# PUF3FAka: PUF-Enhanced Lightweight Three-Factor Authentication and Key Agreement Protocol for Mobile Edge Computing

Shuming Qiu, Yu Liang, Peizhen Hong, and Ding Wang

***Abstract*—In the era of Internet of Things (IoT), the data generated by a large number of devices usually relies on cloud servers for processing, but the traditional cloud server architecture has problems such as high latency and bandwidth bottlenecks. Mobile edge computing (MEC) effectively enhances the real-time data processing capability by sinking computing power to the network edge. However, the heterogeneity of MEC devices brings complex security challenges, such as risks of information leakage and privacy infringement, and it is urgent to build an efficient identity authentication mechanism to ensure system security. Nevertheless, the existing identity-based and physical unclonable function (PUF)-based identity verification protocols show limitations, such as an inability to ensure anonymity and nonreproducibility, vulnerability to the impact of challenge response pairs on leakage, and susceptibility to various attacks caused by temporary secret leakage and key leakage. To address these issues, this article proposes a PUF-enhanced lightweight three-factor authentication and key agreement protocol named PUF3FAka tailored for MEC environments. For the first time, a cross-ring oscillator (CRO)-PUF resilient to machine learning attacks is integrated with a fuzzy-verification mechanism into a three-factor authentication framework. Without storing any challenge-response pairs (CRPs) on the server side, the protocol eliminates the risk of CRP leakage attacks root. The design satisfies ten edge-computing security evaluation criteria and withstands various attacks such as offline password guessing, ephemeral-secret leakage attacks, and key-compromise impersonation attacks. Performance evaluations demonstrate that PUF3FAka outperforms existing protocols by 82.86% in computational cost and 96.7% in communication overhead on average, making it well-suited for resource-constrained IoT terminals and edge servers.**

***Index Terms*—Authentication and key agreement, cross-ring oscillator (CRO)-physical unclonable function (PUF), guessing attack, key compromise impersonation (KCI), three-factor.**

## NOMENCLATURE

| | |
|---|---|
| $\text{MECS}_j$ | Server. |
| $U_i$ | User. |
| RA | Trusted registration authority (RA). |
| $\text{ID}_i, \text{PW}_i, B_i$ | Identity, password, and biometrics of $U_i$. |
| $s$ | Primary key of RA. |
| $\theta_i, \theta_j$ | Helper data. |
| $\text{MID}_j$ | Fixed pseudo-identity of $\text{MECS}_j$. |
| $\oplus$ | Bitwise XOR operation. |
| $H(\cdot)$, $H_0(\cdot), H_1(\cdot)$ | One-way hash-function. |
| $\|\|$ | String concatenation operation. |
| $K_m$ | Long-term key of $S$. |
| $(\text{Ch}_j, \text{Re}_j)$ | Challenge response pair of $\text{MECS}_j'$s CRO – PUF. |
| $\mathcal{A}$ | Malicious adversary. |
| Fu.Gen(·) | Fuzzy extractor generation algorithm. |
| Fu.Rep(·) | Fuzzy extractor reproduction algorithm. |
| SK | Session key between $U_i$ and $S$. |
| $\delta_i$ | Key extracted from the $U_i$'s biometric. |
| $K_j$ | Key extracted from the response. |
| CPUF | CRO-PUF. |

## I. INTRODUCTION

IN THE digital era, the rapid advancement of the Internet of Things (IoT), fifth-generation (5G) mobile communication technology, and smart devices has led to an explosive growth in data volume. The generation and processing demands of this data present unprecedented challenges to network response speed, bandwidth, and computing power. While the traditional cloud computing model has achieved significant success in handling large-scale datasets and providing elastic computing resources, its centralized architecture results in high latency and bandwidth limitations. These drawbacks are particularly problematic in real-time application scenarios such as autonomous driving, telemedicine (see Fig. 1), and augmented reality. To address these issues, mobile edge computing (MEC) has emerged as a transformative solution [1].

MEC is an innovative technology that brings computing, storage, and network services closer to the edge of the network, enabling rapid data processing and response by deploying computing resources near end-users. This distributed architecture not only reduces data transmission latency and enhances quality of service but also alleviates the load on the core network, improving network scalability and reliability. Furthermore, edge computing better supports localized

Received 8 August 2025; revised 3 November 2025 and 15 December 2025; accepted 21 December 2025. Date of publication 26 December 2025; date of current version 26 March 2026. This work was supported in part by Jiangxi Provincial Natural Science Foundation under Grant 20232BAB202012; in part by the Science and Technology Research Project of Education Department of Jiangxi Province under Grant GJJ191680; and in part by the National Natural Science Foundation of China under Grant 12261049. *(Corresponding author: Peizhen Hong.)*

Shuming Qiu and Yu Liang are with the School of Mathematics and Statistics and Jiangxi Provincial Center for Applied Mathematics, Jiangxi Normal University, Nanchang, Jiangxi 330022, China (e-mail: qiushuming@jxnu.edu.cn; 2139965098@qq.com).

Peizhen Hong is with the College of Electronics, Information and Optical Engineering, Nankai University, Tianjin 300350, China (e-mail: hongpeizhen@nankai.edu.cn).

Ding Wang is with the College of Cryptology and Cyber Science and Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China (e-mail: wangding@nankai.edu.cn).

Digital Object Identifier 10.1109/JIOT.2025.3648415

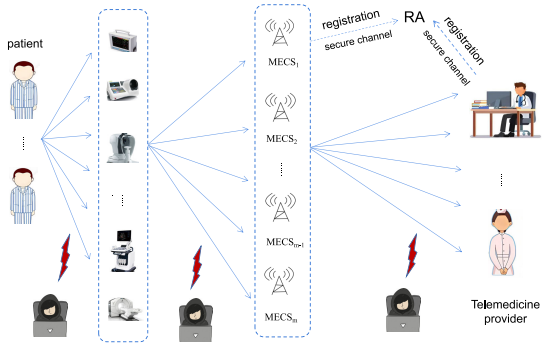


Fig. 1. Telemedicine architecture under MEC environments.

services and applications, delivering a more personalized and responsive user experience. In the edge computing environment, mobile devices, sensors, and other end devices can communicate directly with edge servers, which possess robust computing capabilities and can perform data analysis, storage, and application services. For example, in 5G-enabled MEC remote-medicine scenarios, end-to-end latency can be reduced by more than 60% compared with traditional cloud-centric modes, with field measurements as low as 7–12 ms. By processing data closer to its source, MEC shortens the distance data must travel across the network, significantly improving processing speed. In addition, MEC can integrate with cloud computing to form a hybrid model known as edge cloud [2], [3]. In this model, edge servers handle tasks requiring high real-time performance, while nonreal-time or computation-intensive tasks are offloaded to the cloud.

Despite its technical advantages, MEC faces several challenges, including secure privacy protection, secure data transmission, and limited computing and storage resources. The explosive growth of IoT, 5G, and smart devices imposes stringent requirements on network latency, bandwidth, and computing power. Centralized cloud computing suffers from long back-haul delays and bandwidth bottlenecks, which are unacceptable for real-time applications such as autonomous driving, telemedicine (see Fig. 1), and augmented reality. MEC mitigates these drawbacks by pushing resources close to end-users, but its distributed and heterogeneous nature also introduces new security challenges: CRP leakage, offline guessing, ephemeral-secret exposure, and identity privacy loss.

Existing identity-based [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] or PUF-enhanced protocols [16], [20] either store massive CRPs on servers or fail to balance security and efficiency. To close these gaps, we propose PUF3FAka, a lightweight three-factor authentication and key agreement protocol that includes the following aspects.

1) It integrates CRO-PUF—resilient to machine-learning and side-channel attacks—into a three-factor framework for the first time, eliminating CRP storage on the server side.
2) It replaces traditional password verification with "honeywords + fuzzy-verifiers " to thwart offline password guessing.
3) It satisfies ten edge-specific security criteria and demonstrates strong resistance against multiple advanced attacks.
4) Reduces computational and communication overhead by 82.86% and 96.7% on average compared with seven recent schemes, making it ideally suited for resource-constrained MEC environments.

### A. Related Work

The decade-long evolution of authentication protocols for cloud/edge computing reveals a persistent trilemma: balancing security, efficiency, and scalability. Identity-based protocols (2015–2024) have cyclically addressed vulnerabilities like weak anonymity [4] or replay attacks [12], yet consistently falter in achieving robust multifactor security and forward secrecy while avoiding computational overheads (e.g., bilinear pairings). Meanwhile, PUF-based protocols (2020–2024) leverage hardware uniqueness to bypass digital secret storage but introduce new risks: ephemeral-secret leakage [21], asynchronous attacks [25], and denial-of-service vulnerabilities [24]. Despite iterative advancements, both paradigms suffer from fragmented security models—patching one flaw often exacerbates another, as seen in [12], [14], and [15]. Critically, even "provably secure" designs fail under practical adversarial scrutiny, underscoring a systemic reliance on reactive, rather than anticipatory, threat modeling.

This will naturally lead to an urgent open question: *can one design a PUF-integrated authentication protocol that simultaneously meets the following three conditions?* 1) Achieves provable three-factor security with forward secrecy and attack resistance; 2) eliminates reliance on bilinear pairings and public-key infrastructure; and 3) maintains sublinear computational and communication overhead for resource-constrained edge devices.

Existing studies largely adopt a modular security enhancement strategy, where protocol components are revised to address specific vulnerabilities. However, the integration of physical-layer characteristics with cryptographic protocol design remains relatively underexplored, particularly in scenarios involving both digital and physical adversarial capabilities.

*1) Identity-Based Authentication Protocols:* Research on identity-based authentication protocols has evolved considerably over the past decade, with many proposals aiming to enhance security and privacy in cloud and MEC environments. In 2015, Tsai and Lo [4] introduced a privacy-aware authentication protocol for distributed mobile cloud computing services, though it did not support mutual authentication or anonymity. The following year, Irshad et al. [5] put forth a multiserver authentication protocol improving on prior work, though it still fell short of three-factor security and incurred high computational cost. Also in 2016, Yang et al. [6] employed numerous pseudonyms to achieve anonymity and untraceability for handover authentication, but the storage requirements posed challenges for resource-limited devices. However, Deng et al. [7] identified that this approach did not achieve full untraceability or forward secrecy.

In 2018, He et al. [8] developed an efficient privacy-aware authentication scheme. By 2020, Jia et al. [9] proposed a

protocol using identity and bilinear pairing operations with provable security claims, while Mishra et al. [10] introduced a dynamic identity-based key agreement that, however, lacked untraceability and a revocation mechanism. The following year, Li et al. [11] identified replay attack vulnerabilities and forward secrecy shortcomings in Jia et al.'s scheme and proposed an alternative, though Jia et al. [12] later reported that Li et al.'s proposal remained susceptible to replay, impersonation, and DoS attacks, and did not achieve three-factor security. However, in 2025, Ma et al. [13] discovered and analyzed that Jia et al.'s protocol [12] cannot resist offline key guessing attacks and ephemeral-secret leakage attacks. In 2023, Lee et al. [14] presented an anonymous authentication protocol for MEC, though it was later found to have certain security limitations. Most recently, Tian et al. [15] built upon [12], [14] to design a fully anonymous and unlinkable protocol, though our analysis indicates that some security aspects require further attention.

In general, these identity-based protocols either have some security vulnerabilities, or use bilinear peer-to-peer operations to reduce efficiency, or use public-key cryptography but need to solve the problem of public-key distribution. So researchers thought about an important problem: Whether could one design an authentication protocol that is both secure and relatively efficient?

*2) PUF-Based Authentication Protocols:* Physical unclonable functions (PUFs) have attracted growing interest as a hardware-rooted security primitive that offers uniqueness, low cost, and resistance to physical cloning. Accordingly, numerous PUF-based authentication protocols have been proposed in recent years, particularly for Internet of Things (IoT) and edge computing environments.

In 2020, Yoon et al. [16] introduced a mutual authentication protocol for IoT devices using a central server that stores and updates challenge-response pairs (CRPs) per device, enabling session key establishment. That same year, Ghafi et al. [17] proposed a self-correcting PUF-based protocol incorporating blockchain and smart contracts to facilitate mutual authentication between IoT devices. In 2021, Quershi and Munir [18] put forward PUF-RAKE, a lightweight protocol combining PUF with key establishment, using random nonces and masking to enhance security. Also in 2021, Zhang et al. [19] designed a PUF and blockchain-based privacy protection mechanism, though its complexity may hinder practicality, while Zheng et al. [20] developed a mutual authentication and key exchange protocol that avoids local CRP storage.

In 2023, Zheng et al. [21] proposed a PUF-based key agreement protocol with forward secrecy, though it did not fully address ephemeral-secret leakage. Zhang et al. [22] combined PUF with group signatures and Merkle trees to achieve anonymous batch authentication, reducing overhead. This year, Tian et al. [24] presented a PUF-based authentication method for UAV-assisted IoT systems, though potential DoS risks against UAVs were noted. In 2025, Kalam and Keshri [25] incorporated PUF and fuzzy logic for two-factor authentication, though the protocol was found susceptible to asynchronous attacks and session key compromises. Tanveer and Aldossari [26] also proposed a cryptographic PUF-based protocol for edge-enabled IoT, though it may not be extendable to three-factor authentication.

A recurring theme among these works is the challenge of maintaining security against physical and logical attacks—such as CRP leakage, desynchronization, and ephemeral-secret exposure—while preserving low overhead and usability. The limitations of both identity-based and PUF-based protocols underscore the need for a new design methodology that integrates physical security properties into cryptographic primitives in a cohesive manner. To clearly compare the identity-based and PUF-based authentication protocols described in the previous text, Table I summarizes and sorts out their respective advantages and disadvantages.

In response, we propose PUF3FAka, a three-factor authenticated key agreement protocol incorporating PUF technology. In this design, an edge server is equipped with a PUF module, but unlike conventional approaches, no CRPs are stored on the server, mitigating risks of database leakage. To enhance server protection, a dual-key strategy is adopted: one long-term key and another derived from the PUF response. This architecture aims to achieve provable security without relying on bilinear pairings or traditional public key infrastructure (PKI), while maintaining low overhead suitable for edge devices. This integrated approach represents a step toward inherently secure, efficient, and scalable authentication—addressing limitations of previous work through a unified design grounded in both cryptographic and physical security principles.

*3) Insights Into Cross-Environmental Protocol Design:* The core design principles of security protocols often exhibit remarkable generality across different environments. A proven secure and efficient authentication and key agreement protocol from one specific application scenario can often be successfully adapted and migrated to other environments with similar security requirements but different network architectures. This capability for "cross-application" serves as a crucial dimension for evaluating the robustness and innovative value of a protocol's design.

For instance, the anonymous authentication protocol for generic MEC environments proposed by Mahmood et al. [27] established a lightweight privacy protection framework using identity-based cryptography. This foundational design philosophy was extended and specialized in their subsequent work. Specifically, their protocol, designed for the edge-centric maritime transportation system (MTS) [28], not only maintained the core objective of anonymous authentication, but also introduced significant enhancements, including the integration of a PUF for robust defense against physical attacks and an optimized authentication flow. This evolution demonstrates how design principles are reinforced and augmented when moving from a general-purpose to a domain-specific context.

Similarly, the lightweight remote data authentication protocol for cloud storage environments proposed by Ghaffar et al. [29] centers on constructing an efficient access control model governed by a data delegator. By leveraging symmetric encryption to ensure data confidentiality and extensively employing lightweight operations like hash functions for secure mutual authentication and key agreement, it embodies a universal design principle for achieving controlled and secure data

sharing in resource-constrained cloud-edge environments. Furthermore, the anonymous authenticated key agreement scheme for multiserver infrastructure by Akram et al. [30] exemplifies another paradigm that balances security with efficiency. The scheme ingeniously employs elliptic curve cryptography combined with a dynamic pseudo-identity mechanism to ensure user anonymity and untraceability while maintaining low computational and communication overhead. This approach, designed to address the core challenges of privacy and security in "single-registration, multiservice" scenarios, holds significant referential value for other distributed systems like the IoT and vehicular networks, which also require cross-domain authentication and privacy preservation.

Particularly noteworthy is the migration of design concepts from fundamental communication environments. For example, an identity-based authentication and key agreement protocol utilizing puncturable pseudorandom functions is proposed for mobile clients in IoT environments [31], while targeting resource-constrained mobile devices and servers, showcases a general design framework. This framework combines lightweight cryptographic constructs with privacy-enhancing technologies, offering a generic approach to significantly strengthen forward secrecy and user privacy protection without compromising efficiency. Such a design pattern provides a highly promising and transferable architectural reference for other scenarios with stringent demands for both low overhead and high security, such as edge computing and distributed sensor networks.

Collectively, these research efforts highlight that a key to protocol innovation and cross-domain application lies in deeply analyzing the common security requirements and unique constraints of different application environments (e.g., edge computing, cloud storage, multiserver architecture, and fundamental IoT), and then extracting transferable design patterns from existing protocols—such as PUF enhancement for hardware trust roots, lightweight cryptographic construction, dynamic anonymization, and enhanced forward secrecy mechanisms. The PUF3FAka protocol proposed in Section IV of this article, although primarily designed for telemedicine in MEC, offers a potentially valuable reference architecture based on its framework of PUF-based hardware security, three-factor authentication, and efficient key agreement. This framework can inspire solutions to device authentication and key security problems in broader scenarios, including MEC, industrial IoT, and distributed sensor networks.

### B. Motivations and Contributions

After conducting an in-depth analysis of a large number of authentication protocols based on identity or PUF, we have found that although some protocols meet the required security standards, their efficiency is far from satisfactory. More concerning is the fact that many authentication protocols share common security vulnerabilities. This phenomenon has prompted us to delve into the underlying causes of these common vulnerabilities. By uncovering the root causes, we hope to develop effective design principles to prevent such issues from recurring. This is not only of great significance for guiding the design of more secure and efficient authentication and key agreement protocols for MEC but also helps us establish more targeted evaluation criteria and design objectives, thereby promoting the continuous progress and development of related technologies.

At present, the authentication protocols on mobile cloud computing, edge computing, and some PUF-based authenticated key agreement protocols either have some security vulnerabilities, or they cannot balance security and efficiency well. Therefore, we propose a PUF-based authenticated key agreement protocol for MEC. We avoid potential attacks by some clever designs, and the efficiency of our protocol is better. The following are the main contributions of this article.

1) The security of the latest three representative protocols Jia et al. [12] at IEEE IoTJ'22, Lee et al. [14] at IEEE IoTJ'23, and Tian et al. [15] at IEEE IoTJ'24 is analyzed, and the common security vulnerabilities in identity authentication protocols for MEC are identified including offline password guessing attacks ephemeral-secret leakage attacks and Key compromise impersonation (KCI) attacks.
2) In order to avoid password-guessing attacks, current password-authenticated key agreement protocols usually store the password authentication parameters in smart cards or smart devices. However, there is a security risk in this method. Once the password verification information is leaked, it cannot effectively resist the password-guessing attack. Therefore, this article uses fuzzy verification technology, which can effectively avoid offline password guessing attacks.
3) To resist key compromise and PUF response leakage, the edge server is equipped with CRO-PUF to protect the important user information in the server database; it also stores critical parameters that are safeguarded by the registrar's primary private key and the key extracted from CRO-PUF, eliminating the need to keep the server's long-term private key in plaintext, so adversaries cannot obtain them even through differential power analysis (DPA).
4) We provide a formal security proof under the ROR model, and use BAN logic to validate the correctness of mutual authentication.
5) We perform a comparative analysis of the security, computation, and communication cost of the proposed protocol PUF3FAka with seven related protocols. Comparison results demonstrate that the PUF3FAka protocol for mobile lightweight devices can achieve a better balance between security and usability.

### C. Roadmap of This Article

The roadmap of this article is as follows. Section II introduces necessary preliminaries, including the adversary model and the ten evaluation criteria for edge computing three-factor AKA protocols. In Section III, we analyze the Jia et al., Lee et al., and Tian et al.'s protocols and the vulnerabilities of their protocols in detail. In Section IV, we present our new protocol. The formal security proof and heuristic security arguments are presented in Sections V-A and V-C, respectively.

TABLE I
SUMMARY OF IDENTITY-BASED AND PUF-BASED AUTHENTICATION PROTOCOLS

| Ref. | Year | Theme | Technology/Advantages | Limitations/Disadvantages |
|---|---|---|---|---|
| [4] | 2015 | Privacy-aware authentication for mobile cloud | Privacy-aware design for distributed mobile cloud | No mutual authentication or anonymity support |
| [5] | 2016 | Multi-server authentication | Improved multi-server authentication | No three-factor security, high computational cost |
| [6] | 2016 | Handover authentication with anonymity | Pseudonyms for anonymity and untraceability | High storage requirements, not suitable for resource-limited devices |
| [7] | 2016 | Analysis of anonymity approaches | Critical analysis of pseudonym-based methods | Does not achieve full untraceability or forward secrecy |
| [8] | 2018 | Privacy-aware authentication | Efficient privacy-aware design | Not specified in text |
| [9] | 2020 | Authentication using identity and bilinear pairing | Identity-based, bilinear pairing, provable security | Vulnerable to replay attacks, lacks forward secrecy |
| [10] | 2020 | Dynamic identity-based key agreement | Dynamic identity-based approach | Lacks untraceability and revocation mechanism |
| [11] | 2021 | Improvement on [9] | Addresses replay and forward secrecy issues | Susceptible to replay, impersonation, DoS attacks; no three-factor security |
| [12] | 2021 | Analysis of [11] | Critical security analysis | No three-factor security, Susceptible to ephemeral Secret Leakage and Key Compromise attack |
| [14] | 2023 | Anonymous authentication for MEC | Anonymous authentication | Susceptible to ephemeral Secret Leakage and Key Compromise attack, No three-factor security |
| [15] | 2023 | Fully anonymous and unlinkable protocol | Based on [12], [14]; fully anonymous and unlinkable | Some security aspects need further attention |
| [16] | 2020 | Mutual authentication for IoT | Server-managed CRPs, session key establishment | Requires server to store/update CRPs |
| [17] | 2020 | Self-correcting PUF authentication | Blockchain, smart contracts, mutual authentication | Complexity may be high |
| [18] | 2021 | Lightweight PUF key establishment | PUF, random nonces, masking, lightweight | Vulnerable to a single point of failure |
| [19] | 2021 | Privacy protection with PUF and blockchain | PUF, blockchain for privacy | High complexity hinders practicality |
| [20] | 2021 | Mutual authentication and key exchange | PUF-based, no local CRP storage | Vulnerable to a single point of failure |
| [21] | 2023 | PUF key agreement with forward secrecy | PUF, forward secrecy | Does not fully address ephemeral secret leakage |
| [22] | 2023 | Anonymous batch authentication | PUF, group signatures, Merkle trees, reduced overhead | The access frequency of the Merkle tree is inherently limited, and the strong trust assumption imposed on the gateway may introduce potential security risks. |
| [24] | 2024 | PUF authentication for UAV-assisted IoT | PUF-based for UAV-assisted IoT | Potential DoS risks against UAVs |
| [25] | 2025 | Two-factor authentication with PUF and fuzzy logic | PUF, fuzzy logic, two-factor authentication | Susceptible to asynchronous attacks and session key compromise |
| [26] | 2025 | Cryptographic PUF protocol for edge-enabled IoT | PUF-based for edge-enabled IoT | Not extendable to three-factor authentication |

Section VI summarizes the comparison of security, communication, and computation cost. Section VII concludes the article.

## II. PRELIMINARIES

In this section, we present some relevant preliminaries required for the understanding of the remaining of this article, as well as the security evaluation criteria. A summary of notations used in this article is presented in Nomenclature.

### A. Elliptic Curve Cryptography

Suppose $q$ is a large prime number, and the elliptic curve over a finite field $F_q$ is defined as $E(a,b): y^2 = x^3 + ax + b$, where $a, b, x, y \in F_q$. Here, are two classic difficult problems in elliptic curve cryptography that need to be used in the PUF3FAka protocol.

*Definition 1 (Elliptic Curve Discrete Logarithm Problem (ECDLP) [32]):* The ECDLP states that, given an elliptic curve over a finite field $F_q$, given a point on the elliptic curve $E(a,b)$, where $Q = xP$, $P$ is a generator point on $E(a,b)$, which usually written as $G$, it is computationally infeasible for a probabilistic polynomial time (PPT) adversary to calculate $x$ by given only $P, Q$.

*Definition 2 (Elliptic Curve Diffie Hellman Problem (ECDHP) [32]):* The ECDHP states that, given an elliptic curve over a finite field $F_q$, given two points on the elliptic curve $E(a,b)$, where $Q_1 = x_1P$, $Q_2 = x_2P$, it is computationally infeasible for a PPT adversary to compute $Q = x_1x_2P$ by given only $P, Q_1, Q_2$.

### B. Fuzzy Extractor

A fuzzy extractor [4] consists of two integral functions: a probabilistic key generation function denoted as Fu.Gen(·) and a deterministic reproducible function represented by Fu.Rep(·). These functions are described below.

1) Fuzzy.Gen(·) accepts a string $R$ as an input and returns two outputs, a secret key $K \in (0,1)^n$ and helper data $h \in (0,1)^*$, which may be stated as follows: Fuzzy.Gen($R$) = $(K, h)$.
2) Fuzzy.Rep(·) will recover the secret key $K$ using $R'$ and the helper data $h \in (0,1)^*$ as inputs, which may be stated as follows: Fuzzy.Rep($R', h$) = $K$, If the Hamming distance between $R$ and $R'$ is small.

### C. Physical Unclonable Function

A PUF is a security technique based on physical properties that is used to generate unique and unreplicable authentication keys. It was first proposed by Pappu [33] in 2002. PUFs exploit natural variations in semiconductor manufacturing processes, such as the threshold voltage of transistors, delay, or impedance of circuits, to create a unique response $R$ = PUF($C$). When presented with a challenge, typically an electronic signal or data, the PUF produces a response based on its physical properties. These responses are random and unique for each chip, thus they can be used as a kind of hardware "fingerprint" [34].

As a result, many authentication negotiation fields are secured with PUF. Among the many PUF categories, the ring oscillator (RO) PUF is known for its simplest structure and is well-suited for both ASIC and FPGA implementations. However, RO-PUF does not produce large amounts of CRP. And many PUFs have problems with key randomness and are vulnerable to attacks such as machine learning modeling (MLMo) and DPA. Miah and Hossain [38] described a cross-RO (CRO)-PUF to generate strong encryption keys. The proposed CRO provides an optimal key selection technique to select the most resilient key, demonstrating significant resistance to MLMo and DPA attacks. In addition, CRO PUFs have lower hardware overhead than other lightweight PUFs, highlighting their high efficiency and lightweight nature. Therefore, we equip the mobile edge server (MESC) with an improved structure based on RO-PUF: (CRO)-PUF to protect the security of the server. The main features of CRO-PUFs include the following aspects.

1) *Uniqueness:* Each PUF is unique, like a human fingerprint, which makes them suitable for device authentication and key generation.
2) *Unclonability:* Due to the physical properties of PUFs based on hardware, a PUF with the same response cannot be replicated even using the same design and fabrication process. This makes PUF-generated keys difficult to clone or predict.
3) *Tamper Resistance:* The PUF is sensitive to any tampering or alteration of the hardware, and any unauthorized modification may cause a change in the PUF output, thus increasing the security of the key.
4) *No Stored Key:* The PUF does not store the key itself, but generates the key dynamically each time it is needed. This means that even if the adversary has access to the storage area, he cannot find the statically stored key, which describes the challenge and represents the reaction. In order to maintain a constant PUF output during temperature fluctuations, we use a fuzzy extractor to improve the reliability and applicability of PUF in different applications.
5) *Well-Resistance of MLMo and DPA Attacks:* CRO-PUF is well resistant to machine learning attacks and power analysis attacks.
6) *Low Cost:* CRO-PUF costs less than some other PUFs.

*Remark 1:* To evaluate the resistance of the CRO-PUF against MLMo attacks, [38] designed a systematic security validation scheme. This approach not only examines the inherent security of the fundamental PUF structure but also verifies the practical effectiveness of the optimal key selection technique in enhancing system robustness. The experiment employed six widely used machine learning classifiers to conduct modeling attacks on the keys generated by the CRO-PUF. These algorithms cover diverse learning paradigms, including decision tree (DT), random forest (RF), support vector machine (SVM), naive bayes (NB), $K$-nearest neighbors (KNNs), and logistic regression (LR). The training set to test set ratio was set at 80:20, ensuring the statistical significance of the evaluation results. Table II shows the prediction accuracy of the CRO-PUF keys, screened by the optimal key selection technique, when confronted with different machine learning attacks. The data indicates that the filtered keys exhibit exceptional resistance to modeling attacks.

To objectively assess the security performance of the CROPUF, [38] also conducted a comparative analysis with various existing PUF design schemes. Table III details the resistance capabilities of different PUF structures when facing machine learning attacks.

The comparative results demonstrate that the CRO-PUF significantly outperforms other design schemes in resisting MLMo attacks. The exceptional resistance of the CRO-PUF against machine learning attacks primarily stems from two key design features: first, the crossover RO structure, by incorporating multiplexers to select different inverter paths within the intercrossing paths, significantly enhances the complexity and randomness of the CRP space. Second, the optimal key selection technique based on Shannon entropy can proactively identify and exclude keys that may contain predictable patterns, thereby ensuring that the final keys used possess the strongest unpredictability. Experimental data confirm that unserialized CRP data indeed carries the risk of being identified by specific machine learning algorithms (such as LR and RF). However, after processing with the optimal key selection technique, this risk is effectively controlled within an acceptable range.

*Remark 2:* The stability of CRO-PUF is grounded in its fixed-path delay structure and process-variation entropy. With consistent gate delays (700 ps per path at 28 nm) and manufacturing mismatch as the sole randomness source, the response is reproducible intrachip and unique interchip. Entropy-based screening retains only the highest-entropy subset, eliminating weak instances. Empirically, no response collision is observed (mean HD $\approx 61.22$, $\sigma = 7.82$) and prediction rates under ML and DPA attacks remain $\leq 22\%$, confirming stable operation across voltage, temperature, and adversarial conditions [38].

TABLE II

RESISTANCE PERFORMANCE OF CRO-PUF OPTIMAL KEYS AGAINST MACHINE LEARNING ATTACKS (FIVE DATASETS) [38]

| ML Algorithm | Prediction Accuracy (%) | | | | |
|---|---|---|---|---|---|
| | Set 1 | Set 2 | Set 3 | Set 4 | Set 5 |
| Decision Tree (DT) | 10.85 | 10.95 | 11.35 | 12.95 | 10.55 |
| Random Forest (RF) | 9.70 | 10.25 | 10.30 | 12.60 | 11.75 |
| Support Vector Machine (SVM) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Naive Bayes (NB) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| K-Nearest Neighbors (KNN) | 0.05 | 0.10 | 0.05 | 0.15 | 0.25 |
| Logistic Regression (LR) | 19.75 | 19.05 | 20.35 | 20.95 | 20.2 |
| **Average per set** | **10.08** | **10.09** | **10.51** | **11.66** | **10.69** |

TABLE III

MACHINE LEARNING ATTACK RESISTANCE COMPARISON OF DIFFERENT PUFS [38]

| PUF Type | DT | RF | SVM | NB | KNN | LR |
|---|---|---|---|---|---|---|
| CRO-PUF | 10.85 | 9.70 | 0.00 | 0.00 | 0.05 | 19.75 |
| MAR-PUF | — | 58.98 | 57.30 | 56.30 | — | 53.50 |
| FLAM-PUF | — | — | 48.1-53.3 | — | — | 50.6-52.8 |
| Inv-RO | 66.20 | 67.30 | 67.80 | 60.30 | 69.60 | 64.30 |
| Arb-PUF | 64.30 | 65.60 | 66.80 | 60.50 | 68.20 | 63.30 |
| RO-PUF | 71.20 | 73.50 | 74.10 | 67.40 | 76.20 | 69.30 |
| Cfg-RO | 80.10 | 81.30 | 80.90 | 75.90 | 82.50 | 78.70 |

***Note:*** *CRO-PUF: Crossover Ring Oscillator PUF (this work); MAR-PUF: Multi-round Arbiter PUF; FLAM-PUF: Feedback-based Lightweight Anti-ML PUF; Inv-RO: Inverter-based Ring Oscillator PUF; Arb-PUF: Arbiter PUF; RO-PUF: Traditional Ring Oscillator PUF; Cfg-RO: Configurable Ring Oscillator PUF. All values represent prediction accuracy (%). Dashes (—) indicate data not reported in the respective studies.*

### D. System Model

The system model of the proposed three-factor authentication and key agreement protocol is illustrated in Fig. 1. This model constructs a secure communication framework tailored for medical edge computing scenarios. The system comprises three core participating entities: the trusted RA, the user (U), and the MESC. These entities complement each other in terms of resource capabilities, trust assumptions, and security responsibilities, collectively forming a secure, efficient, and distributed system suitable for resource-constrained environments. The detailed definitions and functions of each entity are described below.

*1) Trusted RA:* The trusted RA serves as the security cornerstone and trust anchor of the system. As a fully trusted third-party institution, it plays a central role during the system initialization phase and the registration phase of all entities. Its specific responsibilities include the following aspects.

1) **System Primary Key Generation and Management:** During the system start-up phase, the RA is responsible for generating and securely storing the global primary key for the entire system. This key is fundamental to all subsequent security operations.
2) **Entity Registration and Credential Distribution:** It is responsible for completing the identity registration of all users and edge servers. During this process, the RA generates unique security credentials (such as pseudo-identities, tokens, or partial keys) for each entity and distributes them through a secure channel.
3) **PUF Challenge Secure Configuration:** It is responsible for initializing edge servers integrated with PUF modules. The RA securely generates and distributes a set of initial PUF challenge values to the corresponding edge servers, while the PUF responses are generated only locally on the server, ensuring the secrecy of the responses. Crucially, after initialization is complete, the RA itself does not store any challenge-response pairs (CRPs), thereby fundamentally avoiding the large-scale security risks associated with central database breaches.
4) **Lifecycle Management:** When necessary, the RA handles dynamic entity addition, revocation, or key update operations, maintaining the long-term security of the system.

*2) User:* The user is the terminal entity in the system that requires access to edge computing services. In the specific application of this model, the user typically refers to mobile terminal devices (such as smartphones or tablets) held by doctors, nurses, or other healthcare personnel. These devices exhibit the following distinctive characteristics.

1) **Resource Constraints:** They are limited in computing power, storage capacity, and battery life, and cannot undertake complex public-key cryptographic operations (such as bilinear pairing operations).
2) **High Mobility:** Users may move within the coverage areas of different edge servers; therefore, the protocol needs to support secure, seamless service handover and reauthentication.
3) **Sensitive Operation Executor:** The user end initiates the authentication process and needs to negotiate a secure session key with the edge server to protect subsequently transmitted sensitive medical data (such as patient vital signs and electronic medical records).

*3) Mobile Edge Server:* The MESC is a computing node deployed at the network edge (e.g., within hospital ward floors and community clinics), acting as a bridge connecting users and the cloud medical data center. It is defined as a **"Semi-Trusted" entity** in the trust model.

1) **Protocol Execution Trust:** When executing the authentication and key agreement protocol, it strictly follows the predefined steps and will not actively disclose its long-term keys.
2) **Data Privacy Distrust:** It might be curious about user identity information and communication content, attempting passive analysis, or it could be subjected to physical capture or logical attacks by external adversaries.

**Core Security Features:** To enhance its own security protection, each edge server is embedded with a PUF module. The use of this hardware security primitive offers two core advantages as follows.

1) **Key Isolation:** The server's core secrets (e.g., session key derivation material) are bound to the physical characteristics of the PUF. Even if an attacker physically obtains the server and extracts its stored data, they cannot clone or reconstruct the PUF response, thus achieving effective key isolation.
2) **Storage-Less Security:** As mentioned earlier, the server does not need to store a complete CRP database locally.

Instead, it utilizes the real-time computation characteristics of the PUF to dynamically generate responses based on challenges distributed by the RA when needed, significantly reducing the risk of key leakage due to theft of the storage medium.

### E. Adversary Model

Based on some current adversary models, such as those presented in [35], [36], and [37], the adversary capacity can be summarized as follows.

1) The adversary $\mathcal{A}$ can reveal all the parameters stored in the user's smart device, using power analysis techniques.
2) $\mathcal{A}$ can control the information transmitted through the public channel, such as intercepting, modifying, replaying, deleting, and so on.
3) $\mathcal{A}$ can offline enumerate all pairs of $(\mathrm{PW}_i, \mathrm{ID}_i)$ from $(\mathcal{D}_{\mathrm{PW}}, \mathcal{D}_{\mathrm{ID}})$ within polynomial time, where $\mathcal{D}_{\mathrm{ID}}$ and $\mathcal{D}_{\mathrm{PW}}$ refer to the space of identities and passwords, respectively.
4) $\mathcal{A}$ can reveal any two of the legitimate user's passwords, biometrics, and information from the smart device.
5) $\mathcal{A}$ may obtain the keys from the previous session.
6) When carrying out ephemeral-secret leakage attacks, $\mathcal{A}$ can obtain the short-term parameters or the response of the PUF.
7) When considering perfect forward secrecy or carrying out KCI attacks, $\mathcal{A}$ can obtain the long-term private key of the edge server.

### F. Security Evaluation Criteria

To judge the security of PUF-based three-factor protocols in edge computing, based on the literature [38], [39], [40], we propose an evaluation criterion that considers PUFs and other security properties.

EC1 *Password Friendly:* Legitimate users can update and modify passwords locally without the involvement of the server.
EC2 *No Password Verifier Table:* The server does not store the password and password's validation in plaintext.
EC3 *No Password Exposure:* During registration, the legitimate user does not send a password to prevent a privileged insider from knowing the user's password.
EC4 *Three-Factor Security:* Even if an adversary had access to a legitimate user's password, biometrics, and the information in two of the smart devices, it would be difficult to calculate the third.
EC5 *Resistance to Known Attacks:* The protocol should be able to resist some potential attacks, such as replay attack, impersonation attack, offline guessing attack, DoS attack, de-synchronization attack, and stolen verifier-attack.
EC6 *Mutual Authentication and Key Agreement:* The user and the edge server should first authenticate each other and then negotiate a session key for subsequent secure communication.
EC7 *Ephemeral-Secret Leakage:* Even if the adversary obtains some temporary secret parameters used by the current session, the session key cannot be calculated.
EC8 *KCI Attack:* Even if the adversary obtains the long-term key of the server, he cannot obtain the important information of the legitimate user to impersonate the user.
EC9 *Forward Secrecy:* Even if the adversary obtains the long-term key of the server, it cannot calculate the session key of the current session and the previous session.
EC10 *Anonymity and Untraceability:* An adversary cannot obtain or track a user's real identity from the information passed on the open channel.

## III. Review and Analysis of Representative Protocols

### A. Review and Analysis of Jia et al.'s Protocol

In the Jia et al.'s protocol [12], there are three entities: MU, MS, and RC, and the protocol is mainly divided into three phases: initialization phase, registration phase, and authentication key agreement phase.

1) *Initialization Phase: RC* chooses a cyclic group $G$ with order $q$ over a nonsingular elliptic curve $E/F_p$ and a cyclic multiplicative group $G_T$ with the same order $q$. $P$ is the generator of $G$. The RC also randomly picks $s, \hat{s} \in Z_q^*$ and computes $P_{\mathrm{pub}} = sP$, $\hat{P_{\mathrm{pub}}} = \hat{s}P$, as well as four hash functions: $h_0 : \{0,1\}^* \to Z_q^*$, $h_1 : \{0,1\}^* \to \{0,1\}^\lambda$, $h_2 : \{0,1\}^* \to \{0,1\}^\lambda \times G \times G$, $h_3 : \{0,1\}^* \to \{0,1\}^l$, where $\lambda$ is security parameters and $l$ is the required length of the session key. RC keeps $(s, \hat{s})$ as the primary private key, others are published as public parameters.
2) *Registration Phase: MU* input $\mathrm{ID}_{\mathrm{mu}}$, $\mathrm{PW}_{\mathrm{mu}}$, $B_{\mathrm{mu}}$, $(\sigma_{\mathrm{mu}}, \tau_{\mathrm{mu}}) = \mathrm{Gen}(B_{\mathrm{mu}})$, computes his pseudonym $\mathrm{PID}_{\mathrm{mu}} = h_1(\mathrm{ID}_{\mathrm{mu}} \| \mathrm{PW}_{\mathrm{mu}} \| \sigma_{\mathrm{mu}})$, then MU sends $\mathrm{ID}_{\mathrm{mu}}, \mathrm{PID}_{\mathrm{mu}}$ and to RC in secure channel; RC randomly chooses $r_{\mathrm{mu}} \in Z_q^*$ and computes $R_{\mathrm{mu}} = r_{\mathrm{mu}}P$, $h_{\mathrm{mu}} = h_0(\mathrm{PID}_{\mathrm{mu}} \| R_{\mathrm{mu}})$, $\mathrm{SID}_{\mathrm{mu}} = (r_{\mathrm{mu}} + \mathrm{sh}_{\mathrm{mu}}) \bmod q$, RC returns $R_{\mathrm{mu}}$ and $\mathrm{SID}_{\mathrm{mu}}$ to MU.
RC chooses a unique identity $\mathrm{ID}_{\mathrm{ms}}$ to MS and computes $h_{\mathrm{ms}} = h_1(\mathrm{ID}_{\mathrm{ms}})$ and $\mathrm{SID}_{\mathrm{ms}} = (1/(\hat{s} + h_{\mathrm{ms}}))P$, then RC sends $\{\mathrm{SID}_{\mathrm{ms}}\}$ to MS, $\mathrm{SID}_{\mathrm{ms}}$ is the private key of MS.
3) *Authentication Key Agreement Phase:* In this phase, MU logs into the mobile sever MS and completes mutual authentication. MU first inputs $\mathrm{ID}_{\mathrm{mu}}$, $\mathrm{PW}_{\mathrm{mu}}$, and $B_{\mathrm{mu}}$ to verify the mobile device, then MU chooses $x$ and computes $M$, $N$, and $\delta$, where $M = x(\hat{P_{\mathrm{pub}}} + h_0(\mathrm{ID}_{\mathrm{ms}})P)$, $N = h_2(g^x) \oplus (\mathrm{PID}_{\mathrm{mu}} \| X \| R_{\mathrm{mu}})$, $\delta = \mathrm{SID}_{\mathrm{mu}} + xh_0(\mathrm{PID}_{\mathrm{mu}} \| X \| R_{\mathrm{mu}} \| \mathrm{TS}_{\mathrm{mu}})$, $\mathrm{PID}_{\mathrm{mu}} = h_1(\mathrm{ID}_{\mathrm{mu}} \| \mathrm{PW}_{\mathrm{mu}} \| B_{\mathrm{mu}})$. Then, user MU sends $\{M, N, \delta, \mathrm{TS}_{\mathrm{mu}}\}$ to MS. After MS receives it, MS first computes $g^x$ by $\mathrm{SID}_{\mathrm{ms}}$ and gets the user's information by $g^x$ and $N$. Then, MS computes $W$ to verify the signature $\delta$, where $W = R_{\mathrm{mu}} + h_0(\mathrm{PID}_{\mathrm{mu}} \| R_{\mathrm{mu}})P_{\mathrm{pub}}$. Finally MS chooses $y$ and computes $t, \mathrm{SK}$, where $t = h_1(\mathrm{PID}_{\mathrm{mu}} \| \mathrm{ID}_{\mathrm{ms}} \| \mathrm{yW} \| \mathrm{TS}_{\mathrm{ms}})$, $\mathrm{SK}_{\mathrm{ms}} = h_3(\mathrm{ID}_{\mathrm{ms}} \| \mathrm{PID}_{\mathrm{mu}} \| yX)$. And then, MS send $\{t, Y, \mathrm{TS}_{\mathrm{ms}}\}$ to MU. After MU receives it, MU checks whether $t = h_1(\mathrm{PID}_{\mathrm{mu}} \| \mathrm{ID}_{\mathrm{ms}} \|$

$\mathrm{SID}_{\mathrm{mu}}Y \parallel \mathrm{TS}_{\mathrm{ms}}$) and computes the session key $\mathrm{SK}_{\mathrm{mu}}$, where $\mathrm{SK}_{\mathrm{ms}} = h_3(\mathrm{ID}_{\mathrm{ms}} \parallel \mathrm{PID}_{\mathrm{mu}} \parallel xY)$. It is easy to see that $\mathrm{SK}_{\mathrm{mu}} = \mathrm{SK}_{\mathrm{ms}}$ since $xY = yX$.

4) *Analysis of Jia et al.'s Protocol:* According to [13], since the parameters $\{\mathrm{PID}_{\mathrm{mu}}, xY\}$ used for computing the session key rely solely on and are protected only by the ephemeral parameter $x$, once $x$ is compromised, an adversary can easily calculate the session key. Therefore, the protocol by Jia et al. cannot resist ephemeral-secret leakage attacks. And based on evaluation indicators, we find that Jia et al.'s protocol cannot resist offline password guessing attacks and insider attacks, besides being unable to resist offline key guessing attacks and ephemeral-secret leakage attacks. We elaborate on these below.

a) *Offline password guessing attacks:* It is assumed that the adversary $\mathcal{A}$ can obtain the parameter information $\{\mathrm{PW}^*_{\mathrm{mu}}, \mathrm{SID}^*_{\mathrm{mu}}, \mathrm{Gen}, \mathrm{Rep}, R_{\mathrm{mu}}, \tau_{\mathrm{mu}}\}$ in the intelligent device through the channel test attack, and obtain the biometric characteristics $B'_{\mathrm{mu}}$, then he can launch an offline password guessing attack to get MU's password following the process given below:

Step 1: $\mathcal{A}$ can choose a possible identity and password $\mathrm{ID}'_i, \mathrm{PW}_i \in \mathcal{D}_{\mathrm{ID}} \times \mathcal{D}_{\mathrm{PW}}$.

Step 2: $\mathcal{A}$ invoke Rep algorithm, $\sigma_{\mathrm{mu}} = \mathrm{Rep}(B'_{\mathrm{mu}}, \tau_{\mathrm{mu}})$ and computes $\mathrm{SID}'_{\mathrm{mu}} = \mathrm{SID}^*_{\mathrm{mu}} \oplus h_0(\mathrm{PW}'_{\mathrm{mu}} \parallel R_{\mathrm{mu}} \parallel \sigma_{\mathrm{mu}})$.

Step 3: $\mathcal{A}$ computes $(\mathrm{PW}^*_{\mathrm{mu}})' = h_1(\mathrm{ID}'_{\mathrm{mu}} \parallel \mathrm{PW}'_{\mathrm{mu}} \parallel \mathrm{SID}'_{\mathrm{mu}})$, then check $\mathrm{PW}^*_{\mathrm{mu}}? = (\mathrm{PW}^*_{\mathrm{mu}})'$.

If they are not equal, the adversary $\mathcal{A}$ reselects a set of possible identities and passwords in the identity password space and repeats step 2–step 3 until successful verification, the computational complexity is $\mathcal{O}(|\mathcal{D}_{\mathrm{ID}}| \times |\mathcal{D}_{\mathrm{PW}}| \times 2T_h)$, where $|\mathcal{D}_{\mathrm{PW}}|$ denotes the size of password space $\mathcal{D}_{PW}$, $|\mathcal{D}_{\mathrm{ID}}|$ denotes the size of identity space $\mathcal{D}_{ID}$ and $T_h$ denotes the running time for hash operation [40].

b) *Insider attacks:* Suppose an internal adversary $\mathcal{A}$ obtains the registration information and biometrics of legitimate user, i.e., $B_{\mathrm{mu}}, \mathrm{ID}_{\mathrm{mu}}, \mathrm{PID}_{\mathrm{mu}}$, $\mathrm{PID}_{\mathrm{mu}} = h_1(\mathrm{ID}_{\mathrm{mu}} \parallel \mathrm{PW}_{\mathrm{mu}} \parallel \sigma_{\mathrm{mu}})$, then $\mathcal{A}$ can guess user's password.

### B. Review and Analysis of Lee et al.'s Protocol

In Lee et al.'s protocol [14], there are three entities: $U_i$, $\mathrm{MEC}_j$, and RC, and the protocol is mainly divided into three phases: initialization phase, registration phase, and authentication key agreement phase.

1) *Initialization Phase: RC* chooses a cyclic group $G$ with order $q$ over a nonsingular elliptic curve $E/F_p$ and a cyclic multiplicative group $G_T$ with the same order $q$. $P$ is the generator of $G$, RC also randomly picks $s, \hat{s} \in Z^*_q$ and computes $s, \hat{s} \in Z^*_q$, RC keeps $(s, \hat{s})$ as the primary private key and publishes $\{G, G_T, g, q, P, P_{\mathrm{pub}}, \hat{P_{\mathrm{pub}}}, h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8\}$ as public parameters.

2) *Registration Phase:* The user $U_i$ selects the identity $\mathrm{ID}_i$ and corresponding password $\mathrm{PW}_i$ and then inputs the biometric $\mathrm{BIO}_i$. Then, by using the fuzzy extractor, $U_i$ computes $(\varsigma_i, \theta_i) = \mathrm{Gen}(\mathrm{BIO}_i)$ and $\eta_i = h_0(\mathrm{ID}_i \parallel \varsigma_i)$. Then, $U_i$ sends a registration request $\{\mathrm{ID}_i, \varsigma\}$ to RC through a secure channel. RC randomly chooses $r_i \in Z^*_q$ and computes $R_i = r_iP$, $h_i = h_1(\mathrm{ID}_i \parallel R_i)$, $\mathrm{SID}_i = (r_i + \hat{s}h_i) \bmod q$ and $\mathrm{RID}_i = h_2(\eta_i \parallel \hat{s})$, RC returns $R_i$ and $\mathrm{SID}_i$ to $U_i$. After receiving the response messages, $U_i$ computes $C_i = h_3(\mathrm{ID}_i \parallel \mathrm{PW}_i \parallel \varsigma_i)$, $D_i = C_i \oplus \mathrm{SID}_i$, $E_i = R_i \oplus h_0(\mathrm{ID}_i \parallel C_i)$, $F_i = h_4(C_i)$, $U_i$ stores $\{\theta_i, D_i, E_i, F_i\}$ in the device. $\mathrm{MEC}_j$ sends a registration request to RC with its identity $\mathrm{ID}_j$, RC computes $h_j = h_1(\mathrm{ID}_j)$, $\mathrm{SID}_j = (1/\hat{s} + h_j)P$ and sends $\{\mathrm{SID}_j\}$ to $\mathrm{MEC}_j$. After receiving the response, $\mathrm{MEC}_j$ stores $\mathrm{SID}_j$ in the memory.

3) *Authentication Key Agreement Phase:* In this phase, $U_i$ logs into MESC $\mathrm{MEC}_j$ and completes mutual authentication. $U_i$ first inputs $\mathrm{ID}_i, \mathrm{PW}_i$, and $\mathrm{BIO}_i$ to verify the mobile device, then $U_i$ chooses $x_i$ and computes $M_i, \delta_i$, and $\mathrm{Auth}_i$, where $M_i = x_i(\hat{P_{\mathrm{pub}}} + h_5(\mathrm{ID}_j)P)$, $\sigma_i = \mathrm{SID}_i + h_7(\mathrm{ID}_i \parallel \mathrm{ID}_j \parallel L_i \parallel T_i)$, $\mathrm{Auth}_i = \mathrm{Enc}_{K_i}(\mathrm{ID}_i, R_i, L_i, \mathrm{UX}_i)$, $L_i = h_0(\mathrm{ID}_i \parallel K_i)$, $K_i = h_6(e(P, P)^{x_i})$, $\mathrm{UX}_i = x_iP$. Then, user $U_i$ sends $\{M_i, \sigma_i, \mathrm{Auth}_i, T_i\}$ to $\mathrm{MEC}_j$. After $\mathrm{MEC}_j$ receives it, $\mathrm{MEC}_j$ first computes $K_i$ by $\mathrm{SID}_j$ and decrypts $\mathrm{Auth}_i$ to get the user's information by using $K_i$. Then, $\mathrm{MEC}_j$ verifies the signature $\sigma_i$. Finally $\mathrm{MEC}_j$ chooses $y_j$ and computes $\mathrm{SK}_j, \mathrm{Auth}_j$, where $\mathrm{SK}_j = h_9(\mathrm{ID}_i \parallel \mathrm{ID}_j \parallel L_i \parallel \mathrm{UX}_i y_j \parallel T_i \parallel T_j)$, $\mathrm{Auth}_j = h_3(K_i \parallel R_i \parallel \mathrm{SK}_j)$. And then, $\mathrm{MEC}_j$ sends $\{\mathrm{Auth}_j, \mathrm{MY}_j, T_j\}$ to $U_i$. After $U_i$ receives it, $U_i$ computes the session key $\mathrm{SK}_i$, where $\mathrm{SK}_i = h_9(\mathrm{ID}_i \parallel \mathrm{ID}_j \parallel L_i \parallel x_i\mathrm{MY}_j \parallel T_i \parallel T_j)$ and checks whether $\mathrm{Auth}_j = h_3(K_i \parallel R_i \parallel \mathrm{SK}_i)$. It is easy to see that $\mathrm{SK}_i = \mathrm{SK}_j$ since $x_i\mathrm{MY}_j = y_j\mathrm{UX}_i$.

4) *Analysis of Lee et al.'s Protocol:* Based on evaluation indicators, we find that Lee et al.'s protocol cannot resist offline password guessing attacks, ephemeral-secret leakage attacks. We will elaborate on these below.

a) *Offline password guessing attack:* It is assumed that the adversary $\mathcal{A}$ can obtain the parameter information $\{\theta_i, D_i, E_i, F_i\}$ in the intelligent device through the channel test attack, and obtain the biometric characteristics $\mathrm{BIO}^*_i$, then he can launch an offline password guessing attack to get $\mathrm{U_i}$'s password following the process given below:

Step 1: $\mathcal{A}$ can choose a possible identity and password $\mathrm{ID}'_i, \mathrm{PW}_i \in \mathcal{D}_{\mathrm{ID}} \times \mathcal{D}_{\mathrm{PW}}$.

Step 2: $\mathcal{A}$ invoke algorithm Rep, $\varsigma_i = \mathrm{Rep}(\mathrm{BIO}_i, \theta_i)$ and computes $C'_i = h_3(\mathrm{ID}_i \parallel \mathrm{PW}_i \parallel \varsigma_i)$.

Step 3: $\mathcal{A}$ computes $F'_i = h_4(C'_i)$, then check $F'_i? = F_i$.

If they are not equal, the adversary reselects a set of possible identities and passwords in the identity password space and repeats step 2–step 3 until successful verification, the computational complexity is $\mathcal{O}(|\mathcal{D}_{\mathrm{ID}}| \times |\mathcal{D}_{\mathrm{PW}}| \times 2T_h)$.

b) *Ephemeral-secret leakage attack:* Assuming temporary parameter $x_i$ are leaked, then adversary $\mathcal{A}$

can compute $X_i = x_iP$ and $K_i = h_6(X_i)$, then $\mathcal{A}$ can decrypt $\text{Auth}_i$ with $K_i$ and get the important parameters $\{\text{ID}_i, R_i, L_i\}$ which are used to compute $\text{SK}_i$, since $(\text{ID}_i, R_i, L_i, \text{UX}_i) = \text{Dec}_{K_i}(\text{Auth}_i)$; finally, the adversary computes $\text{SK}_i = h_9(\text{ID}_i \parallel \text{ID}_j \parallel L_i \parallel x_i\text{MY}_j \parallel T_i \parallel T_j)$.

c) *KCI:* Assuming $\text{MEC}_j$'s private key $\text{SID}_j$ are leaked, then then adversary $\mathcal{A}$ can compute $K_i = h_6(e(M_i, \text{SID}_j))$ and decrypt $\text{Auth}_i$ with $K_i$ to get the important parameters $\{\text{ID}_i, R_i, L_i\}$. And then adversary $\mathcal{A}$ can get user's private key $\text{SID}_i$, since $\text{SID}_i = \sigma_i - h_7(\text{ID}_i \parallel \text{ID}_j \parallel L_i \parallel T_i)$. The adversary can then forge a new $M_i, \sigma_i, \text{Auth}_i$ to impersonate the user $U_i$.

### C. Review and Analysis of Tian et al.'s Protocol

In Tian et al.'s protocol [15], there are three entities: MU, MS, and RC, and the protocol is mainly divided into three phases: initialization phase, registration phase, and authentication key agreement phase. The authentication and key agreement phase includes the process of updating user information.

1) *Initialization Phase: RC* chooses a cyclic group $G$ with order $q$ and a cyclic multiplicative group $G_T$ with the same order which satisfy $e : G \times G \rightarrow G_T$. $P$ is the generator of $G$, RC also randomly picks $s, \hat{s} \in Z_q^*$ and computes $P_{\text{pub}} = sP$, $\hat{P_{\text{pub}}} = \hat{s}P$, $g = e(P, P)$. RC keeps $(s, \hat{s})$ as the primary private key and publishes $\{G, G_T, e, g, q, P, P_{\text{pub}}, \hat{P_{\text{pub}}}, H_1, H_2, H_3, H_4, \text{FzExt}\}$ as public parameters, where $\text{FzExt} = (\text{Gen}, \text{Rep})$.

2) *Registration Phase:* The user MU selects the identity $\text{ID}_u$ and corresponding password $\text{PW}_u$ and then inputs the biometric $B_u$. Then, by using the fuzzy extractor, MU computes $(\sigma_u, \tau_u) = \text{FzExt.Gen}(B_u)$ and $\text{FPID}_u = H_2(\text{ID}_u \parallel \text{PW}_u\sigma_u)$. Then, MU sends a registration request $\{\text{ID}_u, \text{FPID}_u, \text{type}\}$ to RC through a secure channel, where $\text{type} = 0$. RC checks if the user is registrated and then randomly chooses $r_u \in Z_q^*$. RC generates one-time passwoed otp and computes $\text{TPID}_u = H_2(\text{FPID}_u \parallel \text{otp})$, $Ru = r_uP$, $Q_u = H_1(\text{TPID}_u \parallel Ru)$, $\text{SID}_u = (r_u + \text{sQ}_u) \bmod q$. RC returns $R_u$, otp and $\text{SID}_u$ to User and save $(\text{FPID}_u, R_u, r_u)$. After receiving the response messages, MU computes $\text{VPW}_u = H_2(\text{PW}_u \parallel \text{ID}_u \parallel \text{otp} \parallel \text{SID}_u)$, $\text{SID}_u^* = \text{SID}_u \oplus H_1(\text{PW}_u \parallel \sigma_u \parallel \text{otp})$. Then, MU saves $(\text{VPW}_u, \text{SID}_u^*, R_u, \tau_u, \text{otp}, \text{FzExt})$ in its memory. MS sends its identity $\text{ID}_{\text{ms}}$ to RC, RC computes $\text{SID}_{\text{ms}} = (1/\hat{s} + H_1(\text{ID}_{\text{ms}}))P$ and sends $\text{SID}_{\text{ms}}$ to MS. After recieving the response, MS stores $\{\text{SID}_{\text{ms}}\}$ as its private key.

3) *Authentication Key Agreement Phase:* In this phase, MU logs into MESC MS and completes mutual authentication. MU first inputs $\text{ID}_u, \text{PW}_u$, and $B_u$ to verify the mobile device, then MU chooses $x$ and computes $M, N$, and $\delta$, where $M = x(\hat{P_{\text{pub}}} + H_1(\text{ID}_{\text{ms}})P)$, $N = H_3(g^x) \oplus (\text{TPID}_u \parallel X \parallel R_u)$, $\delta = \text{SID}_u + xh_1(\text{TPID}_u \parallel X \parallel R_u \parallel T_u)$. Then, MU sends $\{M, N, \delta, T_u\}$ to MS. After MS receives it, MS first computes $g^x$ by $\text{SID}_{\text{ms}}$ and gets the user's information $\{\text{TPID}_u, X, R_u\}$ by $g^x$ and $N$. Then, MS computes $W$ to verify the signature $\delta$, where $W = R_u + P_{\text{pub}}H_1(\text{TPID}_u \parallel R_u)$. Finally, MS chooses $y$ and compute $t, \text{SK}$, where $\eta = H_2(\text{TPID}_u \parallel \text{ID}_{\text{ms}} \parallel \text{yW} \parallel T_{\text{ms}})$, $\text{SK} = H_4(\text{TPID}_u \parallel \text{ID}_{\text{ms}} \parallel yX)$. And then MS send $\{\eta, Y, T_{\text{ms}}\}$ to User. After MU receives it, MU checks whether $\eta = H_2(\text{TPID}_u \parallel \text{ID}_{\text{ms}} \parallel \text{yW} \parallel T_{\text{ms}})$ and compute the session key SK, where $\text{SK} = H_4(\text{TPID}_u \parallel \text{ID}_{\text{ms}} \parallel yX)$.

4) *Analysis of Tian et al.'s Protocol:* Based on evaluation indicators, we find that Tian et al.'s protocol cannot resist offline password guessing attack, ephemeral-secret leakage attack, and cannot satisfy unlinkability. We will elaborate on these below.

a) *Offline password guessing attack:* It is assumed that the adversary $\mathcal{A}$ can obtain the parameter information $(\text{VPW}_u, \text{SID}_u^*, R_u, \tau_u, \text{otp}, \text{FzExt})$ in the memory through the channel test attack, and obtain the biometric characteristics $B_u^*$, then $\mathcal{A}$ can launch an offline password guessing attack to get User's password following the process given below:

Step 1: $\mathcal{A}$ can choose a possible identity and password $\text{ID}_u', \text{PW}_u \in \mathcal{D}_{\text{ID}} \times \mathcal{D}_{\text{PW}}$.

Step 2: $\mathcal{A}$ invoke the fuzzy extrator algorithm Fz.Rep, $\sigma_u = \text{Fz.Rep}(B_u, \tau_u)$ and computes $\text{FPID}_u' = H_2(\text{ID}_u' \parallel \text{PW}_u'\sigma_u)$, $\text{TPID}_u' = H_2(\text{FPID}_u' \parallel \text{otp})$, $\text{SID}_u' = \text{SID}_u^* \oplus H_1(\text{PW}_u' \parallel \sigma_u \parallel \text{otp})$.

Step 3: $\mathcal{A}$ computes $\text{VPW}_u' = H_2(\text{PW}_u' \parallel \text{ID}_u' \parallel \text{otp} \parallel \text{SID}_u')$, then check $\text{VPW}_u'? = \text{VPW}_u$.

If they are not equal, the adversary reselects a set of possible identities and passwords in the identity password space and repeats step 2–step 3 until successful verification, the computational complexity is $\mathcal{O}(\mid \mathcal{D}_{\text{ID}} \mid \times \mid \mathcal{D}_{\text{PW}} \mid \times 4T_h)$.

b) *Ephemeral-secret leakage attack:* Assuming temporary parameter $x$ are leaked, then adversary $\mathcal{A}$ can compute $g_x = g^x$ and $\text{TPID}_u \parallel X \parallel R_u = H_3(g_x \oplus N)$, where $N$ is passed on the open channel; finally, the adversary computes $\text{SK}_u = H_4(\text{TPID}_u \parallel \text{ID}_{\text{ms}} \parallel xY)$.

c) *No unlinkability:* Even though Tian et al. updated the important credentials of the user in the authentication phase to protect the complete anonymity and unlinkability of the user, even the server could not link to the same user through these credentials, but we actually found that in the authentication process, the server could obtain $R_u$, which was invariant for the same user, unless reregistered. So the server can link different sessions to the same user through $R_u$, which does not satisfy the unlinkability stated by Tian et al.

### D. Comparative Summary

Jia et al. [12], Lee et al. [14], and Tian et al. [15] recently proposed identity-based AKA schemes for MEC. All three protocols achieve lightweight computation and resist replay attacks, but they share the following common weaknesses: 1) the server must store or update a large set of CRPs, which introduces a single-point leakage risk; 2) the password-verifier or biometric helper data stored in the device allows an

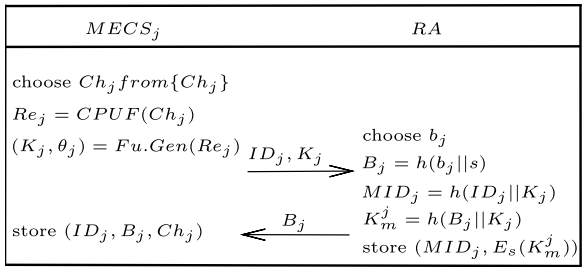


Fig. 2. Registration of edge server.

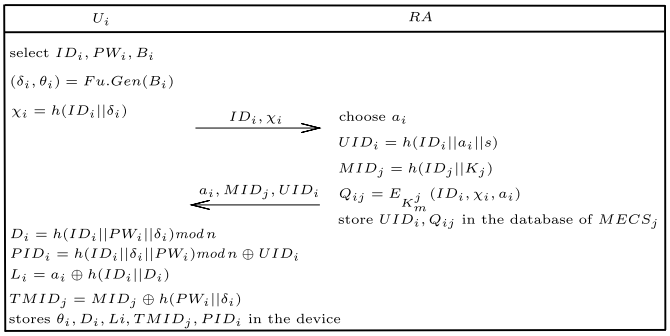


Fig. 3. Registration of user.

adversary to perform offline password guessing once the card is lost; 3) ephemeral secrets (e.g., random exponents $x$ and $y$) are transmitted or used in a way that their leakage immediately exposes the session key; and 4) at least one static parameter (e.g., $R_u$) is reused across sessions, breaking user unlinkability. These fatal defects motivate our design of PUF3FAka, which eliminates CRP storage, adopts fuzzy verification to defeat guessing attacks, and achieves full forward secrecy and unlinkability without sacrificing efficiency.

## IV. Proposed PUF3FAka Protocol

### A. Protocol Overview

Before delving into the detailed steps, we provide a high-level overview of the proposed PUF3FAka protocol to facilitate understanding. As illustrated in Figs. 2–4, the protocol involves three entities: the user ($U_i$), the server ($\text{MECS}_j$), and the trusted authority (RA). The primary design objectives are to address the security vulnerabilities in existing schemes (e.g., protocols by Jia et al. [12], Lee et al. [14], and Tian et al. [15]), where critical parameters stored on either the user or server side could be exploited to compute session keys. To this end, our protocol aims to construct a secure and efficient authentication and key agreement scheme for MEC environments.

To achieve these objectives, the PUF3FAka protocol introduces two core innovations as follows.

1) Deployment of CRO-PUF and long-term keys on the server side to fundamentally safeguard session key generation.
2) Incorporation of fuzzy verification techniques to protect user identity and password information.

The protocol operates sequentially through three phases as follows.

1) *Initialization Phase: RA* performs system setup, generates public parameters, and configures initial security settings. This phase establishes the foundational cryptographic environment for the protocol.
2) *Registration Phase:* Both $U_i$ and $\text{MECS}_j$ register with the RA through secure channels. During this phase, $U_i$ submits identity and authentication factors, while $\text{MECS}_j$ registers its CRO-PUF instance and receives corresponding credentials. The RA issues necessary security parameters to each entity.
3) *Authentication and Key Agreement Phase:* This is the core online interaction phase. The User initiates a session request, after which all three entities exchange messages containing nonces, timestamps, and cryptographic constructs (e.g., hash values and PUF responses). Mutual authentication is achieved by verifying the preshared credentials and leveraging the physical unclonability of the PUF. Concurrently, $U_i$ and $\text{MECS}_j$ negotiate a common session key based on ephemeral shared secrets.

In summary, by integrating the hardware security features of PUFs, a three-factor authentication mechanism, and fuzzy verification, the PUF3FAKA protocol is designed to achieve a robust authentication scheme resistant to parameter leakage attacks. The subsequent subsections detail the specific steps and mathematical operations of each phase.

### B. System Initialization Phase

In the initialization phase, RA chooses a cyclic group $G$ with order $q$ over a nonsingular elliptic curve $E/F_q$ and chooses $s$ as its primary private key. $P$ is the generator of $G$. RA also randomly generates a lot of challenges $\{\text{Ch}_j\}$, $j = 1, \ldots, t$ for the MEC servers $\text{MECS}_j$ and sends these challenges $\{\text{Ch}_j\}$ to $\text{MECS}_j$, as well as hash function: $h : \{0,1\}^* \to \{0,1\}^*$, $\mathcal{O}$ is the infinity point of $G$. RA publishes $\{P, G, h()\}$.

### C. Registration Phase

During this phase, the user and the edge server register with the registration center independently. Upon successful registration, the user locally stores $\{\theta_i, D_i, Li, \text{PID}_i, \text{TMID}_j\}$ and the edge server stores $\{\text{ID}_j, B_j, \text{Ch}_j\}$, which are subsequently used for login and authentication negotiation. The complete procedure is illustrated in Figs. 2 and 3.

$\text{MECS}_j$ chooses its identity $\text{ID}_j$ and randomly chooses $\text{Ch}_j$ from $\{\text{Ch}_j\}$, $j = 1, \ldots, t$, then $\text{MECS}_j$ computes $\text{Re}_j = \text{CPUF}(\text{Ch}_j)$ and invokes the fuzzy extractor to get the secret key $K_j$ and sends $\text{ID}_j, K_j$ to RA through secure channel. After receiving those, RA chooses $b_j$ randomly and computes $B_j = h(b_j \| s)$, $K_m^j = h(B_j \| K_j)$. Then, RA sends $B_j$ to $\text{MECS}_j$ and stores $\{\text{ID}_j, E_s(K_m^j)\}$. $\text{MECS}_j$ only stores $\{\text{ID}_j, B_j, \text{Ch}_j\}$ in database.

The user $U_i$ selects his/her identity $\text{ID}_i$ and password $\text{PW}_i$, and $U_i$ inputs the biometric $B_i$. Then, $U_i$ invokes th generation algorithm Fu.Gen() and computes $(\delta_i, \theta_i) = \text{Fu.Gen}(B_i)$, $U_i$ computes $\chi_i = h(\text{ID}_i \| \delta_i)$ and sends the registration request $\text{ID}_i, \chi_i$ to RA securely. Then, RA randomly chooses $a_i$ and computes $\text{UID}_i = h(\text{ID}_i \| a_i \| s)$ and $\text{MID}_j = h(\text{ID}_j)$.

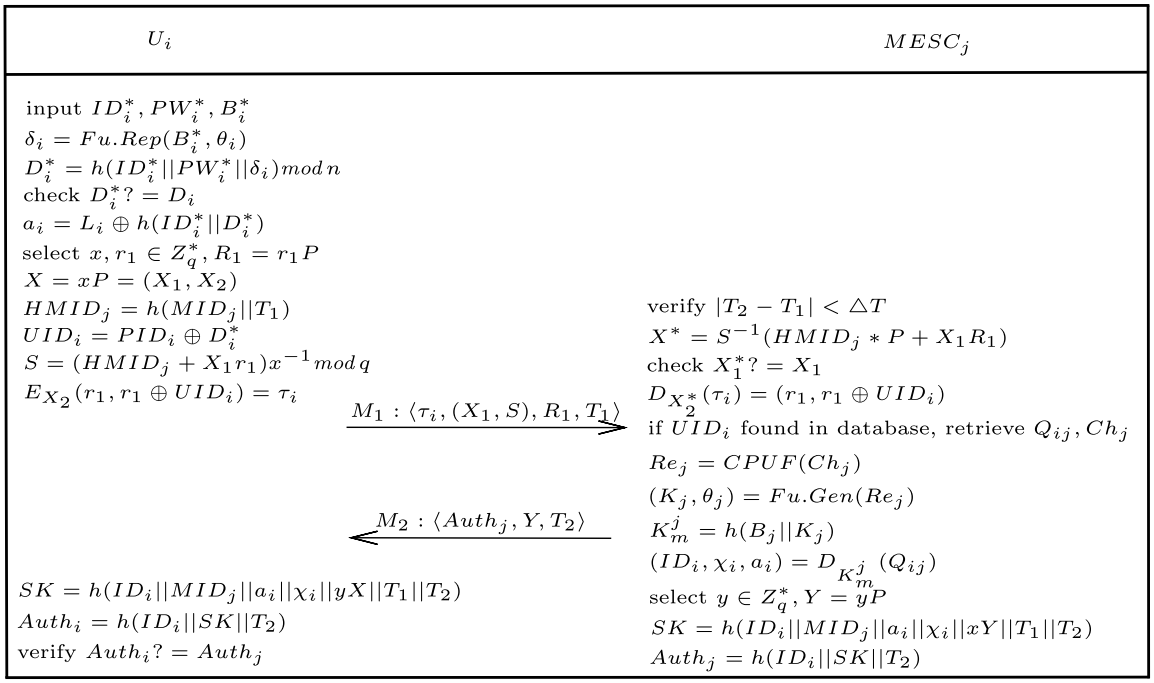


Fig. 4. Login and authentication phase of PUF3FAka protocol.

At the same time, RA encrypts $\{\mathrm{ID}_i, \chi_i, a_i\}$ with the server's key $K_m^j$ and stores $\mathrm{MID}_j, Q_{ij}$ in the $\mathrm{MECS}_j$ 's database; and then RA returns $a_i, \mathrm{UID}_i$ and $\mathrm{MID}_j$ to $U_i$. After receiving these, $U_i$ computes $D_i = h(\mathrm{ID}_i \parallel \mathrm{PW}_i \parallel \delta_i) \bmod n$, $\mathrm{PID}_i = h(\mathrm{ID}_i \parallel \delta_i \parallel \mathrm{PW}_i) \bmod n \oplus \mathrm{UID}_i$ and $L_i = a_i \oplus h(\mathrm{ID}_i \parallel D_i)$, where $|\mathcal{D}_{\mathrm{ID}}| = |\mathcal{D}_{\mathrm{PW}}| = 10^6$, $2^4 \leq n \leq 2^8$ [40]. $U_i$ stores $\{\theta_i, D_i, Li, \mathrm{PID}_i, \mathrm{TMID}_j\}$ in the device.

### D. Login and Mutual Authentication Phase

In this phase, the legitimate user $U_i$ logs into the edge computing server $\mathrm{MECS}_j$, and the two parties authenticate each other and negotiate a session key SK for subsequent secure communication. Fig. 4 further provides a comprehensive flowchart that delineates the step-wise execution of the proposed protocol.

S1: $U_i$ inputs identity $\mathrm{ID}_i$, password $\mathrm{PW}_i$, and biometric $B_i^*$. The device then has the following aspects.

a) Invokes the fuzzy reproduction function $\mathrm{Fu.Rep}(B_i^*)$ to recover $\delta_i$.
b) Computes $D_i^* = h(\mathrm{ID}_i \parallel \mathrm{PW}_i \parallel \delta_i) \bmod n$.
c) Verifies if $D_i^* = D_i$. If not equal, the session is terminated. Otherwise, it proceeds.
d) Computes $a_i = L_i \oplus h(\mathrm{ID}_i \parallel D_i^*)$.

S2: $U_i$ then has the following aspects.

a) Selects random numbers $x, r_1$.
b) Computes

$$X = xP, \quad R_1 = r_1 P$$
$$\mathrm{UID}_i = D_i^* \oplus \mathrm{PID}_i, \quad \mathrm{HMID}_j = h_1\left(\mathrm{MID}_j \parallel T_1\right).$$

c) Generates a digital signature $(X_1, S)$, where $S = (\mathrm{HMID}_j + X_1 r_1) x^{-1} \bmod q$.
d) Encrypts the tuple $(r_1, r_1 \oplus \mathrm{MID}_i)$ using key $X_2$ to produce $\tau_i = E_{X_2}(r_1, r_1 \oplus \mathrm{MID}_i)$.
e) Sends the authentication request message $\{\tau_i, (X_1, S), T_1, R_1\}$ to $\mathrm{MECS}_j$.

S3: Upon receiving the request, $\mathrm{MECS}_j$ has the following aspects.

a) Checks the freshness of timestamp $T_1$.
b) Computes $X^* = S^{-1}(\mathrm{HMID}_j \cdot P + X_1 R_1) = (X_1^*, X_2^*)$.
c) Verifies if $X_1^* = X_1$. If not equal, the session is aborted. Otherwise, it proceeds.
d) Decrypts $\tau_i$ using $X_2^*$ to recover $\mathrm{UID}_i$.
e) Searches its database for $\mathrm{UID}_i$. If found, it retrieves the associated $Q_{ij}$ and $\mathrm{Ch}_j$.
f) Queries its C-PUF with challenge $\mathrm{Ch}_j$ to get the response $\mathrm{Re}_j = C\text{-PUF}(\mathrm{Ch}_j)$.
g) Generates the server's key material $(K_j, \theta_j) = \mathrm{Fu.Gen}(\mathrm{Re}_j)$.

S4: $\mathrm{MECS}_j$ then has the following aspects.

a) Computes its primary key $K_m^j = h_1(B_j \parallel K_j)$.
b) Decrypts $Q_{ij}$ using $K_m^j$ to obtain $\{\mathrm{ID}_i, \chi_i, a_i\}$.
c) Selects a random number $y \in Z_q^*$ and computes $Y = \mathrm{yP}$.
d) Generates the session key:$\mathrm{SK} = h_1(\mathrm{ID}_i||\mathrm{MID}_j||a_i||\chi_i||yX||T_1||T_2)$
e) Computes an authentication token $\mathrm{Auth}_j = h_1(\mathrm{ID}_i \parallel \mathrm{SK} \parallel a_i \parallel R_1 \parallel T_2)$.
f) Sends the response message $\{\mathrm{Auth}_j, Y, T_2\}$ to $U_i$.

S5: Finally, $U_i$ has the following aspects.

a) Checks the freshness of timestamp $T_2$.
b) Computes the same session key SK (using its local parameters and received $Y$).

c) Computes its own authentication token $\text{Auth}_i = h_1(\text{ID}_i \| \text{SK} \| a_i \| R_1 \| T_2)$.
d) Verifies if $\text{Auth}_i = \text{Auth}_j$.
e) If the equality holds, mutual authentication is successful, and the session key SK is established for secure communication.

*Remark 3:* In our PUF3FAka protocol, $X^* = X$ because $X^* = S^{-1}(\text{HMID}_j^* + X_1R_1) = S^{-1}(\text{HMID}_j^* + X_1r_1) * P = x(\text{HMID}_j + X_1r_1)^{-1}(\text{HMID}_j + X_1r_1) * P = x * P = X$, which immediately yields $(X_1^*, X_2^*) = (X_1, X_2)$, that is, $X_1^* = X_1$ and $X_2^* = X_2$.

### E. Password Update Phase

$U_i$ inputs $\text{ID}_i, B_i^*$, the old password $\text{PW}_i$ and sends the update request to the device, the device invokes the reproduction algorithm $\text{Fu.Rep}(\cdot)$ to get $\delta_i$ and computes $D_i^* = h_1(\text{ID}_i \| \text{PW}_i \| \delta_i) \bmod n$, and then the device checks whether $D_i^*$ and $D_i$ are equal. If it is equal, the device passes the user's request. $U_i$ inputs $\text{ID}_i, B_i^*$ and the new password $\text{NPW}_i$, the device invokes the reproduction algorithm $\text{Fu.Rep}(\cdot)$ to get $\delta_i$ and computes $\text{ND}_i = h_1(\text{ID}_i \| \text{NPW}_i \| \delta_i) \bmod n$, $\text{NPID}_i = D_i \oplus \text{PID}_i \oplus \text{ND}_i$, and $\text{NL}_i = A_i \oplus h_2(\text{ID}_i \| \delta_i)$. Then, the device replaces $\{D_i, \text{PID}_i, L_i\}$ with $\{\text{ND}_i, \text{NPID}_i, \text{NL}_i\}$.

### F. Reregistration Phase

If the number of times the user cannot be authenticated by the server $\text{MECS}_j$ exceeds the maximum threshold value, then $U_i$ is required to reregister. In this case, $U_i$ will not be able to log into the system even though he/she inputs the correct values $\{\text{ID}_i, \text{PW}_i, B_i\}$. However, $U_i$ can reregister by performing the following steps:

Step 1: $U_i$ submits the reregistration request $\{\text{ID}_i, \text{Re} - reg - \text{request}\}$ to RA through a private channel.
Step 2: After receiving the request from $U_i$, the registration center RA uniquely identifies the user $U_i$ by checking her identity information. Afterward, if $\text{ID}_i$ is found in the database, RA accepts this request and performs the user registration phase to complete reregistration. Otherwise, RA rejects this request.

## V. Security Analysis

In this section, we conduct a thorough security analysis of the PUF3FAka protocol. Using both formal and informal methods, we prove that our protocol achieves the desired security properties.

### A. Formal Security Proof

*1) Security Model:* Based on the Bellare et al. [41] and Bresson et al. [42] models, we propose a three-factor authentication protocol security model for MEC, and prove the security of our proposed protocol PUF3FAka under this model.

*Participants:* The participants of the protocol include edge servers $\text{MEC}_j$ and users $U_i$, and each participant contains multiple instances $U_i^P$, $\text{MECS}_j^P$.

*Queries:* adversary interacts with the participants through the following query, thus simulating the adversary's capabilities.

- $\text{Execute}(U_i^P, \text{MECS}_j^P)$. This query simulates an eavesdropping attack, where the oracle returns a copy of the message on the open channel to the adversary.
- $\text{Send}(U_i^P/\text{MECS}_j^P, M)$. The adversary sends a message $M$ to the instance $U_i^P/\text{MECS}_j^P$, and the instance honestly returns the response of the message $M$ to the adversary according to the protocol.
- Reveal(sid). The adversary can obtain the session key of the completed session sid.
- Test(sid). The adversary can choose a fresh session sid as a test session, the oracle flips a fair coin $b$. If $b = 1$, returns the session key SK; if $b = 0$, returns a random string of the same length as the session key. This query can be run at most once.
- $\text{Guess}(b')$. After Test(sid), $\text{Guess}(b')$ is executed. The adversary inputs the guess $b'$, if $b' = b$, the oracle returns 1. Otherwise, return 0.
- $\text{Corrupt}(P, \mu)$. This oracle reveals some authentication messages of the user or server.
  a) When $P = U_i^P$, the adversary $\mathcal{A}$ can break any two of the three factors.
     - If $\mu = 0$, oracle returns the parameters in the device.
     - If $\mu = 1$, oracle returns the user's password.
     - If $\mu = 2$, oracle returns the user's biometric.
  b) When $P = \text{MECS}_j^P$, the adversary $\mathcal{A}$ can break the data $\{\text{ID}_j, B_j, \text{Ch}_j\}$ in database and verifier table $\{\text{MID}_j, Q_{ij}\}$.
- $\text{Hash}(M)$: This query allows the adversary to obtain hash values as if accessing a random oracle. If the corresponding record already exists in the $H$-table, the same hash value is returned; otherwise, a fresh random value is generated and returned, and the new entry is immediately written into the $H$-table.

*Partnering:* Two oracle instances $U_i^P$ and $\text{MECS}_j^P$ are said to be *partnered* if and only if the following three conditions hold simultaneously.

1) Both instances have reached the accepted state.
2) Their session identifiers coincide: $\text{sid}_{U_i^P} = \text{sid}_{\text{MECS}_j^P}$.
3) Each instance names the other as its intended partner: $\text{pid}_{U_i^P} = \text{MECS}_j^P$ and $\text{pid}_{\text{MECS}_j^P} = U_i^P$.

*Freshness:* A session is fresh if the following three conditions are met.

- $U_i^P$ and $\text{MECS}_j^P$ in the session have run the protocol to negotiate a session key SK.
- The adversary does not execute Reveal(sid) query and $\text{Corrupt}(\text{MECS}_j^P)$ query on the session and its corresponding instance.

*Semantic Security:* Under the security model, a protocol $\Pi$ is said to be secure for any probabilistic polynomial-time adversary, if the advantage $\text{Adv}_\Pi^{\text{aka}}$ of the adversary attacking the protocol is negligible, where $\text{Adv}_\Pi^{\text{aka}} = |\Pr[b' = b] - (1/2)|$.

*2) Security Proof:*

*Theorem 1:* If the password space $\mathcal{D}_{\text{PW}}$ obeys the Zipf distribution. Assuming that the adversary $\mathcal{A}$ of PUF3FAka protocol can make $q_s$ send queries, $q_h$ hash queries, $q_e$ execute

queries, $q_f$ probabilistic generation function queries of fuzzy extractor, and $q_{\text{ed}}$ symmetric encryption/decryption queries in polynomial time $t$, then we have

$$\text{Adv}^{\text{aka}}_{\text{PUF3FAka}}(t) \le \frac{q_h^2}{2^{l_h+1}} + \frac{q_f^2}{2^{l_f+1}} + \frac{q_{\text{ed}}^2}{2^{l_{\text{ed}}+1}} + \frac{(q_e+q_s)^2}{2q} + \frac{q_s^2}{2^{l_h-1}} + C' q_s^{s'} + q_h \text{Adv}^{\text{ECDHP}}(t')$$

where $C' = 0.062239$ and $s' = 0.155478$ are Zipf parameters [40], $l_h$, $l_f$, and $l_{\text{ed}}$ are the output length of hash function, the fuzzy extractor's probabilistic generation function and symmetric encryption/decryption, $\text{Adv}^{\text{ECDHP}}(t')$ is the advantage for adversary $\mathcal{A}$ to compute abP within polynomial time $t'$, and $t' \le t + (q_e + q_s + 1)T_{\text{sm}}$, $T_{\text{sm}}$ is time for scalar multiplication operation in $G$.

*Proof:* To prove Theorem 1, we define a sequence of games $\Psi_t, 0 \le t \le 5$. $\Pr[\Psi_t]$ represents the probability that $\mathcal{A}$ successfully guesses the value of $b$ in the game $\Psi_t$.

*Game* $\Psi_0$*:* The game simulates an adversary's attack on a real protocol, so we have

$$\text{Adv}^{\text{ake}}_{\Pi} = \left| \Pr[\Psi_0] - \frac{1}{2} \right|. \quad (1)$$

*Game* $\Psi_1$*:* The game simulates passive attacks by executing Execute queries to eavesdrop on an open channel, intercepting all exchanged information during login and authentication, and then executing test queries to confirm whether the output is a real session key or a random number. Even if the adversary $\mathcal{A}$ obtains the publicly passed messages $\{\tau_i, (X_1, S), T_1, R_1, \text{Auth}_j, Y, T_2\}$, the important parameters $\{\text{ID}_i, \chi_i, a_i\}$ to distinguish and calculate the session key cannot be obtained due to the protection provided by PUF, so the probability of winning games 1 and 0 remains the same, then we have

$$\Pr[\Psi_0] = \Pr[\Psi_1]. \quad (2)$$

*Game* $\Psi_2$*:* The game stimulates collisions by asking hash query, probabilistic generation function queries of the fuzzy extractor, and encryption/decryption queries. According to the birthday paradox:

1) the probability of collisions caused by hash queries is $(q_h^2/2^{l_h+1})$;
2) the probability of collisions caused by encryption/decryption queries is $(q_{\text{ed}}^2/2^{l_{\text{ed}}+1})$;
3) the probability of collisions caused by probabilistic generation function queries of fuzzy extractor is $(q_f^2/2^{l_f+1})$; and
4) the probability of collisions of $x, y$ is $((q_e+q_s)^2/2q)$.

Thus, we have

$$|\Pr[\Psi_2] - \Pr[\Psi_1]| \le \frac{q_h^2}{2^{l_h+1}} + \frac{q_f^2}{2^{l_f+1}} + \frac{q_{\text{ed}}^2}{2^{l_{\text{ed}}+1}} + \frac{(q_e+q_s)^2}{2q}. \quad (3)$$

*Game* $\Psi_3$*:* In this game, we consider such an attack which $\mathcal{A}$ luckily and correctly guesses the following messages without querying the hash oracle.

1) For $M_1$, the simulator checks if $((\text{ID}_j \| K_j) \| T_1)$ is present in the hash list, and the probability is $(q_s^2/2^{l_h})$.
2) For $M_2$, the simulator checks if $(\text{ID}_j \| \text{MID}_j \| a_i \| \chi_i \| xY \| T_1 \| T_2)$ is present in the hash list and the probability is $(q_s^2/2^{l_h})$. Therefore, we can have

$$Pr[\Psi_2] \le \frac{q_s^2}{2^{l_h-1}}. \quad (4)$$

*Game* $\Psi_4$*:* In this game, $\mathcal{A}$ asks Corrupt$(U_i^P, \mu)$ query. If $\mu = 0, 1$, $\mathcal{A}$ can obtain the parameters $\{\theta_i, D_i, Li, \text{PID}_i, \text{MID}_j\}$ in the device and the password $\text{PW}_i$, but $\mathcal{A}$ can only obtain multiple guessed values from $D_i$, but cannot determine the correct one. So $\mathcal{A}$ guesses the user's biometrics with probability $(q_s/2^{l_f})$. If $\mu = 0, 2$, $\mathcal{A}$ can obtain the parameters $\{\theta_i, D_i, Li, \text{PID}_i, \text{MID}_j\}$ in the device and biometric $B_i$, $\mathcal{A}$ cannot determine the correct $D_i$, thus $\mathcal{A}$ guesses the user's password with probability $C' q_s^{s'}$. If $\mu = 1, 2$, $\mathcal{A}$ can obtain these parameters biometric $B_i$ and password $\text{PW}_i$, $\mathcal{A}$ guesses an important parameter $D_i$ with probability $(q_s/2^{l_h})$. we have

$$|\Pr[\Psi_4] - \Pr[\Psi_3]| \le \text{MAX}\left\{ \frac{q_s^2}{2^{l_s}}, \frac{q_s^2}{2^{l_f}}, C' q_s^{s'} \right\} = C' q_s^{s'}. \quad (5)$$

*Game* $\Psi_5$*:* The adversary asks Corrupt$(\text{MECS}_j^P)$, test queries, and guess queries, so the probability of winning this game is $\Pr[\Psi_4] = (1/2)$. And if the adversary asks a hash query for $h_1(\text{ID}_i \| \text{MID}_j \| A_i \| \chi_i \| yX \| T_1 \| T_2)$, to calculate the session key, $\mathcal{A}$ faces the ECDHP assumption as described in Section II. Therefore, we have

$$|\Pr[\Psi_5] - \Pr[\Psi_4]| \le q_h \text{Adv}^{\text{ECDHP}}(t') \quad (6)$$

where $t' \le t + (q_e + q_s + 1)T_{\text{sm}}$, $T_{\text{sm}}$ is the time for the scalar multiplication operation in $G$.

Based on the above five inequalities, we finally have $\text{Adv}^{\text{aka}}_{\text{PUF3 FAka}}(t) \le (q_h^2/2^{l_h+1}) + (q_f^2/2^{l_f+1}) + (q_{\text{ed}}^2/2^{l_{\text{ed}}+1}) + ((q_e+q_s)^2/2q) + (q_s^2/2^{l_h-1}) + C' q_s^{s'} + q_h \text{Adv}^{\text{ECDHP}}(t')$. ∎

### B. Security Analysis Based on BAN Logic

This section describes the logical analysis of the proposed protocol PUF3FAka by using BAN logic, which was defined and presented by [43]. The following symbols and rules (R1–R5) govern our analytical framework.

1) $P \triangleleft X$: Participant $P$ sees statement $X$.
2) $P| \equiv X$: $P$ believes $X$.
3) $\#(X)$:The statement $X$ is fresh.
4) $P| \sim X$: Participant $P$ express statement $X$.
5) $P_1 \overset{K}{\longleftrightarrow} P_2$: Participants $P_1$ and $P_2$ share the same key $K$.
6) $P \Longrightarrow X$: Participant $P$ controls statement $X$.
7) $(X)_K$: Use $K$ to perform a cryptographic operation on $X$.

R1: $(P_1| \equiv P_1 \overset{K}{\longleftrightarrow} P_2, P_1 \triangleleft (X)_K / P_1| \equiv P_2 \sim X)$, if $P_1$ trusts that the key $K$ is shared with $P_2$, $P_1$ sees $X$ combined with $K$, then $P_1$ trusts $P_2$ once expressed $X$.

R2: $(P_1| \equiv \#(X), P_1| \equiv P_2 \sim X / P_1| \equiv P_2 \equiv X)$, if $P_1$ trusts the freshness of $X$ and $P_1$ trusts $P_2$ once expressed $X$, then $P_1$ trusts that $P_2$ trusts $X$.

R3: $(P| \equiv X_1, P| \equiv X_2 / P| \equiv (X_1, X_2))$, if $P$ trusts $X_1$ and $X_2$, then $P$ also trusts $(X_1, X_2)$.

R4: $(P| \equiv \#(X_1)/P| \equiv \#(X_1, X_2))$, if the freshness of $X_1$ is trusted by P, then P can trust the freshness of the statement $(X_1, X_2)$.

R5: $(P_1| \equiv P_2 \Longrightarrow X, P_1| \equiv P_2 \equiv X/P_1| \equiv X)$, if $P_1$ trusts that $P_2$ has control on X and $P_1$ trusts that $P_2$ trusts statement $X$, then $P_1$ also trusts $X$.

Through the analysis, we need to achieve the following four goals.

G1: $U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$.

G2: $\text{MECS}_j| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$.

G3: $U_i| \equiv \text{MECS}_j| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$.

G4: $\text{MECS}_j| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$.

In the PUF3FAka protocol, all messages transmitted through public channels can be converted into an idealized form as follows.

1) $M_1 = \{\tau_i, (X_1, S), T_1, R_1\}$, $\tau_i = E_{X_2}(r_1, r_1 \oplus \text{MID}_i)$, $S = x^{-1}(\text{HMID}_j + r_1 X_1) \bmod q$, $R_1 = r_1 P$. This is reduced to the following: $\text{Msg}_1 = \{\text{ID}_i, \text{ID}_j, A_i, x, T_1\}$.
2) $M_2 = \{\text{Auth}_j, Y, T_2\}$, $\text{Auth}_j = h_1(\text{ID}_i \parallel \text{SK} \parallel A_i \parallel R_1 \parallel T_2)$, $Y = \text{yP}$. This is reduced to the following: $\text{Msg}_2 = \{\text{ID}_i, \text{ID}_j, A_i, \chi_i, y, T_1, T_2\}$.

In addition, we define the following assumptions to demonstrate the freshness of the session key in the PUF3FAka protocol.

A1: $U_i| \equiv \#(T_1)$.

A2: $\text{MECS}_j| \equiv \#(T_2)$.

A3: $U_i| \equiv \#(x)$.

A4: $\text{MECS}_j| \equiv \#(y)$.

A5: $U_i| \equiv (U_i \overset{X}{\longleftrightarrow} \text{MECS}_j)$.

A6: $\text{MECS}_j| \equiv (U_i \overset{X}{\longleftrightarrow} \text{MECS}_j)$.

A7: $U_i| \equiv (U_i \overset{Y}{\longleftrightarrow} \text{MECS}_j)$.

A8: $\text{MECS}_j| \equiv (U_i \overset{Y}{\longleftrightarrow} \text{MECS}_j)$.

A9: $U_i| \equiv \text{MECS}_j \Longrightarrow (U_i \overset{\text{SK}}{\longleftrightarrow} \text{MECS}_j)$.

A10: $\text{MECS}_j| \equiv U_i \Longrightarrow (U_i \overset{\text{SK}}{\longleftrightarrow} \text{MECS}_j)$.

We then demonstrate our main proof using the above rules and assumptions.

1) From $\text{Msg}_1$, we obtain $V_1$: $\text{MECS}_j \triangleleft (\text{ID}_i, \text{ID}_j, A_i, x, T_1)_X$.
2) Based on A5 and Rule1, we derive $V_2$ : $\text{MECS}_j| \equiv U_i| \sim (\text{ID}_i, \text{ID}_j, A_i, x, T_1)_X$.
3) Based on A1 and Rule4, we derive $V_3$ as follows: $\text{MECS}_j| \equiv \#((\text{ID}_i, \text{ID}_j, A_i, x, T_1)_X)$.
4) Based on $V_2, V_3$ and Rule2, we derive $V_4$: $\text{MECS}_j| \equiv U_i| \equiv (\text{ID}_i, \text{ID}_j, A_i, x, T_1)_X$.
5) From $\text{Msg}_2$, we derive $V_5$: $U_i \triangleleft (\text{ID}_i, \text{ID}_j, A_i, \chi_i, y, T_1, T_2)_Y$.
6) Based on A8 and Rule1, we derive the following $V_6$ : $U_i| \equiv \text{MECS}_j| \sim (\text{ID}_i, \text{ID}_j, A_i, \chi_i, y, T_1, T_2)_Y$.
7) Based on A4 and Rule4, we derive $V_7$ as follows: $U_i| \equiv \#((\text{ID}_i, \text{ID}_j, A_i, \chi_i, y, T_1, T_2)_Y)$.
8) Based on $V_4, V_5$ and Rule2, we derive $V_8$: $U_i| \equiv \text{MECS}_j| \equiv (\text{ID}_i, \text{ID}_j, A_i, \chi_i, y, T_1, T_2)_Y$.
9) Based on $V_8$ and $\text{SK} = h_1(\text{ID}_i \parallel h_1(\text{ID}_j) \parallel A_i \parallel \chi_i \parallel yX \parallel T_1 \parallel T_2)$, we derive $V_9$ as follows: $U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$(G1).
10) Based on $V_6$ and $\text{SK} = h_1(\text{ID}_i \parallel h_1(\text{ID}_j) \parallel A_i \parallel \chi_i \parallel yX \parallel T_1 \parallel T_2)$, we derive $V_{10}$: $\text{MECS}_j| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$(G2).
11) Based on A9, $V_8$ and Rule5, we derive $V_{11}$ as follows: $U_i| \equiv \text{MECS}_j| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$ (G3).
12) Based on A10, $V_9$ and Rule5, we derive $V_{12}$ as follows: $\text{MECS}_j| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} \text{MECS}_j)$ (G4).

Therefore, $U_i$ and $\text{MECS}_j$ achieve mutual authentication and share the session key SK securely by satisfying $G1, G2, G3$, and $G4$.

### C. Heuristic Analysis

In this section, we provide a comprehensive assessment of the security of the proposed protocol, PUF3FAka. This analysis is based on the evaluation criteria and threat model outlined in Section II. The threat model details the capabilities and limitations of potential adversaries. By dissecting the nature of these threats, we can better assess a protocol's resilience against risks and ensure its security attributes are robustly maintained. Overall, the goal of this evaluation is to identify any potential flaws or areas for improvement in the protocol design, ensuring it meets the necessary requirements for secure communication in the target application scenario.

*1) Resist Offline Password Guessing Attacks:* It is assumed that the adversary $\mathcal{A}$ can obtain the parameter information $\{\theta_i, D_i, Li, \text{MID}_j\}$ in the intelligent device through the channel test attack, and obtain the biometric characteristics $B_i$; the adversary can invoke fuzzy extractor to get $\delta_i$, if it's assumed that the user's identity and password are both 20-bit when $|\mathcal{D}_{\text{ID}}| = |\mathcal{D}_{\text{PW}}| = 10^6$, one can be assured that there exists $(2^{20} \times 2^{20}/2^8)$ candidates of pair to frustrate which satisfy $D_i^* = D_i$, where $2^4 \leq n \leq 2^8$. There is no other way than launching an online password guessing attack to determine the exactly correct one, thus our proposed PUF3FAka protocol can resist offline password guessing attacks.

*2) Resist Replay Attacks:* In the authentication phase, the messages passed on the channel $M_1 : \{\tau_i, (X_1, S), T_1, R_1\}$ and $M_2 : \{\text{Auth}_j, Y, T_2\}$ use timestamps, so the adversary cannot replay previous messages to obtain authentication.

*3) Resist Impersonation Attacks:* In our proposed protocol PUF3FAka, we use the PUF to protect the server's key $K_m^j$ from being compromised due to the unclonable and tamper-resistant properties of PUF. Therefore, the adversary cannot impersonate the server to send the message $M_2 : \{\text{Auth}_j, Y, T_2\}$ to the user, because the message is generated by decrypting the parameters obtained from $Q_i$ with the key $K_m^j$. At the same time, the adversary cannot impersonate the user, because the message $M_1 : \{\tau_i, (X_1, S), T_1, R_1\}$ needs the legitimate user's identity, password and biometrics, and the adversary does not know the legitimate user's random number.

*4) Resist Ephemeral-Secret Leakage Attacks:* Suppose the temporary parameters $x, r_1, y$ are leaked, the adversary $\mathcal{A}$ only can compute $X = xP = (X_1, X_2)$ and $D_{X_2}(\tau_i) = (r_1, r_1 \oplus \text{MID}_i)$, then the adversary gets $\text{MID}_i$. However, the adversary $\mathcal{A}$ cannot get $\text{ID}_i, A_i$ and $\chi_i$, thus the adversary cannot compute the session key $\text{SK} = h_1(\text{ID}_i \parallel \text{MID}_j \parallel A_i \parallel \chi_i \parallel yX \parallel T_1 \parallel T_2)$.

TABLE IV
AVERAGE TIME EACH CRYPTOGRAPHIC OPERATION USING 1000 RUNS [44], [45]

| Cryptographic Primitives | Symbol | Execution Time(ms) |
|---|---|---|
| Hash Function | $T_h$ | 0.0507 |
| Symmetric Enc/Dec | $T_s$ | 0.22 |
| Bilinear Pairing Operation | $T_b$ | 12.45 |
| Scalar Multiplication on $G$ | $T_{sm}$ | 0.37 |
| Point Addition on $G$ | $T_{pa}$ | 0.145 |
| Modular Exponential Operation | $T_e$ | 0.87 |
| Fuzzy Extractor Operation | $T_{fe}$ | 0.0072 |

*5) Puf Response Friendly:* We do not explicitly store challenge response pairs on the edge server, which avoids the insecure session key caused by the disclosure of CRPS. At the same time, we equip the server with CRO-PUF that can resist machine modeling attacks and power differential attacks, which further guarantees the security of CPRS, so our protocol is friendly to PUF's response.

*6) Perfect Forward Secrecy:* In our proposed protocol PUF3FAka, we use the PUF to protect the server's key $K_m^j$. The adversary cannot fake the same PUF under to get the correct response $\mathrm{Re}_j$ and get the correct secret $K_j$. Therefore, the adversary cannot compute the decryption key $K_m^j$ to obtain important information for computing the session key. And the adversary does not know $x, y$. Thus, the proposed protocol can provide perfect forward secrecy.

*7) Resist KCI Attacks:* In our proposed protocol PUF3FAka, We used two secrets to protect the decryption key $K_m^j$, which are used by the server. The adversary cannot fake the same PUF under to get the correct response $\mathrm{Re}_j$ and get the correct secret $K_j$. So the adversary cannot use the key $K_m^j$ to decrypt $Q_{ij}$ and obtain some user information to impersonate the user.

*8) Mutual Authentication:* In the proposed protocol PUF3FAka, $\mathrm{MECS}_j$ authenticates $U_i$ by verifying whether $X_1^* = X_1$, while $U_i$ authenticates $\mathrm{MECS}_j$ by checking whether $\mathrm{Auth}_i = \mathrm{Auth}_j$. After mutual authentication, $U_i$ and $\mathrm{MECS}_j$ negotiate a common session key SK. Thus, the proposed protocol PUF3FAka achieves secure mutual authentication.

*9) Anonymity and Untraceability:* In the proposed protocol PUF3FAka, the adversary $\mathcal{A}$ may attempt to determine the user's identity $\mathrm{ID}_i$ by intercepting the message $M_1$ and $M_2$ transmitted through public channels. However, $\mathrm{ID}_i$ is not transmitted in plaintext in message $M_1$ and $M_1$, and $\mathrm{ID}_i$ is protected by the server's secret key $K_m^j$, thus, our protocol PUF3FAka satisfies user anonymity.

*10) Resist MLMo Attacks:* The edge server in our proposed design incorporates a CRO-PUF, which demonstrates significant resilience against MLMo attacks. This claim is substantiated by the detailed analysis in Section II-C, Remark 1: the intrinsic optimal key selection technique of the CRO-PUF ensures that the generated cryptographic keys possess high randomness and unpredictability, thereby fundamentally mitigating the risk of being modeled by adversaries using machine learning algorithms. Consequently, within the defined threat model and an acceptable risk threshold of the current system, this design provides reliable authentication security guarantees for the edge server.

## VI. COMPARATIVE SUMMARY: SECURITY AND PERFORMANCE

To demonstrate the good balance relations of the proposed protocol PUF3FAka in security and usability, this section provides a comparative measurement of the security, computation and communication cost of [4], [5], [9], [10], [12], [14], [15], and the proposed protocol. To analyze the computational complexity of the login and authentication step, the lightweight operations, including exclusive-OR and string concatenation operations, are omitted. But the following additional notations are used, i.e., $T_{\mathrm{pa}}$ is the computational time required for executing an elliptic curve point addition, $T_s$ is the computational time required for symmetric cryptography operation, $T_h$ is the computational time required for operating a one-way hash operation, $T_b$ is the computational time required for operating a bilinear pairing operation, $T_{\mathrm{sm}}$ is the computational time required for executing a elliptic curve scalar multiplication, and $T_e$ is the computational time required for executing a modular exponentiation. The time cost of each operation required for the login and authentication phase protocol studied is shown in Table IV. For measuring the average execution time, we adopted the hardware benchmark used by Seifelnasr et al. [44]: a Raspberry Pi 4 Model B (8-GB RAM) equipped with a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor and running the 64-bit Raspbian operating system. On this platform, we replicated and extended their software stack—utilizing bplib 0.0.6 for bilinear pairing operations, fastecdsa for elliptic curve point operations, and Python's standard crypto and socket modules for hash functions, encryption/decryption, and network simulation. Each cryptographic primitive was executed 1000 times consecutively, and the average value was taken as the baseline for all computational cost evaluations and comparative experiments in this study. And $T_{\mathrm{fe}}$ is the computational time required for executing a fuzzy extractor generation function or reproducible function, which is derived from [45].

*Remark 4:* As reported in [38], the CRO-PUF exhibits an average response generation of 0.7 ns, which is on the order of $10^{-7}$ ms. Since this response speed far surpasses the benchmark for comparison, the access time metric was excluded from the comprehensive evaluation in Table V to better highlight the differences in other key performance indicators.

Table V presents the performance of each protocol under a series of evaluation criteria (EC1 to EC10), as well as their computational costs on the user and server-side and communication costs. Tsai and Lo [4] performed well in most evaluation criteria, but failed to meet the requirements in EC6 and EC10. The computational cost and communication cost were relatively high. Irshad et al. [5] also performed well in most evaluation criteria, but failed to meet the requirements in EC4 and EC10. The computational cost and the communication cost were the highest. Jia et al. [9] met the requirements of all evaluation criteria, except for EC6 and

TABLE V
SECURITY COMPARISON AMONG RELEVANT PROTOCOLS

| Protocols | Evaluation Criteria | | | | | | | | | | Computational Cost | | | CC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $EC_1$ | $EC_2$ | $EC_3$ | $EC_4$ | $EC_5$ | $EC_6$ | $EC_7$ | $EC_8$ | $EC_9$ | $EC_{10}$ | User | Server | Total(ms) | Bits |
| Tsai et al. [4] | √ | √ | √ | √ | √ | × | √ | √ | √ | × | $5T_h + 3T_{sm} + T_b + 2T_{pa}$ | $4T_h + T_{sm} + 3T_b + 2T_{pa}$ | 52.3163 | 4768 |
| Irshad et al. [5] | √ | √ | √ | × | √ | √ | √ | √ | √ | √ | $8T_h + 3T_{sm} + 2T_{pa} + 3T_b$ | $5T_h + 4T_{sm} + 3T_b + 3T_{pa}$ | 66.2241 | 5632 |
| Jia et al. [9] | √ | √ | √ | √ | × | √ | √ | √ | × | √ | $5T_h + 3T_{sm} + T_{pa}$ | $5T_h + 5T_{sm} + 3T_{pa} + T_b$ | 16.497 | 4832 |
| Mishra et al. [10] | √ | √ | √ | √ | √ | √ | √ | √ | √ | × | $8T_h + 2T_{sm} + T_b + T_{pa}$ | $7T_h + 3T_{sm} + T_b + 2T_{pa}$ | 27.9962 | 3392 |
| Jia et al. [12] | √ | √ | √ | × | × | √ | × | √ | √ | √ | $8T_h + 5T_{sm} + T_{pa}$ | $5T_h + 6T_{sm} + T_b + 2T_{pa}$ | 17.6648 | 4832 |
| Lee et al. [14] | √ | √ | √ | × | × | √ | × | × | √ | √ | $9T_h + T_b + 4T_{sm} + T_s + T_{pa}$ | $5T_h + T_b + 3T_{sm} + T_s + 2T_{pa}$ | 29.0748 | 2656 |
| Tian et al. [15] | √ | √ | √ | × | × | √ | × | √ | √ | × | $8T_h + 4T_{sm} + T_{pa}$ | $5T_h + 4T_{sm} + T_b + 2T_{pa}$ | 16.5041 | 4832 |
| PUF3FAka | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | $6T_h + 4T_{sm} + 2T_e + T_s + T_{fe}$ | $2T_h + 3T_{sm} + 2T_s + T_{pa} + T_{fe}$ | 5.555 | 3520 |

√ means the property is satisfied; × means the property is not satisfied; − means the property is not considered. CC denotes communication cost. $T_h, T_s, T_b, T_{sm}, T_{pa}, T_e$ denote the operation time for hash, symmetric encryption, bilinear pairing, scalar multiplication on $G$, point addition on $G$ and modular exponentiation respectively. The length of the output of hash Functions (SHA) is 256 bits. The length of the elements of $G$ and $|G_T|$ is 1024 bits. The length of the timestamp is 32 bits. The length of the elements of $Z_q$ 160 bits. The length of the output of Symmetric Enc/Dec (AES) is 128 bits.

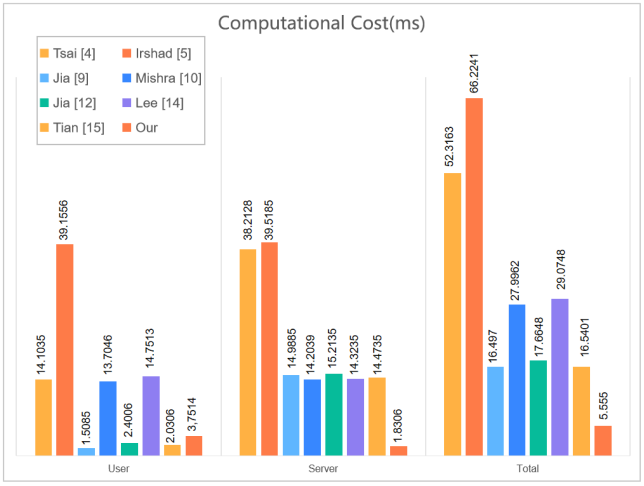


Fig. 5. Computation cost comparison.

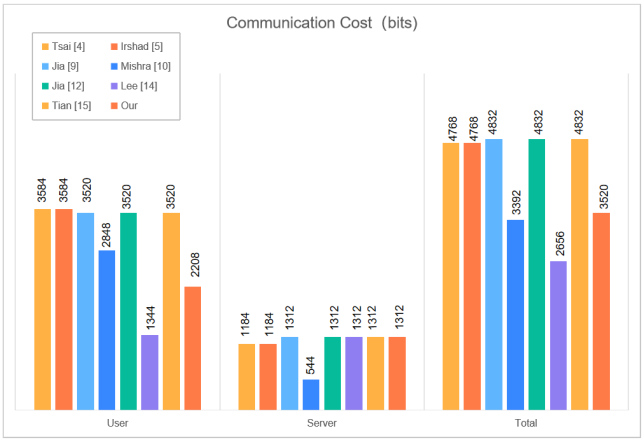


Fig. 6. Communication cost comparison.

EC10. The computational cost and communication cost were moderate. Mishra et al. [10] met the requirements in all evaluation criteria, except for EC10. The computational cost and communication cost were relatively low. Jia et al. [12] and Lee et al. [14] failed to meet the requirements under most evaluation criteria, with relatively low computational cost and communication cost. Tian et al. [15] failed to meet the requirements of most evaluation criteria, but the computational cost and the communication cost were very low. In contrast, our protocol does not use the bilinear pairing complex operation commonly used in previous protocols, also reduces the use of scalar multiplication on G, and equips the server with a very low-cost PUF instead of IBE to protect the security of the server, which makes our protocol perform well in terms of computational cost and communication cost. This can also be intuitively perceived from the comparison of the diagrams in Figs. 5 and 6. Moreover, through calculation, the average increase is 82.86% and 96.7%, showing high efficiency and practicability.

## VII. CONCLUSION

In this article, we design an AKA protocol by adopting the techniques of fuzzy-verifiers and PUF, and then prove its security from formal and evaluation criteria-based security analysis. The results of the security analysis indicate that the proposed protocol PUF3FAka is capable of achieving semantic security and meets the ten security evaluation criteria ($EC_1$–$EC_{10}$). By comparing the security, computation, and communication costs of our protocol with those of the state-of-the-art protocols, we show that our new AKA protocol is more practical for the edge computing environment. Our future work is to combine the design idea of the proposed protocol PUF3FAka with some techniques of postquantum cryptography to design a postquantum authenticated key agreement protocol for the MEC environment.

## REFERENCES

[1] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.

[2] T. X. Tran, M.-P. Hosseini, and D. Pompili, "MEC: Recent efforts and five key research directions," *IEEE COMSOC MMTC Commun.-Front.*, vol. 12, no. 4, pp. 29–33, Apr. 2017.

[3] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[4] J. L. Tsai and N. W. Lo, "A privacy-aware authentication protocol for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 9, no. 3, pp. 805–815, Mar. 2015.

[5] A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, S. A. Chaudhry, and R. Kumar, "An improved multi-server authentication protocol for distributed mobile cloud computing services," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 12, pp. 5529–5552, 2016.

[6] X. Yang, X. Huang, and J. K. Liu, "Efficient handover authentication with user anonymity and untraceability for mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 62, pp. 190–195, Sep. 2016.

[7] Y.-Y. Deng, C.-L. Chen, J. Shin, and K.-H. Wang, "Cryptanalysis of Yang et al.'s handover authentication scheme for mobile network environment," in *Proc. Int. Symp. Comput. Sci. Intell. Controls (ISCSIC)*, Oct. 2017, pp. 152–157.

[8] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy aware authentication protocol for mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1621–1631, Feb. 2018.

[9] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication protocol for MEC," *IEEE Syst. J.*, vol. 14, no. 1, pp. 560–571, Jan. 2020.

[10] D. Mishra, D. Dharminder, P. Yadav, Y. S. Rao, P. Vijayakumar, and N. Kumar, "A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102648.

[11] Y. Li, Q. Cheng, X. Liu, and X. Li, "A secure anonymous identity-based protocol in new authentication architecture for MEC," *IEEE Syst. J.*, vol. 15, no. 1, pp. 935–946, Jan. 2021.

[12] X. Jia, M. Luo, K.-K.-R. Choo, L. Li, and D. He, "A redesigned identity-based anonymous authentication scheme for mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10108–10120, Jun. 2022.

[13] Q. Ma, M. Deng, Q. Song, and C. Zhang, "A novel pairing-free authentication scheme with anonymity for mobile infrastructure," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 53308–53320, Dec. 2025, doi: 10.1109/JIOT.2025.3616304.

[14] H. Lee, J. Ryu, and D. Won, "Secure and anonymous authentication protocol for MEC environments," *IEEE Internet Things J.*, vol. 14, no. 8, pp. 5798–5815, Aug. 2023.

[15] J. Tian, Y. Wang, and Y. Shen, "An identity-based authentication protocol with full anonymity and unlinkability for MEC," *IEEE Internet Things J.*, vol. 11, no. 13, pp. 23561–23576, Jan. 2024.

[16] S. Yoon et al., "Puf-based authentication protocol for IoT devices," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, New York, NY, USA, Feb. 2020, pp. 1792–1794.

[17] B. K. Ghafi and B. M.-N. Maybodi, "A distributed PUF-based mutual authentication system with self-correction," in *Proc. 28th Iranian Conf. Electr. Eng. (ICEE)*, New York, NY, USA, Aug. 2020, pp. 1–5.

[18] M. A. Qureshi and A. Munir, "PUF-RAKE: A PUF-based robust and lightweight authentication and key establishment protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2457–2475, Jul. 2022.

[19] Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng, "A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13958–13974, Sep. 2021.

[20] Y. Zheng and C.-H. Chang, "Secure mutual authentication and key-exchange protocol between PUF-embedded IoT endpoints," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[21] Y. Zheng, W. Liu, C. Gu, and C.-H. Chang, "PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 3299–3316, Jul. 2023.

[22] Q. Zhang, J. Wu, H. Zhong, D. He, and J. Cui, "Efficient anonymous authentication based on physically unclonable function in industrial Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 233–247, 2023.

[23] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 9–14.

[24] C. Tian, J. Ma, T. Li, J. Zhang, C. Ma, and N. Xi, "Provably and physically secure UAV-assisted authentication protocol for IoT devices in unattended settings," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 4448–4463, 2024.

[25] S. Kalam and A. K. Keshri, "Advancing IoMT security: A two-factor authentication model employing PUF and fuzzy logic techniques," *Comput. Secur.*, vol. 148, Jan. 2025, Art. no. 104138.

[26] M. Tanveer and S. A. Aldossari, "RAM-MEN: Robust authentication mechanism for IoT-enabled edge networks," *Alexandria Eng. J.*, vol. 112, pp. 436–447, Jan. 2025.

[27] K. Mahmood, M. F. Ayub, S. Z. Hassan, Z. Ghaffar, Z. Lv, and S. A. Chaudhry, "A seamless anonymous authentication protocol for mobile edge computing infrastructure," *Comput. Commun.*, vol. 186, pp. 12–21, Mar. 2022, doi: 10.1016/j.comcom.2022.01.005.

[28] K. Mahmood, S. Shamshad, M. F. Ayub, Z. Ghaffar, M. K. Khan, and A. K. Das, "Design of provably secure authentication protocol for edge-centric maritime transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14536–14545, Dec. 2023, doi: 10.1109/TITS.2023.3295818.

[29] Z. Ghaffar, S. Shamshad, K. Mahmood, M. S. Obaidat, S. Kumari, and M. K. Khan, "A lightweight and efficient remote data authentication protocol over cloud storage environment," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 1, pp. 103–112, Jan. 2023, doi: 10.1109/TNSE.2022.3205443.

[30] M. A. Akram, Z. Ghaffar, K. Mahmood, S. Kumari, K. Agarwal, and C.-M. Chen, "An anonymous authenticated key-agreement scheme for multi-server infrastructure," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 22, Dec. 2020, doi: 10.1186/s13673-020-00227-9.

[31] M. A. Saleem, Z. Ghaffar, K. Mahmood, A. K. Das, J. J. P. C. Rodrigues, and M. K. Khan, "Provably secure authentication protocol for mobile clients in IoT environment using puncturable pseudorandom function," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16613–16622, Nov. 2021, doi: 10.1109/JIOT.2021.3075158.

[32] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, pp. 203–209, Jan. 1987.

[33] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002, doi: 10.1126/science.1074376.

[34] Y. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electron.*, vol. 3, no. 2, pp. 81–91, Feb. 2020, doi: 10.1038/s41928-020-0372-5.

[35] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," *Provable Secur. Comput. Sci.*, vol. 4784, pp. 1–16, May 2007.

[36] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 2045, B. Pfitzmann, Ed., 2001, pp. 453–474.

[37] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Feb. 1983.

[38] A. Miah and F. S. Hossain, "CRO-PUF: Resilience to machine learning and differential power attacks," *Comput. Secur.*, vol. 151, Apr. 2025, Art. no. 104313.

[39] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.

[40] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 708–722, Jul. 2018.

[41] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," *Comput. Sci.*, vol. 1807, pp. 139–155, Apr. 2000.

[42] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proc. 10th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2003, pp. 241–250.

[43] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *Proc. Roy. Soc. London A*, vol. 426, pp. 233–271, Jan. 1989.

[44] M. Seifelnasr, R. AlTawy, A. Youssef, and E. Ghadafi, "Privacy-preserving mutual authentication protocol with forward secrecy for IoT–edge–cloud," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8105–8117, Mar. 2024.

[45] Z. Ghaffar et al., "A lightweight and robust access control protocol for IoT-based e-healthcare network," *IEEE Trans. Mobile Comput.*, vol. 24, no. 9, pp. 9080–9091, Sep. 2025.

**Shuming Qiu** received the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2019.

He is currently a Lecturer with Jiangxi Normal University, Nanchang, China. His research interests include authentication protocols, lattice-based cryptography, noncommunicative cryptography, and privacy in the Internet of Things.

Dr. Qiu won the 2018 China Internet Development Foundation Network Security Scholarship. He is a reviewer of some journals, such as IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, SCN, WCMC, and IJCS.

**Yu Liang** received the bachelor's degree in mathematics and applied mathematics from Jiangxi Normal University, Nanchang, China, in 2021, where she is currently pursuing the primary's degree in applied cryptography with the School of Mathematics and Statistics.

Her main research interests include password-based authentication, cryptographic protocols, and provable security.

**Peizhen Hong** received the Ph.D. degree in microelectronics and solid-state electronics from the University of Chinese Academy of Sciences, Beijing, China, in 2020.

Currently, she is an Associate Professor with Nankai University, Tianjin, China. As the first author (or corresponding author), she has published more than ten papers. Her research interests include hardware encryption and nanofabrication processes.

Dr. Hong is serving as a Council Member of the Collaborative Innovation Platform for Industry-Academia Research in Electronic Materials and Components in China, a member of the Professional Committee on Key Materials and Technologies for Electronic Components of the China Society for Instrument Functional Materials, and a Young Editorial Board Member of the *Journal of Advanced Dielectrics*.

**Ding Wang** received the Ph.D. degree in information security from Peking University, Beijing, China, in 2017.

He was supported by the "Boya Postdoctoral Fellowship" in Peking University from 2017 to 2019. Currently, he is a Full Professor at Nankai University. As the first author (or corresponding author), he has published more than 100 papers at venues like IEEE S&P, ACM CCS, NDSS, Usenix Security, IEEE TDSC, and IEEE TIFS. His main research interests focus on passwords, authentication, and provable security. His research has been reported by over 200 media like Daily Mail, Forbes, IEEE *Spectrum, and Communications of the ACM*, appeared in the Elsevier 2017 "Article Selection Celebrating Computer Science Research in China," and resulted in the revision of the authentication guideline NIST SP800-63-2.

Dr. Wang has been involved in the community as a PC Chair/TPC member for over 60 international conferences such as NDSS 2023-2025, ACM CCS 2022, PETS 2022-2024, ACSAC 2020-2024, RAID 2024/2023, IEEE EuroS&P 2025, FC 2026/2025, ACM AsiaCCS 2025/2022/2021, ICICS 2018-2025, and SPNCE 2020-2022. He has received the "ACM China Outstanding Doctoral Dissertation Award," the Best Paper Award at INSCRYPT 2018, the First Prize of Cryptologic Innovation Award of the China Association for Cryptologic Research, and the First Prize of Natural Science Award of Ministry of Education.