



1 of 1

[Download](#) [Print](#) [Save to PDF](#) [Save to list](#) [Create bibliography](#)

Lecture Notes in Networks and Systems • Volume 699 LNNS, Pages 401 - 412 • 2023 • Intelligent Computing and Networking - Proceedings of IC-ICN 2023 • Mumbai • 24 February 2023 through 25 February 2023 • Code 299239

Document type

Book Chapter

Source type

Book Series

ISSN

23673370

ISBN

978-981993176-7

DOI

10.1007/978-981-99-3177-4_29

[View more](#)

PUF-Based Lightweight Authentication Protocol for IoT Devices

Shah, Amita^a ; Pandya, Hetal^a ; Soni, Mukesh^b ; Karimov, Akramjon^c ;
Maaliw, Renato R.^d ; Keshta, Ismail^e

[Save all to author list](#)

^a Department of Computer Engineering, L D College of Engineering, Gujarat, Ahmedabad, India

^b Department of CSE, University Centre for Research and Development Chandigarh University, Punjab, Mohali, 140413, India

^c Department of Corporate Finance and Securities, Tashkent Institute of Finance, Tashkent, Uzbekistan

^d College of Engineering, Southern Luzon State University, Quezon, Lucban, Philippines

[View additional affiliations](#)

[Full text options](#) [Export](#)

Cited by 0 documents

Inform me when this document is cited in Scopus:

[Set citation alert >](#)

Related documents

On the Security of a PUF-Based Authentication and Key Exchange Protocol for IoT Devices

Sun, D.-Z. , Gao, Y.-N. , Tian, Y. (2023) *Sensors*
Improvement and Cryptanalysis of a Physically Unclonable Functions Based Authentication Scheme for Smart Grids

Safkhani, M. Bagheri, N. Ali, S. (2023) *Mathematics*
PUF-Based Authentication Protocol with Physical Layer-Based Obfuscated Challenge-Response Pair

Alkanhal, M. Alali, A. Younis, M.

(2023) *IEEE International Conference on Communications*
[View all related documents based on references](#)

Find more related documents in Scopus based on:

[Authors](#) [Keywords >](#)

Abstract

Author keywords

Indexed keywords

SciVal Topics

Metrics

Abstract

The Internet of Things (IoT) carries the secure transmission and storage of a large amount of sensitive information. This paper uses the anti-tampering and anti-cloning features of the hardware physical unclonable function (PUF) to generate a shared key, combined with security primitives such as MASK algorithm and Hash function and proposes a lightweight Anonymous key-sharing security authentication protocol. Through the security analysis and verification of Ban logic and the proper tool

Valentina Emilia Balas
Vijay Bhaskar Semwal
Anand Khandare *Editors*


Intelligent Computing and Networking

Proceedings of IC-ICN 2023

Lecture Notes in Networks and Systems

Volume 699

Series Editor

Janusz Kacprzyk , Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA,
School of Electrical and Computer Engineering—FEEC, University of
Campinas—UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering,
Bogazici University, Istanbul, Türkiye

Derong Liu, Department of Electrical and Computer Engineering, University of
Illinois at Chicago, Chicago, USA

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of
Alberta, Alberta, Canada

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering,
KIOS Research Center for Intelligent Systems and Networks, University of Cyprus,
Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong,
Kowloon, Hong Kong

PUF-Based Lightweight Authentication Protocol for IoT Devices



Amita Shah, Hetal Pandya, Mukesh Soni, Akramjon Karimov,
Renato R. Maaliw, and Ismail Keshta

Abstract The Internet of Things (IoT) carries the secure transmission and storage of a large amount of sensitive information. This paper uses the anti-tampering and anti-cloning features of the hardware physical unclonable function (PUF) to generate a shared key, combined with security primitives such as MASK algorithm and Hash function and proposes a lightweight Anonymous key-sharing security authentication protocol. Through the security analysis and verification of Ban logic and the proper tool ProVerif. It is proved that the protocol can defend against man-in-the-middle attacks, desynchronization attacks, impersonation attacks, modeling attacks, etc. By comparing other protocols, it is verified that the protocol has the advantages of low computing cost, small communication overhead and storage capacity, and high-security performance; it is suitable for secure communication transmission of resource-constrained devices.

A. Shah (✉) · H. Pandya

Department of Computer Engineering, L D College of Engineering, Ahmedabad, Gujarat, India
e-mail: Amitashah@ldce.ac.in

H. Pandya

e-mail: hetalbpandya@ldce.ac.in

M. Soni

Department of CSE, University Centre for Research and Development Chandigarh University,
Mohali 140413, Punjab, India

A. Karimov

Department of Corporate Finance and Securities, Tashkent Institute of Finance, Tashkent,
Uzbekistan
e-mail: akram_karimov@tfi.uz

R. R. Maaliw

College of Engineering, Southern Luzon State University, Lucban, Quezon, Philippines
e-mail: rmaaliw@slsu.edu.ph

I. Keshta

Computer Science and Information Systems Department, College of Applied Sciences,
AlMaarefa University, Riyadh, Saudi Arabia
e-mail: imohamed@mcst.edu.sa

Keywords Lightweight authentication · Internet of things · Authentication protocol · Desynchronization attacks · Anti-tampering

1 Introduction

With the rapid development of sensing, automation, and communication technologies, the massive data generated by the IoTs (IoT) has dramatically threatened the effective and safe transmission, storage and protection between devices [1]. Traditional network Security protocols will use complex security primitives such as encryption algorithms [2], digital signatures, Hash functions [3], and message verification codes to ensure the confidentiality, integrity, and non-repudiation of information transmission [4]. However, usually small size, strong resource constraints, and low hardware processing capability, so the security primitives with slow transmission rate and high communication overhead are not suitable for the authentication of lightweight devices [5].

The secure communication of the IoTs assumes that hardware devices and systems are safe. Still, malicious attackers can destroy confidential device information using chip cloning and dissection [6]. Information encryption is an effective way to protect information security certification. Still, encryption of the algorithm's key is usually stored in non-volatile memory (NVM), and the attacker can successfully read the private information in the memory through a side channel and physical attacks [7]. Physically unclonable function (PUF) is an emerging The hardware security primitives of the chip use the random process deviation that cannot be controlled in the manufacturing process of the chip to generate the unique digital signature of the device [8]. It is susceptible to physical tampering, does not need to be stored, has low hardware overhead, can solve the security problems faced by traditional keys, and is suitable for lightweight IoTs device security authentication protocols [9–11].

Key-sharing authentication protocols are usually completed using public key algorithms and digital signatures at the software layer. Still, these encryption primitives run slowly and have high communication overhead, and new quantum computing methods can effectively crack public key algorithms. To solve the problem of IoTs devices in terms of security issues in channel transmission and critical storage, this paper uses reconfigurable CROPUF anti-tampering and anti-cloning features to generate shared keys, replacing asymmetric encryption algorithms and digital signatures with high communication overhead, combined with MASK algorithm and Hash function. A lightweight anonymous key-sharing security authentication protocol is proposed for encryption methods. The protocol ensures security properties such as anonymity, availability, integrity, and forward/backward secrecy.

2 Mathematical Theoretical Knowledge and Related Work

2.1 PUF-Based Key-Sharing Mechanism

Author [12] used CRO PUF to generate the same shared key for devices, which is suitable for one-to-many security authentication protocols. This mechanism obtains shared keys between devices through two stages.

Phase 1: Generate a reliable response to the shared key. Obtain the high-precision delay matrix S of the CRO PUF through modeling; calculate the delay difference between all paths and sort them in descending order of absolute value. Consider the influence of different temperatures to determine threshold U . When the total value of the delay difference between the two paths is greater than the threshold, the output response is stable.

Phase 2: Generate incentives for the shared key. Store all the paths of the delay matrix S in the set E , and enumerate all the bits of the shared key L . Randomly select two different ways from the set E for each of the shared key L . One bit L_i , if L_i is equal to 1 and the delay difference is more significant than U , it means that a configuration stimulus D_i that can produce a stable response of 1 has been found; if L_i is equal to 0 and the delay difference is less than $-U$, it means that a configuration that can generate a stable response of 0 has been found Incentive D_i ; otherwise, reselect the shared key L .

2.2 MASK and UNMASK Algorithms

The MASK algorithm proposed by author [13] contains three parameters: the input vector $s = [s_1, s_2, s_3, \dots, s_l]$ with a length of l bits, and a set of l positive integers $L = \{l_1, l_2, l_3, \dots, l_l | l_i \in A^+\}$ and generate l -bit output vector $s_m = [s_{m1}, s_{m2}, s_{m3}, \dots, s_{ml}]$. MASK algorithm uses positive integer set L as auxiliary data and inputs vector r with length l -bit to generate an Output function s_m of equal length. A set of positive integers L generates $L \leftarrow \text{PRNG}_i(x)$ through a pseudo-random number generator, where y is an input vector of length n bits, and $y = [y_1, y_2, y_3, \dots, y_n]$. Similarly, the reversible transformation UNMASK function of the MASK function uses the positive integer set L to convert the output function s_m into a restored output function r .

(1) Integer set generation: The vector y is used as the seed of the pseudo-random number generator PRNG circuit to generate a set of positive integers $l \{l_1, l_2, l_3, \dots, l_l | l_i \in A^+\}$. The integer set L contains l n -bit positive Integer; the maximum value of any positive integer is $2^n - 1$.

(2) Function range transformation: It defines a range function $\text{Range}()$ as a linear mapping transformation, given an l -digit integer $\{k | k \in K\}$, whose value range is $[0,$

$2^1 - 1]$, generate an m -digit The new set $R = \{ r | r \in Z^+ \}$ with integers in the Range $[0, 2^m - 1]$, where $m \leq 1$. The following equation governs the linear range mapping:

$$M_{\text{new}} = \left\lceil \frac{(M_{\text{old}} - M_{\text{oldamin}}) \times (M_{\text{neemax}} - M_{\text{neemin}})}{(M_{\text{oldmax}} - M_{\text{oldanin}})} \right\rceil$$

$M_{\text{old}} \in L$ is the input of the Range () function, M_{oldamin} and M_{oldmax} are the minimum and maximum values in the Range $[0, 2^1 - 1]$, M_{newmin} and M_{newmax} are the minimum values in the new Range $[0, 2^m - 1]$ with the maximum value.

(3) Bit obfuscation: The MASK function finally completes the bit obfuscation of the sequence based on the Fisher-Yates Shuffler shuffling algorithm, and a finite series of n different elements generates an $n!$ a random permutation algorithm.

The MASK algorithm has two advantages: (1) It effectively hides the relationship between device PUF excitation and response; (2) It verifies the device’s input. The PUF will not be activated without verifying the input stream, so the device will not generate any response, effectively preventing the device from any brute force attack.

3 Protocol Design and Analysis

This paper proposes a lightweight key-sharing authentication protocol based on the IoT device embedded with PUF, including the protocol registration phase, two-way authentication, and firmware update phase. The related symbols of the protocol are shown in Table 1.

Table 1 Description of protocol-related symbols

Symbol	Meaning
CRO PUF	Reconfigurable Ring Oscillator PUF
Delay Matrix M_A / M_B	Delay Matrix M_A or M_B
(C,R)	Stimulus C and response R generated by PUF
Timestamp()	Timestamp function
(n_{i1}, n_{i2})	Pseudo-random number
Hash(·)	One-way hash function
Fisher - Shuffler()	Shuffle Confusion Algorithm
PRNG()/TRNG()	Pseudo/True Random Number Generator

4 Formal Security Proof

4.1 Formal Security Analysis

This protocol ensures the channel transmission security of IoT devices and can also defend against physical attacks on PUF by attackers. The specific security analysis is as follows.

- (1) Modeling attack. The machine learning modeling attack is aimed at a strong PUF with a publicly accessible CRP interface. The attacker collects many CRPs, trains, learns, and optimizes an accurate model to predict response.
- (2) However, the reconfigurable CRO PUF is a weak PUF used for key generation. There is no access interface to read the key generated inside the chip, and the key will not be exposed to attackers. At the same time, due to the protection of the protocol mechanism, it is protected by the MASK algorithm (D_A , D_B) value. The Fisher-Shuffler confusion algorithm divides the response value into two parts (s_{A1} , s_{A2}) and (s_{B1} , s_{B2}) and uses the Hash algorithm and random number generator to protect part of the response value s_{A1} in the device. Since the Hash function is one-way, the attacker cannot obtain the real CRP value by eavesdropping on the content e_A , so it is difficult for the attacker to carry out machine learning modeling attacks on PUF.
- (3) Untraceability. In the IoT device identity authentication process, if the attacker cannot effectively associate the request and response information of the two authentications, the input and output results cannot be mapped, and the device is considered untraceable. The attacker passes when eavesdropping to obtain messages E_A and E_B because the incentives D_A and D_B are encrypted and protected by the MASK function; the attacker cannot infer the value of the incentive D_A and D_B . After the attacker eavesdrops on the messages e_A and e_B , that is, $e_A \leftarrow \text{Hash}_A(s_{A1}, \text{PRNG}(n_{B2}))$, $e_B \leftarrow \text{Hash}(s_{A2}, \text{PRNG}(n_{A2}))$. Due to the uniqueness of the hash function, it cannot obtain the values of the shared keys S_A and S_B . Therefore, this protocol can prevent location tracking, as shown in Fig. 1.
- (4) Desynchronization attack. In the update phase of the protocol, the device generates new random numbers (n_{A1}^{new} , n_{A2}^{new}) and (n_{B1}^{new} , n_{B2}^{new}) and also stores the random numbers (n_{A1}^{old} , n_{A2}^{old}) and (n_{B1}^{old} , n_{B2}^{old}). When device B is attacked by desynchronization, the random number of device A will be updated typically, but the random number of device B will not be updated. When the next authentication round is performed, the server will return the value E_A , n_{B2}^{old} , given to device A, and the value E_B , n_{A2}^{new} , is returned to device B. Since the verification value e_A^{old} returned by device A is consistent with the previous authentication round, the new B returned by device B is different from the last round. Therefore it is possible to detect that the devices are not synchronized.

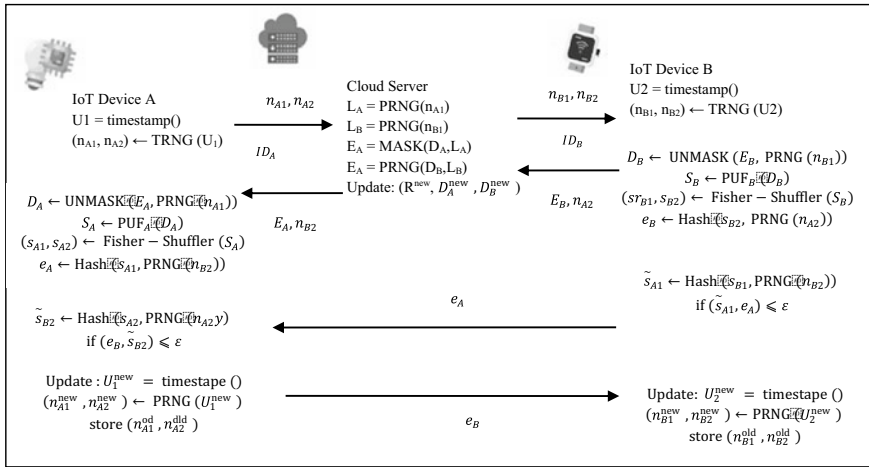


Fig. 1 Two-way authentication and firmware update phase

- (5) **Replay attack.** The protocol mechanism uses timestamps and updated random numbers (n_{A1}, n_{A2}) and (n_{B1}, n_{B2}) to defend against replay attacks. Taking the authentication of device A as an example, assuming that the i-th and i + 1 random numbers are (n_{A1}ⁱ, n_{A2}ⁱ) and (n_{B1}ⁱ⁺¹, n_{B2}ⁱ⁺¹), when the attacker obtains the i-th session message E_Aⁱ⁺¹, e_Bⁱ⁺¹, and performs the i + 1th authentication, device A receives the session message of has been updated to E_Aⁱ⁺¹, e_Bⁱ⁺¹. Therefore, the attacker authentication will fail, and the replay attack will be detected.
- (6) **Counterfeit attack.** When the attacker pretends to be a legitimate device, it must send valid messages e_A and e_B. Using device A as an example, because the generation of information e_A requires valid s_{A1}, the input of the PUF is protected by the MASK function to stimulate CA. At the same time, the response S_A is protected by the obfuscation algorithm. Due to the uniqueness of the Hash function, effective information s_{A1} cannot be obtained even if the information e_A is obtained. Therefore, in this protocol mechanism, the attacker cannot pretend to be a legitimate device and authenticate with the server. Similarly, when the attacker Masquerades as a server, the value of incentive D_A and D_B cannot be obtained, so mutual authentication with the device cannot be accepted.
- (7) **Man-in-the-middle attack.** This protocol can perform a man-in-the-middle attack in defense, the attacker cannot obtain valid information by eavesdropping the messages E_A, n_{B2}, E_B, n_{A2}, e_A, and e_B. Because the eavesdropped messages are all encrypted information, if the attacker replaces the new message, the devices will not be able to identify each other causing authentication failure. Furthermore, suppose the attacker wants to parse the encrypted information. In that case, it can be known from the above impersonation attack that the attacker cannot obtain the CRP pair (D_A, S_A) of the PUF, so the authentication between devices will fail.

4.2 Protocol Proof

This section uses BAN logic to prove the security of the shared key generated by PUF. BAN logic is a formal analysis method for authentication protocols, which can prompt some defects that are difficult to find by non-formal methods. The logic symbols commonly used in BAN logic are shown in Table 2. Table 3 shows the logic rules widely used in BAN logic related to this section.

For the convenience of analysis, devices A and B are denoted as E1 and E2, and the server is designated as S. After idealizing the protocol, the message-sending rules are adjusted as follows:

$$\begin{aligned} N1 : E1 &\rightarrow T : n_{A1}, n_{A2}; N2 : E2 \rightarrow T : n_{B1}, n_{B2} \\ N3 : T &\rightarrow E1 : \{d_A, n_{A1}\}_{n_{A1}}, n_{B2}; N4 : T \rightarrow E2 : \{d_B, n_{B1}\}_{n_{B1}}, n_{A2} \\ N5 : E1 &\rightarrow E2 : \{s_{A1}, n_{B2}\}_{n_m}; N6 : E2 \rightarrow E1 : \{s_{B1}, n_{A2}\}_{n_e} \end{aligned}$$

Protocol initialization assumes the following:

$$\begin{aligned} A1 : E1 &|\equiv \neq \{n_{A1}, n_{A2}\} A2 : E2 |\equiv \neq \{n_{B1}, n_{B2}\} \\ A3 : E11 &\equiv E1 \xleftrightarrow{n_n} E11 \equiv E1 \xleftrightarrow{n_{B_j}} T \\ A4 : E21 &\equiv E2 \xleftrightarrow{n_m} T; E21 \equiv E2 \xleftrightarrow{n_m} T \\ A5 : E11 &\equiv E1 \xleftrightarrow{n_m} E2, E21 \equiv E1 \xleftrightarrow{n_m} E2 \end{aligned}$$

The target formula is as follows:

Table 2 Description of logic symbols commonly used in BAN logic

Symbol	Meaning	Symbol	Meaning
$P \models X$	P believes that X is true	$P \mid \Rightarrow X$	Entity P has jurisdiction over X
$P \mid \sim X$	P used to send message X	$\#(X)$	X is fresh
$P \triangleleft X$	Entity P receives message X	(X, Y)	X or Y is part of (X, Y)
$\{X\}_K$	Use key K to message X perform cryptographic operations	$P \xleftrightarrow{K} Q$	K is between P and Q shared key

Table 3 Logic rules widely used in BAN logic

Rules	Logical expression
Rule 1: Message Meaning Rules	$\frac{P \models P \xleftrightarrow{k} Q, P \triangleleft \{X\}_k}{P \models Q \mid \sim X}$
Rule 2: The Freshness Rule	$\frac{P \mid \sim \#(X)}{P \mid \sim \#(X, Y)}$
Rule 3: Temporary Value Validation Rules	$\frac{P \mid \sim \#(X), P \mid \sim Q \mid \sim X}{P \mid \equiv Q \mid \sim X}$
Rule 4: Trust Rules	$\frac{P \mid \equiv Q \mid \sim (X, Y)}{P \mid \equiv Q \mid \sim X}$

$$\begin{aligned} G1 : E1 \equiv T \equiv \{d_A\}; H3 : E2 \equiv E1 \equiv \{s_{A1}\}; \\ G2 : E2 \equiv T \equiv \{d_B\}; H4 : E1E \equiv E2 \equiv \{s_{B1}\}; \end{aligned}$$

Protocol initialization assumes the following:

If the target formula is established, both the device and the server T negotiate and confirm the secret key with each other, and the private key is bound to the platform integrity report and the communication channel. The following logical reasoning can be made according to the rules in Table 3.

From rule 1 and A_3, N_3 we know.

Therefore, the target formula G1 can be proved: $D1 \equiv S \equiv \{cA\}$, device A and server S share key cA.

From rule 1 and A_4, M_4 we know

$$\frac{E1 \equiv E1 n_{A1} T, E1 \triangleleft \{d_A, n_{A1} A1\}}{E1 \equiv T \equiv \{d_A, n_{A1}\}}. \quad (1)$$

From rules 2 and A_2 , we know

$$\frac{E1 \equiv \#(n_{A1})}{E1 \equiv \#(d_A, n_{A1})} \quad (2)$$

According to rule 3 and formula (1) and formula (2):

$$\frac{E1 \equiv \#(d_A, n_{A1}), E1 \equiv T \sim \{d_A, n_{A1}\}}{E1 \equiv T \equiv \{d_A, n_{A1}\}} \quad (3)$$

From rule 4 and formula (3), it can be seen that

$$\frac{E1 \equiv T \equiv \{d_A, n_{A1}\}}{E1 \equiv T \equiv \{d_A\}} \quad (4)$$

Therefore, it can be proved that the target formula $H1: E1 \equiv T \equiv \{d_A\}$, equipment A shares secret key d_A with server T.

From rule 1 and A_4, N_4 we know

$$\frac{E2 \equiv E2 n_{m1} T, E2 \triangleleft \{d_B, n_{B1}\}_{n_{n1}}}{E2 \equiv T \equiv \{d_B, n_{B1}\}} \quad (5)$$

From rule 2 and message A_2 , we can know

$$\frac{E2 \equiv \#(n_{B1})}{E2 \equiv \#(d_B, n_{B1})} \quad (6)$$

Similarly, according to rules 3 and 4, and formula (5) and formula (6), the target formula can be proved: $H2: E2 \equiv T \equiv \{d_B\}$.

From rule 1 and A5, N5 we know

$$\frac{E2 \models E1 \xleftrightarrow{n_m} E2, E2 \triangleleft \{s_{A1}, n_{B2}\} n_m}{E2 \models E1 \mid \sim \{s_{A1}, n_{B2}\}} \quad (7)$$

From rule 2 and message A2 we know

$$\frac{E2 \mid \equiv \# \{n_{B2}\}}{E2 \mid \equiv E1 \mid \equiv \{s_{A1}, n_{B2}\}} \quad (8)$$

From rule 3 and formula (7) and formula (8), it can be seen that

$$\frac{E2 \mid \equiv \# \{s_{A1}, n_{B2}\}, E2 \mid \equiv E1 \mid \sim \{s_{A1}, n_{B2}\}}{E2 \mid \equiv E1 \mid \equiv \{s_{A1}, n_{B2}\}} \quad (9)$$

From rule 4 and formula (9), it can be seen that

$$\frac{E2 \mid \equiv E1 \mid \equiv \{s_{A1}, n_{B2}\}}{E2 \mid \equiv E1 \mid \equiv \{s_{A1}\}} \quad (10)$$

Therefore, the target formula G3 can be proved: $E2 \models E1 \models \{s_{A1}\}$, device B and device A share key s_{A1} .

From rule 1 and A5, N6 we know

$$\frac{E1 \models E1 \xleftrightarrow{n_{A2}} E2, E1 \triangleleft \{s_{B1}, n_{A2}\} n_{A2}}{E1 \models E2 \mid \sim \{s_{B1}, n_{A2}\}} \quad (11)$$

From rule 2 and A1 we know

$$\frac{E1 \mid \equiv \# \{n_{A2}\}}{E1 \mid \equiv \# \{s_{B1}, n_{A2}\}} \quad (12)$$

Similarly, from rules 3 and 4, and formula (11) and formula (12), it can be proved that.

Target formula: H4: $E1 \mid \equiv E2 \mid \equiv \{s_{B1}\}$.

5 Protocol Performance Analyses

This section analyzes and evaluates the authentication protocol in terms of security attributes, storage capacity, and communication costs. The authentication protocol program between the device and the server is written in Python. The network interaction is completed by abstracting the socket connected by the TCP client/server. It makes the server wait for a connection with the device on the specified IP address.

Once the device establishes a relationship with the server, the protocol performs a mutual authentication session. The server and the machine run on Windows 10, using an Intel Core i7-9750 CPU with a frequency of 2.60 GHz, Equipped with 8 GB RAM, simulating the proposed authentication scheme.

Regarding security attributes, it is compared with other protocols. In the protocol [14–17] mechanism, the attacker can obtain the CRP pair of PUF through eavesdropping, counterfeiting, and physical attack. Therefore, it is impossible to defend against modeling attacks. The protocol [18] can protect against PUF attacks through the d-time locking mechanism. Still, the information in channel transmission is not encrypted, which leads to security threats such as eavesdropping, resynchronization, and replay attacks in device authentication. Protocol [19, 20] device stores the old and new identities. The attacker obtains the current identity information by accessing the memory to trace the authentication information of the previous or next round, so the protocol is not irretrievable. However, this protocol uses the shared key generated by PUF. It uses the encryption of the MASK algorithm and Hashes function to ensure the privacy and non-traceability of the device.

Figures 2 and 3 list the comparison with other protocols regarding device storage and communication overhead. Referring to the paper by Literature [11], the pseudo-random identity PIDiD is 128 bits, the word length of the CRP pair (D_i , S_i) is 128 bits, and the byte length of the key is 96 bits. This protocol only stores 128-bit random numbers (no1ld, no1ld), which is far lower than the storage capacity of other protocols. Furthermore, the protocol only transmits information (n_{A1} , n_{A2} , E_A , n_{B2} , e_A , e_B) and the communication overhead is 640 bits. Compared with other protocols [3, 5, 9–11, 16, 17], the communication cost of the proposed protocol is lower than that of different schemes (as shown in Fig. 3), and it is suitable for lightweight devices—Security authentication scenarios.

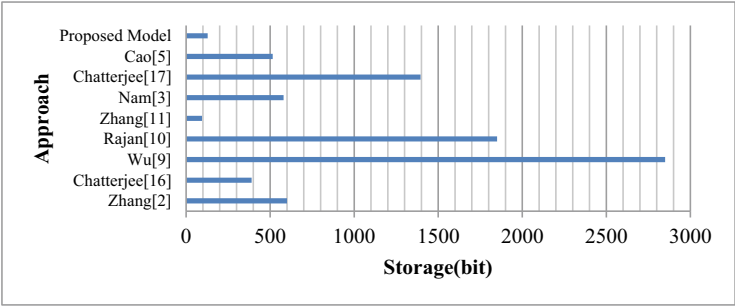


Fig. 2 Performance comparison of device storage

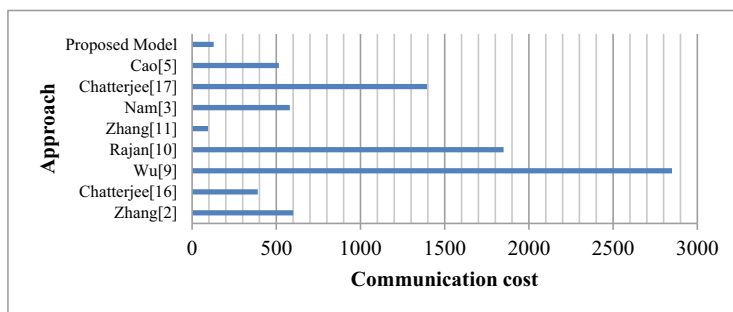


Fig. 3 Communication cost in the proposed protocol

6 Conclusion

The massive data generated by the IoTs brings information transmission security threats to resource-constrained terminal devices. At the same time, hardware devices usually face security issues such as chip cloning, device forgery, and key storage. Therefore, the traditional network protocol is not suitable for the security authentication of lightweight IoT devices. This paper proposes a lightweight anonymous key-sharing authentication protocol for IoT devices. This mechanism uses the characteristics of PUF anti-tampering and anti-cloning to generate shared keys on the hardware side combining security primitives such as obfuscation algorithm, MASK algorithm, and Hash function to ensure security attributes such as anonymity, untraceability, non-repudiation, and forward/backward secrecy of information transmission. Through formal verification tools, ProVerif, BAN logic, and informal, the security analysis and verification of the protocol prove the security, reliability and anti-channel attack ability of the protocol operation. Compared with other existing protocols, the proposed protocol has low computing cost, small communication overhead and storage capacity, and high security, suitable for secure communication transmission of lightweight IoT devices.

References

1. Farha F, Ning H, Ali K, Chen L, Nugent C (2021) SRAM-PUF-based entities authentication scheme for resource-constrained IoT devices. *IEEE IoTs J* 8(7):5904–5913, <https://doi.org/10.1109/JIOT.2020.3032518>
2. Zhang J, Shen C, Guo Z, Wu Q, Chang W (2022) CT PUF: configurable Tristate PUF against machine learning attacks for IoT security. *IEEE IoTs J* 9(16):14452–14462, <https://doi.org/10.1109/JIOT.2021.3090475>
3. Nam J-W, Ahn J-H, Hong J-P (2022) Compact SRAM-Based PUF chip employing body voltage control technique. *IEEE Access* 10:22311–22319. <https://doi.org/10.1109/ACCESS.2022.3153359>







4. Idriss TA, Idriss HA, Bayoumi MA (2021) A lightweight PUF-based authentication protocol using secret pattern recognition for constrained IoT devices. *IEEE Access* 9:80546–80558. <https://doi.org/10.1109/ACCESS.2021.3084903>
5. Cao J, Li S, Ma R, Han Y, Zhang Y, Li H (2022) RPRIA: reputation and PUF-based remote identity attestation protocol for massive IoT devices. *IEEE IoTs J* 9(19):19174–19187, <https://doi.org/10.1109/JIOT.2022.3164174>
6. Lounis K, Zulkernine M (2021) T2T-MAP: a PUF-based thing-to-thing mutual authentication protocol for IoT. *IEEE Access* 9:137384–137405. <https://doi.org/10.1109/ACCESS.2021.3117444>
7. Ebrahimabadi M, Younis M, Karimi N (2022) A PUF-based modeling-attack resilient authentication protocol for IoT devices. *IEEE IoTs J* 9(5):3684–3703, <https://doi.org/10.1109/JIOT.2021.3098496>
8. Sadana S, Lele A, Tsundus S, Kumbhare P, Ganguly U (2018) A highly reliable and unbiased PUF based on differential OTP memory. *IEEE Electron Device Lett* 39(8):1159–1162. <https://doi.org/10.1109/LED.2018.2844557>
9. Wu L, Hu Y, Zhang K, Li W, Xu X, Chang W (2022) FLAM-PUF: a response–feedback-based lightweight anti-machine-learning-attack PUF. *IEEE Trans Comput Aided Des Integr Circuits Syst* 41(11):4433–4444. <https://doi.org/10.1109/TCAD.2022.3197696>
10. Rajan C, Samajdar DP (2020) Design principles for a novel lightweight configurable PUF using a reconfigurable FET. *IEEE Trans Electron Devices* 67(12):5797–5803. <https://doi.org/10.1109/TED.2020.3030868>
11. Zhang Y, Li B, Liu B, Hu Y, Zheng H (2021) A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain. *IEEE IoTs J* 8(18):13958–13974, <https://doi.org/10.1109/JIOT.2021.3068410>
12. Lee J, Kim M, Jeong M, Shin G, Lee Y (2022) A 20F2/Bit current-integration-based differential nand-structured PUF for stable and V/T variation-tolerant low-cost IoT security. *IEEE J Solid-State Circuits* 57(10):2957–2968. <https://doi.org/10.1109/JSSC.2022.3192903>
13. Amsaad F et al (2021) Enhancing the performance of lightweight configurable PUF for robust IoT hardware-assisted security. *IEEE Access* 9:136792–136810. <https://doi.org/10.1109/ACCESS.2021.3117240>
14. Li S, Zhang T, Yu B, He K (2021) A provably secure and practical PUF-based end-to-end mutual authentication and key exchange protocol for IoT. *IEEE Sensors J* 21(4):5487–5501, <https://doi.org/10.1109/JSEN.2020.3028872>
15. Labrado C, Thapliyal H (2019) Design of a piezoelectric-based physically unclonable function for IoT security. *IEEE IoTs J* 6(2):2770–2777. <https://doi.org/10.1109/JIOT.2018.2874626>
16. Chatterjee B, Das D, Maity S, Sen S (2019) RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using In-Situ Machine Learning. *IEEE IoTs Journal* 6(1):388–398. <https://doi.org/10.1109/JIOT.2018.2849324>
17. Chatterjee U et al. (2019) Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database. *IEEE Trans Dependable Secure Comput* 16(3):424–437, <https://doi.org/10.1109/TDSC.2018.2832201>
18. Mall P, Amin R, Das AK, Leung MT, Choo K-KR (2022) PUF-based authentication and key agreement protocols for IoT, WSNs, and smart grids: a comprehensive survey. *IEEE IoTs J* 9(11):8205–8228, <https://doi.org/10.1109/JIOT.2022.3142084>
19. Song B, Lim S, Kang SH, Jung S-O (2021) Environmental-variation-tolerant magnetic tunnel junction-based physical unclonable function cell with auto write-back technique. *IEEE Trans Inf Forensics Secur* 16:2843–2853. <https://doi.org/10.1109/TIFS.2021.3067173>
20. Gao B, Lin B, Li X, Tang J, Qian H, Wu H (2022) A unified PUF and TRNG design based on 40-NM RRAM with high entropy and robustness for IoT security. *IEEE Trans Electron Devices* 69(2):536–542. <https://doi.org/10.1109/TED.2021.3138365>

FACULTY POSITION RECLASSIFICATION FOR SUCS
(DBM-CHED Joint Circular No. 3, series of 2022)


CERTIFICATION OF PERCENTAGE CONTRIBUTION
(Research Output with Multiple Authors)

Title of Research: PUF-Based Lightweight Authentication Protocol for IoT Devices
Type of Research Output: Book Chapter (Scopus-Indexed, Publisher: Springer Nature)

Instruction: Supply ALL the names of the authors involved in the publication of Research output and indicate the contribution of each author in percentage. Each author shall sign the conforme column if he/she agrees with the distribution. The conforme should be signed by all the authors in order to be considered. Please prepare separate Certification for each output.

	Name of Authors	Current Affiliation	% Contribution	Conforme <i>(Sign if you agree with the % distribution)</i>
1	Amita Sha	LD College of Engineering	16.67%	
2	Hetal Pandya	LD College of Engineering	16.67%	
3	Mukesh Soni	Chandigarh University	16.67%	
4	Akramjon Karimov	Tashkent Institute	16.67%	
5	Renato R. Maaliw III	SLSU	16.67%	
6	Ismail Keshta	AlMaarefa University	16.67%	
* Should have a total of 100%			100.00%	

Prepared by:


Renato R. Maaliw III, DIT
(Name and Signature)
Faculty

Certified by:

Nicanor L. Guinto, Ph.D
(Name and Signature)
Director, Office of Research Services