Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6367579, 9 pages
https://doi.org/10.1155/2022/6367579

Research Article

PUFDCA: A Zero-Trust-Based IoT Device Continuous Authentication Protocol

Shrooq Alshomrani and Shancang Li

School of Computer Science and Informatics, Cardiff University, Wales, UK

Correspondence should be addressed to Shancang Li; shancang.li@ieee.org

Received 29 June 2022; Accepted 9 September 2022; Published 11 November 2022

Academic Editor: Muddesar Iqbal

It is very challenging to secure the Internet of Things (IoT) systems, which demands an end-to-end approach from the edge devices to cloud or hybrid service. The exponential growth besides the simple and low-cost nature of IoT devices has made IoT system an attractive target for several types of security attacks such as *impersonation*, *spoofing*, and *DDoD* attacks. This work is aimed at enhancing the IoT security using a zero-trust (ZT) approach by proposing a physical unclonable function-based device continuous authentication (PUFDCA). The PUFDCA provides two kinds of authentications to verify the identity of the IoT device, static authentication to verify the identity before starting the session using PUF technology and continuous authentication to verify the location of the device during the session to ensure the authenticated device is not changed. The security analysis and verification tool results demonstrate that the proposed protocol is secure against a range of common IoT attacks. In addition, PUFDCA is considered lightweight and consumes low energy and storage.

1. Introduction

The Internet of Things (IoT) is significantly changing the way we are living by making our lives smarter [1], which links different systems (such as *smart home*, *intelligent building*, *vehicle-to-roadside*, and *smart cities*). The IoT is rapidly transforming and innovating the business models for many sectors by better managing for both users' and service providers' benefits [2]. However, the growing connected IoT devices and the complicated IoT ecosystem also increase security vulnerability in the edge-computing environment [3], in which a growing volume of private (e.g., personally identifiable information (PII)) or sensitive contents are involved.

It is important to secure IoT systems from the edge device to the remote cloud or hybrid services cross the IoT ecosystem. An IoT system may involve a large number of IoT devices, such as smart sensors, infrastructures, and cloud servers, which makes it very challenging to fully secure the complicated system because of the diversity of the ecosystem, e.g., *vendors*, *hardware*, *operating systems*, and *locations* [4]. Many current IoT systems involve a number of infrastructures or devices not designed for IoT or connectivity that may increase the potential security risks. Smart devices, e.g., *intelligent sensor* and *RFID devices*, usually lack resources in terms of computing, storage, and power supply, which are unable to afford security solutions used in network. On the other hand, IoT devices are typically deployed in a diverse environment, from the smart home, roadside, to critical infrastructure worksite, which can increase the attack surfaces [5].

In the past few years, the zero-trust security model shows great potential in securing the complicated IoT systems [6]. Unlike the traditional castle-and-moat security models, which deploy firewalls, intrusion detection/prevention system (IPS/IDS) to block external attacks of the network perimeter, the zero-trust models do not rely on the firewall based security perimeter but require strict verification for every access by following the "never trust, always verify" principle [7, 8]. The zero-trust security model assumes everything in the network (e.g., *users*, *devices*, *applications*, *networks*, *infrastructure*, and *data*) are untrusted, and explicit verification is required for every access from both inside and outside [9].

Zero-trust security model can ensure the secure identity management and limit access, including the *user/device authentication*, *resource access control*, and *risk detection* [10, 11]. Zero-trust authentication solution, e.g., *push notification authentication*, can ensure IoT service/resource providers be aware that an authentication attempt is taking place, which usually can be implemented via apps on mobile. Unlike authentication in traditional security model, where an authenticated user might be able to operate unauthorized operations, in zero-trust, each request to access data or service needs to be reauthenticated and reauthorized. This helps limit device/user to gain unauthorized access to data or services and avoid the potential over authorization. Specifically in the complicated IoT ecosystem, a zero-trust authentication solution needs to address the following requirements:

(1) Strong device identity. Devices need to be registered and issued with renewable tokens before making a decision for authentication requests

(2) Least privilege access. Granular access control should be implemented to limit the access privilege in case it may have been compromised or request unapproved service

(3) Continuous authentication. Robust and continuous authentication can ensure the devices are up to date

Based on the above requirements, in this work, we developed a continuous zero-trust IoT device authentication solution based on physical unclonable function. The main contributions are summarised as follows:

(i) A physical unclonable function based device continuous authentication (PUFDCA) is proposed for device authentication in IoT ecosystem

(ii) An identity of IoT devices verification algorithm is developed that can conduct static authentication to verify the identity before starting the session using PUF

(iii) Comprehensive security analysis verified the proposed solution

In the following sections, we introduced the related works and detailed PUFDCA framework.

2. Related Works

In the past few years, lots of research efforts have been conducted on zero-trust authentication. Shah et al. proposed a lightweight continuous device-to-device authentication (LCDA) for zero-trust architecture (ZTA), which includes three stages, *initialisation*, *mutual authentication*, and a *continuous authentication* stage [7]. (1) In the *initialisation phase*, the edge device and the gateway exchange each other's identities and generate authentication information that will be used in the following phases. (2) In the *mutual authentication phase*, the device and gateway devices mutually authenticate each other as well as agreeing on a session key based on the Channel State Information (CSI). (3) In the final *continuous authentication phase*, the devices use the previously derived CSI-based keys to agree on a shared secret and then incorporate the key into the algorithm to apply continuous mutual authentication. However, the protocol requires the IoT device to store the identity of the gateway in addition to its own identity, which could consume a large amount of storage at the IoT device [7].

A novel zero-trust authentication scheme for IoT communication is proposed by Bhattacharjya and Saiedian [12], in which an IoT system may consist of four actors: *IoT device*, *user*, *gateway*, and a *delegate*, which includes other users or devices that may connect to the IoT device. In this solution, Bhattacharjya assumed that every device has only one user and any other users are considered delegates. The framework proposes all communication to the device happens through the gateway; in this way, the device is prevented from responding to any external requests. In terms of methodology, each pair of devices and the user is provided with a unique key. The device and the gateway validate each other through the signature that is generated using their private keys. During transmission, SSL/TLS is used to secure the data against replay and other attacks. Bhattacharjya and Saiedian tested and evaluated the proposed framework in IoT environment that was built from scratch based on the four actors, which means that the stability of this system would need to be tested in a typical IoT system [13].

Several recent studies have introduced blockchain technology to provide authentication solutions in a ZT framework for IoT systems. Chen et al. propose a ZT scheme based on blockchain for the power IoT [14]. The scheme uses blockchain in identity management to satisfy the unification requirements since blockchain provides many benefits such as the decentralised and confidential storage of the identity information of all blocks. The timestamp and random number in the header of each block allow replay attacks to be prevented, as well as stop attackers from cracking identity information offline. Blockchain-based solutions require further study in its application to IoT systems for two reasons. First, it requires a high storage capacity, while storage is limited in IoT devices; second, it cannot store the unprecedented amount of data produced by the IoT.

Connected IoT devices can now be found everywhere, including homes, cafes, and factories. The zero-trust approach is a strategic initiative that contributes to preventing data breaches by eliminating the concept of trust. The core principle is "never trust, always verify," which is based on the fact that there are no trustworthy areas and each access must be evaluated and approved. According to Rose et al. [8], ZT provides several ideas that are designed to reduce uncertainty and provide least privilege for each access request in information systems, while zero-trust architecture is a cyber-security plan that uses ZT concepts and composes component relationships, access policies, and workflow planning. The definition highlights the essence of the issue, which is eliminating unauthorized access to data and services, as well as restricting access control as accurately as

possible. To minimise uncertainty, authentication should be addressed, and the number of implicit trust areas reduced.

ZT is based on a collection of principles to achieve the concept of "never trust, always verify." Rose et al. define the major principles as follows [8]:

(i) The resource includes data sources, services, and computational capabilities, which must be protected

(ii) Trust is not granted automatically, and there is no trusted access by default; and the least privilege concept must be enforced

(iii) There is no constant access in zero-trust, and access to the resource is allowed only per session

(iv) Devices behaviour, identities, and environmental properties are the core of granting access

(v) Access request to an IoT resource is not granted statically but also reevaluated

In this work, we address the above principles and propose new zero-trust-based authentication. Specifically, PUF is used to create an identity for each device.

3. Proposed Method

The proposed PUFDCA protocol uses static and continuous authentication for IoT devices under the concept of ZT. The static authentication relays on PUF as a unique fingerprint for each device, while the CA uses wireless channel characteristics in the form of CSI as wireless fingerprints to verify the location of the IoT device during the session.

3.1. PUF-Based Identify. The biometric system can effectively identify people's identities because of the uniqueness of these features. Similarly, the PUF provides a unique method for verifying integrated circuits (ICs). In PUF, the inherent variability in IC manufacturing is used to apply challenge-response (CR) functions, where the output is based on the input and physical microstructure of the device [15, 16]. Typically, IoT devices perform authentication using traditional methods such as digital signature and secret keys. However, two reasons make these methods unsuitable for IoT devices, the first of which is that the simplicity and low cost of IoT devices prevent them from performing digital signature and encryption, which require high power. The second reason is that managing secrets in IoT devices is unfeasible since secrets are stored in nonvolatile memories or battery-backed RAMs that may be read by various attacks [17].

In ZT, the PUF can be used as an effective authentication technique for IoT devices for several reasons [16]. First, the PUF provides efficient, low-cost security, and second, it can also provide security for IoT systems without storing secrets in the devices, which is considered a lightweight solution. Finally, PUF is unique at device level, the variation in the physical factors during the fabrication process of ICs makes it practically impossible to replicate the microstructure. This feature provides a unique identity for each device, which is the core of any successful authentication scheme. Unlike traditional periodic authentically model, in which users can only authenticate once to access network, the continuous authentication (CA) enables users continuously authenticate accessing resources in zero-trust scenario.

3.2. Channel State Information. While Wi-Fi communication is used to exchange data between devices, it can also be used for security purposes, such as the use of CSI to identify the location of the device. Shah et al. conducted a feasibility analysis to demonstrate that CSI is changing by changing the location of the devices [7]. Shah et al. concluded that the change in the location of the devices was directly reflected in CSI values. Thus, the receiver can estimate the CSI to detect any impersonation or tampering attempts.

In particular, the preamble in each 802.11n Wi-Fi frame allows the receiver to estimate the impact of the wireless channel when the sender and receiver are on the signal. According to [18], the estimation of CSI and the transmitted data on each subcarrier (sc) can be formalised by a linear system. Let T denote the signals strength of antennas of the sender on each subcarrier, and let R^{sc} represent the corresponding received signals, including the channel coefficient matrix. Then, the linear equation can be described as

$$R^{sc} = H^{RC}_{R\times T} \times T^{sc}. \tag{1}$$

During communication, let M^{sc}_R denote the CSI metric, as the receiver collects CSIs to obtain and store τ, which is the CSI measurements $M^{sc}_R, i \in 1 \cdots \tau$. The purpose of collecting these CSI measurements is to use them in the comparison of the new CSI $M^{sc}_R, i > \tau$.

To identify the location of the device, the distance D_i between the stored and the new CSI measurements can be calculated as follows:

$$D_i = \frac{1}{\tau}\sum_{j=1}^{\tau}\sqrt{\sum_{r=1}^{R}\sum_{sc=1}^{SC}\left(\frac{\left|M^{sc}_{r,i}\right|}{\left\|M^{sc}_{r,i}\right\|_2^{sc,r}} - \frac{\left|M^{sc}_{r,j}\right|}{\left\|M^{sc,j}_{r,j}\right\|_2^{sc,r}}\right)^2}. \tag{2}$$

To determine if the location is changed, a threshold can be set. Here, the maximum distance is a simple and straightforward method of selecting the proper threshold:

$$D^{\tau}_{i,j} = \sqrt{\sum_{r=1}^{R}\sum_{sc=1}^{SC}\left(\frac{\left|M^{sc}_{r,s,i}\right|}{\left\|M^{sc}_{r,s,i}\right\|_2^{sc,r}} - \frac{\left|M^{sc}_{r,s,j}\right|}{\left\|M^{sc}_{r,s,j}\right\|_2^{sc,r}}\right)^2}. \tag{3}$$

Then, we have

$$\max\left(D^{\tau}\right) = \max_{\forall i,j}\left(D^{\tau}_{i,j}\right). \tag{4}$$

If D_i is greater than the threshold γ, this means the suspicious event is detected $(q_i = 1)$. Otherwise, the change in

the location is considered accepted behaviour $(q_i = 0)$.

$$q_i = \begin{cases} 0 & \text{if } D_i < \gamma \\ 1 & \text{if } D_i \geq \gamma. \end{cases} \quad (5)$$

3.3. PUFDCA Procedures. The PUFDCA protocol has three stages, which can be summarised as follows:

(1) Initial stage. In this stage, the challenge-response pair (CRP) and ID pairs are obtained in a secure manner. It is assumed that before the protocol starts, the server stores ID and CR pairs for each IoT device

(2) Static authentication stage. In this phase, the server authenticates the IoT device using PUF technology

(3) Continuous authentication stage. In this stage, the CSI measurements are used to verify the location of the IoT device during the session

Figure 1 indicates the detailed flow chart for the proposed PUFDAC framework.

Figure 2 depicts the PUFDCA framework through the timeline, in which the green blocks indicate the static authentication using PUF before starting the session, while the black blocks indicate the continuous authentication during the session. After passing the static authentication phase, the time of the current session T is determined, and the continuous authentication is applied after each t until the end of the session. These stages are applied on each access request, which means to start another session, the device requires reauthentication and so on. In Figure 2, t is randomly specified for each session, the PUFDCA framework, the green block denotes static authentication using PUF, the black block represents continuous authentication using CSI, and the red block ends the authentication session.

In PUFDCA framework, we have the following assumptions:

(1) The IoT device is assumed as a device with limited storage and processing abilities, while the server is secured and has no such limitation

(2) The protocol considers the PUF and the device's microcontroller on the same chip, and it is not possible to separate them, meaning that it is impossible to remove PUF

(3) The initial stage is assumed to be completed in a secure environment

The PUFDCA framework includes the following four properties as summarised:

(1) Static authentication: the server verifies the identity of the IoT device and only grants per session the access. This is fulfilled by using PUF at the beginning of the establishing of a session

(2) Continuous authentication: the communication should be consistently verified to ensure that the IoT device that started the session remains unchanged during the session

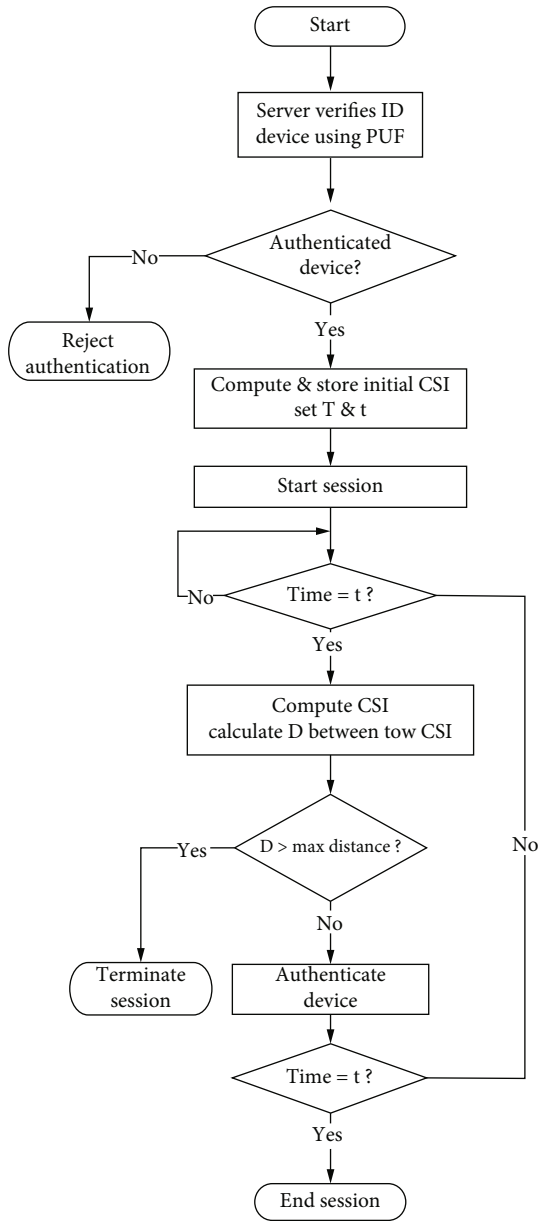

Figure 1: Flowchart for PUFDCA framework.

(3) Message encryption: the confidentiality of exchanged secrets should be protected

(4) Data integrity: the exchanged data should be assured that it is not changed

The PUFDCA authentication is a multiple stage process, which includes three key phases: *initial phase*, *static authentication phase*, and *continuous authentication phase*. The following subsections will provide details these three stages of the PUFDCA. The notations defines the notations of the protocol.

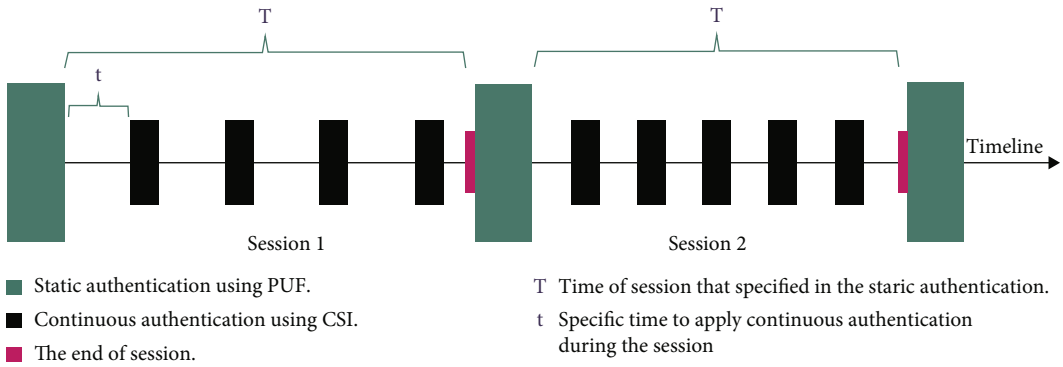

Figure 2: PUFDCA framework.

3.3.1. Initial Phase. This phase is assumed to be applied before the protocol starts in a secure environment. Figure 3 illustrates that the server obtains and securely stores a (C_i, R_i) and (ID_i) pair for each IoT device. This is considered an important step for successful and secure subsequent authentication phases.

3.3.2. Static Authentication Phase. In this stage, the server verifies the identity of the IoT device to decide whether it is a legitimate device. Figure 4 illustrates the steps of this stage, as follows.

(1) As a first step, the IoT device sends the server its (ID_i) accompanied by a random number $nonce_i$. The $nonce_i$ is based on the pseudorandom number generation (PRNG) method

(2) The server searches for the (ID_i) of the IoT device and retrieves the corresponding $CRP(C_i, R_i)$ from its memory. In the case that the server does not find the ID_i in its memory, the authentication request will be declined. Otherwise, a secret random number (N_A) is generated and used to hide R_i in the next message of the protocol. To assure data integrity, the protocol uses a message authentication code (MAC)

(3) In the third step, the IoT device uses its PUF to retrieve R_i from C_i. Then, R_i is used to get N_A, while MAC is used to verify the freshness and integrity of the message. After that, the IoT device produces a new challenge C_{i+1} using the secret random numbers N_A and N_B. The new challenge C_{i+1} is input to the device's PUF to get the new secret response R_{i+1}. Then, the IoT device sends both NB and R_{i+1} to the server

(4) In the last step of the static authentication, the server obtains NB and R_{i+1} using R_i and verifies the message using the MAC; if the MAC is not verified, communication will be terminated, and otherwise, the device will be authenticated. Consequently, the server computes and stores the initial CSI of the authenticated device to use it in the CA, setting T and t, in which T is the time of the communication session and t is the time to apply the CA during the session. Moreover, the server constructs a new challenge and stores the new $CRP(C_{i+1}, R_{i+1})$ using the N_A and N_B against ID_i in its memory. In the above steps, if the MAC is not verified at any point, the authentication request will be declined

At the end of the static authentication stage, the IoT device and server will clear all the temporary numbers such as N_A, N_B, and $nonce_i$.

3.3.3. Continuous Authentication Stage. This phase starts after the IoT device has been authenticated in the static authentication stage. Once the time T of the session starts, the device will be constantly authenticated. After each period of time t, the server uses the received CSI and compares it with the CSI information of the legitimate device that was measured in the static authentication. As shown in Figure 5, the server authenticates the device by computing the distance Di between these two CSI measurements. In case the distance is less than the maximum allowed distance γ, the server will authenticate the device. The main purpose of this process is to ensure whether the location of the device has changed. If the distance exceeds the maximum allowed distance, the session will be terminated as it is considered an unauthenticated device, and otherwise, the session will be continued. This step is continuously repeated after each t until the end of the time session.

4. Experimental Results Analysis

This section describes the simulation that is used to verify the security properties of the proposed static authentication. Since the CA stage is performed constantly from the server side while it receives the sensed data without requiring acknowledgement from the IoT device side. In other words, the server senses the channel, computes the CSI, and determines the location with no need to send or receive any identity or secrets from the IoT device. Thus, the design is sufficient to analyze and verify the security properties of the CA.

In this work, we use the security protocol description language (SPDL) and *Scyther* to model the roles of the authentication parties in the zero-trust scenarios. *Scyther* is a secure automatic verification tool that outperforms other

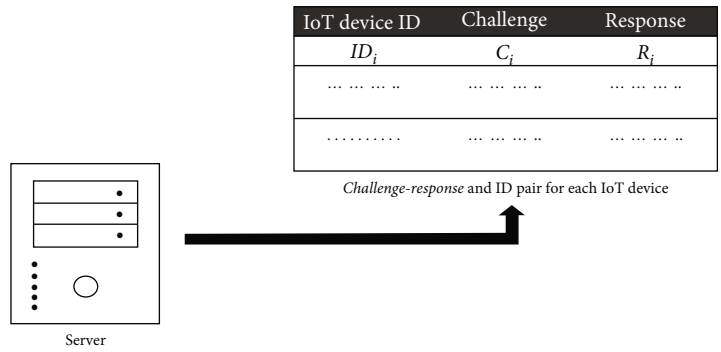

Figure 3: Initial phase of PUFDCA.

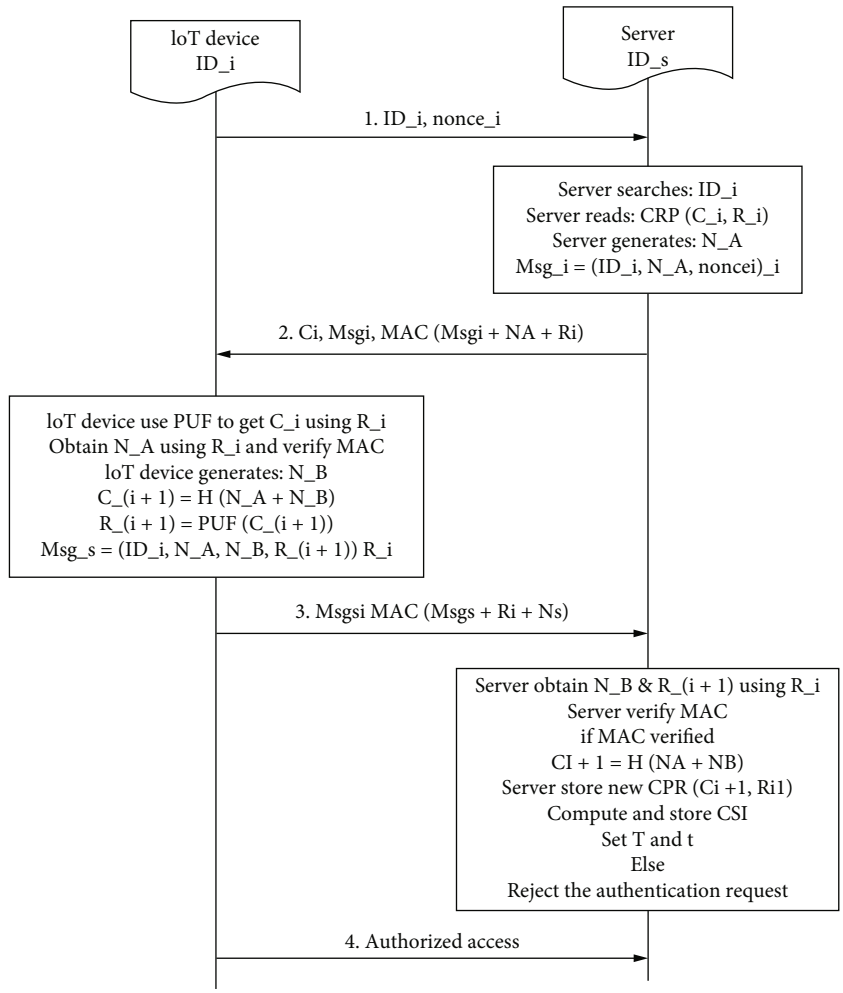

Figure 4: Static authentication phase.

tools such as *ProVerif* [19]. Scyther includes a formal method to simulate the protocol and identify potential vulnerabilities, while it uses a security protocol description language (SPDL) to model the roles of the communicating parties. To specify the security requirements, the SPDL uses claims including alive, secret, commit noninjective synchronisation (NiSynch) and noninjective agreement (NiAgree).

4.1. Security Analysis. Formal security analysis is based on the *Scyther* tool result which demonstrated that there were no potential attacks. Figure 6 indicates the formal verification result of PUFDCA in *Scyther*. To analyze the result, the *Scyther* claims were classified into security properties, involving *confidentiality*, *freshness*, *forward secrecy*, and resistance to impersonation and replay attacks [20, 21].

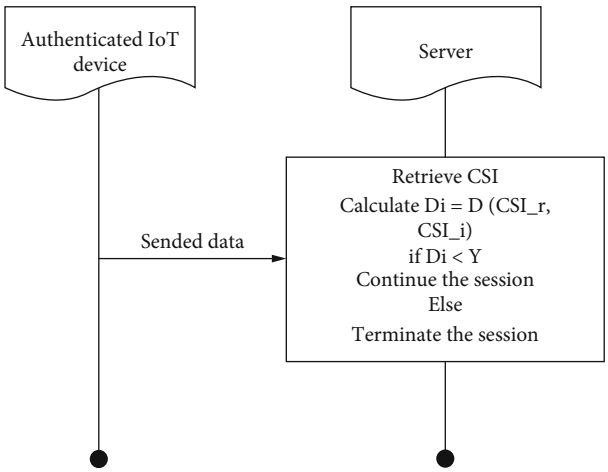

Figure 5: Continuous authentication state.

The security properties can be analyzed for the proposed solution.

4.2. Confidentiality. This claim is achieved when the exchanged messages are confidential. The claim secret is used to evaluate the secrecy of keys and exchanged messages between the sender and receiver.

4.3. Freshness. This claim is achieved when the two parties are synchronised and agree about the exchanged variables. It can be defined using noninjective synchronisation and noninjective agreement claims as NiSynch and NiAgree [22].

4.4. Forward Secrecy. This claim is achieved when the keys are unique and frequently changed. It can be defined using the secret claim.

4.5. Resistance to Impersonation Attack. This claim is achieved when the sender and receiver communicate each other, which enhances the identification of any impersonation attacks. It can be defined using the Weakagree claim.

In resistance to replay attack, this claim is achieved when each party is alive in the same run of the protocol and not just replaying old messages. This requires the two parties to agree on both synchronisation and aliveness in the same run and is formally defined using NiSynch and NiAlive claims.

4.6. Informal Security Analysis. This section addresses a detailed analysis of the security features of the PUFDCA. The purpose of this analysis is to demonstrate that PUFDCA is secure and efficient and aligns with ZT principles.

4.6.1. Resistance against MITM Attack. In a MITM attack, an active attacker intercepts and controls the communication between two communicating parties. This results in compromising privacy by manipulating the exchanged message while the parties believe that they are directly communicating with each other. The PUFDCA is considered resistant to this kind of attack since the MITM attacker needs to know the R_i, N_A, and N_B to construct valid data in order to start the attack. Knowing all these is not possible for an active adversary that places itself on the communication line between the two communicating parties. Moreover, in the CA phase, the server constantly validates the location of the device, which can identify whether the source of the message has changed.

4.6.2. Forward Secrecy. The forward secrecy property is aimed at protecting current and future communication sessions by generating a unique secret for every session, meaning that compromising a single session will not impact the following sessions. In the static phase, the PUFDCA achieves forward secrecy property by generating a new random number for each message, each time. For example, each message of 1, 2, and 3 messages includes a new random number $nonce_i$, N_A, and N_B, respectively. Therefore, if the current session is compromised by the attacker, the subsequent sessions cannot be compromised. Moreover, even if the session is compromised during the CA, this will not affect the next session because the device must pass through a new static authentication to start the next session.

4.6.3. Resistance against Impersonation Attack. In an impersonation attack, the attacker successfully impersonates the identity of one of the communicating parties; since the adversary is able to play the role of one of the legitimate parties, this causes the other party to complete the session while accepting the fake identity. The proposed scheme is immune against this attack for two reasons, the first of which is that the protocol uses the PUF technology, which cannot be reproduced and has its own unique CRP. This means that it is impossible for any two IoT devices to have the same PUF. Second, the protocol uses the location to constantly ensure the legitimacy of the device, so if the attacker attempts to impersonate the device from anywhere, the CA will detect the suspicious location and deny the communication.

4.6.4. Resistance against Physical Attack. The simple nature of the IoT devices makes them easily accessed by adversaries. In a physical attack, the attacker can physically access the IoT device to extract secret keys and consequently clone the device. The PUFDCA is considered secure against this kind of attack for two principal reasons. The first reason is that the IoT device does not store any secrets in its memory, and the second is that the devices' microcontroller and the PUF are on the same chip and communicate securely. Therefore, even if the IoT is physically available to the attacker, it cannot extract any of the secret keys from the IoT device.

4.6.5. Resistance against Replay Attack. In a replay attack, the transmitted messages are eavesdropped on by the attacker, which replays some of these messages to impersonate a legitimate party. In other words, the adversary eavesdrops on the communication and later resends it to misdirect the receiver into doing what the attacker wants. In PUFDCA, if the attacker replays the previous messages, this will not violate the security of the protocol because each time, in each message; the protocol uses a new random number such as nonce

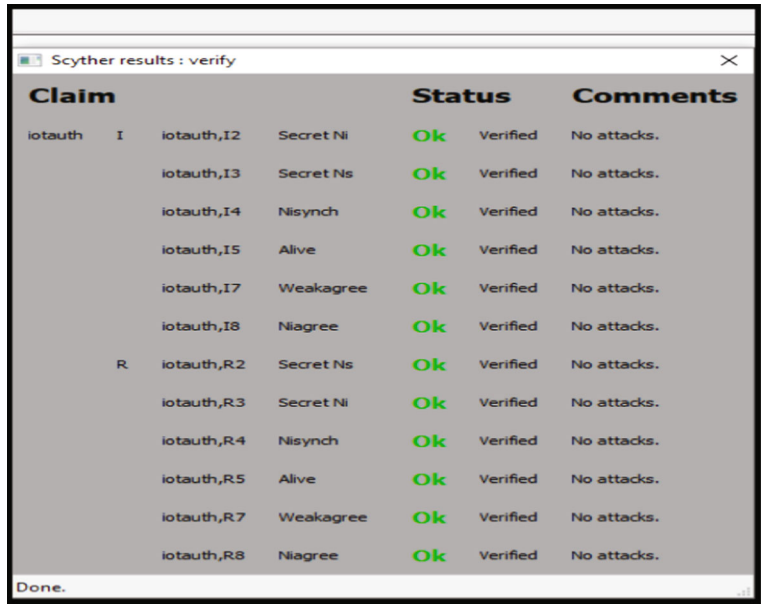

Figure 6: Static authentication phase.

i, *NA*, and *NB* in messages 1, 2, and 3, respectively. Furthermore, the CA reinforces the resistance against this attack by verifying the sender's location and comparing it with the location of the authenticated device.

4.6.6. Low-Cost Energy. Since the IoT device has low energy, it is important to ensure that the technologies used in this protocol respect this limitation. In PUFDCA, PUF technology is fast, with a very small silicon footprint and low energy consumption. All these features make it the best choice for low capability devices. In terms of the CA, the IoT device does not consume energy for computation. It is assumed that the server has high capability, so it is responsible for the computation of the distance between the CSI and deciding to continue or reject the session.

4.6.7. Lightweight. Being lightweight is the most important property in authentication protocols due to the low-energy nature of IoT and its low computation capabilities. This importance increases in the ZT concept where the device must be authenticated at every session. PUFDCA has both static and continuous authentication, which means that it has to use lightweight technologies to enhance the efficiency of the protocol. The proposed authentication is considered efficient in regard to the lightweight feature, using message authentication code (MAC), hash function, and XOR operation to ensure security. These methods are computational efficient and suitable for resource-constrained devices compared to other alternatives.

In particular, the protocol uses SHA-3 of hash function, which is secure and lightweight for IoT environments. Also, the MAC size used is 128 bits; this is very low compared to other signature schemes such as RSA signature, which uses a range of 128 to 256 bytes. In addition, the protocol can be considered to have low storage as it stores only one CRP for each IoT device, while the IoT device only needs to store its ID. During the CA, the IoT device does not need to make any computations as the high-capability server is responsible for this.

4.6.8. Data Integrity. Data integrity ensures that the receiver gets the original message from the sender without any changes. Typically, the attacker attempts to manipulate the content of the message and change it to a different message. In PUFDCA, the data integrity is assured by using MACs with new secrets in every authentication request, which makes the protocol immune to content manipulation.

5. Conclusion and Discussion

The PUFDCA is aimed at applying the ZT concept in IoT environments to provide secure communication without granting implicit trust to the IoT device. The aim was achieved by authenticating the IoT device at the beginning of the session using PUF technology, in addition to continuously verifying the location of the device during the session using CSI. The formal and informal analysis assured that PUFDCA is resistant to common attacks. Also, PUFDCA took into consideration the balance between the lightweight and security properties to be suitable to be applied in each session in IoT environments.

Notations

$\mathbf{ID_i}$: ID of an IoT device
$\oplus$: XOR operation
$\mathbf{H(X)}$: Hash of X
$\mathbf{C_i}$: challenge for the i'th iteration
$\mathbf{R_i}$: response of a PUF to input C_i
$\mathbf{C_{i+1}}$: New challenge
$\mathbf{R_{i+1}}$: New response
$\mathbf{nonce_i}$: Random number generated by PRNG

$\mathbf{N_A}, \mathbf{N_B}$: The secret random numbers
$\mathbf{T}$: The time of the session
$\mathbf{t}$: Certain point of time
$\mathbf{D_i}$: The distance between two CSI
$\boldsymbol{\gamma}$: The maximum allowed distance

Data Availability

No datasets were generated or analyzed during the current study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Domingo-Ferrer, J. Soria-Comas, and R. Mulero-Vellido, "Steered microaggregation as a unified primitive to anonymize data sets and data streams," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3298–3311, 2019.

[2] Y. Gong, S. Mabu, C. Chen, Y. Wang, and K. Hirasawa, "Intrusion detection system combining misuse detection and anomaly detection using genetic network programming," in *2009 ICCAS-SICE*, pp. 3463–3467, Fukuoka, Japan, 2009.

[3] M. Yuan, L. Chen, P. S. Yu, and T. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 633–647, 2013.

[4] D. Kushner, "The real story of stuxnet," *IEEE Spectrum*, vol. 50, no. 3, pp. 48–53, 2013.

[5] N. Helwig, E. Pignanelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 210–215, Pisa, Italy, 2015.

[6] G. Falco, C. Caldera, and H. Shrobe, "IIoT cybersecurity risk modeling for scada systems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4486–4495, 2018.

[7] S. W. Shah, N. F. Syed, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "LCDA: lightweight continuous device-to-device authentication for a zero trust architecture (ZTA)," *Computers & Security*, vol. 108, article 102351, 2021.

[8] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero trust architecture*, National Institute of Standards and Technology, 2020, Techechnical Report.

[9] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, "Never trust, always verify: a multivocal literature review on current knowledge and research gaps of zero-trust," *Computers & Security*, vol. 110, article 102436, 2021.

[10] S. Li, "Editorial: zero trust based internet of things," *EAI Endorsed Transactions on Internet of Things*, vol. 5, no. 20, 2020.

[11] S. Zhao, S. Li, F. Li, W. Zhang, and M. Iqbal, "Blockchain-enabled user authentication in zero trust internet of things," in *International Conference on Security and Privacy in New Computing Environments*, pp. 265–274, Lyngby, Denmark, 2020.

[12] S. Bhattacharjya and H. Saiedian, *A novel simplified framework to secure iot communications*, ICISSP, 2021.

[13] S. Li, M. Iqbal, and N. Saxena, "Future industry internet of things with zero-trust security," *Information Systems Frontiers*, pp. 1–14, 2022.

[14] Z. Chen, L. Yan, Z. Lü et al., "Research on zero-trust security protection technology of power IoT based on blockchain," *Journal of Physics: Conference Series*, vol. 1769, no. 1, p. 012039, 2021.

[15] Y. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electronics*, vol. 3, no. 2, pp. 81–91, 2020.

[16] A. Babaei and G. Schiele, "Physical unclonable functions in the internet of things: state of the art and open challenges," *Sensors*, vol. 19, no. 14, p. 3208, 2019.

[17] A. Durand and J. Pasquier, "Physical unclonable functions for iot security using free software," in *11th International Conference on the Internet of Things*, pp. 173–176, St.Gallen, Switzerland, 2021.

[18] I. E. Bagci, U. Roedig, I. Martinovic, M. Schulz, and M. Hollick, "Using channel state information for tamper detection in the internet of things," in *Proceedings of the 31st Annual Computer Security Applications Conference*, pp. 131–140, Los Angeles, CA, USA, 2015.

[19] P. Lafourcade, V. Terrade, and S. Vigier, "Comparison of cryptographic verification tools dealing with algebraic properties," in *International Workshop on Formal Aspects in Security and Trust*, pp. 173–185, Springer, 2009.

[20] H. Yang, V. A. Oleshchuk, and A. Prinz, "Verifying group authentication protocols by scyther," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 7, no. 2, pp. 3–19, 2016.

[21] Z. Mohammad, "Cryptanalysis and improvement of the YAK protocol with formal security proof and security verification via Scyther," *International Journal of Communication Systems*, vol. 33, no. 9, article e4386, 2020.

[22] S. Mondal, S. Mukherjee, and S. Banerjee, "Machine learning based malicious node detection in iot environment," in *Advanced Techniques for IoT Applications*, J. K. Mandal and D. De, Eds., pp. 316–326, Springer, Singapore, 2022.