The Improvement of PUF-Based Authentication in IoT Systems

E. Haodudin Nurkifli [1,2] and Tzonelih Hwang [1,*]

[1] Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan
[2] Faculty of Computer Science, Universitas Singaperbangsa Karawang, West Java, Indonesia;
Email: dudi.nurkifli@staff.unsika.ac.id (E.H.N.)
*Correspondence: hwangtl@mail.csie.ncku.edu.tw (T.H.)

***Abstract*—The Internet of Things (IoT) has been widely utilized in many fields, including healthcare, manufacturing, and intelligent transportation. IoT has many advantages, including the ease of collecting, accessing, and editing data remotely. Unfortunately, IoT has also given rise to many risks due to its lack of security features. Authentication protocols for IoT environments have been proposed using Physical Unclonable Functions (PUFs) to resolve security problems in IoT environments. However, we have found that these protocols have weaknesses, where the attacker steals the IoT device, obtains data from the communication channel, and has the ability to program for activating PUF, hence the attacker gets the challenge and generates a response. Therefore, we propose a new authentication protocol that resolves the issues with the existing protocol to safeguard against the attack mentioned above with PUF and masking method. We present the results of the Real or Random model (RoR model), which shows that our protocol is secure. The programming model (Scyther tool) also shows that our protocol is safe and can withstand attacks. Finally, our protocol has a low computational time.**

***Keywords*—adversary model, internet of things, authentication protocol, Physical Unclonable Function (PUF), masking method**

I. Introduction

Internet of Things (IoT) has changed the daily lives of humans, individuals can work from home and collect and process data remotely using their devices. IoT has also introduced new technologies such as intelligent transportation systems, robotic healthcare systems, and intelligent drones. The security aspect is imperative in IoT systems. Furthermore, the public key cryptosystem is unsuitable for IoT applications because the IoT device is tiny with limited resources.

Aman *et al*. [1] proposed mutual authentication using Physical Unclonable Function (PUF) in IoT systems. PUF is an alternative cryptography for tiny devices. Gope and Sikdar [2] also proposed a protocol utilizing PUF and a hash function in IoT environments. However, the authors found that Aman *et al*.'s scheme cannot achieve Perfect Forward and Backward Secrecy (PFBS), anonymity, and untrace ability, and cannot resist impersonation and Denial-of-Service (DoS) attacks. Additionally, Gope and Sikdar's protocol cannot achieve PFBS and untraceability. Other research has also utilized PUFs for authentication protocols [3–6]. However, their schemes have similar loopholes to the schemes proposed in [1, 2]. If the attacker can program the activation of PUF, they can obtain the secret and conduct dangerous attacks such as impersonation attacks. On the other hand, if the attacker obtains the challenge-response pattern using a machine learning attack, they can clone the device.

Recently, Lee *et al*. [7] proposed secure sensing utilizing PUFs. Unfortunately, it has been found that the authentication protocol fails to achieve an unclonable sensor when the attacker steals the sensor and activates PUFs. Hence, the attacker has the ability to obtain the challenge response pattern. to get pattern the challenge response.

This article proposes an authentication protocol that resolves authentication problems in IoT environments using PUFs. In addition, our protocol is generic and can also address other security issues in IoT environments.

The remainder of this manuscript is organized as follows: Section II presents related work and our contributions, while Section III provides materials and methods. Our proposed authentication protocol is presented in Section IV, followed by informal analysis in Section V, mathematical model (Real or Random (RoR) model) in Section VI, programming model in Section VII, competencies and complexity comparison in Section VIII, and finally, Section IX concludes the paper.

II. Literature Review

There have been many kinds of research in IoT systems, such as [8–10]. However, their papers only present a survey of the threats and challenges in IoT systems. Hence, their articles need to provide solutions to resolve the security issues. Security features, including mutual authentication, anonymity, untrace ability, resolution of desynchronization issues, and secure session keys, are imperative for an authentication protocol in an IoT environment. In addition, the protocol must withstand well-known attacks, such as impersonation, DoS, physical attacks, etc.

Manuscript received March 17, 2023; revised May 4, 2023; accepted May 16, 2023; published September 18, 2023.

Many fields apply the IoT system, including healthcare, industrial, intelligent transportation systems, and the military. The devices can communicate through wireless systems. The fact is that wireless is a public channel where adversaries may eavesdrop on all transmitted messages. Additionally, an attacker with solid capability can conduct fatal attacks such as impersonation, tracking, and cloning attacks.

The tracking attack is a crucial issue where the adversary can track users of devices based on their identity if the authentication protocol does not provide anonymity. This can endanger the patient's life, especially if the patient is a public figure, such as a president or artist, who requires anonymity while receiving treatment [11–15].

PUF is introduced as an alternative cryptography algorithm that is utilized for a tiny device [16–18]. The PUF has a simple operation R = PUF(C), that is, the PUF generates a response from a challenge. Most researchers utilize the PUF for generating the key [19]. However, now several researchers have extended the usage to encrypt and decrypt processes [20–22].

Gope *et al*. [23] proposed an authentication protocol in Industrial Wireless Sensor Networks. However, Nurkifli and Hwang [24] showed that the protocol proposed by Gope *et al*. [23] and Gope and Sikdar [25] do not achieve anonymity and Perfect Forward and Backward Secrecy (PFBS). In another research, Serkan [26] proposed an authentication protocol using PUF in RFID environments, claiming that their protocol can achieve anonymity. However, Kardas pointed out that the protocol proposed by Serkan [26] is susceptible to the cold boot attack. After the attacker conducts a cold boot attack, they can obtain the credential data, allowing them to impersonate the tag and trace the RFID tag's past and future messages [27].

Aman *et al.* [1] and Gope [2] proposed authentication protocols using PUF, claiming to have achieved security features such as mutual authentication, anonymity, untraceability, and the ability to withstand impersonation and tracking attacks. Other researchers have also utilized PUF in their schemes, such as [3–6]. However, their schemes are vulnerable to modified DY adversary attacks. Furthermore, Lee *et al*. [7] utilized PUF to provide secure sensing in IoT systems, but their protocol fails to achieve the required security competencies. Therefore, this article proposes an authentication protocol using PUF with the following contributions:

- Proposing an adversary model that adopts the Dolev-Yao (DY) structure by adding the powerful capability to attack the PUF-based authentication.
- The authors propose a new authentication protocol to resolve authentication protocol using PUF issues and generic IoT environment security issues.
- The authors conducted a mathematical model analysis to confirm that our authentication protocol is secure under the RoR model.
- The authors conducted a programming model (Scyther tool) to confirm that our authentication protocol is secure and can resist attacks.
- Demonstrating the computational complexity demonstrates that our authentication protocol is suitable for IoT systems.

III. Materials and Methods

This section will briefly explain adversary capabilities, PUFs, secure PUFs, hash functions, security of hash functions, and masking methods.

A. Adversary's Capabilities

The authors propose an adversary model that adopts the Dolev-Yao (DY) structure [28, 29], which has the following capabilities:

C1: The adversary can eavesdrop and intercept all data transmitted in the public channel.

C2: The adversary can alter, delete, modify, and replay data.

C3: The adversary can steal the IoT device.

C4: The adversary can extract data from the IoT device's memory using a side-channel attack [30, 31].

In addition, the authors also modify the DY adversary model by adding the following capabilities:

C5: The adversary can operate the PUF by programming it without separating the PUF from the onboard unit.

C6: The adversary can operate the Fuzzy Extractor (FE) without separating it from the Onboard Unit (OBU).

B. Physical Unclonable Function (PUF)

PUF is a fabrication product that utilizes a function to map a challenge to a response [19, 32], formulated as R = PUF(C). However, in reality, the response R may contain noise or the PUF yields unstable response R. To obtain a stable output, a Fuzzy Extractor (FE) can be applied to the PUF [33, 34]. However, the usage of FE can increase computational complexity. Nowadays, several researchers [35–38] offer stable PUFs, and they produce a stable response R. Therefore, in this article, the authors use a stable PUF.

The PUF is secure if C1 insert into PUF1 produces R1 and C2 insert into PUF1 has R2. In addition, C1 insert into PUF1 yields R1, and C1 insert into PUF2 gains R2. Furthermore, the authors obtain the formulations

$$\Pr\left[HD\left(PUF_1(C_1), PUF_1(C_2)\right) > d_1\right] = 1 - \varepsilon, \Pr\left[HD\left(PUF_1(C_1), PUF_2(C_1)\right) > d_2\right] = 1 - \varepsilon;$$

Pr denotes the Probability of yielding the different outputs of PUF, HD represents Hamming distance, d_1 denotes the variety of challenges, d_2 denotes the variety of PUF, and ε marks negligible with low value.

C. Hash Function

The function maps the arbitrary input to the fixed-length output. The formula is $h: \{0,1\}^* \rightarrow \{0,1\}^n$. The advantages of adversary breaking the hash function and getting collision based on bits x_1 and x_2 in particular time *t*. $Adv_A^{Hash}(t) = \Pr\left[(x_1, x_2) \in_R \boldsymbol{A}: x_1 \neq x_2, h(x_1) = h(x_2)\right]$.

D. Masking Method

The masking method is a countermeasure for side-channel attacks, where the method eliminates the relationship between confidential data and leakage information. There are two types of masking methods: Boolean masking and Arithmetic masking. Boolean masking uses the XOR operation, while arithmetic masking uses arithmetic operations. The formulation of each masking is as follows: Boolean masking: $x' = x \oplus r$, Arithmetic masking: $A = x - x_r \ mod \ 2^k$. Our protocol utilizes a masking method to safeguard against side-channel attacks [39, 40].

E. Security Requirements

The authentication protocol must achieve security requirements under the DY model treatment.

- Anonymity: the capability of authentication to hide the actual identity of the user. Even if the attacker obtains all data from the communication channel, they cannot get the actual identity of the user.
- Untraceability and unlinkability: The capability of an authentication protocol to update secrets in every session, hence safeguarding against attackers tracing the location of the user and utilizing the data obtained from the public channel to link and guess the owner of the message.
- Perfect Forward and Backward Secrecy (PFBS): The capability of an authentication protocol to ensure that even if an attacker steals the device, they cannot obtain past or future secret data.
- Resolve Desynchronization and Withstand DoS Attack: The capability of the authentication protocol to withstand DoS attacks, even if the attacker attempts to flood the communication channel with fake data to make it stop. However, the authentication must handle it to ensure that the communication can still operate.
- Unclonable device and withstanding Cloning Attack: the capability of the authentication protocol to achieve an unclonable device and withstand cloning attacks. Even if the attacker obtains the data, they cannot clone the device.
- Machine Learning attack: The machine learning attack is an attacker that collects challenge-response pairs to create a pattern of challenge responses and create a clone of PUF. The capability of the protocol is to withstand machine learning attacks by updating the challenge response in every session. In addition, the challenges are protected in every communication, hence the attacker cannot get challenge-response pairs.

IV. Our Proposed Authenticaion Protocol

This section details our protocol, starting with the system structure and cryptography notations, followed by the assumptions and our proposed authentication protocol.

A. System Structure and Cryptography Notations

The system architecture consists of two parties: the user/device of IoT and the Server. Firstly, the user/IoT device registers with the Server. In the registration phase, each party stores the secret which will be used in the next phase. The second phase conducts mutual authentication between the device and Server to establish a session key. Based on the possession of the key, each party can authenticate mutually and make the secure communication. In this architecture, the IoT device could be a smartphone, tablet, personal computer, wearable sensor, implantable sensor for patients, smart drone, smartwatch, or smart card. The system architecture is shown in Fig. 1.

Figure 1. System architecture.

In addition, the authors use the cryptography notations in Table I throughout our proposed protocol.

TABLE I. The Notations of Proposed Scheme

Notation	Descriptions
ID_D	The IoT Device Identity
K_{DS}	Secret key to communication between server and IoT Device
PUF_D	The device is equipped with a Physically unclonable function
$\|\|$	Concatenation Operation
C_D, R_D	Challenge—Response Pairs
PID_{DS}	Pseudo Identity
$h()$	Hash function
$\oplus$	Exclusive-OR operation
PID_{Syn}	Pseudo identity of synchronization is used if occur loss of synchronization where $\boldsymbol{PID_{Syn}} = \{\boldsymbol{pid_1}, \dots, \boldsymbol{pid_n}\}$
K_{Syn}	Key of synchronization is used if occur loss of synchronization where $\boldsymbol{K_{Syn}} = \{\boldsymbol{pk_1}, \dots, \boldsymbol{pk_n}\}$
C_{Syn}, R_{Syn}	Challenges-response synchronization of synchronization, where $C_{Syn} = \{c_1, c_2, \dots, c_n\}$ and $R_{Syn} = \{r_{syn_1}, r_{syn_2}, \dots, r_{syn_n}\}$
$Execute(\pi^d, \pi^s)$	The adversary can intercept all message
$Send(\pi^d, m)$	The adversary can alter, delete, and reply message
$CorruptDevice(\pi^d)$	The adversary steals device A and extract the credential using side channel attack.
$Programming(\pi^d)$	The adversary operates the PUF and FE by programming without them from OBU
$Test(\pi^d, \pi^s)$	The adversary operates testing oracle is like the flips of the coin. The attacker guests the C, and if the C=C', the adversary is the winner. Otherwise, the adversary is lost.
π^d	Oracle of device
π^s	Oracle of Server
m	message
q_h	Query of hash
q_p	Query of PUF
q_s	Query of sending message
l_1	Bits of biometric
l_2	Bits of Identity
$\|Hash\|$	Range Space of Hash Function
$\|PUF\|$	Range Space of PUF

B. The Assumptions

Our protocol has the following assumptions:

- The IoT device is equipped with a PUF.
- The masking method is implemented to safeguard against side-channel attacks.
- The IoT device has limited resources.
- The server has unlimited resources.

C. The Authentication Protocol

Our protocol has two steps: Registration and Mutual Authentication between the IoT device and Server.

1) Registration

This subsection presents the registration phase, where the registration is done via a secure channel. The authors also present the registration phase in Fig. 2.

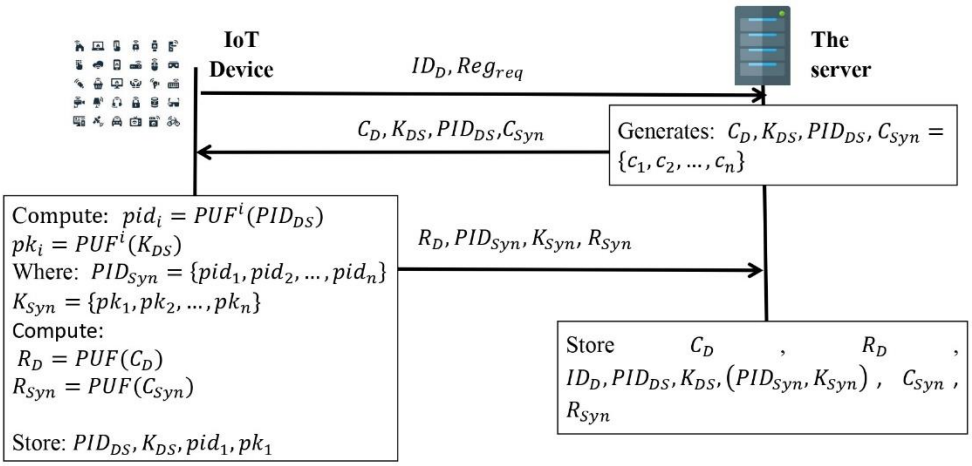

Figure 2. The registration between IoT device and server.

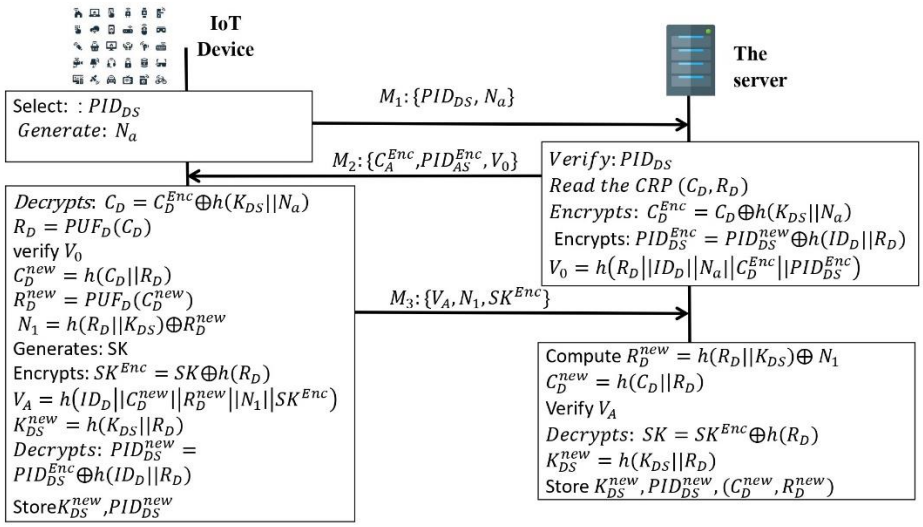

Figure 3. Mutual authentication and establishing session key between IoT device and server.

Step 1: Initially, the user inputs their identity, and sends the identity along with the registration request $\{ID_D, Reg_{req}\}$ to the Server

Step 2: The Server generates the challenge, shared key, pseudo identity, and synchronization of challenge C_D, K_{DS}, PID_{DS}, $C_{Syn} = \{c_1, c_2, \dots, c_n\}$. The server sends $\{C_D, K_{DS}, PID_{DS}, C_{Syn}\}$ to the IoT device.

Step 3: The IoT device generates synchronization of {Pseudo identity and Key} using PUF, where $PID_{Syn} = \{pid_1, pid_2, \dots, pid_n\}$ and $K_{Syn} = \{pk_1, pk_2, \dots, pk_n\}$. $pid_i = PUF^i(PID_{DS})$ and $pk_i = PUF^i(K_{DS})$. In addition, IoT device generates response and synchronization response $R_D = PUF(C_D)$ and $R_{Syn} = PUF(C_{Syn})$. The IoT device sends the response and synchronization {Pseudo identity, key, and response} $R_D, PID_{Syn}, K_{Syn}, R_{Syn}$ to the Server—finally, the IoT device Stores: $PID_{DS}, K_{DS}, pid_1, pk_1$.

Step 4: After the server receives R_D, PID_{Syn}, K_{Syn}, and R_{Syn}, the server stores C_D, R_D, ID_D, PID_{DS}, K_{DS}, (PID_{Syn}, K_{Syn}), C_{Syn}, and R_{Syn}.

2) Mutual authentication

This subsection presents the mutual authentication phase between the IoT device and the server, as well as the established session key. The authors also show the authentication phase in Fig. 3. The mutual authentication phase proceeds as follows.

Step 1: IoT device selects pseudo identity PID_{DS}, generates Nonce N_a, and sends them to the server

Step 2: The server receive $M_1: \{PID_{DS}, N_a\}$, and then, the server verifies PID_{DS} in database and located

challenge-response pairs (C_D, R_D). The server protected challenge by a secret key and Nonce where the formulation $Encrypts$: $C_D^{Enc} = C_D \oplus h(K_{DS}||N_a)$. The server also updated the pseudo-identity and protected it by secret identity and private response; the formulation is Encrypts: $PID_{DS}^{Enc} = PID_{DS}^{new} \oplus h(ID_D||R_D)$. The server create authentication code $V_0 = h(R_D||ID_D||N_a||C_D^{Enc}||PID_{DS}^{Enc})$. The server sends M_2:$\{C_A^{Enc}$,PID_{AS}^{Enc},$V_0\}$ to the IoT device

Step 3: The IoT device receives M_2:$\{C_A^{Enc}$,PID_{AS}^{Enc},$V_0\}$, and then the device of IoT decrypts C_D^{Enc} to obtain the actual challenge, the decryption formula is $Decrypts$: $C_D = C_D^{Enc} \oplus h(K_{DS}||N_a)$. The IoT generates a response using PUF $R_D = PUF_D(C_D)$. The IoT device verifies V_0 to ensure the data is from the legitimate server; the communication will be terminated if the verification process is invalid. Otherwise, The IoT device updates the challenge-response $C_D^{new} = h(C_D||R_D)$, $R_D^{new} = PUF_D(C_D^{new})$ and protected new response by secret {key and response} $N_1 = h(R_D||K_{DS}) \oplus R_D^{new}$ and also N_1 as a nonce to ensure the freshness. The IoT device generated session key SK and protected it by private response $SK^{Enc} = SK \oplus h(R_D)$. The IoT device computes the verification code $V_A = h(ID_D||C_D^{new}||R_D^{new}||N_1||SK^{Enc})$, updates the new secret key $K_{DS}^{new} = h(K_{DS}||R_D)$ and obtain new pseudo identity by decrypting $PID_{DS}^{new} = PID_{DS}^{Enc} \oplus h(ID_D||R_D)$. The IoT device sends M_3:$\{V_A, N_1, SK^{Enc}\}$ and stores new {pseudo identity and secret key} K_{DS}^{new},PID_{DS}^{new}.

Step 4: The Server received M_3:$\{V_A, N_1, SK^{Enc}\}$. The server obtains new response by decrypting $R_D^{new} = h(R_D||K_{DS}) \oplus N_1$ and compute the new challenge $C_D^{new} = h(C_D||R_D)$. The server verifies $V_A$to ensure the message is from a legitimate user/IoT device. The server will be terminated communication if the verification is invalid. Otherwise, the server obtains the session key by decrypting $Decrypts$: $SK = SK^{Enc} \oplus h(R_D)$ and computes a new secret key $K_{DS}^{new} = h(K_{DS}||R_D)$. Finally, the server stores $K_{DS}^{new}, PID_{DS}^{new}$, (C_D^{new}, R_D^{new}).

V. The Informal Analysis

This section presents the analysis of the security properties fulfilled by the proposed protocol. The analysis follows the capability of the attacker in Section III-A, and the proposed protocol achieves the security requirements in Section III-E. The security properties are as follows:

A. The Proposed Protocol Achieves Mutual Authentication

The Server authenticates the IoT device based on Pseudo Identity, actual Identity, and Challenge-Response $\{PID_{DS}, ID_D, (C_D, R_D)\}$. The IoT device authenticates the Server based on real Identity, challenge-response $\{ID_D, (C_D, R_D)\}$. In addition, our protocol preserves freshness based on the nonce. Therefore, the proposed protocol achieves mutual authentication.

B. The Proposed Protocol Achieves Anonymity

Our authentication protocol uses pseudo-identity in communication. Hence, the attacker cannot unveil the user's identity even if applying the DY adversary model provides the attacker with all the data from the communication channel.

C. The Proposed Protocol Achieves Untraceability and Unlinkability

The proposed protocol updates every data and sends different data each session. Even if the attacker gets all data from the communication channel, they cannot link the past and future data. Hence, the proposed protocol achieves unlikability. In addition, based on the DY structure (in Section III-A), if the challenge C_D is made public, the attacker may have an opportunity to trace the user. The proposed protocol protects the challenge C_D using the secret key K_{DS}, and encrypts it as follows $C_D^{Enc} = C_D \oplus h(K_{DS}||N_a)$. Furthermore, the proposed protocol is equipped with masking to avoid side-channel attacks, so the attacker cannot obtain the private key or a response. Therefore, the proposed protocol achieves both untrace ability and unlikability.

D. The Proposed Protocol Achieves PFBS

PFBS is achieved if the adversary cannot obtain past and future secrets. Since the proposed authentication protocol updates the private key, pseudo-identity, and challenge-response in every session, the attacker cannot obtain the confidential data even if they get access to all data from the communication channel. If the adversary operates the DY model adversary, they can obtain private data under a side-channel attack. However, because the proposed protocol uses masking, it is protected from side-channel attacks, and the attacker cannot access confidential data from the device's memory. Therefore, the proposed protocol achieves PFBS.

E. Our Protocol Resolves Desynchronization Issues

The proposed protocol offers synchronization of secret, pseudo-identity, and challenge-response. If desynchronization occurs, one of the parties initiates to send the synchronization of secret, pseudo-identity, and challenge-response to the other, and following the authentication phase, the proposed protocol resolves the desynchronization issues.

F. Resilience against Impersonation Attack

The proposed protocol uses pseudo-identity to communicate between IoT devices and servers. In addition, the proposed protocol protects the challenge with the secret key in every communication process. Even if the attacker gets all data from the communication channel and steals the IoT device, it is difficult for the attacker to impersonate the IoT device because the proposed protocol uses a masking method to safeguard against side-channel attacks, making it tricky to obtain confidential data. The attacker cannot compute the response and obtain the actual identity. Therefore, the proposed protocol resists impersonation attacks.

G. Resilience against Dos Attack

The attacker floods the communication network with fake data to make it seem like communication has stopped. However, the proposed protocol uses synchronization of

{Key, Challenge, Pseudo Identity}. If communication stops or a loss of synchronization occurs, one of the parties in the proposed protocol initiates sending synchronization {Key, Challenge, Pseudo Identity}, and communication continues. Therefore, the proposed protocol can resist a DoS attack.

H. Resilience against Cloning Attack

The IoT device in the proposed protocol is equipped with a PUF, which makes it resistant to cloning attacks. Additionally, the proposed protocol updates the challenge-response in every session and protects the challenge with a secret key. Thus, the attacker cannot collect the challenge-response from the communication channel. Moreover, the protocol is designed to withstand a learning attack aimed at cloning a device.

I. Resilience against Tracking Attack

The proposed protocol replaces the actual identity with a pseudo-identity to enable communication between the parties and the server. Even if the attacker obtains all the data from the communication channel, they cannot track the IoT device. Therefore, the proposed protocol is resistant to tracking attacks.

J. Resilience against Machine Learning Attack

The proposed protocol protects the challenge with a key and utilizes the masking method to resist side-channel attacks. Even if the attacker steals the IoT device and attempts to extract the secret from the memory device, the attacker cannot obtain the private data. The proposed protocol safeguards under DY model modification (as shown in Section III-A). The attacker cannot collect the challenge response. Therefore, the proposed protocol resists machine learning attacks.

VI. Mathematical Model Using RoR Model

The mathematical model was conducted to ensure that our protocol using PUF is secure. In the RoR model [41, 42], the attacker's capabilities are as described in [28], and The authors modify the adversary model as detailed in Section III-A.

Theorem 1. In the proposed protocol, the identity and response are kept secret, and the adversary operates the oracle, which adopts the knowledge password of Zipf's law [43]. The adversary estimates the Session Key based on a secret identity and response by guessing l_1 and l_2. The equation for the estimates mentioned above is as follows.

$$Adv_{P,A}^{AKE}(t) \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}$$

The proposed oracle has four game steps, denoted by G_i and $i \in (0,4)$ [23, 44]. If the adversary correctly guesses C'=c in game G_i, the adversary is successful. The game is as follows.

$\boldsymbol{G_0}$: Game 0 is the beginning of the game.

$$Adv_{P,A}^{AKE}(t) = |2.\Pr[SUCCESS_0] - 1| \quad (1)$$

$\boldsymbol{G_1}$: The adversary executes $Execute(\pi^d, \pi^s)$ and obtains all the messages: M_1:$\{PID_{DS}$, $N_a\}$, M_2:$\{C_A^{Enc}, PID_{AS}^{Enc}, V_0\}$, and M_3:$\{V_A, N_1, SK^{Enc}\}$. In the proposed protocol, the session key is protected by the response: $SK^{Enc} = SK \oplus h(R_D)$. It is impossible for the adversary to obtain R_D. In game one, the adversary does not increase the probability of winning.

$$\Pr[SUCCESS_1] = \Pr[SUCCESS_0] \quad (2)$$

$\boldsymbol{G_2}$: The adversary sends an oracle. Because the proposed protocol does not have a collision of the hash function (as shown in Section III-C) [45], the relationship, according to the birthday paradox, is as follows.

$$|\Pr[SUCCESS_2] - \Pr[SUCCESS_1]| \leq \frac{q_h^2}{2|Hash|} \quad (3)$$

$\boldsymbol{G_3}$: Similar to $\boldsymbol{G_2}$, the proposed protocol has secure PUF; the definition of secure PUF in Section III-B and the equation follows.

$$|\Pr[SUCCESS_3] - \Pr[SUCCESS_2]| \leq \frac{q_P^2}{2|PUF|} \quad (4)$$

$\boldsymbol{G_4}$: The adversary conducts the corrupt oracle and attempts to extract confidential data from the device's memory. Unfortunately, the attacker cannot obtain private data because the proposed protocol utilizes a masking method. Additionally, the attacker conducts the oracle model. However, in the proposed protocol, Challenge C is encrypted by a secret key, hence the attacker cannot generate the private response R, even if the attacker conducts the programming oracle. The result of game four is as follows.

$$|\Pr[SUCCESS_4] - \Pr[SUCCESS_3]| \max\{C', q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\} \quad (5)$$

Finally, the adversary conducts the test oracle by flipping a coin and guessing c'.

$$\Pr[SUCCESS_4] = \frac{1}{2} \quad (6)$$

The combination of Eqs. (1), (2), and (6).

$$\frac{1}{2}Adv_{P,A}^{AKE}(t) = \left|\Pr[SUCCESS_0] - \frac{1}{2}\right| = \left|\Pr[SUCCESS_1] - \frac{1}{2}\right| = |\Pr[SUCCESS_1] - \Pr[SUCCESS_4]| \quad (7)$$

By applying the triangle inequality to Eqs. (3), (4), and (5), the proposed protocol obtains:

$$|\Pr[SUCCESS_1] - \Pr[SUCCESS_4]| \leq |\Pr[SUCCESS_1] - \Pr[SUCCESS_3]| + |\Pr[SUCCESS_3] - \Pr[SUCCESS_4]| \leq |\Pr[SUCCESS_1] - \Pr[SUCCESS_2]| + |\Pr[SUCCESS_2] - \Pr[SUCCESS_3]| + |\Pr[SUCCESS_3] - \Pr[SUCCESS_4]| \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\} \quad (8)$$

The resulting Eq. is obtained from Eqs. (7) and (8).

$$Adv_{P,A}^{AKE}(t) \leq \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2\max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}$$

The proposed protocol is secure under the RoR model.

VII. Programming Model Using Scyther Tool

The proposed programming model uses the Scyther tool [46–48] with two steps to obtain verification results in Scyther: Create Programming and Generate the Program.

The program in Fig. 4 and the verification protocol shown in Fig. 5 are used. The verification demonstrates that the proposed protocol can withstand attacks.

```
usertype String; hashfunction H; const XOR:Function;
const CON:Function; const PUF:Function;
macro Cdenc=XOR(Cd,H(CON(Kds,Na)));
macro PIDdsenc=XOR(PIDdsnew,H(CON(IDd,Rd)));
macro V0=H(CON(CON(CON(CON(Rd,IDd),Na),Cdenc),PIDdsenc));
macro Rd=PUF(Cd);
macro Cdnew=H(CON(Cd,Rd));
macro Rdnew=PUF(Cdnew);
macro N1= XOR(Rdnew,H(CON(Rd,Kds)));
macro SKenc=XOR(SK,H(Rd));
macro VA=H(CON(CON(CON(CON(IDd,Cdnew),Rdnew),N1),SKenc));
protocol ideal-PUF-IoT(IoTDevice,Server){
        role IoTDevice{
                secret PIDds, Rd, Cd, IDd,  PIDdsnew, Rdnew,Kds, SK;
                var Rdnew;
                fresh N1, Na;
                send_1(IoTDevice,Server,PIDds, Na);
                recv_2(Server,IoTDevice,Cdenc,PIDdsenc,V0);
                send_3(IoTDevice,Server,VA,N1,SKenc);
                claim(IoTDevice,Secret,Rdnew);
                claim(IoTDevice,Secret,Rd);
                claim(IoTDevice,Secret,SK);
                claim(IoTDevice,Niagree);
                claim(IoTDevice,Nisynch);
                claim(IoTDevice,Alive);
                claim(IoTDevice,Weakagree); }
        role Server{
                secret PIDds, Rd, Cd, IDd,  PIDdsnew, Rdnew,Kds, SK;
                fresh N1,Na;
                var PIDdsnew;
                recv_1(IoTDevice,Server,PIDds, Na);
                send_2(Server,IoTDevice,Cdenc,PIDdsenc,V0);
                recv_3(IoTDevice,Server,VA,N1,SKenc);
                claim(Server,Secret,SK);
                claim(Server,Secret,Rd);
                claim(Server,Niagree);
                claim(Server,Nisynch);
                claim(Server,Alive);
                claim(Server,Weakagree); } }
```

Figure 4. Programming of our protocol in Scyther tool.

Claim				Status	Comments
Ideal_PUF_IoT	IoTDevice	Ideal_PUF_IoT,IoTDevice1	Secret PUF(H(CON(Cd,PUF(Cd))))	Ok	No attacks within bound
		Ideal_PUF_IoT,IoTDevice2	Secret PUF(Cd)	Ok	No attacks within bound
		Ideal_PUF_IoT,IoTDevice3	Secret SK	Ok	No attacks within bound
		Ideal_PUF_IoT,IoTDevice4	Niagree	Ok	No attacks within bound
		Ideal_PUF_IoT,IoTDevice5	Nisynch	Ok	No attacks within bound
		Ideal_PUF_IoT,IoTDevice6	Alive	Ok	No attacks within bound
		Ideal_PUF_IoT, IoTDevice7	Weakagree	Ok	No attacks within bound
Server		Ideal_PUF_IoT,Server1	Secret SK	Ok	No attacks within bound
		Ideal_PUF_IoT,Server2	Secret PUF(Cd)	Ok	No attacks within bound
		Ideal_PUF_IoT,Server3	Niagree	Ok	No attacks within bound
		Ideal_PUF_IoT,Server4	Nisynch	Ok	No attacks within bound
		Ideal_PUF_IoT,Server5	Alive	Ok	No attacks within bound
		Ideal_PUF_IoT,Server6	Weakagree	Ok	No attacks within bound

Figure 5. The verification results.

VIII. Competencies and Time Complexity Comparison

This section presents a comparison of the capabilities and time complexity of authentication protocols, including Aman *et al*. [1], Gope and Sikdar [2], Bian *et al.* [5], Gope *et al.* [6], Lee *et al.* [7], and the proposed authentication protocol.

A. Authentication Protocol's Competencies

This subsection shows the comparison of authentication protocol' competencies as follows:

TABLE II. Security Properties Comparison

Security Properties	Aman *et al*. [1]	Gope and Sikdar [2]	Bian *et al.* [5]	Gope [6]	Lee *et al*. [7]	Our Authentication protocol
Mutual Authentication	Y	Y	Y	Y	Y	Y
Anonymity.	N	N	Y	Y	Y	Y
Untraceability and Unlinkability	N	N	Y	N	N/A	Y
PFBS	N	N	N/A	N	N	Y
Resolving desynchronization	N	Y	N	Y	N/A	Y
Resilience against impersonation attack	N	Y	Y	N	Y	Y
Resilience against DoS attack	N	Y	N	Y	N/A	Y
Resilience against cloning attack	N	Y	Y	Y	N	Y
Resilience against tracking attack	N	N	Y	N	N	Y
Resilience against Machine Learning attack	N	N	N/A	N	N	Y

Note: Y denotes Yes, N denotes No, N/A denotes Not Available

Based on the DY adversary model (Section III-A), Table II indicates that Aman *et al.*'s [1] authentication protocol only achieves mutual authentication. Gope and Sikdar's protocol performs mutual authentication, resolves desynchronization issues, and resists impersonation and DoS attacks. Bian *et al.* [5] protocol achieves mutual authentication, anonymity, resilience against impersonation, cloning, and tracking attacks. Gope's [6] protocol achieves mutual authentication, untraceability, unlinkability, anonymity, resilience against DoS and cloning attacks. Lee *et al.* [7] protocol achieves anonymity, mutual authentication, and resilience against impersonation attack. However, only the proposed authentication protocol fulfills all security properties in Table II, such as mutual authentication, anonymity, untraceability, unlinkability, resolving desynchronization issues, PFBS, and withstanding impersonation, DoS, cloning, tracking, and machine learning attacks.

B. Time Complexity Comparison

Using Java Cryptography Library [49], The authors demonstrate various cryptography algorithms derived from the protocols in (i.e., PUF, MAC, hash function, and AES) and (i.e., PUF, hash function, and FE). The authors use a virtual machine to demonstrate the actual environment. The specification is processor RAM 32 G, CPU: 5×5.0 GHz ARM Cortex-A45 for the IoT device, Intel Core i7 CPU, RAM 8 GB for the server. The result of comparisons in time complexity in Fig. 6 shows that our protocol is the lowest.

The notation of the Execution Time of each cryptography algorithm, ET: Execution Time, ET_H: Execution Time of Hash Function, ET_{MAC} : Execution Time of Message Authentication Code, $ET_{SE/SD}$: Execution Time of Encryption and Decryption from Symmetric Algorithm (AES CBC), $ET_{FE.Gen}$: Execution Time of Fuzzy Extraction Generator, $ET_{FE.Rec}$: Execution Time of Fuzzy Extraction Reconstructor, ET_{PUF}: Execution Time of Physical Unclonable Function.

The execution time of each algorithm in IoT devices, ET_{PUF}: $0.410\ s$, ET_H: $0.314\ s$, ET_{MAC}: $3.721\ s$, $ET_{SE/SD}$: $1.275\ s$, $ET_{FE.Gen}$: $2.369\ s$. The execution time of each algorithm in the server, ET_H: $0.184\ s$, ET_{MAC}: $2.064\ s$, $ET_{SE/SD}$: $0.931\ s$, $ET_{FE.Rec}$: $1.471\ s$. The total execution time of Aman *et al.*'s protocol is $4ET_H + 6ET_{MAC} + ET_{SD/SE} + 2ET_{PUF} = 21.377\ s$, Gope and Sikdar's protocol is $10ET_H + ET_{FE.Gen} + ET_{FE.Rec} + 2ET_{PUF} = 7.15\ s$, Bian *et al.* protocol is $2ET_{FE.Gen} + 2ET_{FE.Rec} + 14ET_h + ET_{PUF} = 12.667\ s$, Gope's protocol is $ET_{FE.Gen} + ET_{FE.Rec} + 12ET_h + 2ET_{PUF} = 7.648\ s$, Lee *et al.*'s protocol is $34ET_h + 2ET_{FE.Gen} + ET_{PUF} = 13.744\ s$, and our protocol is $5ET_H + 2ET_{PUF} = 2\ s$. Our protocol has the lowest execution time, as shown in Table III and Fig. 6.

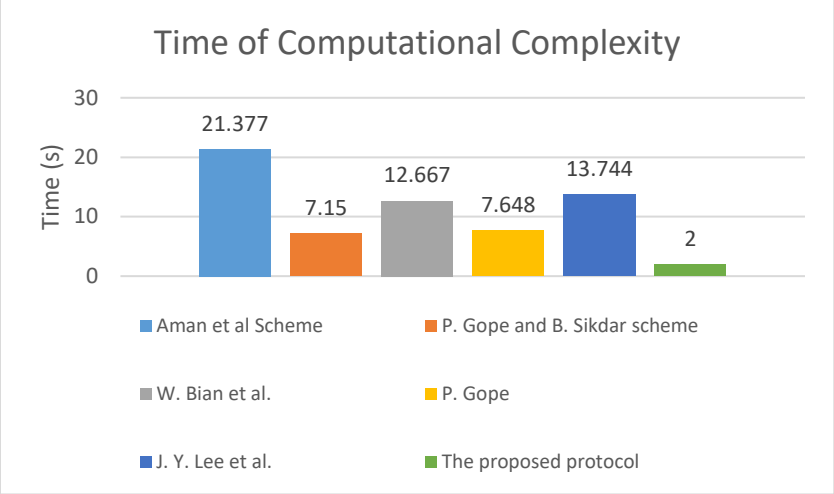

Figure 6. Comparison of computational complexity.

TABLE III. EXECUTION TIME COMPARISON

Authentication Protocol	IoT Device		The Server		Total (second)
Aman *et al.* [1]	$2ET_H + 3ET_{MAC} + ET_{SD} + 2ET_{PUF}$	13.886 s	$2ET_H + 3ET_{MAC} + ET_{SE}$	7.491 s	21.377 s
Gope and Sikdar [2]	$5ET_H + ET_{FE.Gen} + 2ET_{PUF}$	4.759 s	$5ET_H + ET_{FE.Rec}$	2.391 s	7.15 s
Bian *et al.* [5]	$2ET_{FE.Gen} + ET_{FE.Rec} + 7ET_h + ET_{PUF}$	9.908 s	$ET_{FE.Rec} + 7ET_h$	2.759 s	12.667 s
Gope [6]	$ET_{FE.Gen} + 6ET_h + 2ET_{PUF}$	5.073 s	$ET_{FE.Rec} + 6ET_h$	2.575s	7.648 s
Lee *et al.* [7]	$18ET_h + 2ET_{FE.Gen} + ET_{PUF}$	10.8 s	$16ET_h$	2.944 s	13.744 s
The proposed authentication protocol	$2ET_H + 2ET_{PUF}$	1.448 s	$3ET_H$	0.552 s	2 s

IX. CONCLUSION

In this manuscript, the authors initially proposed a powerful adversary model by modifying the DY adversary model. Then, they showed that the existing authentication protocol is vulnerable to modifications in the DY model analysis. The existing protocols fail to achieve PFBS, anonymity, and untraceability, and are unable to resist impersonation and machine learning attacks. Additionally, the previous protocols fail to achieve untraceability and PFBS. The authors propose an authentication protocol that provides security properties, including PFBS, mutual authentication, untraceability, anonymity, and unlinkability. The proposed protocol also resists tracking, impersonation, machine learning, and cloning attacks. The mathematical model's result (RoR model) shows that the proposed protocol is secure. The programming model (Scyther tool) also shows that the proposed protocol is safe and can withstand attacks. Moreover, the comparison of competencies and time complexity indicates that the proposed protocol fulfills security properties and has the lowest computational complexity with a computational time of 2 s. Therefore, the proposed protocol resolves the security issues of the authentication protocol using PUF and can generally solve the security issues in the IoT environment.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

E. Haodudin Nurkifli and Tzonelih Hwang conducted the research, literature review, analysis and wrote the manuscript; finally, all authors approved the final version.

FUNDING

National Cheng Kung University funded this research, No. MOST 109-2221-E-006-168-; No. MOST 108-2221-E-006-107.

ACKNOWLEDGMENT

The authors thank National Cheng Kung University, Tainan, Taiwan.

REFERENCES

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

[2] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, 2019.

[3] M. N. Aman, U. Javaid, and S. Member, "A privacy-preserving and scalable authentication protocol for the internet of vehicles," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1123–1139, 2021.
[4] M. N. Aman, M. H. Basheer, S. Member, and S. Dash, "HAtt : Hybrid remote attestation for the internet of things with high availability," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7220–7233, 2020.
[5] W. Bian, P. Gope, Y. Cheng, and Q. Li, "Bio-AKA: An efficient fingerprint based two factor user authentication and key agreement scheme," *Futur. Gener. Comput. Syst.*, vol. 109, pp. 45–55, 2020.
[6] P. Gope, "PMAKE: Privacy-aware multi-factor authenticated key establishment scheme for advance metering infrastructure in smart grid," *Comput. Commun.*, vol. 152, pp. 338–344, 2020.
[7] J. Y. Lee *et al.*, "PUFTAP-IoT: PUF-based three-factor authentication protocol in IoT environment focused on sensing devices," *Sensors*, vol. 22, no. 18, pp. 1–24, 2022.
[8] M. El-hajj, A. Fadlallah, M. Chmoun, and A. Serchrouchni, "A survey of Internet of Things (IoT) authentication schemes," *Sensors (Basel).*, vol. 19, no. 5, pp. 1–43, 2019.
[9] F. Meneghello *et al.*, "IoT : Internet of threats ? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, 2019.
[10] R. Román-castro and J. López, "Evolution and trends in IoT security," *Computer (Long. Beach. Calif).*, vol. 51, no. 7, pp. 16–25, 2018.
[11] Y. Deng, C. Chen, W. Tsaur, Y. Tang, and J. Chen, "Internet of Things (IoT) based design of a secure and lightweight Body Area Network (BAN) healthcare system," *Sensors (Basel).*, vol. 17, no. 12, 2919, 2017.
[12] B. D. Deebak, F. Al-turjman, M. Aloqaily, and O. Alfandi, "An authentic-based privacy preservation protocol for smart e-healthcare systems in IoT," *IEEE Access*, vol. 7, pp. 135632–135649, 2019.
[13] M. Shuai, B. Liu, N. Yu, and L. Xiong, "Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks," *Secur. Commun. Networks*, vol. 2019, 14, 2019.
[14] K. Yeh, "A secure IoT-based healthcare system with body sensor networks," *IEEE Access*, vol. 4, pp. 10288–10299, 2017.
[15] A. Sivasangari, S. Bhowal, and R. Subhashini, "Secure encryption in wireless body sensor networks," in *Proc. IEMIS (Emerging Technol. Data Min. Inf. Secur*, 2018, vol. 3.
[16] T. Mcgrath *et al.*, "A PUF taxonomy A PUF taxonomy," *Appl. Phys. Rev.*, vol. 6, 011303, 2019.
[17] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A Tutorial," *IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.
[18] A. Babaei and G. Schiele, "Physical unclonable functions in the internet of things: State of the art and open challenges," *Sensors (Basel).*, vol. 19, no. 14, 3208, 2019.
[19] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 2007 44th ACM/IEEE Des. Autom. Conf. San Diego*, 2007, pp. 9–14.
[20] W. Choi, S. Kim, Y. Kim, Y. Park, and K. Ahn, "PUF-based encryption processor for the RFID systems," in *Proc. 2010 10th IEEE Int. Conf. Comput. Inf. Technol.*, 2010, pp. 2323–2328.
[21] S. Kleber, F. Unterstein, M. Matousek, F. Kargl, F. Slomka, and M. Hiller, "Secure execution architecture based on puf-driven instruction level code encryption," *IACR Cryptol. ePrint Arch. 2015*, 651, 2015.
[22] Y. Guo, T. Dee, and A. Tyagi, "Barrel shifter physical unclonable function based encryption," *Cryptoraphy*, vol. 2, no. 3, pp. 1–20, 2018.
[23] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. INFORMATICS*, vol. 15, no. 9, pp. 4957–4968, 2019.
[24] E. H. Nurkifli and T. Hwang, "A secure Lightweight authentication scheme in IoT environment with perfect forward and backward secrecy," in *Proc. 2022 7th Int. Work. Big Data Inf. Secur.*, 2022, pp. 113–118.
[25] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, 2019.
[26] C. Serkan, "PUF-enhanced offline RFID security and privacy," *J. Netw. Comput. Appl.*, vol. 35, pp. 2059–2067, 2012.
[27] K. Suleyman and K. S. Mehmet, "A novel RFID distance bounding protocol based on physically unclonable functions," in *Proc. International Workshop on Radio Frequency Identification: Security and Privacy Issues*, vol. 7055, pp. 78–93, 2012.
[28] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, pp. 198–208, 1983.
[29] S. Garg, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Ind. Informatics*, vol. 16, no. 5, pp. 3548–3557, 2020.
[30] S. Bhunia and M. Tehranipoor, "Side-channel attacks," *Hardw. Secur.*, pp. 193–218, 2019.
[31] M. N. I. Khan, S. Bhasin, B. Liu, A. Yuan, A. Chattopadhyay, and S. Ghosh, "Comprehensive study of side-channel attack on emerging non-volatile memories," *J. Low Power Electron. Appl.*, vol. 11, no. 4, 2021.
[32] C. Bohm and M. Hofer, *Physical Unclonable Functions in Theory and Practice*, New York: NY, USA: Springer, 2012.
[33] P. Tuyls and L. Batina, "RFID-Tags for Anti-counterfeiting," in *Proc. Top. Cryptol. CT-RSA (LNCS 3860)*, 2006, pp. 115–131.
[34] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Adv. Cryptology—EUROCRYPT'2004*, 2004, pp. 523–540.
[35] D. Jeon, J. H. Baek, D. K. Kim, and B. Choi, "Toward zero bit-error-rate physical unclonable function: Mismatch-based vs. physical-based approaches in standard cmos technology," in *Proc. 2015 Euromicro Conf. Digit. Syst. Des.*, 2015, pp. 407–414.
[36] K. Chuang *et al.*, "A physically unclonable function using soft oxide breakdown featuring 0% native," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2765–2776, 2019.
[37] X. Lu, S. Member, L. Hong, and S. Member, "CMOS optical PUFs using noise-immune process-sensitive photonic crystals incorporating passive variations for robustness," *IEEE J. Solid-State Circuits*, vol. 53, no. 9, pp. 2709–2721, 2018.
[38] W. Wang, Y. Yona, S. N. Diggavi, and P. Gupta, "Design and analysis of stability-guaranteed PUFs," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 4, pp. 978–992, 2018.
[39] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7881, pp. 142–159, 2013.
[40] H. B. Kim, S. Hong, and H. S. Kim, "Lightweight conversion from arithmetic to Boolean masking for embedded IoT processor," *Appl. Sci.*, vol. 9, no. 7, 2019.
[41] M. Abdalla and D. Pointcheval, "Simple password-based encrypted key exchange protocols," *Lect. Notes Comput. Sci.*, vol. 3376, pp. 191–208, 2005.
[42] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. IACR Int. Conf. Public-Key Cryptogr.*, 2005, vol. 3386, pp. 65–84.
[43] D. Wang, S. Member, H. Cheng, P. Wang, and S. Member, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2776–2791, 2017.
[44] S. Roy *et al.*, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Trans. Ind. Informatics*, vol. 15, no. 1, pp. 457–468, 2020.
[45] E. H. Nurkifli and T. Hwang, "Untraceable and unclonable sensor movement in the distributed IoT environment," *IEEE Sens. J.*, 2022.
[46] C. Cremers, "Scyther user manual," *CISPA Helmholtz Cent. Inf. Secur.*, pp. 2–52, 2014.
[47] C. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Proc. Int. Conf. Comput. Aided VeriFcation*, 2022, vol. 72, no. 1, pp. 1195–1212.
[48] F. Zhu, P. Li, and H. Xu, "A lightweight RFID mutual authentication protocol with PUF," *Sensors (Basel).*, pp. 1–22, 2019.
[49] Oracle. Java Cryptography Architecture (JCA). [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html