), then our device authentication scheme is secure in the malicious PUF model. No attacker can forge a device that can be successfully verified.

Proof. The cryptographic security of our proposed authentication scheme is based on the uniqueness of the PUF. Suppose an attacker wishes to obtain a legal device identity, i.e., a CRP. As we assume that the entire process in which the manufacturer instantiates the CRP for a particular chip is in an absolutely safe environment, it is impossible for the adversary to obtain a legal device identity in the enrollment phase. In other words, the attacker has to forge a device D' using the chip replication technology or PUF simulator to be successfully verified in the authentication phase. According to the inherent property of PUF, the uniqueness metric is defined as

$$\text{Uniqueness} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{\text{HD}(R_i, R_j)}{n} \times 100\% \tag{4}$$

where $\text{HD}(R_i, R_j)$ is the Hamming Distance (HD) between the responses of i-th and j-th PUFs embedded in two chips for a particular challenge C and n is the total number of chips under our consideration, so $k \times n$ is the total amount of response bits of the PUF. The metric is used to calculate the Hamming distance of a PUF instance (R_i, R_j) $(i \neq j)$, which is for the same set of applied challenges[14]. In this paper, we use U to represent the uniqueness. The ideal value of the uniqueness metric is 50%. According to the inherent property in PUF, given n responses, which is represented by R_i by each, there is no efficient clone procedure that can build another physical PUF device R' in which the HD between the R' and any other R_i is less than $2U$, $i = 1, 2, \ldots, n$.

Now suppose that a device R' can be successfully verified. According to the theory, the HD between the evaluation result of PUF in R' and any other R_i is less than $2U$. This can follow in two cases. First, if the response in R' is equal to any legal one, then the attacker has successfully cloned a PUF instance. It is obvious that this is a violation of basic physical theory of PUF. Second, if the PUF is in $R' \in (R_1, R_2, \ldots, R_n)$, then it shows that the attacker successfully forged a PUF in

R' such that the output of R' has Hamming distance less than $2U$ from any output of R_i. According to the uniqueness of PUF, this is also an impossible situation. This feature of the PUF circuit embedded in the chip provides the ability to be uniquely identified from the same type of PUF instance.

Theorem 2. If an attacker can control the communication between the sensor device and the server but can obtain neither the session key nor the seed r from the server or the sensor node, then although the attacker can eavesdrop or modify all of the packets at the verification phase, the session key is still considered safe and will not be leaked to the attacker.

Proof. Our verification mechanism focuses on two points:

(1) Effectiveness of sensor device authentication;

(2) Security of session key transmission.

In our scheme, we designed a dual authentication mechanism using MDS and PUF. In the MDS verification phase, the prover sends the hash code of the MDS to the verifier, who then searches its database to obtain the MDS string for generating the value a which is transmitted to the prover. MDS has never been exposed to the outside, so we can recognizance it as safe. As the hash of MDS is public in the network, the possibility of an attacker obtaining it and thus posing as a legitimate node cannot be excluded. To avoid this situation, we design a second verification, which is in the PUF verification phase. The PUF challenge, which is only possessed and sent by the verifier, is encrypted in this action. Thus, we can conclude that as long as the session key used to encrypt the PUF challenge is not grabbed by the attacker, the returned response by the prover is true and credible.

Our method focuses on the security of session keys. Our key exchange algorithm is based on the cryptosystem that uses quadratic residues modulo—a large composite integer. Thus, one way to destroy this security mechanism is to determine the decomposition factor of M. As M is publicly distributed, it is also feasible for $\frac{M+5-(P+Q)}{8}$ to generate the exponent used to compute square roots. However, this is not our concern. We only consider an active attack against each bit of session key. For general consideration, we suppose that an attacker can recover every bit of the session key from the package without knowing either r or the factors of M. Under this consideration, the attacker may compute the following map:

$$F(M, a, w) \to x = \left(\frac{t}{M}\right) \tag{5}$$

where $w = \left(t + \frac{a}{t}\right) \bmod M$ for t.

Then we consider when an attacker obtains the value a, what the value of F could be evaluated where the Jacobi symbol $\left(\frac{a}{M}\right)$ is $+1$, but a is not a square. So the Jacobi symbols $\left(\frac{a}{P}\right)$ and $\left(\frac{a}{Q}\right)$ will both be -1. If t is the value used to calculate s, there will be three other values t_1, t_2, and t_3 giving the same value of w. These are given by

$$t_1 \equiv t \bmod P, \quad t_1 \equiv \frac{a}{t} \bmod Q \tag{6}$$

$$t_2 \equiv t \bmod P, \quad t_2 \equiv \frac{a}{t} \bmod Q \tag{7}$$

$$t_3 \equiv t \bmod P, \quad t_3 \equiv \frac{a}{t} \bmod Q \tag{8}$$

as $\left(\frac{a}{P}\right) = \left(\frac{a}{Q}\right) = -1$, then $\left(\frac{t_1}{M}\right) = \left(\frac{t_2}{M}\right) = -\left(\frac{t}{M}\right) = -\left(\frac{t_3}{M}\right)$. So, there is no unique $\left(\frac{t}{M}\right)$ to recover, as F cannot return $\left(\frac{t}{M}\right)$ correctly more than half the time whenever a is not a square. Hence, we would have a procedure that can distinguish the two cases of $\left(\frac{a}{M}\right) = +1$; that is, determining whether a is a square or a non-square without factoring M. This is the quadratic residuosity problem which is currently unsolved, and also is a problem on which a number of other public key systems are based.

The original encryption scheme is vulnerable to an adaptive chosen cipher text attack. As the prover transmits x that represents one bit of the session key at a time, an attacker may intercept the original x and forge another one and send it to the verifier. This process does not expose secret messages. By using x' provided by the attacker, the verifier cannot restore the forgery key expected by the attacker through its own R. However, it can destroy the next authentication process. To avoid this situation, we have improved the transmission of the session key by appending a hash digest of the entire package. Furthermore, a time stamp can help the verifier judge the facticity of the received session key. If the distance between the timestamp and the current time is beyond a normal threshold, then the verifier has reason to suspect whether the packet is tampered with by an attacker.

6 Evaluation

This section presents the evaluation of our scheme using real Static Random-Access Memory (SRAM) PUF devices. We have implemented the full verification steps of our scheme. As our scheme is based on the physical property of CRP uniqueness of PUFs, we do not need extensive modifications in the embedded chip and most parts of our scheme are implemented on software. The goal of our evaluation is to validate whether our protocol can provide safe and correct authentication while also adapting to the limited computing and storage capabilities of terminal devices. In particular, we want to verify that our scheme is robust even under the targeted attack model that is mentioned in Section 4. A device authentication scheme should be robust even when terminal devices have no protection in the environment.

Our main target is to use PUF for device authentication. Thus, we use the PUF instance for each ALINX board as implemented in the AX7020 board containing an XILINX Zynq7000 SOC chip. This method leverages Advanced RISC Machine (ARM)+Field Programmable Gate Array (FPGA) System On Chip (SOC) technology and integrates dual-core ARM Cortex-A9 and FPGA programmable logic on one chip, which is connected to the particular board. We consider its parameter suitable for the identity of an IoT device. In this evaluation, we use PC as the server and implement a CRP database that is deployed on a PC to store CRP for each of the IoT devices in the enrollment phase. At the authentication phase, the PC sends the challenges to the ALINX board through WiFi and the board applies the challenge to FPGA, collects the response, and sends it back to the PC.

• **Implementation in FPGA.** We first implement the SRAM-based ePUFs on the FPGA platform to authenticate its usability as a hardware security foundation. Figure 3 presents the infrastructure of the development platform, which has two components: the PC as verification server and an FPGA board equipped with SRAM PUFs. The host PC is responsible for maintaining a database for storing CRPs and MDS with its related information. In the verification phase, the host PC creates the seed a and restores the session key k from the PUF. The PUF is responsible for receiving a and generating the session key k.

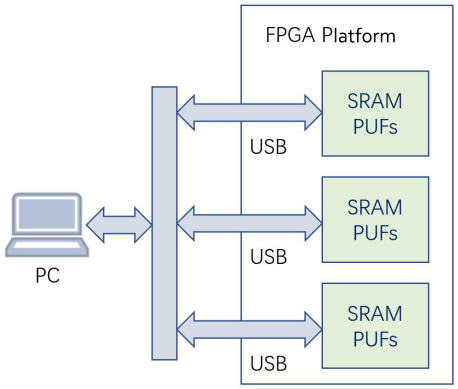


Fig. 3 Architecture of FPGA platform.

• **CRP databases.** The CRP database is stored and maintained in a PC verifier and the CRP for each PUF is generated using SRAM PUF during the enrollment phase. Compared with the original CRP mechanism, our database does not need to store a large number of CRPs but can satisfy the specific security needs in different scenes. Two problems occur in using SRAM PUF to extract keys: (1) The response of SRAM PUF is different between the same chip because of the influence of ambient noise and (2) the initial power value of SRAM is not evenly distributed.

To solve these problems, we introduce a fuzzy extractor model[34, 35]. The function of the extractor is to obtain the same output from two input data with minimal difference, and the output data have a good uniformity distribution.

In this design, we use SRAM start address 0x10000000 size 580-bit continuous storage unit as seed value of BCH coding and start address 0x10001000 size 4096-bit continuous storage unit as SRAM identification code to generate system keys. MDS data are stored in the start address 0x0800000 size 4095-bit continuous storage unit.

Example of CRP database is shown in Table 1, we can conclude that taking the No.1 CRP message as an example in the MDS fragment, we select some IoT device model label strings to simulate different PUFs. In every device, we choose the same SRAM memory location to generate a session key. The key after the

Table 1 Example of CRP database.

Device	MDS	Challenge	Response
Device 1	Apexis-camera-APM-J011-2018-2	0x10001000−0x100011FF	0xA6995421...B3014525
Device 2	TP-Link-WiFi Smart-plug-HS100	0x10001000−0x100011FF	0xCCFb2024...D93022157
Device 3	Netatmo-weather-station-with wind-gauge	0x10001000−0x100011FF	0x6682F2A1...A8C33362

fuzzy exactor is represented by 128 bits.

• **Time complexity.** As the security of the session key exchange process has been proved by mathematical principles, in this evaluation, we focus on the time overhead required to generate different lengths of session keys at the sensor node to find the most suitable key strength for the IoT environment under existing hardware conditions. At the same time, we measure the time required for the encrypted CRP verification process. In this verification scheme, the overhead of a sensor node mainly comes from computing bits for the session key, so the time complexity is proportional to the length of the key $o(n)$. We select different lengths of the key from 48 bits to 128 bits for testing, and the result is shown in Fig. 4. For each secret length of the key, we calculate the total time needed to generate a new P and Q and the secret key, and then we test the average value repeatedly.

From the results, we can conclude that the time required to calculate the secret key is proportional to the length of the key. However, the time to generate secret keys is not significantly increased as the length of the secret key increases. This condition indicates that we can select the suitable length of the secret key to protect the communication security under the premise of computing power.

In the IoT environment, the time required for authenticating the identity of each device is not only decided by the length of the secret key itself, but is also tightly bound to the number of concurrent ending devices connected in IoT. We simulate the number of concurrent nodes with the maximum 1000 by using 96 bits and 128 bits lengths of the key. The result is an average response time of each IoT device, as shown in Fig. 5.

• **Reliability.** Identity authentication requires each PUF to be unique, that is, the response of different PUFs has obvious difference. We tested the ratio-difference rate of Hamming distance and total bit length between two different PUFs of 40 SRAM PUFs used previously.

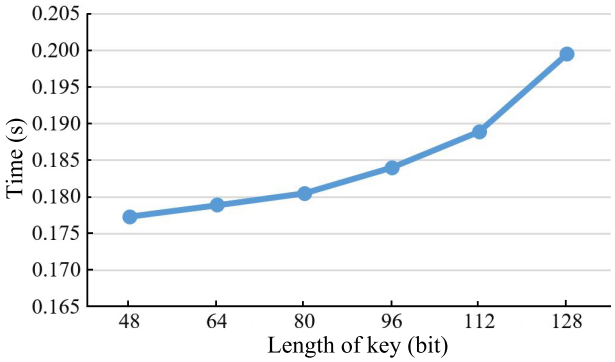


Fig. 4 Time for generating keys.

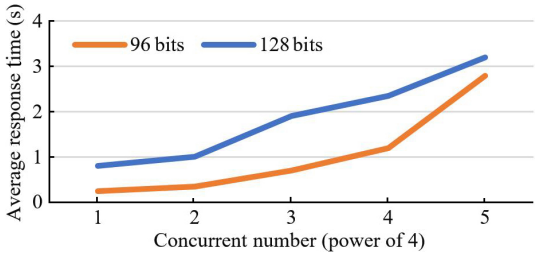


Fig. 5 Complexity comparison in concurrent number.

We obtained the maximum and minimum difference rates of response between every two PUFs. As shown in Fig. 6, the average difference rate between 40 SRAM PUFs is 0.4774 and the standard deviation is 1.1%. The average value is extremely close to the ideal value of 0.500. Thus, we can consider SRAM PUF as unique.

7 Conclusion

In this paper, we propose a new terminal authentication scheme based on PUF, which applies double identity verification using device feature information and PUF unique CRP. We adapt an identity-based encryption scheme that uses quadratic residues to encrypt the PUF authentication process. The scheme that we propose is simple, lightweight, and flexible and is suitable for IoT. Considering the experiment results, we believe that most of the current terminal devices can use PUF technology to perform authentication. These devices also have secret key generation and conversation functions.

Acknowledgment

This work was supported in part by the National Basic Research Program of China (973 Program) (No. 2014CB340600) and in part by the Wuhan Frontier Program of Application Foundation (No. 2018010401011295).

References

[1] M. A. Feki, F. Kawsar, M. Boussard, and L. Trappeniers, The internet of things: The next technological revolution, *Computer*, vol. 46, no. 2, pp. 24&25, 2013.

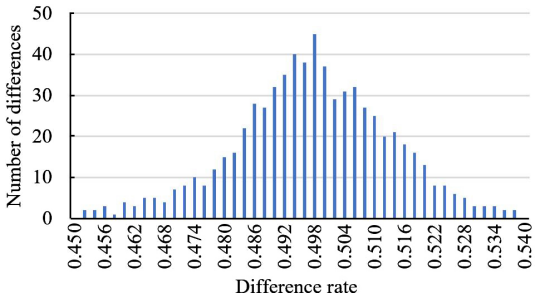


Fig. 6 Difference rate distribution in SRAM PUF.

[2] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, Internet of things (IoT) security: Current status, challenges and prospective measures, in *2015 10th Int. Conf. Internet Technology and Secured Transactions*, London, UK, 2015, pp. 336–341.

[3] K. Zhao and L. Ge, A survey on the internet of things security, in *2013 9th Int. Conf. Computational Intelligence and Security*, Leshan, China, 2013, pp. 663–667.

[4] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, Design and implementation of PUF-based "unclonable" RFID ICs for anti-counterfeiting and security applications, in *2008 IEEE Int. Conf. RFID*, Las Vegas, NV, USA, 2008, pp. 58–64.

[5] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, Physical one-way functions, *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[6] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching, *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 37–49, 2014.

[7] U. Rührmair, J. Sölter, F. Sehnke, X. L. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, PUF modeling attacks on simulated and silicon data, *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[8] J. Tobisch and G. T. Becker, On the scaling of machine learning attacks on PUFs with application to noise bifurcation, in *Proc. Int. Workshop on Radio Frequency Identification*, New York, NY, USA, 2015, pp. 17–31.

[9] F. Ganji, S. Tajik, and J. P. Seifert, Why attackers win: On the learnability of XOR arbiter PUFs, in *Int. Conf. Trust and Trustworthy Computing*, Heraklion, Greece, 2015, pp. 22–39.

[10] G. E. Suh and S. Devadas, Physical unclonable functions for device authentication and secret key generation, in *Proc. 44th Annu. Design Automation Conf.*, San Diego, CA, USA, 2007, pp. 9–14.

[11] H. Akhundov, Design & development of public-key based authentication architecture for IoT devices using PUF, Master dissertation, Delft University of Technology, Delft, Netherlands, 2017.

[12] J. C. Choon and J. H. Cheon, An identity-based signature from gap Diffie-Hellman groups, in *Int. Workshop on Public Key Cryptography*, Miami, FL, USA, 2003, pp. 18–30.

[13] P. Koeberl, J. T. Li, A. Rajan, C. Vishik, and W. Wu, A practical device authentication scheme using SRAM PUFs, in *Int. Conf. Trust and Trustworthy Computing*, Pittsburgh, PA, USA, 2011, pp. 63–77.

[14] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, A PUF-based secure communication protocol for IoT, *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 3, p. 67, 2017.

[15] U. Rührmair, SIMPL systems as a keyless cryptographic and security primitive, in *Cryptography and Security*: *From Theory to Applications*, D. Naccache, ed. Berlin, Heidelberg: Springer, 2012, pp. 329–354.

[16] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, Controlled physical random functions, in *Proc. 18th Annu. Computer Security Applications Conf.*, Las Vegas, NV, USA, 2002, pp. 149–160.

[17] E. Öztextürk, G. Hammouri, and B. Sunar, Towards robust low cost authentication for pervasive devices, in *2008 6th Annu. IEEE Int. Conf. Pervasive Computing and Communications*, Hong Kong, China, 2008, pp. 170–178.

[18] S. Katzenbeisser, Ü. Kocabaş, V. Van Der Leest, A. R. Sadeghi, G. J. Schrijen, and C. Wachsmann, Recyclable PUFs: Logically reconfigurable PUFs, *Journal of Cryptographic Engineering*, vol. 1, no. 3, pp. 177–186, 2011.

[19] A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A. R. Sadeghi, I. Verbauwhede, and C. Wachsmann, Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs, in *Int. Conf. Financial Cryptography and Data Security*, Kralendijk, Sint Eustatius and Saba, 2012, pp. 374–389.

[20] M. Majzoobi, M. Rostami, F. Koushanfar, D. S. Wallach, and S. Devadas, Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching, in *2012 IEEE Symp. Security and Privacy Workshops*, San Francisco, CA, USA, 2012, pp. 33–44.

[21] Ü. Kocabaş, A. Peter, S. Katzenbeisser, and A. R. Sadeghi, Converse PUF-based authentication, in *Int. Conf. Trust and Trustworthy Computing*, Vienna, Austria, 2012, pp. 142–158.

[22] D. Moriyama, S. Matsuo, and M. Yung, PUF-based RFID authentication secure and private under memory leakage, Cryptology ePrint Archive: Report 2013/712, 2013.

[23] A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung, End-to-end design of a PUF-based privacy preserving authentication protocol, in *Int. Workshop on Cryptographic Hardware and Embedded Systems*, Saint Malo, France, 2015, pp. 556–576.

[24] J. Delvaux, D. W. Gu, D. Schellekens, and I. Verbauwhede, Secure lightweight entity authentication with strong PUFs: Mission impossible? in *Int. Workshop on Cryptographic Hardware and Embedded Systems*, Busan, Korea, 2014, pp. 451–475.

[25] G. Avoine, X. Carpent, and J. Hernandez-Castro, Pitfalls in ultralightweight authentication protocol designs, *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2317–2332, 2016.

[26] X. Tan, J. L. Zhang, Y. J. Zhang, Z. Qin, Y. Ding, and X. W. Wang, A PUF-based and cloud-assisted lightweight authentication for multi-hop body area network, *Tsinghua Science and Technology*, doi: 10.26599/TST.2019.9010048.

[27] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, Silicon physical random functions, in *Proc. 9th ACM Conf. Computer and Communications Security*, Washington, DC, USA, 2002, pp. 148–160.

[28] Intrinsic ID, www. intrinsic-id.com/products/2010/, 2019.

[29] Y. Dodis, L. Reyzin, and A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, in *Int. Conf. Theory and Applications of Cryptographic Techniques*, Interlaken, Switzerland, 2004, pp. 523–540.

[30] A. R. Sadeghi, I. Visconti, and C. Wachsmann, PUF-

enhanced RFID security and privacy, *ResearchGate*, doi: 10.2200/S00550ED1VO1Y201311SPT007.

[31] A. Shamir, Identity-based cryptosystems and signature schemes, in *Workshop on the Theory and Application of Cryptographic Techniques*, Santa Barbara, CA, USA, 1984, pp. 47–53.

[32] D. Boneh and M. Franklin, Identity-based encryption from the weil pairing, in *Annu. Int. Cryptology Conf.*, Santa Barbara, CA, USA, 2001, pp. 213–229.

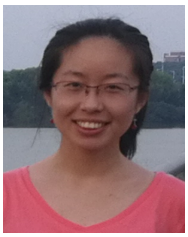
[33] C. Cocks, An identity based encryption scheme based on quadratic residues, in *IMA Int. Conf. Cryptography and Coding*, Cirencester, UK, 2001, pp. 360–363.

[34] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis, Fuzzy extractors for continuous distributions, in *Proc. 2nd ACM Symp. Information, Computer and Communications Security*, Singapore, 2007, pp. 353–355.

[35] A. Arakala, J. Jeffers, and K. J. Horadam, Fuzzy extractors for minutiae-based fingerprint authentication, in *Int. Conf. Biometrics*, Seoul, Korea, 2007, pp. 760–769.

Bo Zhao received the PhD degree from Wuhan University in 2006. He is currently the vice president of the School of Cyber Science and Engineering, Wuhan University. His research interests include information system security, trusted computing, embedded system, and cloud computing security.

Pengyuan Zhao is currently pursuing the PhD degree in information security from Wuhan University, China. His research interests include big data, embedded systems, and IoT security. He is also a teacher in Hebei University and has been conducting research as a lecturer in the School of Cyber Security and Computer, Hebei University since 2008.



Peiru Fan is currently pursuing the PhD degree in Wuhan University. Her current interest lies in the area of virtualization, cloud computing, and security.