Information Sciences 503 (2019) 129–147

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A double PUF-based RFID identity authentication protocol in service-centric internet of things environments

Wei Liang[a], Songyou Xie[b], Jing Long[c], Kuan-Ching Li[d,*], Dafang Zhang[e], Keqin Li[f]

[a] School of Opto-Electronic and Communication Engineering, Xiamen University of Technology, China
[b] School of Computer Science and Engineering, Hunan University of Science and Technology, China
[c] College of Information Science and Engineering, Hunan Normal University, China
[d] Department of Computer Science and Information Engineering, Providence University, Taiwan
[e] College of Computer Science and Electronic Engineering, Hunan University, China
[f] Department of Computer Science, State University of New York New Paltz, USA

ARTICLE INFO

Article history:
Received 24 February 2019
Revised 6 June 2019
Accepted 16 June 2019
Available online 17 June 2019

Keywords:
Authentication
Integrated circuit (IC)
Internet of things (IoT)
Physical unclonable function (PUF)
Radio frequency identification (RFID)

ABSTRACT

The rapid development of sensing, automation and communication technologies has led to the proliferation of the Internet of Things (IoT), providing recognized promising opportunities to build complex industrial systems and applications, leveraging the growing ubiquity of Radio Frequency Identification (RFID) and wireless sensor devices. With the pervasiveness of the interconnected systems encompassed with an ever-growing number of RFID-enabled devices being deployed, RFID security is an issue of high concern. As lightweight security encryption primitive, Physical Unclonable Function (PUF) is used to protect the information security of low-cost devices. Unfortunately, they are vulnerable to attacks, so countermeasures should be employed in the design. Aimed at low-cost and security of connected IoT devices to satisfy various security requirements of RFID technology in IoT, a two-stage multiple-choice arbiter (TSMCA)-based PUF in RFID systems is proposed, referred as TSMCA PUF. It is aimed at the design of a double PUF-based bidirectional RFID identity authentication protocol that permits the realization of bidirectional authentication between a server and a tag for the IC authentication in low-cost RFID systems, where the exclusive-OR (XOR) and character padding operations are adopted to generate the response of the PUF; the string-matching method is used in authentication, without exposing the PUF response to the verifier. Evaluation and analysis show that the advantages over conventional schemes include reduced area, higher randomness, and high stability, yet experimental results depict that the proposed protocol is promising resilient against attacks and practical for the deployment of low-cost hardware.

1. Introduction

Radio Frequency IDentification (RFID) is a wireless technology capable of automatic and unambiguous identification yet essential to the development of the Internet of Things (IoT), making incredible strides for the connected world. RFID systems

[*] Corresponding author.
E-mail addresses: wliang@xmut.edu.cn (W. Liang), syou720@163.com (S. Xie), jlong@hunnu.edu.cn (J. Long), kuancli@pu.edu.tw (K.-C. Li), dfzhang@hnu.edu.cn (D. Zhang), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.ins.2019.06.047
0020-0255/© 2019 Elsevier Inc. All rights reserved.

consist of a tag, a reader, and a backend server, where the tag has a unique identity and can enter the working area of the reader for read/write operations. As depicted in Fig. 1, the tag encrypts the identification information and sends it to the reader, which authenticates the received data and the tag ID. An RFID system is categorized either as passive or active [12], depending on whether the system utilizes tags with an internal power source or powered by energy transmitted from RFID readers to broadcast signals continuously. The low-cost RFID devices underline the challenges of securing systems [21], since traditional encryption primitives increase the cost of such devices, as the power consumption of traditional encryption primitives is excessive high [28]. While the challenges in designing low-cost secured RFID systems include limited capacity and computation capability of the device [5], realizing a safe and inexpensive protection technique to ensure the security of user information is critical [20].

The Physical Unclonable Function (PUF) is a physical entity embodied in a physical structure, a hardware circuit that depends on chip characteristics and profound submicron variations during manufacturing to uniquely characterize each chip [19]. It is unclonable and unpredictable, as they are unable to simulated or copied, providing security assurance for chips with low cost and high security. PUF can also be inserted into integrated chips at minimal cost to recognize RFID modules [23]. The intrinsic delay of PUF and the number of logic gates are unique. As a consequence, responses of two PUFs are different for the same challenges. As PUFs are unpredictable, attackers that attempt to extract responses will encounter additional difficulties compared to those found in traditional cryptosystems. PUF has been used in intellectual property (IP) protection [8] and secure authentication [11], among several others.

Embedded system technologies make the development of devices possible at low-cost, low-power, and small-sized [29]. However, PUF-based RFID identity authentication protocols require an error correction module to stabilize PUF outputs, and the use of many hash operations in some of these protocols (e.g., [38] and [13]) results in additional manufacturing costs and thus economically unviable. Thereafter, the proposed research addresses issues of cost restriction in state-of-the-art RFIDs, and the contributions are listed next.

- A two-stage multiple-choice arbiter (TSMCA) based PUF structure is proposed, that is more stable and reconfigurable than traditional arbiter PUF structures (e.g., four-stage arbiter PUF). Moreover, a double PUF-based bidirectional RFID identity authentication protocol is proposed for low-cost hardware circuits, that consists of a two-stage PUF, a four-way selector, and a selection module,
- A novel PUF string encryption algorithm is proposed, utilizing the latest string-matching authentication mechanism for high security and efficiency,
- The double PUF structure to effectively save RFID system costs and avoiding the store of many incentive response pairs are utilized. The tag does not need non-volatile memory (NVM) to store registration information and no hash operations are performed,
- Ban logic analysis, Proverif tool simulation, and security analyses of the protocol are performed, showing that the proposed protocol can effectively resist to many current mainstream attacks,
- A FPGA version of PUF is implemented to validate the proposed approach and compared with other well-known protocols/filters.

The remaining of this paper is organized as follows. Section 2 presents the related work, Section 3 introduces the preliminaries of PUFs and PUF protocols, and the proposed RFID bidirectional identity authentication protocol based on double

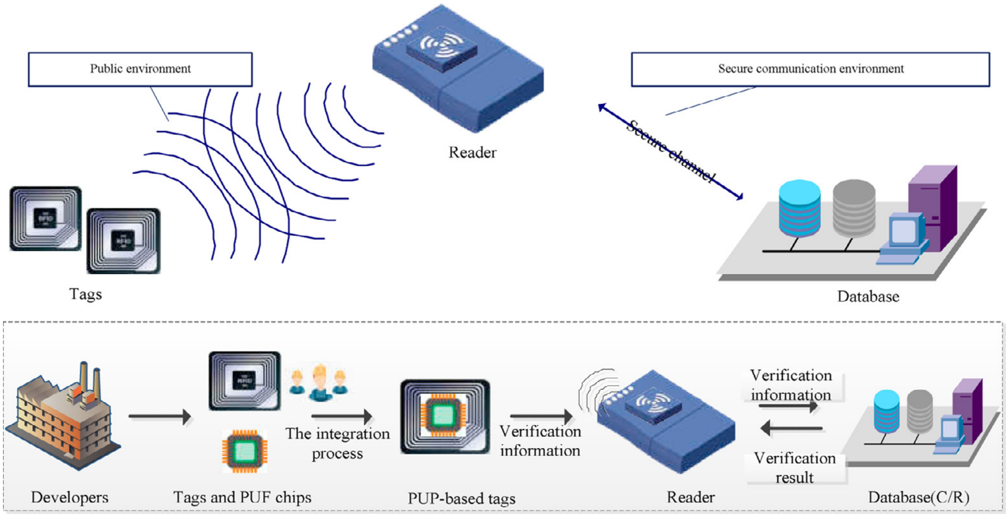


Fig. 1. An example RFID authentication system.

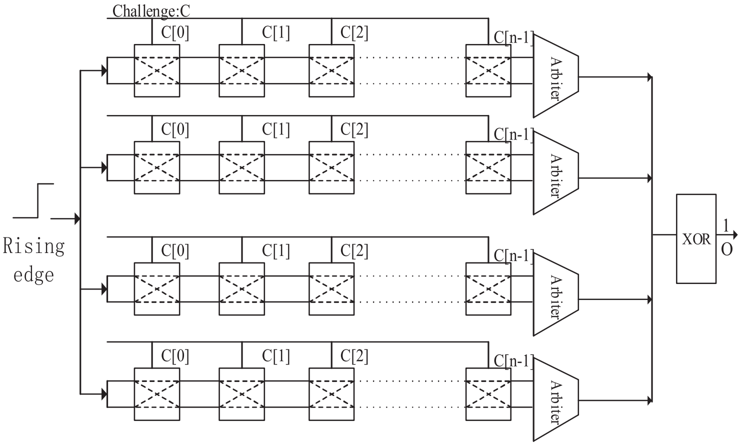


Fig. 2. Structure of four-stage arbiter PUF.

PUF circuits. Section 4 discusses the analyses of the proposed protocol's security and performance evaluation, and finally, Section 5 concludes this paper and presents future directions.

2. Related work

With the widespread use of portable devices, the architecture and structure of systems are even more complex and sophisticated [27]. Many security solutions have been proposed to ensure the security of RFID systems, such as cryptographic approach based on physical hardware (e.g., the fire extinguishing method [26]), cryptography-based secure protocol, and bitwise operation-based lightweight authentication protocols (e.g., LMAP+ protocol and UMAP protocol). Examples of RFID authentication protocols include hash function protocol [26], HB protocol [1], and improved LMAP+ protocol. For instance, a hash function is an encryption algorithm with high hardware cost and used in MD4, MD5, and SHA-256 operations, approximately 7350–10,868 logical gates [24] are required for such an implementation; to implement the AES encryption algorithm [2], 3400 logical gates are needed. That is, about 2000 logic gates in the tag of a typical lightweight RFID system are used for security, increasing the hardware overhead when manufacturing such RFID systems. In such a design, RFID secure mode depends on binary secret keys and encryption algorithms, and the keys are always stored in NVM despite its vulnerability to side-channel attacks [34] and attackers may capture the keys. Most RFID devices are executed in complex environments, they have ultra-small areas and ultra-low power overhead due to resource limitation. Since traditional cryptology-based RFID protocols cannot work with these requirements [15], PUF has widely been considered as a new chip encryption method due to its low implementation cost and high security [26,16,1,24,2,34,15].

PUF utilizes manufacturing variances to generate communication, and this randomness involves transmission delay, hardware structure, and sensitivity to temperature [7]. For instance, a four-stage arbiter PUF consists of four single-arbiter PUFs and one XOR unit, as depicted in Fig. 2. Similar to other PUFs, PUF communication is unique and produces an unpredictable pair of challenge-response messages in RFID devices [24]. The outputs of four single-arbitrators are the input to the XOR module to form a bit output. This structure improves the security and unpredictability of PUF [10] while leading to higher manufacturing costs.

Suh et al. utilized PUF in a RFID system [33], where each response of the PUF tag is stored in the backend database yet drew during authentication. If the new response generated from the tag is the same as that stored in the database, then one-way authentication of the tag valid to the reader is achieved. Though, the method proposed in [13] is not considered in this research due to the manufacturing cost as well as the vulnerability to Machine Learning (ML) attacks [35]. In recent years, such a kind of attacks has severe impact on PUF, since the delay parameters of PUF mainly determine the challenge-response behavior of the arbiter PUF. Machine Learning (ML) algorithms collect and analyze the challenge-response pairs (CRPs) of the arbiter PUF and randomly generates n PUF models with different delay parameters. Furthermore, these models are trained by collected CRPs, so the PUF model with the challenge response behavior closest to the original arbiter PUF is selected among n PUF models. Additionally, the delay parameters of the selected PUF model are randomly mutated so new PUF models are generated, and continuously repeated as many times as needed until the final trained PUF model produces a similar response as the original arbiter PUF.

Ruhrmair et al. proposed a method to model PUF by using ES- and LR-based Machine Learning algorithms [32], where the attackers may guess the internal linear delay vector after capturing many CRPs, so the simulated PUF is realized via software modeling [3]. Majzoobi et al. proposed the authentication verification model and used a PUF model for system security [25] in the following way: as the trusted manufacturer can access the PUF structure in the prover through the I/O pins of the chip and retrieve enough CRPs to train the PUF model, it is successfully built with the I/O pin permanently destroyed, and no longer can access the PUF via the I/O pin. Next, the trusted manufacturer sends the PUF model to the verifier, which stores a PUF model for each prover.

The original PUF and PUF models have similar challenging response behavior. In [25], it used an authentication verification model and proposed a slender PUF identity authentication protocol. In the authentication phase of the protocol, the prover and the verifier jointly generate the challenge information of the PUF. With two response strings generated, the prover extracts the substring in the random position of the response string (the length of the substring is shorter than the response string). As the substring is sent to the verifier, it authenticates the received substring with the response information generated by the PUF model. In [31], Rostami et al. added the process of filling substrings with random bits [25], so that the length of the filled string is equal to the length of the original response string and have the filled string sent to the verifier. Verifier can extract substring information through response information generated by the PUF model and verify the identity of the prover through the substring information. The methods presented in [25] and [31] turn impossible the attacker to directly obtain the information of the substring, which improves the difficulty of ML attacks. However, Mukhopadhyay et al. successfully attacked the protocols [25] and [31] with methods designed using Machine Learning [22], proving that these protocols are not fully resistant to Machine Learning attacks.

To address this issue, Yu et al. improved the algorithm of PRNG, locked the relationship of the response information and set the maximum number of authentications *d* for each device [36]. This protocol achieves the goal of resisting to Machine Learning attacks by limiting the ability of attackers to collect the CRPs. Nevertheless, the protocol cannot resist replaying attacks, as the attacker can intercept and replay the identity information of each device, causing the device's authentication number *c* to be higher than the maximum authentication number *d*. Therefore, the device will no longer be able to access the server. In response to the problems abovementioned, Gope revoked the restrictions on the number of accesses to devices [9] and used the pseudo-identity *PID* method for authentication. In the registration phase, the registry assigns each tag a temporary identity (*TID*) and a set of pseudo-identities (*PID*). As the attacker destroys the authentication process of the protocol and causes an invalid *TID*, the tag randomly selects a PID_i for identity authentication. Nevertheless, the protocol is only applicable to tags with large storage capacity and rich computing resources; moreover, after the attacker exhausts the *PID* information inside the tag by intercepting and playing back the *TID* information, the device needs to be re-registered to be certified. Two critical issues, including ability against Machine Learning attacks and achieving lightweight authentication are mainly considered in this proposed approach.

3. Proposed identity authentication protocol

3.1. TSMCA PUF

The four-stage arbiter PUF is the most popular among the common PUFs. Unfortunately, approximately 40,000 CRPs are required to conduct ML attacks on a four-stage PUF [3]. To increase PUF security and stability and defend against ML attacks, a novel TSMCA PUF is proposed.

A secure PUF structure must satisfy the avalanche effect. That is, as one bit (or several bits) of the challenge changes, more than half bits of the response flip. An illustration of the TSMCA PUF structure with limited resources is shown in Fig. 3.

The PUF structure includes a four-way selector between the arbiter and the multiplexer, in which the selector chooses the path of the signal for the arbiter, and the response must be sent to the arbiter via the selector. The selector is controlled by five bits, so 24 paths are made available for selection each time, thus achieving high reconfigurability. Compared with the four-stage arbiter PUF, the proposed structure requires fewer hardware resources and a smaller area. The challenged order of both PUFs is opposite, and the attacker needs to record two challenge information at the same time, which can effectively increase the difficulties of the attacker attacking the PUF.

To achieve responsive stability, a selection module is added to the PUF structure to follow the rule of "the minority obeying the majority", where the input signal is constructed by a pseudo-random number generator (PRNG). Each signal is inputted five times repeatedly, and the result is transmitted to the arbiter. The output value with the highest count is used as

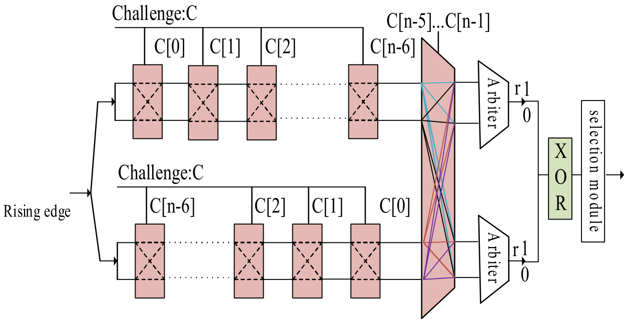


Fig. 3. Structure of a TSMCA PUF.

Table 1
Notation description.

Symbol	Definition
PRNG	Pseudo-random number generator
TRNG	True Random Number Generator
Seed	The seed of *PRNG*
PUF_mod	Legal PUF certification model
T_A	The random number generated by the server
T_B	The random number generated by the tag
$h(\cdot)$	Hash function
$\oplus$	XOR operator
$\|$	The operator to cascade two messages
c,c',c_2,c_2'	The challenge of PUF
R,R',R_2,R_2'	The response of PUF
L	Length of PUF response
Ind	A random index of the bit string
Sea	A function to find string information
match	A function to match strings
Pad	A function for bit-wise padding

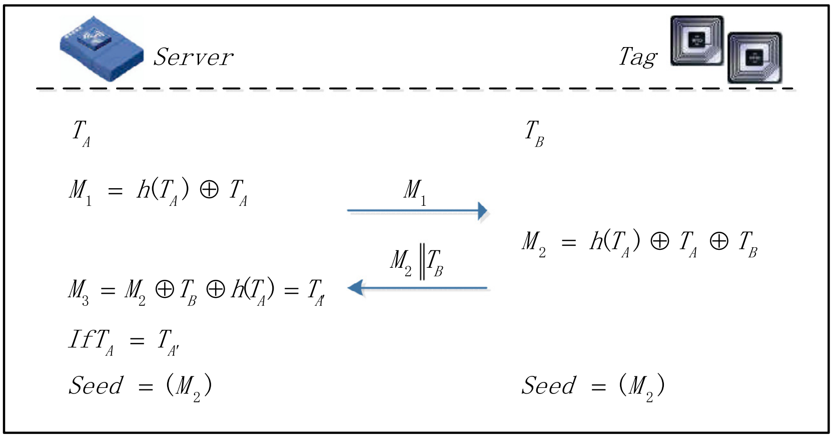

Server
Tag
T_A
T_B
$M_1 = h(T_A) \oplus T_A$
M_1
$M_2 = h(T_A) \oplus T_A \oplus T_B$
$M_2 \| T_B$
$M_3 = M_2 \oplus T_B \oplus h(T_A) = T_{A'}$
$If T_A = T_{A'}$
$Seed = (M_2)$
$Seed = (M_2)$

Fig. 4. Generation of the seed of the PRNG.

the output of the PUF. Compared with traditional error correction modules, the proposed structure requires fewer hardware resources and has excellent stability and reconfigurability, and thus suitable for a system with cost-restricted resources.

The double PUF structure is included in the proposed protocol. The tag contains an original PUF and the server that contains a PUF model revealed and shown that it can be built through Machine Learning. Moreover, this PUF model can be built within a short period, and its behavior is similar to that of the original PUF. Accordingly, the proposed protocol is adequate and effective for identity authentication. The original PUF is adopted in the TSMCA PUF instead of the four-stage arbiter PUF due to its proper response characteristics and suitability for lightweight tags.

3.2. Bidirectional PUF authentication protocol

In this protocol, the communication between the reader and the backend server is considered safe by default, as tags require only simple bit and logical operations to achieve secure authentication. It adds a hash function in the verifier to avoid man-in-middle attacks while altering the PUF and does not expose the response of PUF during the interaction between server and tag, so more secure than the substring matching method of the slender PUF protocol.

The proposed bidirectional authentication protocol includes two parts, namely (1) the server that verifies the tag and (2) the tag that verifies the server. The notations used for the protocol are defined in Table 1.

3.3. Production of PRNG seed

In the proposed protocol, PRNG is utilized to generate the challenge of the PUF, as the output of PRNG is determined by the seed. The procedure for generating the PRNG seed is shown in Fig. 4, where a hash function is adopted to the initial message in the server.

The server and the tag generate private random numbers T_A and T_B by using TRNG, with the server calculating M_1=$h(T_A)$⊕T_A then sending M_1 to the tag. Once having received M_1, the tag calculates M_2=M_1⊕T_B=$h(T_A)$⊕T_A⊕T_B and then

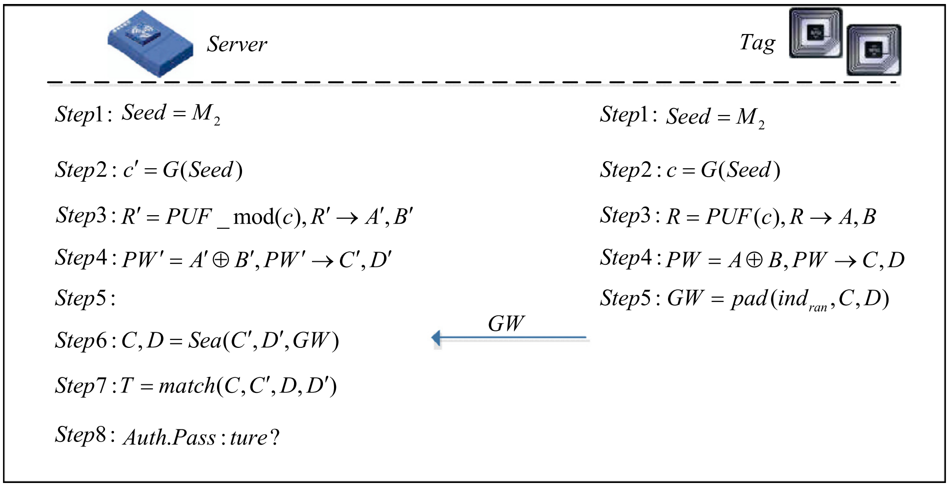

Server
Tag
$Step1: Seed = M_2$
$Step2: c' = G(Seed)$
$Step3: R' = PUF\_\mathrm{mod}(c), R' \rightarrow A', B'$
$Step4: PW' = A' \oplus B', PW' \rightarrow C', D'$
$Step5:$
$Step6: C, D = Sea(C', D', GW)$
$Step7: T = match(C, C', D, D')$
$Step8: Auth.Pass: ture?$
GW
$Step1: Seed = M_2$
$Step2: c = G(Seed)$
$Step3: R = PUF(c), R \rightarrow A, B$
$Step4: PW = A \oplus B, PW \rightarrow C, D$
$Step5: GW = pad(ind_{ran}, C, D)$

Fig. 5. The verification process of the server with *GW* from the tag.

sends back $M_2 \| T_B$ to the server. After received $M_2 \| T_B$, the server calculates $M_3{=}M_2{\oplus}T_B{\oplus}h(T_A){=}T_{A'}$. The result $T_{A'}$ is then compared with the T_A: if the equation $T_{A'}{=}T_A$ is valid, then M_2 is used as the seed of PRNG, denoted by $Seed{=}(M_2)$.

After the successful generation of PRNG seed, the steps of server and bidirectional tag authentication, as well as the string encryption and extraction process, are introduced in Section 3.4.

3.4. Server verifying tag

The process of verifying the tag by the server is shown in Fig. 5, as the PRNG outputs the challenge of PUF by using the generated PRNG seed, as per the discussion presented in the previous section. The detailed process of the server verifying the tag is as follows.

Steps 1 and 2: The first two steps involve seed M_2 and the output of PRNG as the challenge of PUF, denoted by $c'{=}G(Seed)$ and $c = G(Seed)$,

Step 3: Server and tag individually generate the response strings *R*' and *R* by applying challenges *c*' and *c* to the PUF model of server and PUF of the tag. The length of the response string is set to *L*. The response *R*'(*R*) is divided into the two substrings *A*' and *B*' (*A* and *B*),

Step 4: The XOR operation is performed on *A*' and *B*' (*A* and *B*) to obtain *PW*' (*PW*). Then *PW*'(*PW*) is divided into two substrings of *C*' and *D*' (*C* and *D*) with the same length,

Step 5: Strings *C* and *D* of the tag are padded by randomly bits. The *GW* with the padding represents a new string containing *C* and *D* and then sent to the server,

Step 6: Authentication information *C* and *D* are extracted by the server after receiving *GW* and by using its own *C*' and D'. When search *C*' and *D*' in *GW*, *C*' and *D*' are matched from the ind_0 and $ind_{L\text{-}1}$ positions of *GW* to find the substrings *C* and *D*, which have the minimum Hamming distances to *C*' and *D*',

Step 7: *C* and *D* are compared with *C*' and *D*',

Step 8: If the error bits in the comparison are less than the threshold ε, then the server has successfully verified the tag.

3.5. Tag verifying server

After completing the verification of the server to the tag, the tag needs to verify the server as the second part of the protocol. Fig. 6 details the steps of this procedure.

Step 1: Server and tag use the corresponding information *PW*, and the seed of the pseudo-random number generator is denoted by $Seed{=}\{PW\}$,

Step 2: The PRNG with *PW* is used to generate the challenges of *PUF_mod* and PUF,

Step 3: Server and tag individually generate the responses R_2' and R_2 by challenges c_2' and c_2. Server (Tag) divides the response R_2'(R_2) into two substrings A_2' and B_2' (A_2 and B_2),

Step 4: The XOR operation is performed on A_2' and B_2' (A_2 and B_2) to obtain PW_2' (PW_2), and server and tag divide the PW_2' (PW_2) into C_2' and D_2' and (C_2 and D_2),

Step 5: The substrings are padded to a specific length by actual needs. GW_2' denotes a new string, including padding, C_2' and D_2', and then sent to tag for verification,

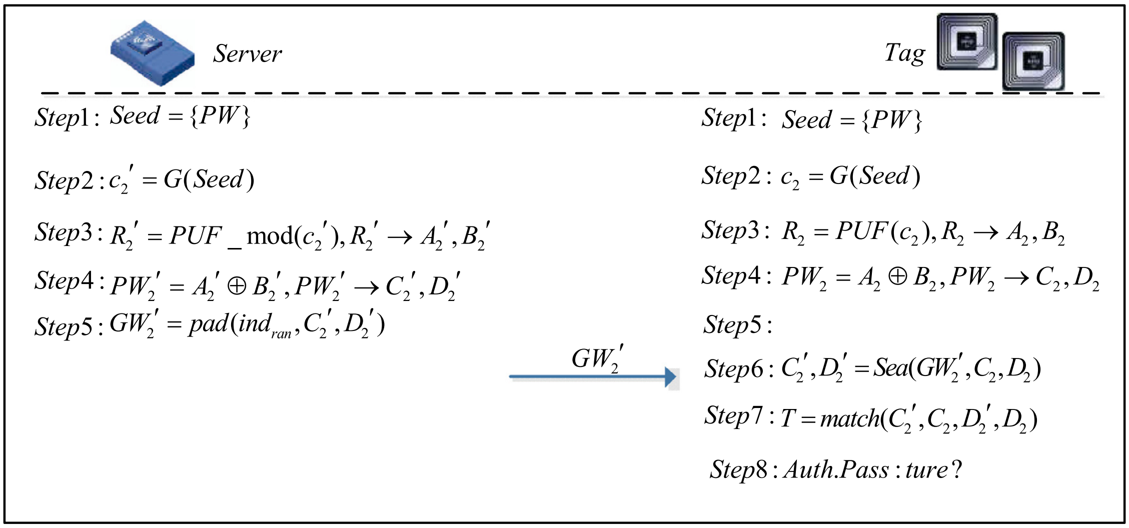

Server
Tag
$Step1: Seed = \{PW\}$
$Step2: c_2' = G(Seed)$
$Step3: R_2' = PUF\_mod(c_2'), R_2' \to A_2', B_2'$
$Step4: PW_2' = A_2' \oplus B_2', PW_2' \to C_2', D_2'$
$Step5: GW_2' = pad(ind_{ran}, C_2', D_2')$
GW_2'
$Step1: Seed = \{PW\}$
$Step2: c_2 = G(Seed)$
$Step3: R_2 = PUF(c_2), R_2 \to A_2, B_2$
$Step4: PW_2 = A_2 \oplus B_2, PW_2 \to C_2, D_2$
$Step5:$
$Step6: C_2', D_2' = Sea(GW_2', C_2, D_2)$
$Step7: T = match(C_2', C_2, D_2', D_2)$
$Step8: Auth.Pass: ture?$

Fig. 6. Verifying process of tag with GW_2' from server.

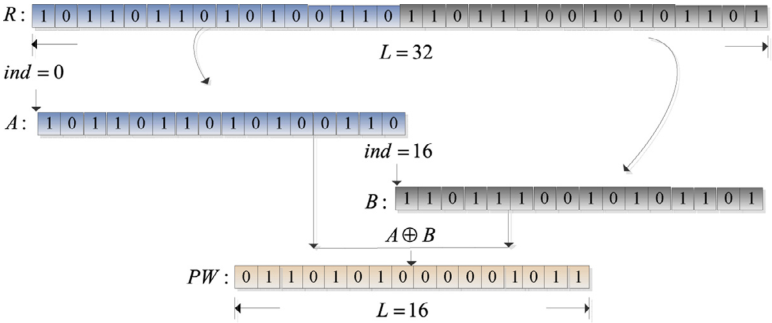

$R:$ 1 0 1 1 0 1 1 0 1 0 1 0 0 1 1 0 1 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1
$L=32$
$ind=0$
$A:$ 1 0 1 1 0 1 1 0 1 0 1 0 0 1 1 0
$ind=16$
$B:$ 1 1 0 1 1 1 0 0 1 0 1 0 1 1 0 1
$A \oplus B$
$PW:$ 0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 1
$L=16$

Fig. 7. Dichotomy XOR method for the string.

Step 6: The authentication information of C_2' and D_2' are extracted. The tag receives GW_2' and makes use of C_2 and D_2. Next, C_2 and D_2 are matched from the ind_0 and $ind_{L\text{-}1}$ positions of GW_2' to find the substrings C_2' and D_2', which have the minimum Hamming distances to C_2 and D_2,

Step 7: C_2 and D_2 are compared with C_2' and D_2',

Step 8: Tag verifies the server consecutively if the error bits in comparison are less than the threshold ε.

3.6. String encryption and extraction

In the proposed authentication protocol, dichotomy XOR and padding methods are used to secure the response, so the verifier must extract and authenticate the encrypted information by using the correct strings. The processes are demonstrated in the following sections.

3.6.1. Dichotomy XOR of string

The PUF_mod and PUF generate two responses, *R*' (R_2') and *R* (R_2), respectively. The XOR operation further transforms these responses, as shown in Fig. 7 that the assumed length of the response in the prover is $L = 32$, so the substring of *R* is selected from Position *ind*=0 to *ind*=15 (denoted by *A*) and the substring from Position *ind*=16 to *ind*=32 is denoted by *B*. The operation *A* XOR *B* is the method to generate *PW*.

This procedure requires no extra hash function since only a simple XOR operation is used. Moreover, information PW does not expose the response behavior of the PUF, as the response behavior of PUF is undetermined, and the attackers cannot model PUF through traditional machine learning attacks. Hence, this approach performs better in terms of security and lighter-weighted properties than the traditional method.

3.6.2. Character padding

This section describes the padding procedure, where the prover divides the *PW* into two substrings *C* and *D*, cascaded next with some random paddings to form the new string *GW*. As depicted in Fig. 8, the authentication information *C* and

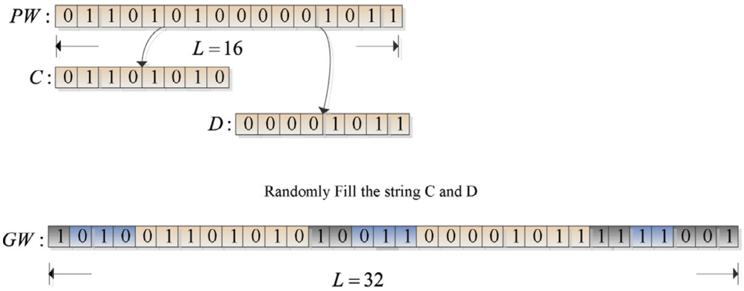

PW : 0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 1
L = 16
C : 0 1 1 0 1 0 1 0
D : 0 0 0 0 1 0 1 1
Randomly Fill the string C and D
GW : 1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1
L = 32

Fig. 8. Random padding method.

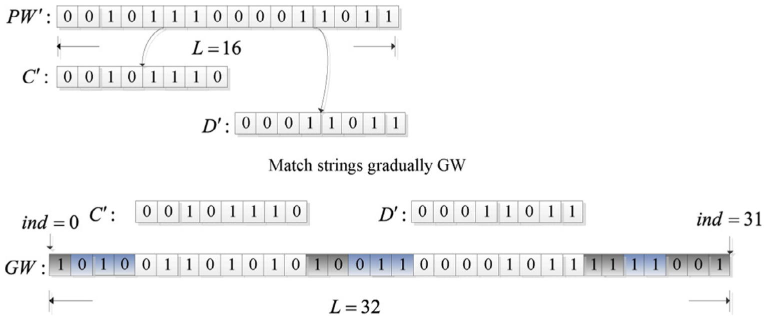

PW′ : 0 0 1 0 1 1 1 0 0 0 0 1 1 0 1 1
L = 16
C′ : 0 0 1 0 1 1 1 0
D′ : 0 0 0 1 1 0 1 1
Match strings gradually GW
ind = 0
C′ : 0 0 1 0 1 1 1 0
D′ : 0 0 0 1 1 0 1 1
ind = 31
GW : 1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1
L = 32

Fig. 9. Extraction of the authentication information.

D are hidden in *GW* such that the attacker cannot directly capture *C* and *D*. The positions of substrings *C* and *D* in *GW* are random yet the length of *GW* is not determined, being adjusted on demand. Though, the extraction of a large *GW* is highly time-consuming.

3.6.3. Extraction of the authentication information

After received *GW*, the server utilizes *C*' and *D*' to find the real *C* and *D* in *GW*.

As shown in Fig. 9, the server divides *PW*' into *C*' and *D*' to perform a match with *GW* and find *C* and *D*. By using the extracted *C* and *D*, the server can individually compare *C* and *D* with *C*' and *D*'. If the comparison results have fewer error bits than the threshold set ε in the system, then the tag can be authenticated.

4. Protocol analysis

4.1. BAN logic analysis

4.1.1. Symbols and rules

BAN logic uses a highly extensive protocol security analysis method, that is widely used in verifying the correctness of key agreement authentication protocols. This protocol is logical and secure only if it meets certain specific objectives, whose symbol description is shown in Table 2. *P* and *Q* represent two principals in the protocol, *X* the information, and *K* the encryption key.

The BAN logic rule is divided into two parts that represent the upper and lower parts of the expression, respectively known as condition and conclusion. Specific rules are strictly followed when performing BAN logic analysis as follows.

Rule 1 (Message-meaning rule): $\frac{P|\equiv P\overset{K}{\leftrightarrow}Q,P\triangleleft\{X\}_K}{P|\equiv Q\sim X}$. If *P* believes that it shares key *K* with *Q*, and *P* receives encrypted information $\{X\}_K$, then *P* believes that *Q* once sent message *X*,

Table 2
Ban symbols.

Symbols	Instructions	Symbols	Instructions
$P\triangleleft X$	P receives X	$P|\equiv X$	*P* believes information *X*
$P|\sim X$	P sent message X	$P|\equiv\#X$	*P* believes *X* is new
$P\overset{K}{\leftrightarrow}Q$	K is the shared key of P and Q	$P\Rightarrow X$	P can control X
(X, Y)	Information about X and Y	$(X,Y)_K$	(X,Y) By K-encryption

Table 3
Protocol idealization.

Messages	Instructions
1.$S{\rightarrow}T{:}h(T_A){\oplus}T_A$	S sends message $h(T_A){\oplus}T_A$ to T.
12.$T{\rightarrow}S{:}h(TA){\oplus}TA{\oplus}TB{,}TB$	T sends a message $h(TA){\oplus}TA{\oplus}TB{,}TB$ to S.
3.$T{\rightarrow}S{:}\{GW\}M2$	T sends a message $\{GW\}M2$ to S.
4.$S{\rightarrow}T{:}\{GW2'\}PW$	S sends a message $\{GW2'\}PW$ to T.

Rule 2 (Nonce-verification rule): $\frac{P|{\equiv}\#(X),P|{\equiv}Q|{\sim}X}{P|{\equiv}Q|{\equiv}X}$. If P believes that X is new and believes that Q once sent message X, then P believes that Q believes X,

Rule 3 (Jurisdiction rule): $\frac{P|{\equiv}Q{\Rightarrow}X,P|{\equiv}Q|{\equiv}X}{P|{\equiv}X}$. If P believes that Q can control X and believes that Q believes X, then P believes X,

Rule 4 (Freshness rule): $\frac{P|{\equiv}\#(X)}{P|{\equiv}\#(X,Y)}$. If P believes that X is new, P believes that (X, Y) is new.

In this research, two servers and tags of the protocol are denoted by S and T. Assuming that the parameters are correct, then the proposed authentication protocol should meet the following goals:

(a) $S| \equiv h(h(T_A) \oplus T_A \oplus T_B$ (S believes message $h(h(T_A) \oplus T_A \oplus T_B)$
(b) $\mathrm{S}| \equiv \{GW\}_{M_2}$ (S believes the message$\{GW\}_{M_2}$)
(c) $\mathrm{T}| \equiv \{GW'_2\}_{PW}$ (T believes the message$\{GW'_2\}_{PW}$)

4.1.2. Idealization

The protocol is inserted into the BAN logic for verification. First, the protocol is converted into a BAN logic symbol according to the message passing sequence, as shown in Table 3.

Messages 3 and 4 are encrypted by $M_2(PW)$ and sent by the tag to the server. $GW(GW_2$') depends on the challenge of PUF, and $M_2(PW)$ is a key to generate the PUF challenge. Before performing the BAN logic analysis, we also make some assumptions based on the state of the protocol, as shown in Table 4.

A1 implies that S believes $h(T_A){\oplus}T_A$ is new, considering that T_A is a random number generated by S, and it can be guaranteed to be new. Next, A2 indicates that S believes that T can control the information $h(T_A){\oplus}T_A{\oplus}T_B$, so A2 is established as the information is determined by T's random number T_B. A3 indicates that S believes the shared key $h(T_A){\oplus}T_A$ with T, this piece of information is determined by the random number T_A generated by S and the corresponding assumption is established. Later, A4 and A5 denote that S(T) believe the shared key $M_2(PW)$ with T(S), and thus, $M_2(PW)$ is obtained after the successful authentication of S(T), so A4 and A5 are also established. A6 and A7 indicate that S(T) believes that T(S) can control message $GW(GW_2$'). Two assumptions are established as $GW(GW_2$') is determined by the PUF structure of S(T). Finally, A8 and A9 are also valid since $M_2(PW)$ is calculated by S(T).

4.1.3. Proof

The BAN logic in this paper demonstrates that the process must achieve the expected goals based on logic rules (Section 4.1.1), idealization (Table 3), and initial assumption (Table 4). The process is described as follows.

1. With rule 4 and assumption A1, we have

$$S|{\equiv}\#h(T_A) \oplus T_A \oplus T_B \tag{4.1}$$

2. By using rule 1, Eq. (4.1), and assumption A3, we have

$$S|{\equiv}\, T| \sim h(T_A) \oplus T_A \oplus T_B \tag{4.2}$$

Table 4
The initial assumption of the protocol.

Assumption	Instructions
A1:$S\vert{\equiv}\#h(T_A){\oplus}T_A$	S believes $h(T_A){\oplus}T_A$ is new
A2:$\mathrm{S}\vert \equiv \mathrm{T} \Rightarrow \mathrm{h}(\mathrm{T_A}) \oplus \mathrm{T_A} \oplus \mathrm{T_B}$	S believes that T can control this information
A3:$\mathrm{S}\vert \equiv \overset{\mathrm{h(T_A)\oplus T_A}}{\leftrightarrow} \mathrm{T}$	S believes that S and T share information
A4:$\mathrm{S}\vert \equiv \mathrm{S} \overset{\mathrm{M_2}}{\leftrightarrow} \mathrm{T}$	S believes that S and T share M_2
A5:$\mathrm{T}\vert \equiv \mathrm{S} \overset{\mathrm{PW}}{\leftrightarrow} \mathrm{T}$	T believes that S and T share PW
A6:$\mathrm{S}\vert \equiv \mathrm{T} \Rightarrow \{\mathrm{GW}\}_{\mathrm{M_2}}$	S believes that T can control $\{GW\}_{\mathrm{M2}}$
A7:$\mathrm{T}\vert \equiv \mathrm{S} \Rightarrow \{\mathrm{GW'_2}\}_{\mathrm{PW}}$	T believes that S can control $\{GW_2’\}_{PW}$
A8:$S\vert{\equiv}\#M_2$	S believes message $M_2$is new
A9:$T\vert{\equiv}\#PW$	T believes message PW is new

3. With rule 2, formula (4.1), and formula (4.2), we have

$$S|\equiv T|\equiv h(T_A)\oplus T_A\oplus T_B \tag{4.3}$$

4. By rule 3, assumption A2, and formula (4.3), we have

$$S|\equiv h(T_A)\oplus T_A\oplus T_B \tag{4.4}$$

By adopting the four steps, we can prove that the goal (a) can be achieved. The proof process of goal (b) is as follows.

1. Eq. (4.5) is established by rule 1, message 3, and assumption A4.

$$S|\equiv T\sim\{GW\}_{M2} \tag{4.5}$$

2. According to rule 4 and assumption A8, we can obtain

$$S|\equiv\#\{GW\}_{M2} \tag{4.6}$$

3. Given rule 2, formula (4.5), and formula (4.6), we have

$$S|\equiv T\equiv\{GW\}_{M2} \tag{4.7}$$

4. Given rule 3, assumption A6, and formula (4.7), we can obtain

$$S|\equiv\{GW\}_{M2} \tag{4.8}$$

Goal (b) can be proven with the above four steps, and the proof of goal (c) is similar to that of goal (b) as follows.

1. Formula (4.9) is established according to rule 1, message 4, and assumption A5.

$$T|\equiv S|\sim\{GW_2'\}_{PW} \tag{4.9}$$

2. Given rule 4 and assumption A9, to obtain

$$T|\equiv\#\left\{GW'_2\right\}_{PW} \tag{4.10}$$

3. Given rule 2 and formulas (4.9) and (4.10), we have

$$T|\equiv S|\equiv\left\{GW'_2\right\}_{PW} \tag{4.11}$$

4. By rule 3, assumption A7, and formula (4.11), we have

$$T|\equiv\left\{GW'_2\right\}_{PW} \tag{4.12}$$

The Ban logic proof process ends with all objectives validated and demonstrates that the proposed protocol conforms to the BAN logic. In ideal work environments, the proposed protocol can meet the logical requirements of the system.

4.2. Proverif simulation

The authentication part of the protocol has been simplified, as shown in Fig. 10. At this point, the PUF module is formalized as a *P* function, which is confidential and not obtainable by attacks. *P* functions and XOR operations are used in the simplified protocol for encryption and authentication. At the end of the protocol execution, the shared keys *sks* and *skt* are added. Moreover, the original protocol requires that the entity be authenticated appropriately under the premise of hiding the PUF delay attribute, as the PUF behavior cannot be predicted. Finally, the simplified protocol also meets the requirement and has a similar algorithm behavior to the original protocol and thus suitable for experimental analysis.

Proverif is an efficient and stable algorithm analytical tool that can infer concepts based on specific rules [4], widely used in the analysis of information that involves the confidentiality and security of protocols.

In the predefinition part of the algorithm shown in Fig. 11, *ch* is the public channel, *sch* the secret channel, *sks* and *skt* the session keys, P the P function, xor the XOR operation, con the symbol concatenation, and the equations are used to define the XOR algorithm. From the perspective of protocol security, the goal in the proposed research is to prove that four queries can be executed safely, where events ServerStart, Event TagStart, Event ServerAuth and Event TagAuth are the events of the protocol.

Fig. 12 shows the codes of the server and the tag, and the simplified protocol is formalized into the Proverif language. As shown in this figure, lines 1 to 3 illustrate the server and tag receive the information of M2 in the secure transmission channel sch, in the server code. Next, lines 4 to 13 code are Step5 to Step11 in Fig. 10, indicating that the server receives the information {TPW, U} from the tag, and starts to verify the identity of the tag. After completed and succeeded the verification, {SGW, V} is calculated and sent to the tag. Finally, the server calculates the session key sks. In the tag code, the lines 4 to 9 are Step1 to Step4 in Fig. 10, indicating that the tag sends the {TPW, U} to the server for authentication. Next, lines 10 to 16 are Step 10 to Step 15 in Fig. 10, indicating that the tag receives the information {SGW, V} from the server, starts authenticating the identity of the server, and if successful the verification, the session key skt is calculated.

The four security queries were successfully executed, as shown in Fig. 13. The results in lines 4 and 8 show the security of sks and skt, proving that the sks and skt information are safe, yet the attacker cannot obtain the shared keys sks and skt. The results of lines 12 and 16 indicate that the proposed protocol successfully executed two-way authentication in Proverif, indicating that the protocol is secure in Proverif environments.

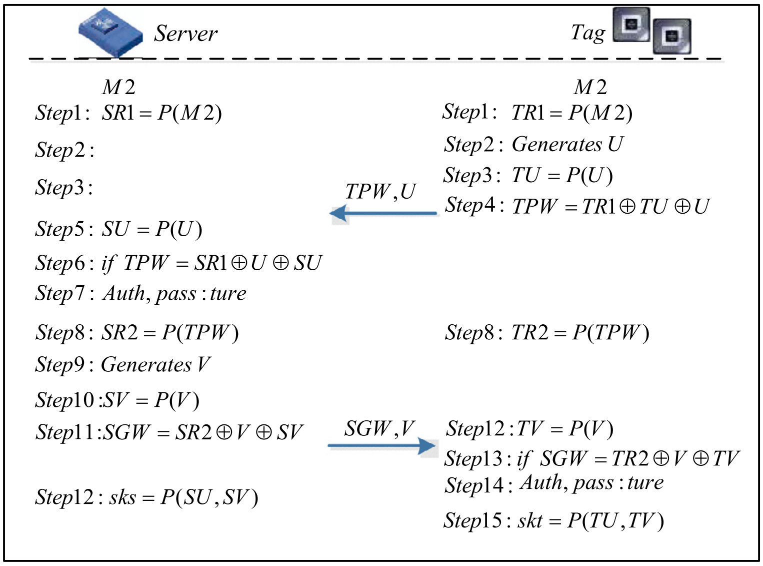
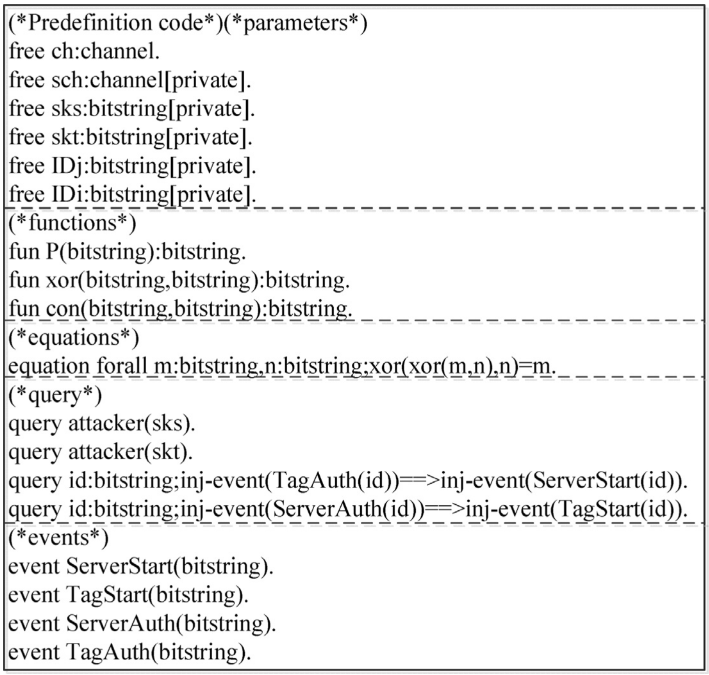


Fig. 10. Simplified protocol.



```
(*Predefinition code*)(*parameters*)
free ch:channel.
free sch:channel[private].
free sks:bitstring[private].
free skt:bitstring[private].
free IDj:bitstring[private].
free IDi:bitstring[private].
(*functions*)
fun P(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.
(*equations*)
equation forall m:bitstring,n:bitstring;xor(xor(m,n),n)=m.
(*query*)
query attacker(sks).
query attacker(skt).
query id:bitstring;inj-event(TagAuth(id))==>inj-event(ServerStart(id)).
query id:bitstring;inj-event(ServerAuth(id))==>inj-event(TagStart(id)).
(*events*)
event ServerStart(bitstring).
event TagStart(bitstring).
event ServerAuth(bitstring).
event TagAuth(bitstring).
```

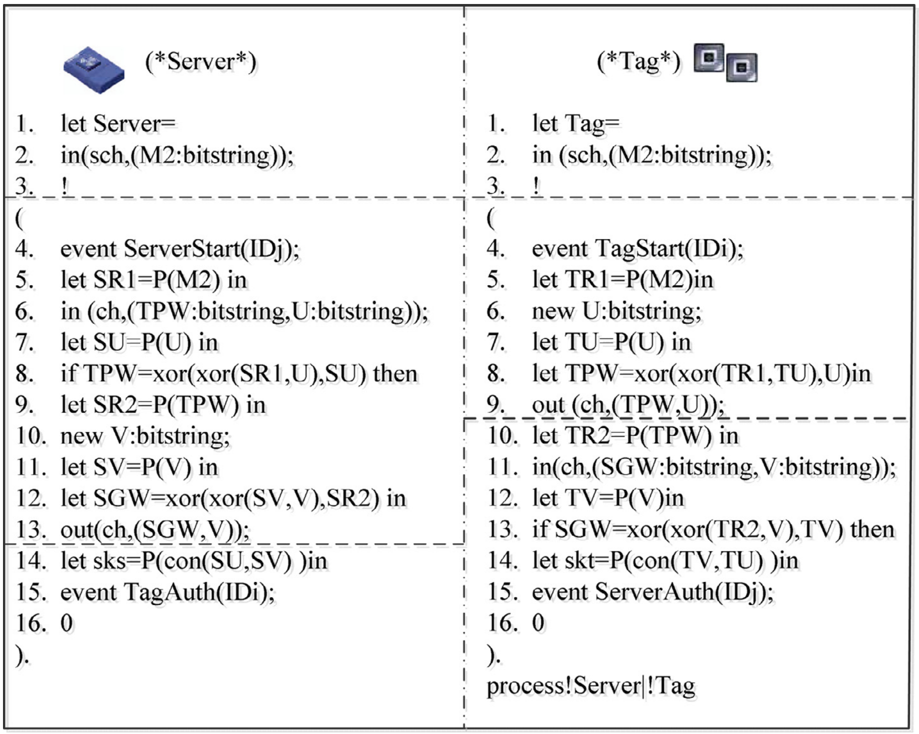
Fig. 11. Predefinition part.

4.3. Security analysis

In this section, the proposed protocol is analyzed on how to ensure system security facing various security threats, and the security is compared with that of the slender PUF protocol.

4.3.1. Machine learning attacks

In the slender PUF protocol, the substring W can expose the response behavior of PUF. Despite the increase in difficulties for attacks, the protocol is not safe against machine learning attacks. Mukhopadhyay et al. proposed a method to attack slender PUF by using ES-based machine learning and proved that slender PUF protocol is vulnerable since machine learning attacks have been realized [22]. For the four-stage arbiter PUF, Mukhopadhyay assumes that the length of the slender PUF is defined as $L = 1024$ with $L_{sub}= 256$, where L is the length of the PUF response string, and L_{sub} is the length of the substring.



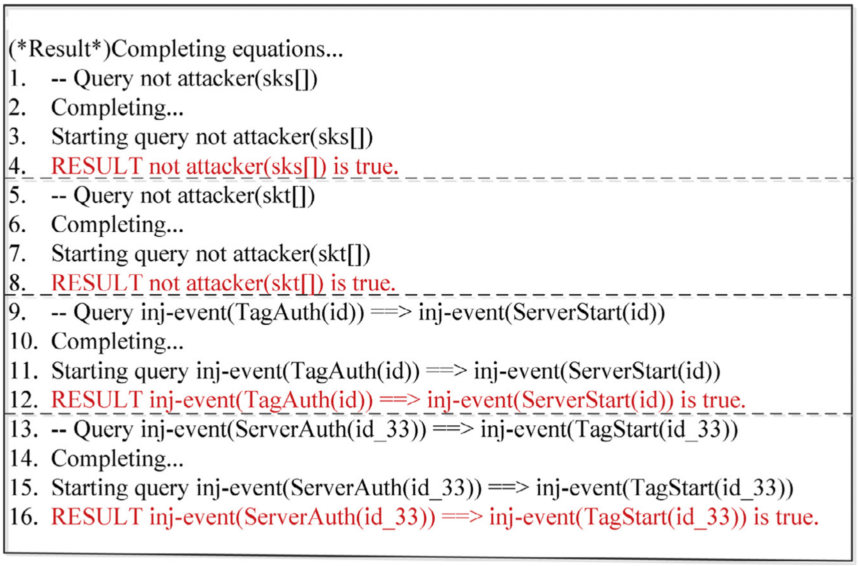
```
(*Server*)

let Server=
in(sch,(M2:bitstring));
!
(
event ServerStart(IDj);
let SR1=P(M2) in
in (ch,(TPW:bitstring,U:bitstring));
let SU=P(U) in
if TPW=xor(xor(SR1,U),SU) then
let SR2=P(TPW) in
new V:bitstring;
let SV=P(V) in
let SGW=xor(xor(SV,V),SR2) in
out(ch,(SGW,V));
let sks=P(con(SU,SV) )in
event TagAuth(IDi);
0
).
```

```
(*Tag*)

let Tag=
in (sch,(M2:bitstring));
!
(
event TagStart(IDi);
let TR1=P(M2)in
new U:bitstring;
let TU=P(U) in
let TPW=xor(xor(TR1,TU),U)in
out (ch,(TPW,U));
let TR2=P(TPW) in
in(ch,(SGW:bitstring,V:bitstring));
let TV=P(V)in
if SGW=xor(xor(TR2,V),TV) then
let skt=P(con(TV,TU) )in
event ServerAuth(IDj);
0
).
process!Server|!Tag
```

Fig. 12. Server and tag implementation.



```
(*Result*)Completing equations...
-- Query not attacker(sks[])
Completing...
Starting query not attacker(sks[])
RESULT not attacker(sks[]) is true.
-- Query not attacker(skt[])
Completing...
Starting query not attacker(skt[])
RESULT not attacker(skt[]) is true.
-- Query inj-event(TagAuth(id)) ==> inj-event(ServerStart(id))
Completing...
Starting query inj-event(TagAuth(id)) ==> inj-event(ServerStart(id))
RESULT inj-event(TagAuth(id)) ==> inj-event(ServerStart(id)) is true.
-- Query inj-event(ServerAuth(id_33)) ==> inj-event(TagStart(id_33))
Completing...
Starting query inj-event(ServerAuth(id_33)) ==> inj-event(TagStart(id_33))
RESULT inj-event(ServerAuth(id_33)) ==> inj-event(TagStart(id_33)) is true.
```

Fig. 13. Proverif results.

They estimated that the attacker could analyze the critical delay vector information ω inside the PUF only with about 600,000 CRPs [22]. Once the PUF delay vector information ω is exposed, attacks are successfully launched by using machine learning.

In the proposed protocol, the server and the tag jointly generate the challenge information of the PUF, and the attacker cannot impersonate either party to achieve the purpose of completely controlling the PUF challenge information. In the phase of tag verifying server, the seed of PRNG is *PW* sent by the tag after the server successfully authenticates the tag. Therefore, attackers cannot directly obtain *PW*. Moreover, the TSMCA PUF satisfies the avalanche effect, and replay attacks do not occur with the duplicate response of PUF. In this work, the response *R* is processed by a dichotomy XOR. The response of PUF *GW* does not directly determine the string. Also, the entropy of *GW* is high, and string padding is random to avoid exposing the original response. Consequently, attackers cannot obtain the response of PUF even if the information *GW* is captured during the interaction. Moreover, machine learning attacks cannot select the optimal offspring of the machine learning, and the delay information in PUF is not obtained. Thus, the proposed protocol is secure in terms of resisting machine learning attacks.

4.3.2. Man-in-Middle attack

Attackers can damage the authentication when specific parameters are obtained or constructed by attackers. The slender PUF agreement cannot resist man-in-middle attacks [22]. In the slender PUF protocol, the prover and the verifier generate random numbers N_v and N_p. Both random numbers are used as the seed of *PRNG*, as denoted by $Seed{=}(N_v \| N_p)$. Assume that the method to connect the numbers is denoted by $PRNG(N_v \| N_p){=}LFSR(N_v) \oplus LFSR(N_\mathrm{p})$. Also, it is assumed that the attackers send the random number $N_v = N_p$ via malicious attackers. After receiving the random number, the verifier performs $PRNG(N_v \| N_p)$, which causes the output of the PUF model to be equal to 0. If the malicious attackers send $R_{sub} = 0$ or the all-zero string, then the success rate of the authentication is 1/2. By playback, it can reach 1.

In the proposed protocol, a hash function is added to the calculation of the server to ensure the security of the challenge. The server generates a random number T_A and encrypts it with the hash function, as denoted by $h(T_A)$. Then, the server calculates $M_1{=}h(T_A) \oplus T_A$ and sends the result to the tag. The tag performs an XOR operation on its random number T_B and computes $M_2{=}h(T_A) \oplus T_A \oplus T_B$. After that, M_2 and T_B are sent to the server. After receiving M_2, the server calculates $M_3{=}M_2 \oplus T_B \oplus h(T_A){=}T_{A'}$. Then, $T_{A'}$ is compared with T_A. If the values are consistent, then M_2 is used as the seed of PRNG to generate the challenge of PUF.

At this point, it is assumed that the attackers can capture the interaction information of server and tag, as denoted by $M_1 \| M_2 \| T_B$. Also, T_A is encrypted by a hash function. The hash function is a one-way irreversible function, and the attacker cannot know the value of T_A through $M_1 \| M_2 \| T_B$. Hence, malicious attackers cannot alter $M_1 \| M_2 \| T_B$ during the server and tag interaction.

4.3.3. Replay attack

The attacker successfully captures the authentication information of the legal tag and requests authentication with this piece of information to cheat the legal tag or server. This type of attack applies to protocols that do not update keys promptly. By contrast, in the proposed protocol, the attackers cannot entirely control the challenge of PUF. In the worst-case scenario, only a half challenge is controlled, as depicted by M_1 or T_B. Nevertheless, TSMCA PUF has a good avalanche effect. If one or multiple bits of the challenge change, then at least half bits of the response flip. Despite that attacker's control half bits of the challenges, the replay attack will not be realized.

4.3.4. Information exposure attack

In [31], Rostami filled a padding string operation based on slender PUF protocol. Although the addition can increase the difficulty of attacks, filled string *PW* inevitably exposes the response of PUF due to the insufficient protocol. For instance, assuming that the length of *PW* after padding is 1024, and the length of *W* is 64. The attacker can guess and enumerate all possible strings with the length of 64 bits, such that 1024 choices are made available to the attacker. Therefore, the attackers can find the rules of PUF CRPs.

In the proposed protocol, a dichotomy XOR operation is performed on the response *R*. This method differs from the operation that uses *W* in the slender PUF protocol. The string in protocol *PW* will not wholly expose the response behavior of PUF, and the information differs from the original response *R* through the XOR operation. According to the protocol, the string is randomly padded after *PW* is divided into the two substrings *C* and *D*, and a new string *GW* with the length of *L* is generated. In *GW*, the positions of *C* and *D* have no order. In this case, *PW* is further hidden.

4.3.5. Spoofing attack

In the slender PUF protocol, one-way authentication is realized while the verifier verifies the prover, though the protocol assumes the verifier is legal. Nevertheless, many deficiencies exist in practice: an attacker can counterfeit a server to retrieve user information, for example.

The protocol proposed in this paper has high levels of security, and bidirectional authentication between server and tag is realized. The authentication is also realized by an original PUF and a PUF model. If bidirectional authentication is successful, then we can ensure that the communication parties are trustable entities. Thus, the proposed scheme can resist spoofing attacks.

4.4. Performance comparison

To perform the testing for the uniqueness and stability parameters of the TSMCA PUF proposed in this paper, we use Xilinx's XUPV5-LX110T development board with a clock signal of 25 MHz as the experimental platform. The circuit module is split into multiple specific LUT regions on the FPGA platform, and the same functions of the TSMCA PUF circuit and the PUF circuits on multiple LUT regions are implemented [6,9]. In the experiment, the challenge signal is set to 64 bits. Each

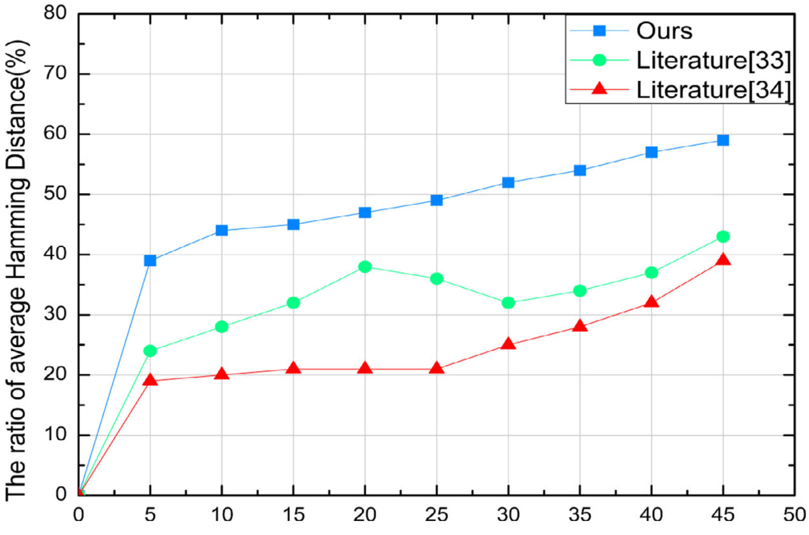


Fig. 14. Uniqueness comparison.

time the 64-bit signal is given, the test platform is cyclically shifted 64 times. Furthermore, each cyclic shift obtains the output result of the 1-bit arbiter, such that a 64-bit response bit value is obtained each time.

The inter-chip Hamming distance can express the uniqueness of the PUF, used to refer the difference of the response bit value of the same challenge signal between different PUF circuits. The higher the inter-chip Hamming distance, the better the uniqueness of the PUF structure. When calculating the inter-chip Hamming distance, the same challenge signal must input the PUF in each LUT region, and the PUF response bit value obtained by each PUF is calculated. By measuring and comparing the response bit values generated by different PUF circuits with these same challenge signals, the uniqueness of the PUF structure can be verified. The average inter-hamming Hamming distance formula for PUF is expressed as:

$$\alpha = \frac{2}{q(q-1)} \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} \frac{D\left(P_i, P_j\right)}{n} \times 100\%, \tag{4.13}$$

q is the number of tested PUFs, where $\mathrm{D}(P_i, P_j) = \sum_{m=1}^{n} (r_{i,m} \oplus r_{j,m})$, and $r_{i,m}$ represents the m th bit information of the n-bit response P_i.

To better understand the behavior of TSMCA PUF structure, we enter 100 64-bit challenge signals in the PUF circuit in each LUT region. The behavior of the TSMCA PUF structure proposed is shown in Fig. 14. As a result, a 6400-bit response bit value is obtained; when $q = 45$, the maximum inter-chip Hamming distance is 59% while when $q = 5$, the minimum inter-chip Hamming distance is 39%. In this case, the average inter-chip Hamming distance is 49.6%, which approximates the ideal value of 50%.

Colombier et al. [6] reported that the maximum inter-chip Hamming distance obtained is 43% when $q = 45$; the minimum inter-chip Hamming distance is 24% when $q = 5$; and the average inter-chip Hamming distance is 33%. In [9], the maximum inter-chip Hamming distance is 39% when $q = 45$; the minimum inter-chip Hamming distance is 19% when $q = 5$; and the average Hamming distance is 25%. Thus, as the value of q increases, the inter-chip Hamming distance increases, and the relative difference of response bit values increases [6,9]. Therefore, the experimental analysis shows that the TSMCA PUF structure has functional uniqueness.

PUF stability can be expressed by intra-chip Hamming distance. The intra-chip Hamming distance of PUF refers to the difference of the response bit value of the same challenge signal between the same PUF circuits. The smaller the intra-chip Hamming distance, the more stable the PUF structure becomes. In the proposed implementation, the PUF circuit's challenge signal is controlled to the same challenge signal and input the challenge signal to the PUF circuit under different environmental conditions (i.e., different noise and voltage values) in different LUT regions. By inputting the same challenge signal for the PUF circuit y times, the stability of the PUF can be verified by measuring and comparing the response bit values generated by the same PUF circuit with the same challenge. The average intra-chip Hamming distance of PUF is:

$$\beta = \frac{1}{q} \sum_{1}^{q} \frac{D\left(R_i, R_{i,j}\right)}{n} \times 100\%, \tag{4.14}$$

where $\mathrm{D}(R_i, P_{i,j}) = \sum_{i=1}^{y} \sum_{m=1}^{n} (r_{i,m} \oplus r_{j,m})$ represents the Hamming distance between the original response R_i and the output response $R_{i,y}$ of the yth test, n represents the number of bits of the response and q is the number of tested PUFs.

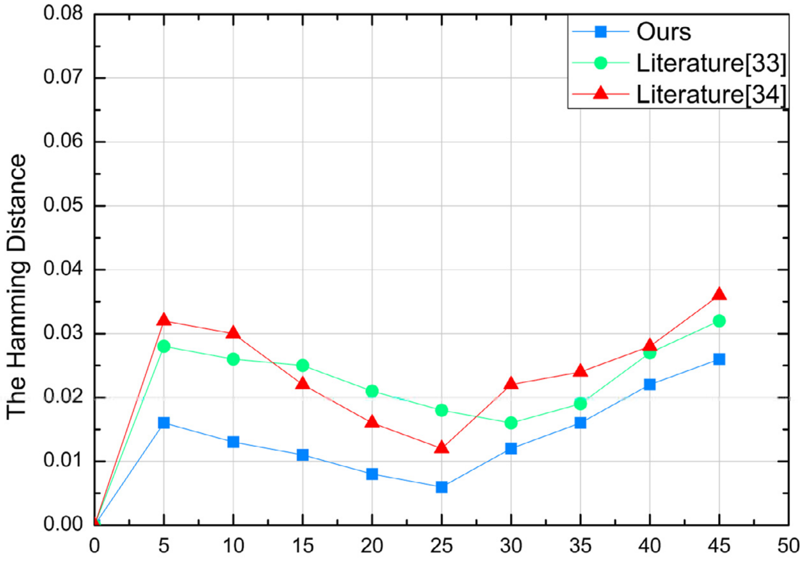


Fig. 15. Stability comparison.

Table 5
Security comparison of the different identity authentication protocol.

Attacks	MQ	GIMAP	PUF-HB#	Slender PUF	Proposed protocol
Machine learning	No	Yes	No	No	Yes
Replay	No	Yes	No	Yes	Yes
Man-in-middle	No	Yes	No	No	Yes
Tracking attack	Yes	Yes	Yes	Yes	Yes
Nonsynchronous	No	Yes	Yes	No	Yes
Side-channel attacks	Yes	No	No	Yes	Yes

For stability issues, an experiment is designed where the same challenge signal is repeatedly input $y=4$ times into each PUF circuit, and a total of 256-bit response information is generated as the output. The results are shown in Fig. 15. For the TSMCA PUF, the maximum intra-chip Hamming distance of 0.026 is obtained when $q=45$, the minimum intra-chip Hamming distance of 0.006 is obtained when $q=25$, and the average intra-chip Hamming distance is 0.014. As depicted by Colombier et al. [6], the maximum intra-chip Hamming distance is 0.032, the minimum intra-chip Hamming distance is 0.016, and the average intra-chip Hamming distance is 0.023. Moreover, Gope et al. [9] shown that the PUF intra-chip Hamming distance maximum is 0.036, the minimum value is 0.012, and the average intra-chip Hamming distance is 0.024. From the analysis of experimental results and comparison to Colombier et al. [6] and Gope et al. [9], the TCMCA PUF structure proposed in this paper has low intra-chip Hamming distance and the robust capability to resist environmental changes.

Results show that the uniqueness and stability of the TSMCA PUF structure are better than the other two PUF structures, which implies better suitability in terms of providing security for low-cost RFID systems in harsh environments.

Many attacks are launched against RFID systems, which include machine learning attacks, replay attacks, and man-in-middle attacks. Also, RFID systems are influenced by tracking and nonsynchronous and side-channel attacks. When tracking these attacks, if the parameter values of a system are set to constant, then an attacker can quickly capture the values of the system and replay them. In nonsynchronous attacks, the attacker may attempt to capture or alter the data if either the reader or the tag updates the data while the other does not update on time. Due to the nonsynchronous data update, the authentication cannot fail between them. In side-channel attacks, the attacker detects the power or collects the leaking information to extract the information in NVM, which is treated as a threat to RFID systems.

We assume that an attacker can eavesdrop and intercept information passing through the network and tamper it to perform attacks on the protocol. Table 5 presents the security of the proposed protocol compared with other protocols: multi-query (MQ) protocol [18], GIMAP [37], PUF-HB# [17], and slender PUF [25]. Still, the proposed protocol can resist various attacks better than others.

Table 6 shows a comparison of the computation complexity of the proposed protocol with other protocols. The proposed protocol has about 150% more computational complexity than the compared ones.

The proposed protocol for bidirectional authentication between server and tag is implemented without storing essential information on NVM. The application of NVM is neglected due to expensive hardware resources requirements and not

Table 6
Computation comparison of different protocols.

Protocol	Times of interaction	NVM	Computation complexity	Bidirectional authentication
Multi-query	10~N	Yes	N	No
GIMAP	5	Yes	13	Yes
PUF-HB#	3	Yes	15	Yes
Slender PUF	3	No	11	No
our protocol	4	No	32	Yes

Table 7
Comparison of hardware resource of different PUF structures.

Types of PUF	Gate resources	LUT
Controlled PUF	3545~N	1558~N
PPUF [30]	545~1090	*
Mixture PUF	836	*
Slender PUF	2180	512
Our PUF	1090	287

"*" represents that the author provides no data.

Table 8
Comparison of storage and communication cost.

Storage and communication requirements	Storage cost(bits)	Communication cost(bits)
PUF-HB#	1840	1746
GIMAP	96+(n*m)128	*
MQ	96	(96×2)+(256+1024)N
Slender PUF	0	128×2 + 512
Proposed protocol	0	160×3 + 1024×2

"*" represents that no data is provided.

conducive implementation in lightweight architecture. Moreover, NVM is vulnerable to physical attacks [34]. Although the computation cost required by the proposed protocol is higher than that of the others, the proposed protocol is better due to its lightweight performance.

The tag is deployed with limited hardware resources, so the resource consumption of hardware in Tag is analyzed in the proposed implementation. For the PUF structure in Tag, the TSMCA PUF structure has a four-way selector and a selection module. In the Virtex II-PRO FPGA device, the implementation of the four-way selector requires nearly 21 NAND gates while the selection module needs 21 registers since approximately 80 gates realize each register.

Given the safety and lightness of the system, it is assumed that the PUF structure in the slender PUF protocol is a four-stage arbiter PUF, since it is the most popular PUF and it has better security than single arbiter PUF. Also, a 64-bit arbiter PUF requires 545 gates, and the four-stage arbiter PUF needs at least 2180 gates. Thus, approximately 512 LUTs and a trigger are required to realize the protocol in FPGA. The TMCCA PUF structure proposed in this paper requires two arbiters, a multiplexer, and a counter. Overall, the implementation of FPGA needs only 287 LUTs and a trigger.

The PUF structure proposed in this paper is compared to the controlled PUF [14], PPUF [30], Mixture PUF [18], and slender PUF [25]. The controlled PUF requires a hash function in encryption, though the implementation of a lightweight hash function requires more than 3000 gates. The comparison results are listed in Table 7. Therefore, the implementation of the proposed protocol is the most efficient in terms of area: it occupies only 56% of the slender PUF implementation, saving 44% of the total area.

Table 8 summarizes the consumption of tag memory and communication of existing protocols and the proposed protocol. To improve security, we assume that the length of the $GW(GW')$ string transmitted is 1024 bits, the length of the random number is 128 bits, the length of the substring L_{sub} is 256 bits, the ID and query information of an entity is 96 bits, and the hash function (the SHA of the output is −1) is 160 bits.

The PUF-HB# protocol incurs minimal communication costs, which is 1746 bits [17]. The Tag memory and communication consumption in the GIMAP protocol depend on the size of the matrix n*m, and an excessively large matrix space is not suitable for low-cost RFID systems. The MQ protocol is relatively lightweight in storage consumption, but it requires multiple information exchanges to complete tag authentication. The slender PUF protocol and the proposed protocol do not store key messages in tags and rely solely on the tag's PUF structure to generate critical information. In general, the proposed protocol has superior performance in terms of security and resource overhead.

5. Conclusions and future work

Designing security solutions in favor of low-cost RFID systems for IoT deployment is challenging due to physical limitations in storage and computational resources and other operational requirements. As discussed in this paper, conventional PUF and cryptographic protocols are generally not suitable for the direct deployment of RFID systems given the resource limitations.

In this investigation, the TSMCA PUF structure is designed and a double PUF-based bidirectional RFID identity authentication protocol is proposed. The PUF structure can be implemented with limited hardware requirements. A server does not need to store the CRPs of PUF. In the proposed approach, the XOR method is used to process strings, while the random padding method is used to encrypt PUF responses. The findings from the security and performance analysis demonstrate the utility of the proposed approach.

Despite their potential in RFID authentication, PUFs remain mostly under-explored. First of all, the stability of the PUF is weak. For the same challenge information, the PUF usually outputs different results under the influence of the environment, which reduces the security of the PUF. In order to enable the PUF to output stable response information, error correction code and majority selection method are proposed. These methods cannot guarantee that the CRPs of the PUF will be precisely the same, though. Therefore, designing a high stability PUF structure is crucial for the development of PUF. Also, PUF is a security encryption primitive based on hardware characteristics. The hardware age and the output of PUF will change as time passes. Unfortunately, such changes will make the legal owner of the PUF unable to pass the verification of the verifier. Therefore, to propose an authentication protocol suitable for aging PUF is an important future direction.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Acknowledgment

The authors would like to thank anonymous reviewers for their helpful advice on various technical issues examined and comments. This research is funded by the National Natural Science Foundation of China (grant 61572188), Xiamen Science and Technology Foundation(grant 3502Z20173035), Scientific Research Program of New Century Excellent Talents in Fujian Province University, Industrial Robot Application of Fujian University Engineering Research Center, Minjiang University (MJUKF-IRA201802), Fujian Provincial Natural Science Foundation of China(grant 2018J01570) and the CERNET Innovation Project(grant NGII20170411).

References

[1] G. Avoine, M.A. Bingol, X. Carpent, et al., Privacy-Friendly authentication in RFID systems: on sublinear protocols based on symmetric-key cryptography, IEEE Trans. Mob. Comput. Vol.12 (, 10) (2013) 2037–2049.
[2] A. Arbit, Y. Livne, Y. Oren, et al., Implementing public-key cryptography on passive RFID tags is practical, Int. J. Inf. Secur. Vol.14 (2015) 85–99.
[3] G.T. Becker, The gap between promise and reality: on the insecurity of XOR arbiter PUFs, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, Heidelberg, 2015, pp. 535–555.
[4] R. Küsters, T. Truderung, Using ProVerif to analyze protocols with Diffie-Hellman exponentiation, in: 2009 22nd IEEE Computer Security Foundations Symposium, IEEE, 2009, pp. 157–171.
[5] P.Y. Cui, An improved ownership transfer and mutual authentication for lightweight RFID protocols, Int. J. Netw. Secur. 18 (6) (2016) 1173–1179.
[6] B. Colombier, L. Bossuet, V. Fischer, et al., Key reconciliation protocols for error correction of silicon PUF responses, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1988–2002.
[7] J. Delvaux, Security analysis of PUF-based key generation and entity authentication, Ph. D. Thesis, Katholieke Universiteit Leuven (KULeuven), Leuven, Belgium, 2017.
[8] J. Delvaux, R. Peeters, D. Gu, et al., A survey on lightweight entity authentication with strong PUFs, ACM Comput Surv 48 (2) (2015) 1–42.
[9] P. Gope, J. Lee, T.Q.S Quek, Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions, IEEE Trans. Inform. Forensics Secur. 13 (11) (2018) 1 PP(99)-1.
[10] Y. Gao, D.C. Ranasinghe, S.F. Al-Sarawi, et al., Emerging physical unclonable functions with nanotechnology, IEEE Access Vol.4 (2016) 61–80.
[11] F. Ganji, S. Tajik, J.P. Seifert, Why attackers Win: on the learnability of XOR arbiter PUFs, in: proceedings of the International Conference on Trust and Trustworthy Computing, 2015, pp. 22–39.
[12] S. Xie, W. Liang, J. Xu J, et al., A Novel Bidirectional RFID Identity Authentication Protocol, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2018, pp. 301–307.
[13] V. Herrewege, S. Katzenbeisser, R. Maes, et al., Reverse fuzzy Extractors: enabling lightweight mutual authentication for PUF-Enabled RFIDs, in: Financial Cryptography and Data Security -, International Conference, FC, 2012, pp. 374–389.
[14] C. Herder, M.D. Yu, F. Koushanfar, et al., Physical unclonable functions and applications: a Tutorial, Proc. IEEE Vol.102 (2014) 1126–1141.
[15] D. He, S. Zeadally, An analysis of RFID authentication schemes for internet of things in healthcare environment using elliptic curve cryptography, IEEE Internet Thing. J. Vol.2 (2015) 72–83.
[16] J. Katz, in: Analysis of a Proposed Hash-Based Signature Standard, Springer International Publishing, 2016, pp. 261–273.
[17] H. Li. PUF-HB#, A lightweight RFID mutual authentication protocol, J. Beijing Univ. Post. Telecommun. 36 (6) (2013) 13–17.
[18] W.H. Liu, Design and implementation of a Low-cost physical Non-cloneable function structure and its RFID applications, J. Electron. 44 (7) (2015) 1772–1776.
[19] W. Liang, B. Liao, J. Long, et al., Study on PUF based secure protection for IC design, Microprocess Microsyst 45 (2016) 56–66.

[20] W. Liang, J. Long, T.H. Weng, et al., TBRS:a trust based recommendation scheme for complex CPS network, Future Gener. Comput. Syst. 92 (2019) 383–398.

[21] W. Liang, M. Tang, J. Long, et al., A secure fabric Blockchain-based data transmission technique for industrial Internet-of-Things, IEEE Trans. Ind. Inf. (2019) 1–9.

[22] D. Mukhopadhyay, PUFs as promising tools for security in internet of things, IEEE Des. Test 33 (3) (2016) 103–115.

[23] H. Xu, J. Ding, P. Li, et al., A Lightweight RFID mutual authentication protocol based on physical unclonable function, Sensors 18 (3) (2018) 760.

[24] C. Manifavas, G. Hatzivasilis, K. Fysarakis, et al., in: Lightweight Cryptography for Embedded Systems - A Comparative Analysis, Springer, Berlin Heidelberg, 2014, pp. 333–349.

[25] M. Majzoobi, M. Rostami, F. Koushanfar, D.S. Wallach, S. Devadas, A lightweight, robust, secure authentication by substring matching, in proc, IEEE Symp. Security Privacy Workshops (2012) 33–44.

[26] E. Nilsson, C. Svensson, Ultra-Low power wake-up radio using envelope detector and transmission line voltage transformer, IEEE J. Emerg. Select. Top. Circuits Syst. Vol.3 (2013) 5–12.

[27] M. Qiu, Z. Jia, C. Xue, et al., Voltage assignment with guaranteed probability satisfying timing constraint for real-time multiproceesor DSP, J. VLSI Signal Process. Syst. Signal Image Video Technol. 46 (1) (2007) 55–73.

[28] M. Qiu, Z. Ming, J. Li, et al., Three-phase time-aware energy minimization with DVFS and unrolling for chip multiprocessors, J. Syst. Archit. 58 (10) (2012) 439–445.

[29] M. Qiu, Z. Ming, J. Li, et al., Informer homed routing fault tolerance mechanism for wireless sensor networks, J. Syst. Archit. 59 (4–5) (2013) 260–270.

[30] J. Rajendran, G S Rose, R Karri, et al., Nano-PPUF: a Memristor-based security primitive, VLSI IEEE (2012) 84–87.

[31] M. Rostami, M. Majzoobi, F. Koushanfar, et al., Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching, IEEE Trans. Emerg. Top. Comput. Vol.2 (2014) 37–49.

[32] U. Rührmair, F. Sehnke, J. Sölter, et al., Modeling attacks on physical unclonable functions, in: Proceedings of the 17th ACM conference on Computer and communications security, ACM, 2010, pp. 237–249.

[33] G.E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Proceedings of the 44th annual design automation conference, ACM, 2007, pp. 9–14.

[34] A. Das, Ü. Kocabaş, A.R. Sadeghi, et al., PUF-based secure test wrapper design for cryptographic SoC testing, in: Proceedings of the Conference on Design, Automation and Test in Europe. EDA Consortium, 2012, pp. 866–869.

[35] G. Hospodar, R. Maes, I. Verbauwhede, Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability, in: 2012 IEEE international workshop on Information forensics and security (WIFS), IEEE, 2012, pp. 37–42.

[36] M.D.M. Yu, M. Hiller, J. Delvaux, et al., A lockdown technique to prevent machine learning on PUFs for lightweight authentication, IEEE Trans. Multi--Scale Comput. Syst. 2 (3) (2017) 146–159.

[37] Jiaq Zhang, A lightweight RFID authentication protocol GIMAP, Small Microcomput. Syst. 34 (3) (2013) 530–534.

[38] J. Zhang, Y. Lin, Y. Lyu, G. Qu, A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing, IEEE Trans. Inf. Forensics Security 10 (6) (Jun. 2015) 1137–1150.

Wei Liang is currenty a Professor at College of Computer Science and Electronic Engineering, Hunan University, China. Prior to joining in 2019, he was a professor at School of Opto-Electronic and Communication Engineering, Xiamen University of Technology, China. He received his Ph.D. degree from Hunan University in 2013, and a postdoctoral scholar at the Department of Computer Science and Engineering at Lehigh university in USA during 2014-2016. He has taken as a guest researcher in State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, China. He is a Senior Member of the IEEE, Application Track Chair of IEEE Trustcom 2015, Workshop Chair of IEEE Trustcom WSN 2015, IEEE Trustcom WSN 2016, and several other conferences. He has published more than 110 journal/conference papers in journals such as IEEE Transaction on Computational Biology and Bioinformatics, Wireless Personal Communications, Computer Science and Information Systems, Microprocessors and Microsystems, International Journal of Communication Systems, and Journal of Sensors, Security and Communication Networks, Nonlinear Dynamics, and International Journal of Communication Systems. His research interests include Networks Security Protection, embedded system and Hardware IP protection, and Fog computing, and Security management in WSN.

Songyou Xie is currently pursuing the M.S. degree with the Hunan University of Science and Technology, Xiangtan, China. He received the B.*Sc.* degree in optical information science and technology from Hunan University of Science and Technology, Xiangtan, china, in 2016. His research interest is mainly in the security analysis of integrated circuits and lightweight security authentication protocols based on PUF circuits.

Jing Long is currently working in College of Information Science and Engineering, Hunan Normal University, China. She received the Ph.D. degree in computer science and technology from Hunan University, China in 2018, and M.S. degree in computer science and technology from Hunan University of Science and Technology, China in 2012. Her research interests include network and information security, hardware security and IP protection.

Kuan-Ching Li is a Distinguished Professor at Providence University, Taiwan. He is a recipient of distinguished and chair professorships from universities in China and other countries and awards and funding support from a number of agencies and high-tech companies. Besides publishing numerous journal articles, book chapters, and refereed conference papers, he is co-author/co-editor of more than 20 technical professional books published by CRC Press/Taylor & Francis, Springer, and McGraw-Hill. His research interests include GPU/many-core computing, Big Data, and Cloud. He is a senior member of the IEEE and a fellow of the IET.

Dafang Zhang is a professor in college of computer science and Electronic Engineering, Hunan University, China. He received his PHD degree in applied mathematics from Hunan University in 1997, he has published more than 300 journal articles and conference papers. He has published 11 books and is the Editor-in-Chief of four books. The published books were utilized in more than 100 universities. He has received five provincial and ministerial level scientific and technological progress awards. His research interests include dependable systems/networks, network security, network measurement and hardware security.

Keqin Li is a SUNY Distinguished Professor of Computer Science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber–physical systems. He has published over 490 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, Distributed Computing. He is an IEEE Fellow.