Authenticated Key Agreement Protocol for Device-to-Gateway Communication in IoT

Cong Pu[‡] Jongho Seol[¶] Nohpill Park[‡] Dragan Korać[⊛]
[‡]Oklahoma State University, United States. Email: cong.pu@ieee.org, n.park@okstate.edu
[¶]Middle Georgia State University, United States. Email: jongho.seol@mga.edu
[⊛]University of Banja Luka, Bosnia and Herzegovina. Email: dragan.korac@pmf.unibl.org

***Abstract*—The advent of Internet of Things (IoT) ushers in a significant potential to integrate individuals, devices, and data, leading to a profound change in our professional and social environments. The small, resource-constrained IoT devices are usually deployed to collect various types of critical data in remote or unmonitored locations. Due to extensive interconnectivity, limited resources, and inadequate security design, IoT systems are vulnerable to communication-specific cyber threats, which aim to disrupt operations, steal sensitive information, or cause damage. To resolve the security concerns in IoT communications, many recent efforts have been devoted to designing authenticated key agreement protocols for IoT systems. However, not only most of the existing solutions fail to adopt cost-effective techniques for resource-limited IoT devices, but also they ignore the differentiation among various data types in the established session keys. A few approaches use traditional physical unclonable functions (PUFs) to address resource concerns, yet they introduce new security issues into IoT systems. Once the PUF cryptographic information is compromised by machine learning attacks, the entire authentication framework collapses. Therefore, in this paper we propose an authenticated key agreement protocol for device-to-gateway communication in IoT systems based on Chebyshev polynomial and probability-based PUF. We examine the proposed protocol's security features through formal security validation. We also conduct performance evaluation through a simulation-oriented study, and the results clearly prove that the proposed protocol offers superior security and privacy, while maintaining low computational overhead.**

***Index Terms*—Mutual authentication, key agreement, security, device-to-gateway, Internet of Things (IoT).**

I. Introduction

The Internet of Things (IoT) applications have seen a substantial rise across different domains in recent years. The cost-effective yet powerful IoT devices hold immense potential for flexible deployment across a variety of applications, including smart wristbands at interactive theme parks, intelligent medical devices for personalized healthcare, and real-time disaster management [1]. For example, the global IoT market is anticipated to attain a value of approximately $1.06 trillion by 2025, with the United States generating the highest revenue of $380 billion [2]. While the prospective benefits of IoT applications are broadly recognized, the breach and compromise of IoT communications could lead to disastrous outcomes, such as personal, financial, or business-critical information breaches, unauthorized access and control of devices and systems [3].

This work was supported by the National Science Foundation (NSF) through SaTC under Award 2333777.

IoT devices come with built-in sensors that collect data from their environment or internal state, and transmit the collected data to the IoT gateway through the Internet [4]. Then, the IoT gateway aggregates the collected data and forwards it to the IoT data center. Finally, the IoT data center evaluates the data to generate meaningful insights, and send the corresponding instructions back to the IoT devices to execute specific tasks or implement changes. However, the frequent interaction between the IoT gateway and IoT devices, and the fact that IoT systems leverage open networks to exchange data and instructions increase their susceptibility to cyber attacks [5]. For instance, hackers often target IoT systems because IoT devices may have weaker security and can become entry points to broader networks. In November 2023, a nation-backed hacking group called ''CyberAv3ngers'' launched repetitive attacks against Internet-exposed IoT devices used in water and wastewater systems in the United States [6]. Once these cyber attacks succeed, they may grant attackers control over these crucial parameters, allowing them to maliciously alter settings, potentially causing malfunctions or even full system shutdowns. As IoT security and citizen security are closely intertwined, it is essential to implement robust and reliable network and Internet security mechanisms to ensure secure communications and establish trusting relationships within IoT systems [7].

As ensuring the security and resilience of IoT systems against cyber threats starts with a robust first line of defense, numerous scholars focus their research on authenticated key agreement protocols [8]–[15]. In [8], [9], blockchain assisted authentication schemes are proposed for IoT networks, where the digital ledger is used to store identity verification details of users and/or IoT devices. The proposed authentication schemes establish a single secret key for the whole communication session, through which the IoT devices will deliver the collected data to the user. However, this one-for-all session key could cause potential data leakage, allowing this user to access data intended for other users. In addition, the existing authenticated key agreement solutions, i.e., [10], [11], have another drawback, which is predominantly based on highly resource-demanding operations, such as pairing and bilinear map operations, to offer security to IoT systems. To lessen the resource demands of cryptographic systems on IoT systems, many researchers opt for lightweight operations, such as traditional PUF [12], [14] and hash functions [13]. Unfortunately, these traditional PUF-based security mecha-

979-8-3315-9673-6/26/$31.00 ©2026 IEEE

nism are vulnerable to machine learning attacks that are able to restore the cryptographic PUF challenge-response pair, causing the entire security system collapse. Moreover, even though the approach in [13] only adopts resource-friendly hash functions, but it requires additional authentication factor such as biometrics, which might not be suitable for all IoT systems. Therefore, it is imperative to develop authenticated key agreement protocols that not only adopt cost-effective techniques, but also differentiate among various data types in the established session keys to ensure the protection of communications in IoT systems.

Driven by the discussion above, in this paper we propose a lightweight, data type-aware, and secure authenticated key agreement protocol for IoT systems. As the predominant communication mode in IoT systems involves IoT devices connecting and communicating with the gateway, the proposed protocol facilitates mutual authentication and key agreement for device-to-gateway communication. The authenticated key agreement for device-to-device communication falls outside the scope of this paper and will be proposed as a topic for future research. The main contributions of this paper can be summarized as follows: (i) the design of an authenticated key agreement protocol using Chebyshev polynomial [16] and probability-based PUF [17]; (ii) the formal security verification of the proposed protocol using AVISPA [18]; and (iii) the comparative performance study of the proposed protocol within a simulation-based experimental framework. The extensive evaluation proves that the proposed protocol is not only secure and dependable in adversarial scenarios but also outshines its peers by delivering better performance in computational efficiency and energy usage.

II. Related Work

In [19], the authors concentrate on the security issues of IoT networks and propose an authentication system based on a zero-trust architecture using radio frequency fingerprinting. Instead of using traditional password-based identity authentication, the radio frequency fingerprinting technique is used to identify the radio transmitter of a legitimate IoT device by its unique signal characteristics. The proposed authentication system does get rid of the trustable center, however, radio frequency fingerprinting is highly affected by environmental conditions such as interference or jamming. Thus, the proposed authentication system might not function properly in non-ideal environments. The authors in [20] use the chameleon hash function and traditional PUF to achieve mutual authentication and key agreement between industrial IoT devices. The rationale behind utilizing these techniques is to reduce resource consumption and enhance the physical security robustness of devices, respectively. Industrial IoT systems are usually employed for collecting a wide range of data concurrently, e.g., operational and workflow efficiency data. Thus, the proposed approach is not suitable for multi-taking industrial IoT systems because it does not distinguish the type of data during the authentication process.

To fix the data type aware issue in the authentication process, the researchers from the Internet of Drones domain share some ideas. In [21], each drone is preloaded with a set of data types to collect, while users register to receive certain types of data. Before establishing the session key between the drone and the user, the ground station needs to verify the eligibility of both the drone and the user for communicating the specified data type. In [22], the authors embed the type of data in the calculation of the session keys, which can only be used to encrypt data of the matching type. The above solutions might not be directly applicable in the IoT environment, as they were designed for mobile networks. However, in IoT systems the devices and the gateway are stationary.

An elliptic curve cryptography (ECC) based authentication protocol is proposed for medical IoT networks in [10], where the healthcare devices transmit the patient's data to the local gateway. In [23], an authentication protocol using bilinear pair mapping is proposed for medical IoT networks. Moreover, in [24] a PUF and ECC based authentication protocol is developed for healthcare networks, where health providers, gateways, and medical devices set up secure session keys before exchanging any sensitive data. However, the authentication and key agreement process of these protocols relies on pairing operations which result in significant computational burden on the resource-constrained medical IoT devices. In addition, these traditional PUF assisted approaches are vulnerable to machine learning attacks which are able to predict the challenge-response pairs.

III. Preliminary Background

In this section, we introduce Chebyshev polynomials [16] and their unique commutative property. Then, we present the design of probability-based physical unclonable function [17].

A. Chebyshev Polynomials

Suppose that α and β are two integers, $gcd(\cdot)$ is the greatest common divisor, p is a prime number, and n is a non-negative integer. A polynomial that meets the criteria below defines the Chebyshev polynomial of the first kind with degree n,

$$T_n(x) = \begin{cases} cos(n \cdot cos^{-1}(x)), 0 \leq x \leq 1 \\ cosh(n \cdot cosh^{-1}(x)), x \geq 1 \end{cases}$$

In accordance with the definition, the first kind Chebyshev polynomial of degree n has the following recurrence property

$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x),$$

where $T_0(x) = 1$ and $T_1(x) = x$. In addition, the first kind Chebyshev polynomials of degree n are commutative, i.e., the compositions of Chebyshev polynomials are also Chebyshev polynomials, as the example shown below,

$$T_n(T_m(x)) = T_m(T_n(x)) = T_{m \cdot n}(x),$$

where m and n are positive integers.

The commutative property of the first kind Chebyshev polynomials allows us to develop Diffie-Hellman key agreement algorithms. For example, Alice and Bob first agree on the first kind Chebyshev polynomials of degree n, $T_n(\cdot)$, the base value g, and a large prime number p. Alice chooses a secret integer pr_a as her secret key and calculates the public key $pu_a = T_{pr_a}(g)\ mod\ p$, and Bob selects his secret key pr_b and

computes $pu_b = T_{pr_b}(g)\ mod\ p$ as his public key. Later, Alice and Bob exchange their public keys, pu_a and pu_b. Finally, Alice and Bob negotiate their secret session keys, sk_a and sk_b, as $sk_a = T_{pr_a}(pu_b)\ mod\ p$ and $sk_b = T_{pr_b}(pu_a)\ mod\ p$. Here, $sk_a = sk_b$ and the proof is presented below,

$$\begin{aligned} \underline{sk_a} &= T_{pr_a}(pu_b)\ mod\ p = T_{pr_a}(T_{pr_b}(g))\ mod\ p \\ &= T_{pr_a \cdot pr_b}(g)\ mod\ p = T_{pr_b}(T_{pr_a}(g))\ mod\ p \\ &= T_{pr_b}(pu_a)\ mod\ p = \underline{sk_b}. \end{aligned}$$

B. Probability-Based Physical Unclonable Function

To combine information from the physical layer with cryptography, the natural randomness present in the physical materials and the environmental impact are considered in the process of generating unpredictable outputs based on the provided unique inputs. One example of realizing this idea is the probability-based physical unclonable function (Prob-PUF) [17]. In Prob-PUFs, the detrapping phenomenon in circuit transistors and the varying voltage applied to circuit transistors are seamlessly modeled to produce unique challenge-response pairs. However, in traditional PUFs the varying environmental factors, e.g., the varying voltage, are completely ignored, which makes them functional only in ideal situations. Because Prob-PUFs can provide enhanced security owing to their unpredictable and noisy outputs, they have emerged as a prime choice for high-security demanded applications.

In Prob-PUFs, the charging voltage for a set of selected circuit transistors is considered as the input, while the detrapping events of these circuit transistors serve as the output. In general, the PUF inputs are called the *challenges*, and the PUF outputs are referred to as the *responses*. The rationale behind this design is that the physical properties of the semiconductor materials and the deviations in the manufacturing process invest each circuit transistor with different detrapping property. However, the detrapping property for the same circuit transistor does not vary with time. Thus, the circuit transistors' detrapping properties can be harnessed to generate unique and unpredictable challenge-response pairs, which could play an important role in the process of authentication between two ends. On the end with Prob-PUF, if detrapping does not occur on a circuit transistor for all charging-sensing cycles, a bit '0' is created. Likewise, a bit '1' is created if detrapping occurs on a circuit transistor for all charging-sensing cycles. If a circuit transistor exhibits random detrapping behaviors throughout all charging-sensing cycles, a random bit, either '0' or '1', is returned. After observing detrapping events, all deterministic bits generated by the circuit transistors consist of the response.

In order to generate the same response at the other end, the detrapping probability of each circuit transistor is calculated according to $P^i_{trap}(t) = [1 - e(\frac{-t}{\tau^i_e})]$. Here, i indicates the i^{th} circuit transistor, $e(\cdot)$ represents the exponential function, τ^i_e denotes the constant coefficient of the i^{th} circuit transistor's detrapping property, and t stands for the elapsed time. And then, the calculated detrapping probability $P^i_{trap}(t)$ is compared with the predetermined low threshold TH^-_{trap} and high threshold TH^+_{trap}. If $P^i_{trap}(t) < TH^-_{trap}$, a bit '0' returned. If $P^i_{trap}(t) > TH^+_{trap}$, a bit '1' returned. If $TH^-_{trap} \leq P^i_{trap}(t) \leq TH^+_{trap}$, a random bit, either '0' or '1', is returned. Finally, the other end discards the random bits, and compares the deterministic bits between the PUF circuit output and the mathematically calculated value for the purpose of authentication. From the preceding discussion, two benefits of Prob-PUFs can be identified: (i) As the selected circuit transistors generate both deterministic and random bits, it is extremely challenging for attackers to restore the PUF response using machine learning techniques; and (ii) The two ends do not need to exchange or agree on challenge-response pairs for the authentication purpose. Instead, one end only needs to store the constant coefficient of the selected transistor's detrapping property, which can significantly reduce storage and communication overheads for security systems.

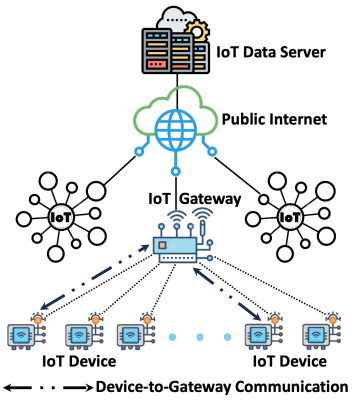

Fig. 1. Network model.

IV. Network and Adversarial Models & Security Requirements

A. Network Model

Fig. 1 shows the network model, where IoT devices are connected to the IoT gateway through public Internet channels (wired or wireless) [25]. It is generally acknowledged that the use of public Internet channels for IoT communications potentially introduces security risks, e.g., unauthorized data access. Thus, it is necessary to establish secret session keys for communications between the IoT devices and the IoT gateway. In the proposed network model, we only focus on device-to-gateway communication as it is the predominant communication mode in IoT systems. The device-to-device communication falls outside the scope of this paper. The communication between the IoT gateway and the IoT data server is assumed to be done through a secure Internet connection. The IoT gateway is considered as a trusted entity with sufficient resources, whose responsibility is to register each IoT device and send the data from the IoT devices to the IoT data server. In addition, we assume that each IoT device is resource-limited and equipped with a built-in Prob-PUF in the circuit. Instead of storing the cryptographic information in memory, the IoT devices can compute it as needed using their Prob-PUFs.

B. Adversarial Model

In the proposed adversarial model, we select the Canetti-Krawczyk (CK) framework [26], [27] to emulate the adver-

sary's behavior. Within the CK framework, the attackers have the capability of reading, creating, modifying, delaying, and replaying messages between communication entities, as well as initiating new authentication sessions and interacting with communication entities. In addition, the attackers can disclose details specific to a communication session, like session states and ephemeral keys, in order to compromise the security of key exchange protocols. Since the IoT devices are often deployed in remote or unattended locations where the physical security is a significant concern. The attackers can physically obtain the IoT devices and attempt to access to the cryptographic information stored on them. However, any attempt by an attacker to probe or alter the IoT device's integrated circuit will irreversibly change its slight physical variations, thereby destroying the circuit transistors' detrapping properties.

C. Security Requirements

Considering that the CK framework is used as the foundation of the adversarial model, the proposed protocol is designed to meet the following security requirements. First, the proposed protocol needs to be secure against various cyber attacks, such as physical probing, impersonation, modification, replay, and man-in-the-middle attacks. Second, the IoT devices should use a different pseudonymous identifier each time they communicate with the IoT gateway. Lastly, the identity verification should be done before negotiating data type-aware session keys between IoT devices and the IoT gateway.

V. The Proposed Protocol

In this section, we introduce an authenticated key agreement protocol for device-to-gateway communication in IoT systems.

A. System Initialization and IoT Device Registration

The IoT gateway G_s starts with two system functions, the first kind Chebyshev polynomial of degree n, $T_n(g)$, and a hash function that converts arbitrary bits into *m*-bit strings, $H(\cdot)$. The IoT gateway G_s also chooses its private key pr_s and computes the corresponding public key $pu_s = T_{pr_s}(g)$. To uniquely identify each IoT device in the database, the IoT gateway G_s uses the media access control (MAC) address as their true identifier. For instance, the i^{th} IoT device D_i's true identifier is D_i^*. In addition, the IoT gateway G_s assigns a set of data types $dt_i = [dt_{i.1}, dt_{i.2}, \cdots, dt_{i.k}]$ to the IoT device D_i to collect, generates a random number r_i and uses it to create the IoT device D_i's initial pseudonym $PD_i = H(D_i^* \parallel r_i)$. After that, the IoT gateway G_s randomly selects k circuit transistors $\Gamma_i = \{\tau_{i.1}, \tau_{i.2}, \ldots, \tau_{i.k}\}$ from the IoT device D_i's integrated circuit, obtains their corresponding detrapping constant coefficients $\Gamma_i^e = \{\tau_{i.1}^e, \tau_{i.2}^e, \ldots, \tau_{i.k}^e\}$. In order to get the same response as the IoT device D_i's Prob-PUF generates, the IoT gateway G_s determines the number of charging-and-sensing cycles Γ_i^{smp} and specifies the size of sampling window Γ_i^{ω} for the IoT device D_i's Prob-PUF. Finally, the IoT gateway G_s sends $\{pu_s, PD_i, \Gamma_i, \Gamma_i^{smp}\}$ to the IoT device D_i via a secure channel, and stores $\{D_i^*, PD_i, dt_i, \{\Gamma_i, \Gamma_i^e, \Gamma_i^{smp}, \Gamma_i^{\omega}\}\}$.

B. IoT Device and IoT Gateway Authenticated Key Agreement

1) The IoT device D_i monitors the detrapping events of the pre-agreed circuit transistors Γ_i for Γ_i^{smp} charging-sensing cycles to obtain the deterministic bits as its Prob-PUF response che_i, generates a random nonce r_i, and calculates $m_{i.1} = T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(g)$. Here, pr_i is the IoT device D_i's private key and the public key is $pu_i = T_{pr_i}(g)$. In addition, che_i and r_i are used to create the session key as well as compute a new pseudonym.
2) The IoT device D_i calculates $m_{i.2} = T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(pu_s)$ and $m_{i.3} = H(D_i^* \parallel r_i \parallel m_{i.1} \parallel dt_{i.k})$, forms an encrypted message $msg_i = \{D_i^* \parallel r_i \parallel m_{i.3} \parallel dt_{i.k}\}_{m_{i.2}}$ using a secure symmetric cryptosystem and $m_{i.2}$ as the key, and sends the message $\{PD_i, m_{i.1}, msg_i\}$ to the IoT gateway G_s via a public channel.
3) The IoT gateway G_s generates the IoT device D_i's Prob-PUF response che_i' using $P_{trap}^i(t) = [1 - e(\frac{-t}{\tau_e^i})]$, Γ_i, Γ_i^e, Γ_i^{ω}, and calculates $m'_{i.2} = T_{pr_s}(m'_{i.1})$. Here, $m'_{i.2} = T_{pr_s}(m'_1) = T_{pr_s}(T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(g)) = T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(T_{pr_s}(g)) = T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(pu_s) = m_{i.2}$. As the IoT gateway G_s's private key pr_s is only known to the gateway itself, $m'_{i.2}$ cannot be restored by anyone other than the gateway.
4) The IoT gateway G_s obtains $D_i^{*'}$, r'_i, $m'_{i.3}$, and $dt'_{i.k}$ through decrypting the encrypted message msg_i using $m'_{i.2}$. If $D_i^{*'}$ is not a registered entry in the database, the IoT gateway G_s terminates the authentication process. Otherwise, the IoT gateway G_s continues as follows.
5) The IoT gateway G_s calculates $m''_{i.3} = H(D_i^{*'} \parallel r'_i \parallel m'_{i.1} \parallel dt'_{i.k})$ and verifies it with $m'_{i.3}$. If the verification fails, the IoT gateway G_s terminates the authentication process. Otherwise, the IoT gateway G_s generates a random nonce r_s, calculates $m_{s.1} = T_{r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^*}(pu_i)$, creates a new pseudonym $PD_i^\star = H(D_i^* \parallel r'_i \parallel r_s)$, and then computes the session key $sk_{s.i} = T_{r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^*}(m'_{i.1})$.
6) The IoT gateway G_s calculates $m_{s.2} = H(D_i^{*'} \parallel sk_{s.i})$, forms an encrypted message $msg_s = \{D_i^{*'} \parallel r_s \parallel m_{s.1} \parallel m_{s.2}\}_{m'_{i.2}}$ using a secure symmetric cryptosystem and $m'_{i.2}$ as the key, and sends the message $\{PD'_i, msg_s\}$ to the IoT device D_i via a public channel.
7) The IoT device D_i obtains $D_i^{*''}$, r'_s, $m'_{s.1}$, and $m'_{s.2}$ through decrypting the encrypted message msg_s using $m_{i.2}$, calculates a new pseudonym $PD_i^\star = H(D_i^* \parallel r_i \parallel r'_s)$, and computes the session key $sk_{i.s} = T_{r_i \cdot che_i \cdot D_i^*}(m'_{s.1})$. Here,
$$\begin{aligned}
\underline{sk_{i.s}} &= T_{r_i \cdot che_i \cdot D_i^*}(m'_{s.1}) \\
&= T_{r_i \cdot che_i \cdot D_i^*}(T_{r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^*}(pu_i)) \\
&= T_{r_i \cdot che_i \cdot D_i^*}(T_{r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^*}(T_{pr_i}(g))) \\
&= T_{r_i \cdot che_i \cdot D_i^* \cdot r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^* \cdot pr_i}(g) \\
&= T_{r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^*}(T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(g)) \\
&= T_{r_s \cdot che_i' \cdot dt'_{i.k} \cdot pr_s \cdot D_s^*}(m_{i.1}) \\
&= \underline{sk_{s.i}}.
\end{aligned}$$
8) The IoT device D_i calculates $m''_{s.2} = H(D_i^* \parallel sk_{i.s})$

Algorithm 1: IoT Device and Gateway Authentication

/* *RandNum*(): random number function */
/* *Send*(·): data transfer */
/* *Wait*(): wait for gateway message */
/* *Terminate*(): terminate authentication */
/* *Eec*(*data*, *key*): encrypt *data* with *key* */
/* *Dec*(*data*, *key*): decrypt *data* with *key* */

Function DevToGtwAuth():
  $che_i \leftarrow$ *Prob-PUF*$(\Gamma_i, \Gamma_i^{smp})$; $r_i \leftarrow$ *RandNum*();
  $m_{i.1} = T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(g)$; $m_{i.2} = T_{che_i \cdot r_i \cdot pr_i \cdot D_i^*}(pu_s)$;
  $m_{i.3} = H(D_i^* \parallel r_i \parallel m_{i.1} \parallel dt_{i.k})$;
  $msg_i = Eec(\{D_i^* \parallel r_i \parallel m_{i.3} \parallel dt_{i.k}\}, m_{i.2})$;
  Send$(G_s^*, PD_i, m_{i.1}, msg_i)$;
  Wait();
  $D_i^{*''}, r_s', m_{s.1}', m_{s.2}' \leftarrow Dec(msg_s, m_{i.2})$;
  $PD_i^\star = H(D_i^* \parallel r_i \parallel r_s')$; $sk_{i.s} = T_{r_i \cdot che_i \cdot D_i^*}(m_{s.1}')$;
  $m_{s.2}'' == H(D_i^* \parallel sk_{i.s})$;
  if $m_{s.2}'' \neq m_{s.2}'$ **then**
    Terminate();
  end
Function GtwToDevAuth():
  $che_i' \leftarrow P_{trap}^i(t) = [1 - e(\frac{-t}{\tau_e^i})]$; $m_{i.2}' = T_{pr_s}(m_{i.1}')$;
  $D_i^{*'}, r_i', m_{i.3}', dt_{i.k}' \leftarrow Dec(msg_i, m_{i.2}')$;
  if $D_i^{*'} \notin$ *Database* **then**
    Terminate();
  else
    $m_{i.3}'' = H(D_i^{*'} \parallel r_i' \parallel m_{i.1}' \parallel dt_{i.k}')$;
    if $m_{i.3}'' \neq m_{i.3}'$ **then**
      Terminate();
    else
      $r_s \leftarrow$ *RandNum*(); $m_{s.1} = T_{r_s \cdot che_i' \cdot dt_{i.k} \cdot pr_s \cdot D_s^*}(pu_i)$;
      $PD_i^\star = H(D_i^* \parallel r_i' \parallel r_s)$;
      $sk_{s.i} = T_{r_s \cdot che_i' \cdot dt_{i.k}' \cdot pr_s \cdot D_s^*}(m_{i.1}')$;
      $m_{s.2} = H(D_i^{*'} \parallel sk_{s.i})$;
      $msg_s = Eec(\{D_i^{*'} \parallel r_s \parallel m_{s.1} \parallel m_{s.2}\}, m_{i.2}')$;
      Send(PD_i', msg_s);
    end
  end

and compares $m_{s.2}''$ with $m_{s.2}'$. If $m_{s.2}'' \neq m_{s.2}'$, the IoT device D_i terminates the authentication process. Otherwise, the authenticated key agreement between the IoT gateway G_s and the IoT device D_i is considered complete and the session key $sk_{i.s}$ (or $sk_{s.i}$) for the data type $dt_{i.k}$ has been established.

The proposed authenticated key agreement protocol for device-to-gateway communication is presented in Algorithm 1.

VI. Security and Performance Evaluation

A. Security Verification

We validate the proposed authenticated key agreement protocol using AVISPA [18]. AVISPA is an automated security testing tool that evaluates communication protocols for security flaws and vulnerabilities. We first implement the IoT device and gateway of the proposed protocol, along with the potential attacker, using AVISPA specification language. After that, we select two security-analyzing back-ends, OFMC and CL-AtSe, to investigate the interactions within the proposed protocol and examine its execution constraints in the AVISPA environment on Ubuntu 10.04. Fig. 2 shows the security verification results from AVISPA, indicating that the proposed

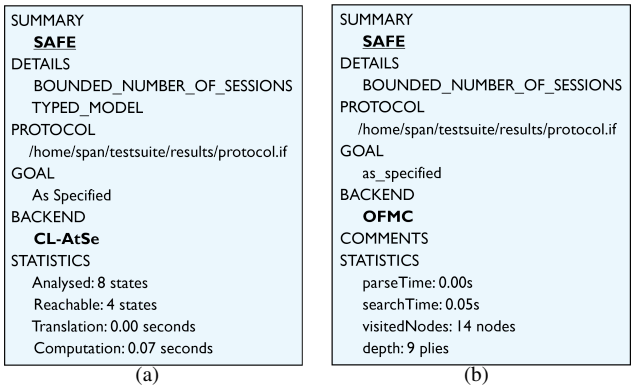

Fig. 2. Security verification results from AVISPA.

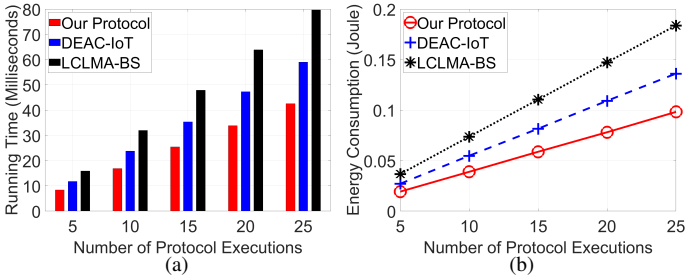

Fig. 3. The performance of running time and energy consumption.

protocol effectively resists replay attacks, protocol falsification, and man-in-the-middle threats, and adheres to the security requirements of the CK adversary model.

B. Performance Evaluation

We conduct a simulation-based study to evaluate the performance of the proposed protocol (marked as ''Our Protocol'' in the result figures) on a Windows 11 Pro computer (Qualcomm Snapdragon X Elite X1E80100 CPU, up to 3.40 GHz). Within the simulation platform, we install and configure Eclipse IDE, implement the proposed protocol, DEAC-IoT [28], and LCLMA-BS [29] in Java, and obtain the results for the running time and energy consumption.

In Fig. 3(a), we present the running time of our protocol, DEAC-IoT, and LCLMA-BS. The results were obtained by varying the number of protocol executions. Generally speaking, our protocol demonstrates promising results and its running time is smaller than that of DEAC-IoT and LCLMA-BS. As our protocol is achieved through lightweight cryptographic primitives, such as Chebyshev polynomial and probability-based PUF, fewer mathematical operations are needed to execute the algorithm. Thus, less amount of time will be required for the execution of our protocol. DEAC-IoT and LCLMA-BS utilize elliptic curve cryptography (ECC) and modular arithmetic (used in bilinear pairing) operations, respectively. As a result, a longer running time is observed for DEAC-IoT and LCLMA-BS. DEAC-IoT demonstrates better performance compared to LCLMA-BS. This is because DEAC-IoT adopts ECC which is more computation-efficient than bilinear pairing.

In Fig. 3(b), we report the energy consumption of our protocol, DEAC-IoT, and LCLMA-BS. DEAC-IoT and LCLMA-BS involve intensive mathematical operations, such as point

addition/doubling on elliptic curves over finite fields and pairing operations over finite fields, which require substantial processing power. Thus, their energy consumption are higher than that of our protocol. LCLMA-BS consumes more energy than DEAC-IoT due to more intricate mathematical operations and operations over extension fields. Overall, our protocol provides the lowest energy consumption as the number of protocol executions increases from 5 to 25. Chebyshev Polynomials can be evaluated using recurrence relations, which reduce the need for complex computations. In addition, unlike other cryptographic methods, Prob-PUFs do not rely on heavy mathematical operations, making them energy-efficient. As a result of the reasons outlined above, our protocol outperforms DEAC-IoT and LCLMA-BS in terms of energy consumption.

VII. Conclusion

IoT applications are becoming more prevalent, making it urgent to determine how to use the existing limited IoT systems/networks for diverse tasks in a secure and efficient manner. However, the lack of lightweight, data type-aware, and secure authenticated key agreement protocols remains as major barriers hindering the further development of next-generation IoT applications. In this paper we chose Chebyshev polynomial and probability-based PUF, and proposed an authenticated key agreement protocol to ensure the protection of data communication within IoT systems. The results of the security and performance evaluation have demonstrated that the proposed protocol not only offers superior security and privacy, but also maintains low computational overhead. Implementing this innovative authenticated key agreement protocol would mark a significant first in the IoT community as no comparable and similar approach currently exists, which makes this research a crucial step toward addressing the existing research gap.

References

[1] O. Aouedi, T. Vu, A. Sacco, D. Nguyen, K. Piamrat, G. Marchetto, and Q. Pham, "A survey on intelligent Internet of Things: Applications, security, privacy, and future directions," *IEEE Communications Surveys & Tutorials (Early Access)*, pp. 1–1, 2024.

[2] *Internet of Things*, https://www.statista.com/outlook/tmo/internet-of-things/worldwide.

[3] C. Pu, I. Ahmed, and S. Chakravarty, "Resource-Efficient and Data Type-Aware Authentication Protocol for Internet of Things Systems," in *Proc. IEEE TPS*, 2023, pp. 101–110.

[4] C. Pu and K. Choo, "Lightweight Sybil Attack Detection in IoT based on Bloom Filter and Physical Unclonable Function," *Computers & Security*, vol. 113, p. 102541, 2022.

[5] C. Pu, "Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4937–4949, 2020.

[6] *IoT / OT Threats*, https://www.microsoft.com/en-us/security/blog/threat-intelligence/iot-ot-threats/.

[7] B. Groves and C. Pu, "A Gini Index-Based Countermeasure Against Sybil Attack in the Internet of Things," in *Proc. IEEE MILCOM*, 2019, pp. 1–6.

[8] R. Ponnuru, S. Kumar, M. Azab, and G. Alavalapati, "BAAP-FIoT: Blockchain Assisted Authentication Protocol for Fog-enabled Internet of Things Environment," *IEEE Internet of Things Journal (Early Access)*, pp. 1–1, 2025.

[9] D. Chhikara, S. Chauhan, and S. Rana, "An Efficient Blockchain-Powered Authentication Scheme for Secure Communication in IoMT," *IEEE Internet of Things Journal (Early Access)*, pp. 1–1, 2025.

[10] M. Servati, M. Safkhani, A. Rahmani, and M. Hosseinzadeh, "ERAS-MIS: An ECC-based robust authentication protocol suitable for medical IoT systems," *Computer Networks*, vol. 258, p. 110938, 2025.

[11] T. Wu, G. Li, J. Wang, B. Xiao, and Y. Song, "PPCA: Privacy-Preserving Continuous Authentication Scheme With Consistency Proof for Zero-Trust Architecture Networks," *IEEE Internet of Things Journal (Early Access)*, pp. 1–1, 2025.

[12] S. Kalam and A. Keshri, "Advancing IoMT security: A two-factor authentication model employing PUF and Fuzzy logic techniques," *Computers & Security*, vol. 148, p. 104138, 2025.

[13] Y. Wu, T. Feng, C. Su, and C. Liu, "MSAUPL: A multi-server authentication and key agreement protocol for industrial IoT based on user privacy level," *Journal of Information Security and Applications*, vol. 89, p. 103991, 2025.

[14] P. Gope, F. Hongming, and B. Sikdar, "Lightweight and Privacy-Preserving Reconfigurable Authentication Scheme for IoT Devices," *IEEE Transactions on Services Computing (Early Access)*, pp. 1–14, 2025.

[15] C. Pu, H. Zerkle, A. Wall, S. Lim, K. Chood, and I. Ahmed, "A Lightweight and Anonymous Authentication and Key Agreement Protocol for Wireless Body Area Networks," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 136–21 146, 2022.

[16] J. Mason and D. Handscomb, *Chebyshev Polynomials*. Chapman and Hall/CRC, 2002.

[17] Z. Tu, Y. Xue, P. Ren, F. Hao, R. Wang, M. Li, J. Zhang, Z. Ji, and R. Huang, "A Probability-Based Strong Physical Unclonable Function With Strong Machine Learning Immunity," *IEEE Electron Device Letters*, vol. 43, no. 1, pp. 138–141, 2021.

[18] *Automated Validation of Internet Security Protocols and Applications*, https://www.ercim.eu/publication/Ercim_News/enw64/armando.html (Accessed: January 28, 2025).

[19] W. Jing, L. Peng, H. Fu, and A. Hu, "An Authentication Mechanism Based on Zero Trust With Radio Frequency Fingerprint for Internet of Things Networks," *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 23 683–23 698, 2024.

[20] Q. Zhang, X. Zhou, H. Zhong, J. Cui, J. Li, and D. He, "Device-Side Lightweight Mutual Authentication and Key Agreement Scheme based on Chameleon Hashing for Industrial Internet of Things," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7895–7907, 2024.

[21] M. El-Zawawy, A. Brighente, and M. Conti, "SETCAP: Service-based energy-efficient temporal credential authentication protocol for Internet of Drones," *Computer Networks*, vol. 206, p. 108804, 2022.

[22] I. Bhattarai, C. Pu, K. Choo, and D. Korać, "A Lightweight and Anonymous Application-Aware Authentication and Key Agreement Protocol for the Internet of Drones," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 790–19 803, 2024.

[23] M. Li, Y. Zhu, R. Du, and C. Jia, "LPCR-IoT: Lightweight and privacy-preserving cross-modal Retrieval in IoT," *IEEE Internet of Things Journal (Early Access)*, pp. 1–1, 2025.

[24] X. Yang and Y. Guo, "IAR-AKA: An Efficient Authentication Scheme for Healthcare Tactile Internet Beyond Conventional Security," *IEEE Transactions on Network and Service Management (Early Access)*, pp. 1–1, 2025.

[25] C. Pu, J. Brown, and L. Carpenter, "A Theil Index-Based Countermeasure Against Advanced Vampire Attack in Internet of Things," in *Proc. IEEE HPSR*, 2020, pp. 1–6.

[26] Q. Do, B. Martini, and K. Choo, "The role of the adversary model in applied security research," *Computers & Security*, vol. 81, pp. 156–181, 2019.

[27] C. Pu, A. Bilal, N. Park, J. Seol, and K. Choo, "A Redactable Blockchain-Assisted Application-Aware Authentication System for Internet of Drones," *IEEE Internet of Things Journal*, vol. 12, no. 14, pp. 27 206–27 221, 2025.

[28] M. Abdussami, S. Dwivedi, T. Al-Shehari, P. Saravanan, M. Kadrie, T. Alfakih, H. Alsalman, and R. Amin, "DEAC-IoT: Design of lightweight authenticated key agreement protocol for Intra and Inter-IoT device communication using ECC with FPGA implementation," *Computers and Electrical Engineering*, vol. 120, p. 109696, 2024.

[29] D. Yin and B. Gong, "A Lightweight Certificateless Mutual Authentication Scheme Based On Signatures for the IIoT," *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 26 852–26 865, 2024.