Secure and Lightweight User Authentication Scheme for Cloud-Assisted Internet of Things

Chenyu Wang, Ding Wang, Yihe Duan, and Xiaofeng Tao, *Senior Member, IEEE*

***Abstract*— Cloud-assisted Internet of Things (IoT) overcomes the resource-constrained nature of the traditional IoT and is developing rapidly in such fields as smart grids and intelligent transportation. In a cloud-assisted IoT system, users can remotely control the IoT devices and send specific instructions to them. If the users' identities are not verified, adversaries can pretend as legitimate users to send fake and malicious instructions to IoT devices, thereby compromising the security of the entire system. Thus, a sound authentication mechanism is indispensable to ensure security. At the same time, it should be noted that a gateway may connect to massive IoT devices with the exponential growth of interconnected devices in a cloud-assisted IoT system. The efficiency of authentication schemes is easily impacted by the computation capability of the gateway. Recently, several schemes have been designed for cloud-assisted IoT systems, but they have problems of one kind or another, making them not suitable for cloud-assisted IoT systems. In this paper, we take a typical scheme (proposed at IEEE TDSC 2020) as an example to identify the common weaknesses and challenges of designing a user authentication scheme for cloud-assisted IoT systems. In addition, we propose a new secure user authentication scheme with lightweight computation on gateways. The proposed scheme provides secure access between remote users and IoT devices with many ideal attributions, such as forward secrecy and multi-factor security. Meanwhile, the security of this scheme is proved under the random-oracle model, heuristic analysis, the ProVerif tool, and BAN logic. Compared with ten state-of-the-art schemes in security and performance, the proposed scheme achieves all the listed twelve security requirements with minimum computation and storage costs on gateways.**

***Index Terms*— User authentication, Internet of Things, cloud computing, offline dictionary attack.**

I. Introduction

THE Internet of Things (IoT) is a dynamic network with self-configuring interconnected objects. It enables these objects to be measured, connected, communicated, understood, and then makes decisions intelligently [1]. According to Gubbi et al. [2], with the popularity of 5G technology, the number of interconnected devices nowadays exceeds the number of users in 2011 and is projected at 24 billion by 2020 [3]. The substantial amount of interconnected devices places a significant computation and storage burden on IoT networks. In this situation, numerous researchers [3], [4], [5] are pursuing the solution of integrating IoT and cloud computing. Cloud computing compensates for IoT networks' computation and storage constraints by providing virtually unlimited storage and computation capability. IoT technology also extends the scope and perception of cloud computing in the real world by offering a mass of environmental data. The integration of IoT and cloud computing techniques maximizes mutual benefits [6]. It dramatically improves applications in smart cities, smart transportation, and smart grids, where large amounts of data and numerous devices are involved, and complex computation are required [3], [7], [8], [9].

Nonetheless, the benefits of integrating cloud computing with IoT techniques are accompanied by new security challenges. Such concerns have been diffused recently about the personal information being acquired in the cloud computing environment by adversaries with ulterior motives. Integrating cloud computing with IoT systems will exacerbate these concerns about privacy protection, as IoT networks bring real-world data to the cloud, and cloud computing increases the number of actions that can be conducted in the real world. Therefore, it is crucial to prevent unauthorized access to these sensitive data. As the first line of defense for system security, user authentication has received extensive attention.

Fig. 1 shows the architecture of a cloud-assisted IoT system in terms of authentication schemes. The authentication for this system involves four different stakeholders: the users with smart mobile devices, the gateways, the IoT devices, and the cloud center. The IoT networks consist of numerous IoT devices and a limited number of gateways. The IoT devices collect real-time data from environments and send the data to their connected gateways. The gateways then upload the collected data into the cloud center. Next, the cloud center processes the environment data and implements intelligent services for users.

In a cloud-assisted IoT system, there are usually two typical authentication scenarios: Auth-Sce I, where the users want to access the services provided by the cloud center, involves the users and cloud center two participants; Auth-Sce II, where the users want to access the real-time data from the IoT devices or deliver instructions to IoT devices via the cloud center (such as via the application programs installed on their phones),

Manuscript received 7 September 2022; revised 19 February 2023 and 17 April 2023; accepted 18 April 2023. Date of publication 3 May 2023; date of current version 15 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62102042; and in part by the Natural Science Foundation of Tianjin, China, under Grant 21JCZXJC00100. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Husrev Taha Sencar. (g.)

Chenyu Wang and Xiaofeng Tao are with the College of Cyber Science and the National Engineering Research Center of Mobile Network Technologies, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Ding Wang and Yihe Duan are with the College of Cyber Science, Nankai University, Tianjin 300350, also with the a, and also with the 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/TIFS.2023.3272772

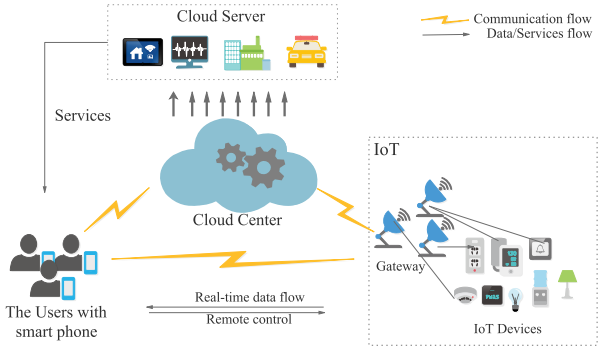

Fig. 1. Architecture of cloud-assisted Internet of Things.

involves three or four participants. In the latter authentication scenario, if there are a limited number of IoT devices and they are directly connected to the cloud center, the authentication phase then involves the users, the cloud center, and the IoT devices; if the number of IoT devices is extensive and they are directly connected to the gateway nodes, the authentication phase then involves the users, the cloud center, the gateways, and the IoT devices.

Usually, the Auth-Sce II containing four participants has various real-world applications. For example, in an industrial predictive maintenance system, IoT devices are deployed to continuously monitor and update the real-time status of critical industrial machines. Once abnormal data are found, the gateway will send them to the cloud center; the cloud center then makes an integrated diagnosis based on the data submitted by each gateway afterwards. On this occasion, responsible persons (the users) may need to access the real-time data directly from IoT devices to further check and deliver the instructions to a specific device to handle the exception. To ensure the security of this occasion, users and target IoT devices should first verify their identities mutually and then build a session key to protect subsequent communications.

Generally, such an authentication scheme consists of three basic phases: registration, login, and authentication. If the cloud center does not participate in the authentication phase, the gateways need to store user-related data and do some computation to authenticate the users. When thousands of IoT devices are connected to one gateway, the efficiency of the authentication scheme is primarily impacted by the performance of the gateway. Thus, the participation of the cloud center in the authentication phase can significantly alleviate the computation pressure at the gateway side.

A. Motivations and Contributions

With the exponential growth of interconnected IoT devices, a single gateway may be connected with thousands of IoT devices [17]. It means that a single gateway node may simultaneously perform mutual authentication with thousands of IoT devices. Hence, the efficiency of authentication schemes is significantly impacted by the computation and storage limitations of the gateway. Thus, seeking a solution to share the gateway's load is crucial to improve the efficiency of authentication schemes. Cloud computing technology is regarded as a promising way to solve this issue [3], [6]. Making proper use of the cloud center's computation power and storage capabilities to alleviate the gateway's load can significantly enhance the efficiency of authentication schemes.

Nevertheless, from the history of user authentication schemes for cloud-assisted IoT, most schemes are not designed for the Auth-Sce II where users can remotely control and access real-time data on IoT devices. Alternatively, the impact of the ever-growing IoT devices on both the performance of gateways and the efficiency of authentication has not been fully taken into account. In most schemes for the Auth-Sce II, the cloud center simply participates in the registration process instead of incorporating it into the authentication process. As such, with the increasing number of connected IoT devices, the gateway in these schemes has to deal with a large number of concurrent user requests, putting tremendous computation pressure on it, which is unsuitable for the Auth-Sce II with four parties.

In this paper, we are committed to designing a suitable authentication scheme for the Auth-Sce II with four parties to adapt to the computation and storage pressure of the gateway caused by the explosive growth of IoT devices. The proposed scheme not only offers protection against various security threats, but also ensures low computation costs at the gateway. Contributions are summarized below.

- We define the adversary model and evaluation criteria for cloud-assisted IoT systems to describe the real adversary capabilities and the security requirements that the authentication schemes need to meet.
- We take a state-of-the-art authentication scheme (published at IEEE TDSC [13]) as a case study to reveal the challenges and subtleties of designing a practical authentication scheme for cloud-assisted IoT systems.
- We propose a secure and efficient authentication scheme for remote control and real-time data access in cloud-assisted IoT systems. The proposed scheme provides many ideal attributes and greatly reduces the computational burden of the gateway by leveraging the capabilities of the cloud center. It is especially suitable for cloud-assisted IoT applications with massive IoT devices.
- We analyze the security of the proposed scheme by provable security analysis, the ProVerif tool, heuristic analysis, and BAN logic, and compare it with ten state-of-the-art relevant schemes in terms of security and performance. The results show that our scheme achieves all listed twelve security requirements with minimum computation and storage costs on the gateway nodes.

Note that this paper extensively expands upon an earlier conference paper [18], with four major differences: 1) The extended version uses a more typical scheme, i.e., Wazid et al.'s scheme at IEEE TDSC'20 [13], as an example to show the difficulties and unreasonableness of most cloud-assisted IoT authentication schemes. 2) The extended version provides formal security proof for the proposed scheme. 3) The extended version improves the original scheme in [18] to achieve better security (i.e., resistance to DDoS attacks). 4) The extended version describes the adversary model and evaluation criteria for cloud-assisted IoT authentication schemes.

TABLE I
THE SKETCH OF USER AUTHENTICATION SCHEMES FOR CLOUD-ASSISTED IOT

Schemes	Years	Auth-Sce*	Participants	Main limitations
Chaudhry et al. [10]	2021	Auth-Sce II	User, gateway, IoT devices	Cannot achieve multi-factor security
Bhuarya et al. [11]	2021	Auth-Sce I	User, cloud centers	Using timestamps, cannot achieve multi-factor security
Srinivas et al. [12]	2020	Auth-Sce II	User, gateway, IoT devices, cloud center	Cannot achieve forward secrecy and multi-factor security
Wazid et al. [13]	2020	Auth-Sce II	User, gateway, IoT devices, cloud center	Cannot achieve forward secrecy and multi-factor security
Sharma et al. [14]	2020	Auth-Sce I	User, cloud center	Cannot resist insider attacks and not achieve user anonymity
Jiang et al. [7]	2020	Auth-Sce II	User, gateway, IoT devices	Cannot achieve forward secrecy and user anonymity
Amin et al. [6]	2018	Auth-Sce I	User, cloud center	Cannot achieve forward secrecy and multi-factor security
Shen et al. [15]	2018	Auth-Sce I	User, cloud center	Using timestamps, cannot achieve multi-factor security
Das et al. [16]	2018	Auth-Sce I	User, cloud center	Cannot achieve forward secrecy and multi-factor security

Auth-Sce*: Authentication Scenarios. In Section I, we mention that there are usually two typical authentication scenarios:
1) In the Auth-Sce I: the users want to access the services provided by the cloud center. The authentication target of this authentication scenario is to allow the user and the cloud center to confirm the authenticity of each other and establish a secure session key between them;
2) In the Auth-Sce II: the users want to access the real-time data from the IoT device or send instructions to the IoT device via the cloud center. The authentication target of this authentication scenario is to allow the user and the IoT device to confirm the authenticity of each other and establish a secure session key between them.
Note that, in the Auth-Sce II, the participants involve resource-constrained IoT devices, while the participants in the Auth-Sce I do not. Based on the difference between the two scenarios in terms of participants and authentication targets, the adversary in Auth-Sce II has more abundant attack entry points, because the adversary in the Auth-Sce II can not only use the attack entry points in the Auth-Sce I, but also find new attack entry points via the vulnerability of IoT devices. For example, since the IoT devices are usually deployed in an unattended environment and are vulnerable to being compromised, the adversary can capture a certain number of IoT device nodes to obtain valuable information and then break the scheme's attributes such as forward secrecy.

II. RELATED WORK

In 2009, to support users in securely accessing the real-time data stored in sensor nodes, Das [19] first proposed a two-factor user authentication scheme for wireless sensor networks (WSNs, one of the essential infrastructures of IoT). Since then, numerous authentication schemes for WSNs have been proposed [7], [20], [21], [22], [23], [24], but most are unsatisfactory in some way. For example, some schemes are identified as vulnerable to offline dictionary attacks; some cannot withstand insider attacks; and some are prone to impersonation attacks. Recently, with the prevalence of IoT techniques, more user authentication schemes for IoT systems have been developed. There are some notable schemes like [12], [20], [21], [25], [26]. However, these schemes still suffer from various attacks. For instance, Wazid et al.'s scheme [26] is vulnerable to offline dictionary attacks and cannot achieve user anonymity and forward secrecy; Wu et al.'s scheme [27] cannot resist offline dictionary attacks.

In 2018, Amin et al. [6] pointed out the importance of integrating the IoT network with the cloud computing center, and then proposed an authentication scheme for cloud-assisted IoT systems. Their scheme contains two parties, i.e., the cloud center and the users. In other words, this scheme is designed for the Auth-Sce I, not for the Auth-Sce II where the user requires to obtain the real-time data of IoT devices and deliver instructions to them. In addition, this scheme is vulnerable to various security threats. As shown in Table I, similar considerations also apply to the schemes of Shen et al. [15], Das et al. [16], Sharma et al. [14] and Bhuarya et al. [11]. In short, none of these schemes supports the interaction between users and IoT devices.

In 2020, Jiang et al. [7] presented a user authentication scheme for cloud-assisted autonomous vehicles (an IoT application). Their scheme realizes the authentication among the users, the cloud center, and the IoT devices. In 2021, Chaudhry et al. [10] also proposed a lightweight scheme for cloud-assisted IoT, which supports the authentication among the users, the gateways, and the IoT devices. Both the schemes of Jiang et al. and Chaudhry et al. are applicable to the Auth-Sce II with three parties. But the security of both schemes is not guaranteed. From the perspective of protocol design, whether the set of participants is {the users, the IoT devices, the gateways} or {the users, the IoT devices, the cloud center}, does not significantly impact the communication architecture and design ideas of the authentication protocols. Thus, we view these two schemes as one category. Recently, a large number of such tripartite authentication schemes have been proposed [25], [26], [28], but they have all been found to have various security issues. Such schemes that are designed for Auth-Sce II with three parties are not our focus.

In 2020, Wazid et al. [13] presented a three-factor user authentication scheme in a smart home environment (an IoT application) with formal security analysis. In this scheme, the registration server, which is responsible for the key distribution and the participants' registration, can be regarded as a cloud center in cloud-assisted IoT environments. This scheme supports authentication among users, the gateway, IoT devices, and the cloud center. It is suitable for the Auth-Sce II with four parties. Unfortunately, after reviewing the scheme of Wazid et al., we note that the registration server (cloud center) simply joins in the registration phase. As such, the gateway has to undertake the huge computation and storage task in the authentication phase to verify the identities of the users, so does the scheme of Srinivas et al. [12].

From the history of user authentication for cloud-assisted IoT, little attention is paid to the Auth-Sce II with four parties. Besides, the current researches emphasize the computation complexity more at the IoT devices, rather than at the gateway. There have been many discussions on reducing the computing load on the IoT device side, but few on reducing the computing load on the gateway in the cloud-IoT environment. However, with the development of electronic technology, on the one hand, the computing and storage capabilities of a single IoT device are constantly improving; on the other hand, more and more IoT devices are connected to a single gateway [3], [17]. As a result, a single gateway has to handle a significant number of concurrent authentication requests, and the computation

TABLE II
NOTATIONS AND ABBREVIATIONS

Symbol	Description	Symbol	Description
U_i	i^{th} user	S_j	j^{th} IoT devices
GWN_k	k^{th} gateway	CloCen/RA.	cloud center/register center
$\mathcal{A}$	the adversary	SK	the session key
ID_i	identity of U_i	SID_j	identity of S_j
GID_k	identity of GWN_k	PW_i, Bio_i	U_i's password, biometrics
Gen/Rep	fuzzy extractor	x/y	$CloCen$'s secret key
X_{S_j}	secret key of S_j	x_{G_k}	GWN's secret key
X_{G_k}	secret key of GWN_k	$K_{\text{GWN}-U_i}$	secret key of GWN for U_i
$\oplus$	bitwise XOR operation	$K_{\text{GWN}-S_j}$	secret key of GWN for S_j
$\rightarrow$	an insecure channel	$\Longrightarrow$	a secure channel
$h(\cdot)$	one-way hash function	$\|$	concatenation operation

complexity of the authentication scheme at the gateway will have a significant impact on the system's efficiency.

Therefore, this paper aims to provide a secure authentication scheme for the Auth-Sce II with four parties. In addition to focusing on the security properties like resisting various attacks, we also emphasize the performance properties like reducing the computation complexity of the gateways through the cloud center, making it applicable to a network environment with massive IoT devices.

III. Adversary Model and Evaluation Criteria

In this section, we depict the adversary model and evaluation criteria of cloud-assisted IoT systems. All the notations used in this paper are presented in Table II.

A. Adversary Model

We explicitly summarize the adversary model that incorporates realistic adversary capabilities as below. It should note that the adversary of this model is not allowed to acquire the temporary secret parameters of sessions.

C-1. According to the Dolev-Yao model [29], the adversary $\mathcal{A}$ can fully control the messages transmitted among users, the cloud center, the gateway, and IoT devices in an insecure channel [30].

C-2. $\mathcal{A}$ can enumerate all of the items in the Cartesian product $\mathcal{D}_{id} \times \mathcal{D}_{pw}$ within polynomial time, where $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$ are the space of identities and passwords, respectively; $\mathcal{A}$ also can obtain users' identities when assessing the security of the schemes. The two capabilities are given from the facts: 1) users' passwords are usually memorable strings and follow a Zipf distribution [31], resulting in the limited space of passwords. 2) Users' identities are usually static and can be gathered from popular forums. Besides, people normally do not keep their identities secret, thus increasing the risk of leakage [32].

C-3. $\mathcal{A}$ can obtain $n-1$ $(n = 2, 3)$ factors in the n-factor user authentication schemes [33], [34], [35], [36]. The factors include passwords, smart mobile devices and biometrics data.

C-4. $\mathcal{A}$ can obtain the secret key of the cloud center or the gateways when evaluating forward secrecy. This capability follows from the definition of forward secrecy, wherein the final compromise of the entire system does not affect the security of previous conversations [37].

C-5. $\mathcal{A}$ can compromise a limited number of IoT devices and extract their stored data, because the IoT devices are usually deployed in an unattended or even hostile environment, and physical access is easy [26], [33].

C-6. $\mathcal{A}$ can obtain previous session keys between users and IoT devices [25], [37].

C-7. When determining the security of the registration phase, $\mathcal{A}$ is able to be the administrator of the cloud center [7], [25]. This capability is allowed to capture insider attacks in the registration phase. In this attack, the adversary can finally compromise users' passwords.

B. Evaluation Criteria

As shown in Table III, we construct our evaluation criteria based on a widely accepted criteria framework [33]. The evaluation criteria are divided into two categories: the ideal attributes and the security requirements. The ideal attributes are evaluated from a functional perspective, i.e., assessing whether the scheme itself has these attributes. The security requirements are evaluated from an attack perspective, i.e., assessing whether $\mathcal{A}$ can succeed in breaking the scheme.

IV. Cryptoanalysis of Wazid et al.'s Scheme

In 2020, Wazid et al. [13] presented a user authentication scheme for a smart home environment (a typical IoT application), and the cloud center of the scheme serves as a registration center to distribute the three parties' secret keys. The authors demonstrate that their scheme is secure against known attacks via formal and informal security analysis. However, the review of this scheme shows that it is subject to critical security weaknesses, such as being vulnerable to desynchronization attacks and not providing multi-factor security and forward secrecy. In this section, we take this typical scheme as an example to identify challenges and subtleties of designing the user authentication schemes for a cloud-assisted IoT environment, i.e., the Auth-Sec II with four parties. We show how an adversary utilizing our practical attack model, as defined in Section III-A, can break Wazid et al.'s scheme. Besides, we also discuss the role of the cloud center in this occasion to explore a better way to deal with the performance limitation of the gateway when the number of connected IoT devices continues to increase. For the review of Wazid et al.'s scheme, please refer to the complete paper at https://bit.ly/3RzdctY.

Here $Gen(\cdot)/Rep(\cdot)$ is a fuzzy extractor algorithm [38]:

- $Gen(\cdot)$: $Gen(Bio_i) = (\delta_i, \tau_i)$. It is a probabilistic generation function. When inputting $Bio_i \in$ metric space $\mathcal{M}$, it outputs an "extracted" string $\delta_i \in \{0, 1\}^l$ and a public string τ_i. This function is used to get two determined values from U_i's biometrics Bio_i.
- $Rep(\cdot)$: $Rep(Bio_i', \tau_i) = \delta_i$. It is a deterministic reproduction function to recover δ_i. For any Bio_i' and $Bio_i \in \mathcal{M}$, if their Hamming distance is negligible, it outputs δ_i.

A. No Multi-Factor Security

Multi-factor security is a crucial requirement of a multi-factor user authentication scheme. It ensures the security of the rest factors even if an adversary has compromised any two of the three factors. However, we find that if an

TABLE III
EVALUATION CRITERIA

	Short term		Definition in WSNs
Ideal Attributes	D1	Password Friendly	It is allowed for users to choose and locally change their passwords.
	D2	Sound Repairability	The IoT devices can join the network dynamically and the smart card can be revoked.
	D3	Key Agreement	The user and IoT devices should build a session key after the authentication.
	D4	No clock synchronization	There is no need for participants to synchronize their time clock.
	D5	Mutual Authentication	All participants should verify others' identities.
	D6	No Password Verifier table	Password-related parameters are only stored in the user side.
Security Requirements	S1	User Anonymity	The users' identities can neither be calculated nor tracked by the adversary.
	S2	No Password Exposure	In the registration phase, the privileged participants (usually the administer of the cloud center) cannot obtain the users' password.
	S3	Forward Secrecy	The agreed session key cannot be acquired by $\mathcal{A}$ even when any one of parties are compromised.
	S4	Resistance to Known Attacks	The scheme can resist impersonation attacks, offline guessing attacks, de-synchronization attacks, replay attacks, stolen verifier-attacks, unknown key share and known key attacks, DDoS attacks. Note that, in these attacks, $\mathcal{A}$ does not compromise the smart mobile device and the Iot device.
	S5	Resistance to Smart Mobile Device Loss Attacks	$\mathcal{A}$ fails to attack the scheme via a user's smart mobile device.
	S6	Resistance to Node Capture Attacks	$\mathcal{A}$ cannot break the scheme via compromising the IoT devices.

adversary in Wazid et al.'s scheme compromises the victim's smart mobile device (to get $\{\tau_i, B_i\}$) and biometrics, then he can launch an offline dictionary attack to get U_i's password following the process given below:

Step 1. Guess the password and identity to be PW_i^*, ID_i^* from $\mathcal{D}_{pw}$ and $\mathcal{D}_{id}$, respectively.
Step 2. Compute $\delta_i^* = Rep(Bio_i^*, \tau_i)$.
Step 3. Compute $a^* = B_i \oplus h(ID_i^* || \delta_i^*)$.
Step 4. Compute $RPW_i^* = h(PW_i^* || \delta_i^* || a^*)$.
Step 5. Compute $C_i^* = h(ID_i^* || RPW_i^* || \delta_i^*)$.
Step 6. Check the correctness of PW_i^* and ID_i^* by verifying whether $C_i^* \stackrel{?}{=} C_i$.
Step 7. Repeat Steps 1∼6 until the correct value is found.

The time complexity of the above attack is $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (3T_H + T_B))$, where T_H denotes the running time for hash functions and T_B denotes the running time for the biometric fuzzy extractor. It indicates the efficiency of this attack. Furthermore, once $\mathcal{A}$ gets U_i's password, he can further impersonate U_i to all other participants. Note that, $\mathcal{A}$ can also make use of M_3 as the verification value to test the correctness of the guessed password and identity, and conduct a similar offline dictionary attack as below. In this new attack, $\mathcal{A}$ is armed with three capabilities: get $\{\tau_i, B_i\}$ from U_i's smart mobile device; obtain U_i's biometric data; eavesdrop $\{TID_i, M_2, M_3, T_1\}$ transmitted between the user U_i and the cloud center.

Step 1. Guess the password and identity to be PW_i^*, ID_i^* from $\mathcal{D}_{pw}$ and $\mathcal{D}_{id}$, respectively.
Step 2. Compute $\delta_i^* = Rep(Bio_i^*, \tau_i)$.
Step 3. Compute $a^* = B_i \oplus h(ID_i^* || \delta_i^*)$.
Step 4. Compute $RPW_i^* = h(PW_i^* || \delta_i^* || a^*)$.
Step 5. Compute $M_1^* = A_i^* \oplus RPW_i^*$.
Step 6. Compute $r_{U_i}^* = M_2 \oplus M_1^*$.
Step 7. Compute $M_5^* = h(M_2 || T_1 || ID_i^* || TID_i || r_{U_i}^*)$
Step 8. Check the correctness of PW_i^* and ID_i^* by verifying whether $M_5^* \stackrel{?}{=} M_5$.
Step 9. Repeat Steps 1-8 until the correct value of (PW_i^*, ID_i^*) is found.

The time complexity of the above attack is also $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (3T_H + T_B))$. The inherent reasons for both attacks are similar: $\mathcal{A}$ can construct and obtain a verification parameter using the victim's biometric data, the parameters in the smart device and a guessed password. To avoid the former attack, a solution integrating the fuzzy-verifier and honey-words has been introduced [37]. The key concept of this method is to let the verification parameter (for example, C_i in Wazid et al.'s scheme) be a fuzzy-verifier, such as $h(ID_i || RPW_i || \delta_i)$ mod n_0, where n_0 is an integer between 2^4 and 2^8. In this way, there are approximately $|\mathcal{D}_{pw}| * |\mathcal{D}_{id}|/n_0 \approx 2^{32}$ candidates of $\{ID_i, PW_i\}$ pairs that satisfy the equation when $n_0 = 2^8$ and $|\mathcal{D}_{pw}| = |\mathcal{D}_{id}| = 10^6$. $\mathcal{A}$ then has to interact with the cloud center online to further verify the correctness of the guessed password. On the other hand, the honey-words will record the user's authentication failures. Once the number of failures times exceeds a pre-set value (such as 10), the victim's account will be locked till he re-registers. Thus, $\mathcal{A}$ can only conduct a limited number of online queries. Thus, the probability that $\mathcal{A}$ obtains the correct password is small.

For the latter attack, Ma et al. [39] have proved the necessity of a public-key algorithm. That is, we can set the verification parameter containing a parameter Pub, and Pub is transmitted by a public-key algorithm. For example, let M_5 be $h(M_2 || T_1 || ID_i || TID_i || r_{U_i} || Pub)$, where Pub can only be computed by the cloud center's secret key. As such, without Pub, $\mathcal{A}$ cannot construct such a M_5^* to verify the guessed password as the above attack, thus preventing the attacks.

B. Desynchronization Attacks

A desynchronization attack occurs when two participants store inconsistent parameters. Thus, even legitimate participants cannot be authenticated successfully. This attack is straightforward but severe, which is hard to avoid by making minor changes simply. Unfortunately, in Wazid et al.'s scheme, once an adversary controls the messages among the four participants, he can make the TID_i on the user side inconsistent with that in the gateway, thus leading to desynchronization issues. The attack steps are shown below:

Step 1. Intercept $\{M_{14}, M_{15}, M_{16}, T_3, T_4\}$.
Step 2. Compute $M_{15}^{*\mathcal{A}} = M_{15} \oplus R^{\mathcal{A}}$, $R^{\mathcal{A}}$ is a random number chosen by $\mathcal{A}$.
Step 3. Send $\{M_{14}, M_{15}^{*\mathcal{A}}, M_{16}, T_3, T_4\}$ to U_i.

The time complexity of the above attack is very small. Note that, following Wazid et al.'s scheme, once the adversary getting $\{M_{14}, M_{15}^{*\mathcal{A}}, M_{16}, T_3, T_4\}$, the smart mobile device

will compute the session key and check M_{16}; then, TID_i is updated with $TID_i^{new'}$, where $TID_i^{new'} = M_{15}^{*\mathcal{A}} \oplus h(TID_i||M_1||T_3||T_4)$. Obviously, here $TID_i^{new'}$ is not equal to the gateway's selected value TID_i^{new}. Therefore, legitimate U_i and GWN_k cannot authenticate each other successfully.

C. No Forward Secrecy

With the increasing attacks on the server side, forward secrecy has become the last significant defense to protect the security of a system. It guarantees the security of the previous conversations even if the entire system is compromised. It is a highly critical security requirement for user authentication schemes. However, Wazid et al.'s scheme does not provide forward secrecy. Once $\mathcal{A}$ obtains the secret key $K_{\text{GWN}-S_j}$ stored in the gateway and controls the open channel, he can compute the previous session keys between the users and the IoT devices as below:

Step 1. Intercept M_7.
Step 2. Decrypt M_7 with $h(SID_j||K_{\text{GWN}-S_j})$ to get $\{ID_i, GID_k, r_{U_i}^*, r_{\text{GWN}}, h(M_4)\}$.
Step 3. Compute the session key between U_i and S_j as $SK = h(ID_i||SID_j||GID_k||r_{U_i}^*||r_{\text{GWN}}||r_{S_j}||h(M_4)||h(h(SID_j||K_{\text{GWN}-S_j})))$.

The time complexity of the above attack is $\mathcal{O}(2T_H)$. From this attack, we can see that it is not trivial to provide forward secrecy. As proved by Ma et al. [39], there are two requirements to achieve forward secrecy: a public-key algorithm and at least two module exponentiation or point multiplication operations on the server side (i.e., the IoT devices). Following this principle, a major modification to Wazid et al.'s scheme is required to ensure forward secrecy.

D. The Role of Cloud Center

In Wazid et al.'s scheme [13], the cloud center mainly participates in the registration processes of IoT devices, the gateway, and the users. As a trusted center, it assigns some core secret parameters to the other three participants to provide a basis for establishing trust among them. Specifically, the cloud center assigns $K_{\text{GWN}-S_j}$ to establish the trust base between the IoT device S_j and the gateway GWN_k, and assigns $K_{\text{GWN}-U_i}$ to establish the trust base between the gateway GWN_k and the user U_i. Then the gateway needs to store the parameters related to the IoT device $\{SID_j, K_{\text{GWN}-S_j}\}$ and the parameters $\{TID_i, ID_i, K_{\text{GWN}-U_i}\}$ related to the user, to assist S_j and U_i to complete mutual authentication. In the authentication phase, the cloud center does not participate, and thus GWN_k needs first to authenticate the identity of the user and the IoT device, and then establish a trust link between them. In this scenario, as the number of IoT devices increases, the computation and storage overhead of the gateway will increase sharply.

From another perspective, if the cloud center participates in the authentication phase, it can store user-related parameters and assist in achieving user-related identity verification, so as to partly share the computation and storage pressure of the gateway, and then improve the efficiency of the scheme.

V. Proposed Scheme

As discussed in Section IV, in Wazid et al.'s scheme [13], the cloud center simply assigns parameters to other participants. As such, when the number of IoT devices grows large, a single gateway may execute thousands of authentication sessions concurrently, and the efficiency of the scheme is easily impacted by the capabilities of the gateway. Therefore, Wazid et al.'s scheme is inadequate for cloud-assisted IoT environments where large numbers of IoT devices are involved. In order to design an authentication protocol that fits the occasion, we improve the efficiency of the scheme by moving heavy computation and storage tasks to the cloud center. Furthermore, as suggested by Ma et al. [39] and Wang et al. [37], we employ a public key cryptography algorithm, fuzzy-verifier, and honey-words techniques to achieve multi-factor security. Besides, we provide forward secrecy by performing two elliptic curve point multiplication operations on the IoT device side [39]. Meanwhile, since temporary-certificate-based schemes are typically prone to desynchronization attacks [32], we adopt the public-key algorithm to meet the same security function. The processes of our scheme are given below.

A. IoT Device and Gateway Registration Phase

In the proposed scheme, to provide better security, we let the cloud center CloCen own two secret long-term keys x and y, relevant to the users and the gateways respectively. The cloud center distributes the gateway GWN_k a secret key X_{G_k} $(=h(x||GID_k))$ to serve as an authenticated credential. The gateway GWN_k and IoT devices S_j share a secret key X_{S_j} $(=h(SID_j||x_{G_k}))$, where x_{G_k} is the gateway's long-term secret key. In this way, the IoT network and the cloud center can run independently, which creates more flexibility and provides more security for the deployment of the scheme in the real world. In addition, our scheme is built on an elliptic curve E (which is generated by P with a large prime order q) over a prime finite field F_p, and the public key is $Y = yP$.

In the proposed scheme, the gateway and the user register to the cloud center, and the IoT device registers to the gateway. After the registration, the cloud center will establish a secret key with the gateway and the user respectively; the gateway will build a secret key with the IoT devices. These secret keys are critical parameters to the authentication process.

In the gateway registration phase, the gateway first sends the registration request to the cloud center, and then the cloud center returns a secret key X_{G_k} as below.

R1. $\text{GWN}_k \Longrightarrow$ CloCen: registration request (including the identity GID_k of the gateway GWN_k).
R2. CloCen $\Longrightarrow GWN_k$: $\{GID_k, X_{G_k}\}$. The cloud center firstly checks the validity of GID_k, then computes $X_{G_k} = h(x||GID_k)$ as the gateway's authenticated credential, where x is the secret key of the cloud center, and finally sends $\{GID_k, X_{G_k}\}$ to GWN_k.
R3. GWN_k keeps X_{G_k}.

In the IoT device registration phase, the IoT device S_j first sends the registration request to the gateway. Then the gateway distributes the IoT device a secret key X_{S_j} as below.

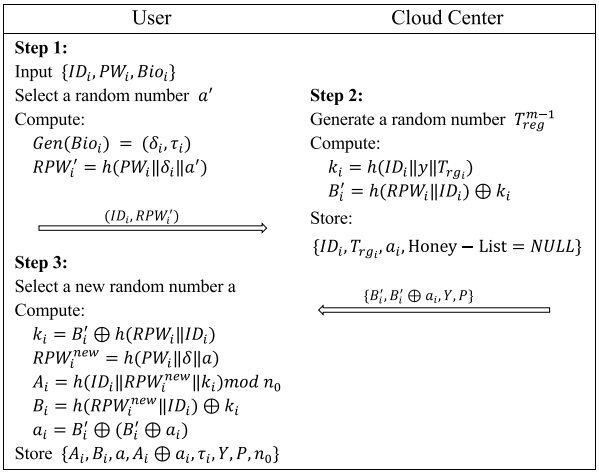

Fig. 2. The user registration phase of the proposed scheme.

R1. $S_j \Longrightarrow \text{GWN}_\text{k}$: registration request (including the identity SID_j of the IoT device S_j).

R2. $\text{GWN}_\text{k} \Longrightarrow S_j$: $\{SID_j, X_{S_j}\}$. GWN_k firstly checks the validity of SID_j, then computes $X_{S_j} = h(SID_j || x_{G_k})$, where x_{G_k} is the secret key of GWN_k.

R3. S_j keeps X_{S_j} as its private key.

B. User Registration Phase

In the user registration phase, as shown in Fig. 2, the user U_i first submits his personal information to the cloud center. Then the cloud center creates an entry to U_i and computes a unique and fixed secret parameter k_i to U_i:

R1. $U_i \Longrightarrow$ CloCen: $\{ID_i, RPW_i\}$.
U_i chooses his identity ID_i and password PW_i, enters his biometric Bio_i. Next, the smart mobile device selects a random number a', computes: $Gen(Bio_i) = (\delta_i, \tau_i)$, $RPW_i = h(PW_i||\delta_i||a')$, and sends the registration request $\{ID_i, RPW_i\}$ to the cloud center CloCen.

R2. $CloCen \Longrightarrow U_i$: $\{B_i', B_i' \oplus a_i, Y, P\}$.
CloCen picks a timestamp T_{rg_i} and a random number a_i, computes $k_i = h(ID_i||y||T_{rg_i})$, $B_i' = h(RPW_i||ID_i) \oplus k_i$, then stores $\{ID_i, T_{rg_i}, a_i,$ Honey−list=NULL$\}$ in the database, and finally sends $\{B_i', B_i' \oplus a_i, Y, P\}$ to U_i.

R3. After receiving $\{B_i', B_i' \oplus a_i, Y, P\}$, the smart device selects a new random number a, computes: $k_i = B_i' \oplus h(RPW_i||ID_i)$, $RPW_i^{new} = h(PW_i||\delta_i||a)$, $A_i = h(ID_i||RPW_i^{new}||k_i) \bmod n_0$, $B_i = h(RPW_i^{new}||ID_i) \oplus k_i$, $a_i = B_i' \oplus (B_i' \oplus a_i)$, and finally keeps $\{A_i, B_i, a, A_i \oplus a_i, \tau_i, Y, P, n_0\}$.

Note that, the reason that we update the random number a is to resist privileged insider attacks. If a is not updated (in this case, a is stored in the smart device), then an administrator of the cloud center who obtains RPW_i and the parameters $\{a, \tau_i\}$ stored in the smart device can conduct the insider attacks following the steps below:

Step 1. Guess the password to be PW_i^* from $\mathcal{D}_{pw}$.
Step 2. Compute $RPW_i^* = h(PW_i^*||\tau_i||a)$,.
Step 3. Check the correctness of PW_i^* by verifying whether $RPW_i^* \stackrel{?}{=} RPW_i$.
Step 4. Repeat Steps 1∼3 until the correct value is found.

The core of this insider attack is: 1) the administrator can obtain the RPW_i, and 2) in the user login phase, passwords and biometric data are often required to derive the RPW_i to complete the authentication of the user's identity by the smart device. In this way, once the administrator gets the parameters (and biometrics) in the smart device, he can deduce a RPW_i^* with the guessed password PW_i^* following the user's login steps, and then verify the values of RPW_i^* and RPW_i to check the correctness of PW_i^*. The key to resisting this insider attack is to make some changes to RPW_i to make it be RPW_i' (Note that RPW_i' is used for users to log into the system). As such, the administrator cannot perform this attack because RPW_i' is not equal to RPW_i. Following this idea, it is not difficult to find that Wazid et al.'s scheme [13] can handle such insider attacks.

C. Login Phase

As shown in Fig. 3, if U_i wants to access an IoT device, he can initiate a login request to the gateway as below:

L1. $U_i \longrightarrow$ CloCen: $\{M_2, M_3, M_4, M_5\}$.
U_i enters $\{ID_i^*, PW_i^*, Bio_i^*\}$, the smart mobile device computes: $\delta_i^* = Rep(Bio_i^*, \tau_i)$, $RPW_i^* = h(PW_i^*||\delta_i^*||a)$, $k_i^* = B_i \oplus RPW_i^*$, $A_i^* = h(ID_i^*||RPW_i^*||k_i^*) \bmod n_0$, then it compares A_i^* with A_i to verifies the authenticity of the user U_i.
If $A_i^* \neq A_i$, the user's request is rejected. Otherwise, the smart device selects r_i, computes: $a_i^* = (A_i \oplus a_i) \oplus A_i$, $M_1 = r_i \cdot Y$, $M_2 = r_i \cdot P$, $M_3 = h(M_2 ||M_1) \oplus (ID_i^*||a_i^*)$, $M_4 = h(M_1||M_2||M_3) \oplus SID_j$, $M_5 = h(k_i^*||ID_i^*||M_1||M_2||SID_j)$, finally transmits $\{M_2, M_3, M_4, M_5\}$ to CloCen.

D. Authentication Phase

As shown in Fig. 3, the authentication phase is the core step of the scheme. It consists of six message flows and involves authentication among four participants.

V1. CloCen $\longrightarrow \text{GWN}_\text{k}$: $\{M_2, M_6, M_7, M_8\}$.
Once obtaining $\{M_2, M_3, M_4, M_5\}$, CloCen first checks the validity of U_i: computes $M_1' = y \cdot M_2$, $ID_i'||a_i' = M_3 \oplus h(M_2||M_1')$, then retrieves $\{T_{rg_i}, a_i\}$ using ID_i', and next compares a_i with a_i'. If $a_i \neq a_i'$, CloCen exits the session. Otherwise, CloCen computes $k_i' = h(ID_i'||y||T_{rg_i})$, $SID_j' = M_4 \oplus h(M_1||M_2||M_3)$, $M_5' = h(k_i'||ID_i'||M_1'||M_2||SID_j')$, and verifies U_i via M_5'.
If $M_5' \neq M_5$, it means that the information provided by the user side does not conform to the data stored in the cloud center. Thus CloCen thinks the message is sent by an adversary, and then rejects the session. Next, CloCen inserts k_i into Honey-list when there are less than 10 items. Once the items in Honey-list exceed 10, CloCen will suspend U_i's account till U_i re-registers. If $M_5' == M_5$, CloCen accepts the authenticity of U_i. Next, CloCen determines which gateway GWN_k the S_j belongs to, and then selects a random number r, computes $X_{G_k}' = h(x||GID_k)$, $M_6 = h(X_{G_k}'||M_2) \oplus r$, $M_7 = h(M_6||r||X_{G_k}') \oplus SID_j'$, $M_8 = h(M_2||M_6||M_7||r||SID_j'||X_{G_k}')$, and sends $\{M_2, M_6, M_7, M_8\}$ to the gateway node GWN_k.

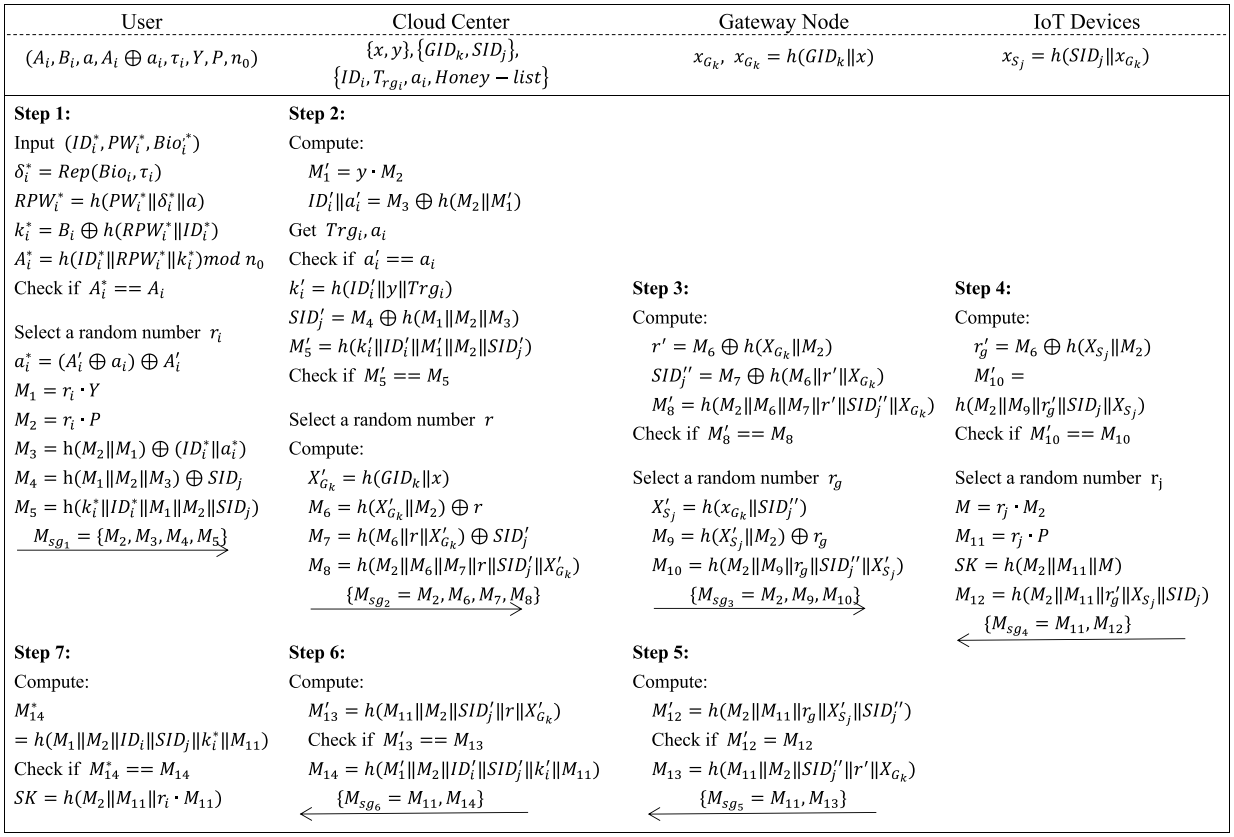

Fig. 3. Login and authentication phase of the proposed scheme.

V2. $\text{GWN}_\text{k} \longrightarrow S_j$: $\{M_2, M_9, M_{10}\}$.
After obtaining the message from CloCen, the gateway GWN_k first computes $r' = M_6 \oplus h(X_{G_k} || M_2)$, $SID_j'' = M_7 \oplus h(M_6 || r' || X_{G_k})$, $M_8' = h(M_2 || M_6 || M_7 || r' || SID_j'' || X_{G_k})$, and then checks whether $M_8' \stackrel{?}{=} M_8$.
If $M_8' \neq M_8$, GWN_k rejects the request. Otherwise, GWN_k computes: $X_{S_j}' = h(x_{G_k} || SID_j'')$, $M_9 = h(X_{S_j}' || M_2) \oplus r_g$, $M_{10} = h(M_2 || M_9 || r_g || SID_j'' || X_{S_j}')$, and sends $\{M_2, M_9, M_{10}\}$, where r_g is a random number chosen by GWN_k.

V3. $S_j \longrightarrow \text{GWN}_\text{k}$: $\{M_{11}, M_{12}\}$.
Once getting $\{M_2, M_9, M_{10}\}$, the IoT device S_j computes $r_g' = M_9 \oplus h(X_{S_j} || M_2)$, $M_{10}' = h(M_2 || M_9 || r_g' || SID_j || X_{S_j})$, and compares the value of M_{10}' and M_{10}. If $M_{10}' \neq M_{10}$, S_j exits the session. Otherwise, S_j chooses a random number r_j, calculates $M = r_j \cdot M_2$, $M_{11} = r_j \cdot P$, $SK = h(M_2 || M_{11} || M)$, $M_{12} = h(M_2 || M_{11} || r_g' || X_{S_j} || SID_j)$, and responds $\{M_{11}, M_{12}\}$ to the gateway GWN_k.

V4. $\text{GWN}_\text{k} \longrightarrow$ CloCen: $\{M_{11}, M_{13}\}$.
On the response from S_j is received, GWN_k computes $M_{12}' = h(M_2 || M_{11} || r_g || X_{S_j}' || SID_j'')$, and compares M_{12}' with M_{12} to check the identity of S_j. If $M_{12}' \neq M_{12}$, GWN_k ends the session. Otherwise, GWN_k calculates $M_{13} = h(M_{11} || M_2 || SID_j'' || r' || X_{G_k})$, sends $\{M_{11}, M_{13}\}$.

V5. CloCen $\longrightarrow U_i$: $\{M_{11}, M_{14}\}$.
After receiving the gateway GWN_k's response, CloCen computes $M_{13}' = h(M_{11} || M_2 || SID_j'' || r || X_{G_k}')$ to test the identity of GWN_k. If $M_{13}' \neq M_{13}$, CloCen ends the session. Otherwise, CloCen computes $M_{14} = h(M_1' || M_2 || ID_i' || SID_j'' || k_i' || M_{11})$, and then returns $\{M_{11}, M_{14}\}$ to the user U_i.

V6. Once obtaining CloCen's reply $\{M_{11}, M_{14}\}$, the smart mobile device computes $M_{14}^* = h(M_1 || M_2 || ID_i || SID_j || k || M_{11})$, and checks whether the value of M_{14}^* is equal to M_{14}. If $M_{14}^* == M_{14}$, U_i accepts $SK = h(M_2 || M_{11} || r_i \cdot M_{11})$ as his session key shared with S_j, and the authentication process finishes successfully. Otherwise, the session is terminated.

Note that, the honey-words technique is used to record the failed logins of users, which is stored in the cloud center by Honey-list. It cooperates with the fuzzy-verifier technology to resist one of the offline password-guessing attacks mentioned in Section IV-A. Additionally, there are some subtleties to be aware of when deploying this technique. Specifically, a_i is a very important parameter that is applied to test whether the smart mobile device has been compromised, and further prevent DDoS attacks. If there is no such a parameter a_i involved, and the number of user's failed logins is simply recorded by verifying the value M_5 like Wang et al.'s scheme [18], then $\mathcal{A}$ can construct a login request arbitrarily and send it to the cloud center. Then following the procedures of their scheme, with a limited number of malicious incorrect login requests, U_i's account will be locked, and a DDoS attack will succeed. Thus, in this proposed scheme, we set an additional parameter a_i to further check whether the user's smart mobile device is compromised: compare the value of a_i' with the stored a_i. Since a_i can only be acquired from the smart mobile device, if a_i is valid, the message sender must have obtained the data

in the smart mobile device. Under this situation, if $M_5' \neq M_5$, it can infer that it is not a legitimate user who has compromised the smart device, which means that $\mathcal{A}$ has the ability and possibility to launch a password-guessing attack at this time.

In our scheme, we suppose that $\mathcal{A}$ has obtained the data $\{A_i, B_i, a, A_i \oplus a_i, \tau_i, n_0\}$ stored in the smart mobile device, and $\mathcal{A}$ tries to guess the value of PW_i' and then verifies the guessed value via A_i like the Steps of Section IV-A. Since A_i is a fuzzy-verifier, as mentioned in Section IV-A, there are approximate $|\mathcal{D}_{pw}| * |\mathcal{D}_{id}|/n_0 \approx 2^{32}$ candidates of $\{ID_i, PW_i\}$ pairs that satisfy the equation when $n_0 = 2^8$ and $|\mathcal{D}_{pw}| = |\mathcal{D}_{id}| = 10^6$. $\mathcal{A}$ then has to interact with the cloud center online to further verify the correctness of the guessed password: $\mathcal{A}$ selects r_i' and constructs $M_1' = r_i Y$, $M_2' = r_i' P$, $M_3' == h(M_1'||M_2') \oplus (ID_i'||a_i^*)$, $M_4' == h(M_1'||M_2'||M_3') \oplus SID_j$, $M_5' == h(k_i'||ID_i'||M_1||M_2||SID_j)$, where k_i' is computed with guessed PW_i' and parameters in the smart mobile device. Note that, $\mathcal{A}$ has gotten correct a_i^* by compute $(A_i \oplus a_i) \oplus A_i$. Thus, if $\mathcal{A}$ guesses a wrong PW_i', then the cloud center will find that $a_i' == a_i$ and $M_5' \neq M_5$, and inserts k_i into Honey-list when there are less than 10 items. Once the items in Honey-list exceed 10, CloCen will suspend U_i's account till U_i re-registers. Thus, $\mathcal{A}$ can only conduct a limited number of online queries, and the probability that $\mathcal{A}$ guesses correct PW_i is very small.

E. Password Change Phase

To achieve user-friendliness, the proposed scheme allows the user U_i to change his password locally as below:

P1. $U_i \longrightarrow$ mobile device: $\{ID_i^*, PW_i^*, Bio_i^*, PW_i^{new}\}$. The user U_i firstly initiates a password change request, and submits $\{ID_i^*, PW_i^*, Bio_i^*, PW_i^{new}\}$.

P2. The smart mobile device computes $\delta_i^* = Rep(Bio_i^*, \tau_i)$, $RPW_i^* = h(PW_i^*||\delta_i^*||a)$, $k_i^* = B_i \oplus RPW_i^*$, $A_i^* = h(ID_i^*|| RPW_i^*||k_i^*) \bmod n_0$.
If $A_i^* \neq A_i$, the device rejects the request; otherwise, it computes $RPW_i^{new} = h(PW_i^{new}||\delta_i^*||a)$, $A_i^{new} = h(ID_i^* || RPW_i^{new}||k_i^*) \bmod n_0$, $B_i^{new} = h(RPW_i^{new}|| ID_i) \oplus k_i$, updates $\{A_i^{new}, B_i^{new}, A_i^{new} \oplus A_i \oplus (A_i \oplus a_i)\}$.

F. Re-Registration Phase

The re-registration phase helps the users whose account has been suspended to recover their services as below:

RR1. $U_i \Longrightarrow$ CloCen: $\{ID_i, RPW_i, revoke-request\}$, where $Gen(Bio_i) = (\delta_i, \tau_i)$, $RPW_i = h(PW_i||\delta_i||a)$.

RR2. CloCen $\Longrightarrow U_i$: $\{A_i^{new}, B_i^{new}, Y\}$.
On receiving the request, CloCen seeks ID_i from the database. If CloCen does not find such an ID_i, the request is rejected. Otherwise, CloCen picks the timestamps $T_{rg_i}^{new}$ and a random number a_i^{new}, computes $k_i = h(ID_i||y||T_{rg_i}^{new})$, $B_i^{new'} = h(RPW_i \ ||ID_i) \oplus k_i^{new}$, stores $\{ID_i, a_i^{new}, T_{rg_i}^{new}$,Honey-list=0$\}$ for U_i, and finally sends $\{B_i^{new'}, B_i^{new'} \oplus a_i, Y, P\}$ to U_i.

RR3. After obtaining the response from CloCen, the device chooses a random number a^{new}, follows the process of the registration phase to calculate, and finally stores $\{A_i^{new}, B_i^{new}, a, A_i^{new} \oplus a_i^{new}, \tau_i, P, Y\}$.

VI. Security Analysis

We show the security of the proposed scheme via provable security analysis, heuristic analysis and BAN logic and the ProVerif tool. These methods have different focuses and limitations, each serving a specific purpose in demonstrating the security of schemes [6], [25], [37], [40], [41], [42].

In general, BAN logic [43] is primarily concerned with the beliefs of principals. Through BAN logic analysis, the differences in security assumptions and design ideas between the two protocols can be clearly compared [43]. However, the security goals of BAN logic analysis are usually to prove the authenticity of the participants and the security of keys shared between them. Furthermore, BAN logic has been recognized by many researchers that there are limitations to its power [44], [45], such as the inability to express certain events.

The ProVerif tool is a mature formal security verification tool to analyze the security of cryptographic schemes [27], [46], [47], [48]. Usually, the researchers apply this tool to prove the three properties of multi-factor authentication schemes: the adversary cannot compromise users' passwords, the security of session keys, or the authentication between the participants. To the best of our knowledge, the ProVerif tool currently does not analyze other properties, such as resistance to node capture attacks. Moreover, the adversary model of multi-factor authentication schemes is not fully characterized in the ProVerif framework at present.

Provable security analysis is based on computational models and has become an indispensable tool in analyzing and evaluating new cryptographic schemes [37], [49], [50]. However, according to Wang et al. [32], provable security analysis fails to capture some complex real-world attack scenarios and security properties.

Heuristic analysis is a crucial way to evaluate the security of a scheme [12], [13], [49]. In heuristic analysis, the adversary capabilities and various security properties are fully considered. However, it heavily relies on the experience of the analyst and there is a risk of oversight in the analysis process, leading to incorrect analysis outcomes.

In conclusion, BAN logic, the ProVerif tool, and provable security analysis are all formal analysis methods. They can efficiently avoid analysis errors caused by human factors in the heuristic analysis. However, they have some limitations in characterizing the security properties and adversary capabilities of multi-factor authentication protocols, but the heuristic analysis can ideally make up for this deficiency. From the perspective of the specific security goals of user authentication schemes, researchers [13], [25], [37] usually analyze the authentication and the security of the session key through BAN logic, the ProVerif tools and provable security analysis, corresponding to the ideal attribute "D5" and the security requirement "S3" in Table III; meanwhile, use heuristic analysis to comprehensively analyze all security requirements in Table III.

It is well known that the design of security protocols is notoriously hard. Therefore, we use these four methods to assess our scheme in the hope that this will thoroughly and accurately examine the security of our protocol. Our scheme

TABLE IV
PLAYERS IN A FOUR-PARTY AUTHENTICATION PROTOCOL

Players	Attributes
$U \in$ **User**	Having personal information $\{PW_i, ID_i, Bio_i\}$ and a smart mobile device SD storing $\{A_i, B_i, a, A_i \oplus a_i, \tau_i, Y, P, n_0\}$ that supports cryptographic operations and biometric inputting,
CloCen $\in$ **Cloud Center**	Having a pair of long-term secret key $\{x, y\}$ with l_s bits length, a user-related table $\{ID_i, T_{rg_i}, a_i\}$, a gateway related table $\{GID_k, SID_j\}$.
GWN $\in$ **Gateway**	Having a secret pair of key $\{x_{G_k}, X_{G_k}\}$, where X_{G_k} is generated by the cloud center.
$S \in$ **IoT device**	Having a secret key $\{X_{S_j}\}$ (generated by GWN).

is proven secure under these four security analysis methods. It provides us with an adequate level of confidence about the security of our protocol. Due to the layout constraints, we present the provable security analysis and heuristic analysis in this paper. For further details on the ProVerif analysis and BAN logic analysis, please refer to the complete paper at https://bit.ly/3RzdctY.

A. Provable Security Analysis

This section formally analyzes the proposed scheme under the random oracle model. Specifically, we extend the BPR00 model [51] from the following two aspects: according to [37], we extend the $Corrupt()$ query to capture smart-devices-loss attacks; like [12], [25], [28], we build a multi-party password authentication model based on [50].

A-I Security Model

As shown below, the adversary's capabilities and behaviour can be modeled via a series of notations and queries.

Players. In a four-party protocol $\mathcal{P}$, four participants, namely users, cloud centers, gateways and IoT devices, are involved. Their attributes are shown in Table IV. In the execution of the protocol, U, CloCen, GWN and S are instantiated as U_i, CloCen_m, GWN and S_j respectively. Let I be the set of instances, I^s be the s-th instance of I.

Queries. We define the queries that depict the adversary's behaviors in real attacks as below:

- $Execute(U_i^r, \text{CloCen}, \text{GWN}_k^s, S_j^t)$: it models the entire protocol flows, and outputs the messages transmitted among the participants $\{U_i, \text{CloCen}, \text{GWN}_k, S_j\}$.
- $Send(I, I^s, m)$: it models an active attack where I sends the message m to I^s. If m is valid, it outputs the response from I^s. If not, this query is ignored.
- $Reveal(I^s)$: it models the leakage of session keys. If the session key has been built, it outputs the session key of I_s, otherwise outputs $\perp$.
- $Corrupt(U_i, a)$: it models the capability of $\mathcal{A}$ to corrupt U_i. a has three different values. It outputs any two of the three factors according to the value of a as below:
  - For a=1, output U_i's password and the data in SD;
  - For a=2, output U_i's biometrics and the data in SD;
  - For a=3, output U_i's biometrics and password.
- $Corrupt(I^s)$: it models the capability of $\mathcal{A}$ to corrupt the cloud center CloCen, the gateway GWN_k and the IoT device S_j. When I^s is instantiated to different objects, the output of the query is also different.
  - When $I^s == \text{CloCen}_m$, output the long-term secret keys $\{x, y\}$, $\{ID_i, T_{rg_i}, a_i\}$ and $\{GID_k, SID_j\}$.
  - When $I^s == \text{GWN}_k$, output $\{x_{G_k}, X_{G_k}\}$.
  - When $I^s == S_j$, output its secret key X_{S_j}.
- $Test(I^s)$: in this query, I^s can only be instantiated to U_i or S_j. This query is used to test session keys' semantic security. If I^s has not yet built a session key or the session key is not fresh, or $Test(I^s)$ has been queried before, it outputs $\perp$; otherwise, the simulator flips a coin b. If $b == 1$, return the session key; if $b == 0$, return a random number with the same length of the session key.

Partnering. Let sid and pid be the identifier of the session and its partner respectively. Then we say two instances U_i^s and S_j^r are partnered when: 1) they have accepted; 2) they share the same sid; 3) U_i^s's pid is S_j^r, S_j^r's pid is U_i^s.

Freshness. It is an essential notion in defining protocol security. It constrains the adversary's capability to get a session key. We say that instance I is freshness when: 1) I has accepted and built a session key; 2) both I and its partner have not been asked for a $Reveal$-query; 3) $Corrupt(U_i, a)$-query is asked at most one time.

Semantic Security. This notion defines the security of session keys. The adversary always tries to break the semantic security of the protocol $\mathcal{P}$. When evaluating the semantic security of $\mathcal{P}$, $\mathcal{A}$ is allowed to ask $Execute$-query, $Send$-query, and $Reveal$-query within a polynomial number, and the $Test$-query to a fresh instance. With the information from these queries, $\mathcal{A}$ attempts to output a guessed bit b' for b in $Test$-query. Let $Succ$ be the event that the adversary $\mathcal{A}$ guesses b correctly. Then the advantage of $\mathcal{A}$ in breaking the semantic security of $\mathcal{P}$ is:

$$Adv_{\mathcal{P},\mathcal{D}}^{ake}\mathcal{A} = 2Pr[Succ\mathcal{A}] - 1. \tag{1}$$

A desirable three-factor user authentication scheme should make online password guessing attacks be adversaries' best way to compute the session keys between users and IoT devices. Therefore, concerning a maximum of q_s times $Send$-query that the adversary asks in any period of polynomial time, we say a protocol $\mathcal{P}$ is semantically secure, when there is a negligible function $\varepsilon(\cdot)$ such that:

$$Adv_{\mathcal{P},\mathcal{D}}^{ake}\mathcal{A} < C' q_{send}^{s'} + \varepsilon(\ell). \tag{2}$$

where l is a system security parameter, $\mathcal{D}$ is the password space whose frequency distribution satisfies a Zipf's law [31], C' and s' are Zipf parameters.

Elliptic Curve Gap Diffie-Hellman (ECGDH) problem: given $(a \cdot P, b \cdot P)$, a and b in $\mathcal{G}$, the advantage for $\mathcal{A}$ to compute $ab \cdot P$ in the polynomial time t is: $Adv^{ECGDH}(t) \leq \varepsilon$.

Theorem 1: Protocol $\mathcal{P}$ is built on a q-order subgroup P on an elliptic curve $\mathcal{E}/\mathcal{F}_p$ over the finite field $\mathcal{F}_p$, p and q are two large primes, and $|q| = l$. $\mathcal{D}$ is a password space following Zipf's law [31]. Then a probabilistic polynomial time adversary against the semantic security of $\mathcal{P}$, making q_s $Send$-query, q_e $Execute$-query and q_h $Hash$-query within time t, have:

$$Adv_{\mathcal{P}}^{ake}(\mathcal{A}) \leq C' q_s^{s'} + \frac{2q_h^2 + 3q_s^2 + 3(2q_h + q_s)^2}{2^l} + \frac{(q_s + q_e)^2}{2(p-1)} + 2q_h((q_s + q_e)^2 + 1)Adv_{\mathcal{A}(t')}, \tag{3}$$

where C' and s' are the Zipf parameter [31], T_m is time for scaler multiplication in $\mathcal{G}$, and $t' \leq t + (2q_s + 6q_e + 1) \cdot T_m$.

A-II Security Proof

Theorem 1 is proved via a sequence of games which model the attack processes of the adversary from a real attack game G_0 to game G_6. The adversary's advantage among these games is gradually decreasing to zero.

Game G_0: G_0 models the real scheme in the random oracles, thus we have

$$Adv_{\mathcal{P}}^{ake} A = 2Pr[Succ_0] - 1 \tag{4}$$

where $Succ_n$ denotes the event that $\mathcal{A}$ in G_n guesses b in $Test$-query correctly.

Game G_1: This game simulates hash oracles $\mathcal{H}$ and creates five lists: $\Lambda_{\mathcal{H}}$ which records the inputs and outputs of hash-query; $\Lambda_{\mathcal{M}}$ which keeps the inputs and outputs of $Execute$-query; $\Lambda_{\mathcal{H}^{\mathcal{A}}}$ which keeps hash-queries asked by the adversary $\mathcal{A}$. In this game, the protocol is conducted as Section V. Then the adversary $\mathcal{A}$ intercepts the messages among the four participants via $Execute$-query, and finally executes $Test$-query to guess b. Obviously, $\mathcal{A}$ armed with the intercepted messages cannot compute the session key ($SK = h(M_2||M_{11}||M)$) between the user and IoT device. Thus compared with G_0, the advantage of $\mathcal{A}$ dose not increase:

$$|Pr[Succ_1] - Pr[Succ_0]| = 0 \tag{5}$$

Game G_2: In this game, the adversary can actively join the conversation via executing $Send$-query and $Hash$-query to construct a forged message that can be accepted. Only when the adversary finds the collisions to make valid messages, the semantic security of the protocol is compromised. In our scheme, there are two kinds of collisions:

- The collisions of the outputs of the hash function, and the probability of it is at most $\frac{q_h^2}{2^{l+1}}$;
- The collisions of random numbers, and the probability of it is at most $\frac{(q_s+q_e)^2}{2(p-1)}$.

Therefore, game G_1 and game G_0 are indistinguishable unless the above collisions occur, we have:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2(p-1)} \tag{6}$$

Game G_3: In this game, $\mathcal{A}$ attempts to guess some parameters to fake messages that can be accepted as below:

- The adversary constructs Msg_1 successfully. In this case, $\mathcal{A}$ needs to ask hash-query to compute Msg_1, thus we have $(M_2||M_1, *)$, $(M_1||M_2||M_3, *)$, $(*||ID_i||M_1||M_2||SID_j, M_5) \in \Lambda_{\mathcal{H}^{\mathcal{A}}}$, and the probability of this event is: $\frac{(2q_h+q_s)^2}{2^l}$;
- The adversary constructs Msg_2 successfully, then similarly, we have $(*||M_2, *)$, $(M_6||r||*, *)$, $(M_2||M_6||M_7||r||SID_j||*, M_8) \in \Lambda_{\mathcal{H}^{\mathcal{A}}}$, and the probability of this event is: $\frac{(2q_h+q_s)^2}{2^l}$;
- The adversary constructs Msg_3 successfully, then we have $(SID_j||*, *)$, $(*||M_2, *)$, $(M_2||M_9 \ ||r_g||SID_j||*, M_{10}) \in \Lambda_{\mathcal{H}^{\mathcal{A}}}$, and the probability of this event is: $\frac{(2q_h+q_s)^2}{2^l}$;
- The adversary constructs Msg_4 successfully, then we have $(M_2||M_{11}||r_g|| * ||SID_j, M_{12}) \in \Lambda_{\mathcal{H}^{\mathcal{A}}}$, and the probability of this is: $\frac{q_s^2}{2^l}$;
- The adversary constructs Msg_5 successfully, then we have $(M_{11}||M_2||SID_j||r||*, M_{13}) \in \Lambda_{\mathcal{H}^{\mathcal{A}}}$, and the probability of this event is: $\frac{q_s^2}{2^l}$;
- The adversary constructs Msg_6 successfully, then we have $(M_1||M_2||ID_i||SID_j|| * ||M_{11}, M_{14}) \in \Lambda_{\mathcal{H}^{\mathcal{A}}}$, and the probability of this event is: $\frac{q_s^2}{2^l}$;

G_2 and G_3 are indistinguishable unless $\mathcal{A}$ successfully constructs above messages, thus we have:

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{3(2q_h + q_s)^2 + 3q_s^2}{2^{l_s}} \tag{7}$$

Game G_4: This game models the corruption capability of the adversary, thus $\mathcal{A}$ can execute $Corrupt(U_i, a)$-query (where $a = 1, 2, 3$) as follows:

- $\mathcal{A}$ queries $Corrupt(U_i, 1)$ to get the victim's password and data in the device. In this case, $\mathcal{A}$ then tries to get the information of the victim's biometric data in two ways: 1) guess δ_i with l_r bits in q_s queries, and its probability is $\frac{q_s}{2^{l_r}}$; 2) use collected biometric data to replace the victim's, and its probability is $q_s \cdot \varepsilon_b$ (ε_b is the probability that two persons' biometric data are similar, and it is negligible). Therefore, the probability of the attack is $\frac{q_s}{2^{l_r}} + q_s \cdot \varepsilon_b$.
- $\mathcal{A}$ queries $Corrupt(U_i, 2)$ to get the victim's biometrics and the data in the device. Then $\mathcal{A}$ tries to guess the victim's password correctly in q_s queries. The probability of this event is $C' q_s^{s'}$ as the distribution of passwords follows Zip'f law, C' and s' are Zip'f parameters [31].
- $\mathcal{A}$ queries $Corrupt(U_i, 3)$ to get the victim's password and biometric. Then $\mathcal{A}$ tries to guess B_i correctly in q_s queries, and the probability of this is: $\frac{q_s}{2^{l_s}}$.

Game G_4 and G_3 are indistinguishable unless $\mathcal{A}$ successfully gets the above parameters, thus:

$$\begin{aligned} |Pr[Succ_4] - Pr[Succ_3]| &\leq max\{\frac{q_s}{2^{l_r}} + q_s \varepsilon_b, C' q_s^{s'}, \frac{q_s}{2^{l_s}}\} \\ &= C' q_s^{s'} \end{aligned} \tag{8}$$

Game G_5: The adversary in this game attempts to compute the session key, as well as solve the ECGDH problem. The probability of picking $ECGDHP(r_i P, r_j P)$ in $\Lambda_{\mathcal{H}^{\mathcal{A}}}$ is $\frac{1}{q_h}$, then we have:

$$Pr[AskH_5] = Pr[AskH_5^0] \leq 2q_h \cdot Adv_p^{ECGDH}(t') \tag{9}$$

where $t' \leq t + (2q_s + 6q_e + 1) \cdot T_m$ and T_m is the time of running a point multiplication.

Game G_6: In this game, forward secrecy is considered. Note that, the adversary in this game can only ask $Send(\cdot)$-query before $Corrupt(\cdot)$-query. The probability of $r_i P$ and $r_j P$ in a session is $\frac{1}{(q_s+q_e)^2}$, then we have:

$$\begin{aligned} Pr[AskH_5] &= Pr[AskH_5^0] \\ &\leq 2q_h(q_s + q_e)^2 \cdot Adv_p^{ECGDH}(t') \end{aligned} \tag{10}$$

where $t' \leq t + (2q_s + 6q_e + 1) \cdot T_m$.

Till now, the advantage of $\mathcal{A}$ to compute session key is zero, thus $Pr[Succ_6] = \frac{1}{2}$. According to Game $G_0 \sim G_6$, we have:

$$\begin{aligned} Adv_{\mathcal{P}}^{(ake)}(\mathcal{A}) &= 2Pr[Succ_0] - 1 \\ &= 2Pr[Succ_5] - 1 + 2(Pr[Succ_0] - Pr[Succ_5]) \\ &\leq C' q_s^{s'} + \frac{2q_h^2 + 3q_s^2 + 3(2q_h + q_s)^2}{2^l} \\ &\quad + \frac{(q_s + q_e)^2}{2(p-1)} + 2q_h((q_s + q_e)^2 + 1) Adv_{\mathcal{A}(t')} \end{aligned}$$

B. Heuristic Analysis

In this section, we provide a heuristic analysis of our scheme from the perspective of a real adversary.

Proposition: User Anonymity. The proposed scheme prevents a user's identity from being computed and tracked.

Proof: For identity protection, we transmit the identity ID_i in the form of $h(M_2||M_1) \oplus ID_i$, where M_1 is only known to the user and the cloud center. Thus, no one except the user and the cloud center can compute ID_i. For user untraceability, all of the parameters transmitted in the open channel change dynamically with the random numbers chosen by the four participants. Therefore, our scheme achieves user anonymity.

Proposition: Forward Secrecy. The compromise of the entire system will not affect the previous sessions.

Proof: Consider that the long-term secret keys x and y are exposed: the adversary eavesdrops on the parameters M_2 and M_{11} that consists of the session key. According to our scheme, the session key is computed as $SK = h(M_2||M_{11}||r_i M_{11})$, thus the adversary still needs to obtain the parameter $M = r_j M_2 = r_i M_{11}$. Note that r_j and r_i are not transmitted in the open channel and are only known to the IoT device and the user, respectively. Therefore, the adversary can only directly compute the value of M using M_2 and M_{11}. That is, the adversary has to solve the elliptic curve computational Diffie-Hellman (ECCDH) problem. Since the ECCDH problem cannot be solved within polynomial time, the adversary cannot compute M. Thus, our scheme achieves forward secrecy.

Proposition: No password Exposure. No Password Exposure, i.e., no privileged insider attacks. In the proposed scheme, the legitimate cloud center administrator gains no advantage in attacking the security of the scheme.

Proof: To achieve this goal, we update the parameter RPW_i to be RPW_i^{new} after the user gets the response from the cloud center in the registration phase. In this way, the administrator of the cloud center cannot obtain a verification parameter to initial an offline dictionary attack. Please refer to Section V for detailed analysis of this attribute.

Proposition: Resistance to Smart Devices Loss Attacks. In the proposed scheme, an adversary cannot conduct an attack using the parameters from a smart mobile device. Also, the proposed scheme achieves multi-factor security. That is, even if any two of the factors are compromised, the security of the scheme could still be promised. (It is obvious that the adversary armed with PW_i and BIO_i cannot get the data in the smart devices. Thus we discuss the smart devices loss attacks to analyze the multi-factor security.)

Proof: In our scheme, if the adversary acquires $\{A_i, B_i, a, A_i \oplus a_i, \tau_i, Y, h(\cdot), Rep(\cdot)\}$ in the smart mobile device or wants to change the password without being noticed by the smart mobile device, he has to construct correct $A_i = h(ID_i ||RPW_i||k_i) \bmod n_0$ to pass the verification of the smart mobile device. Since the knowledge of $\{A_i, B_i, a, A_i \oplus a_i, \tau_i, Y, n_0, h(\cdot), Rep(\cdot)\}$ does not help to compute A_i, the adversary cannot change the password.

However, if the adversary wants to guess the password correctly, he may use either A_i or M_5 as the verification parameter to test the correctness of the guessed password.

For A_i, even if an adversary with a biometric finds such a pair of password and identity that satisfies $h(ID_i^*|| RPW_i^*||k_i) \bmod n_0 == A_i$, he still is not sure whether the password is correct, for there are $|\mathcal{D}_{pw}| * |\mathcal{D}_{id}|/n_0 \approx 2^{32}$ candidates of $\{ID_i, PW_i\}$ pair when $n_0 = 2^8$ and $|\mathcal{D}_{pw}| = |\mathcal{D}_{id}| = 10^6$ according to Wang et al. [37]. To further determine the correctness of the guessed password, the adversary has to conduct an online verification, which will be prevented by the Honey-list of our scheme.

For M_5, as previously explained, M_5 consists of a preset secret shared parameter k_i and a dynamical M_1. k_i can be deduced from the user's password and biometrics, M_1 is only known to the real user who selects r_i and the cloud center who knows y. This means that the adversary can "compute" k_i with the guessed password but cannot "compute" M_1. Therefore, the adversary fails to construct a M_5^*, so he cannot verify the correctness of the guessed password by comparing the value of M_5 and M_5^*. In conclusion, our scheme is secure against such an attack.

Proposition: Resistance to Impersonation Attacks.

Proof: Firstly, we consider user impersonation attacks where the adversary does not acquire data from the victim's smart device. Note that, to impersonate U_i without the parameters in the smart device, the adversary can only try to construct such a valid access request $\{M_2, M_3, M_4, M_5\}$ directly, where M_5 consists of k_i. Since k_i can only be computed via user-sensitive information, such as passwords, biometrics, and parameters in smart mobile devices, or the long-term secret y and verifier table, all these parameters cannot be obtained by the adversary, that is the adversary cannot impersonate U_i.

Next, we discuss the cloud center impersonation attacks. Following our scheme, both the user and the gateway authenticate the cloud center via M_{14} and M_8. Thus, to impersonate the cloud center, the adversary has to compute M_{14} and M_8 correctly. However, to compute these two parameters, the adversary has to know k_i and X_{G_k} simultaneously. Since the two parameters are not transmitted directly or with "$\oplus$" operation in the open channel, the adversary cannot obtain these parameters if he is not a legitimate participant, i.e., the adversary cannot impersonate the cloud center.

Similarly to our analysis above, the gateway and IoT device authenticate each other with X_{S_j}, and X_{S_j} is not transmitted directly or with the "$\oplus$" operation in an open channel. Thus, the adversary cannot obtain X_{S_j} unless he captures the IoT device. However, when the adversary has captured the device, it is no longer appropriate to consider IoT device impersonation attacks anymore. Thus, the adversary cannot impersonate

TABLE V
PERFORMANCE COMPARISON AMONG RELEVANT AUTHENTICATION SCHEMES

Scheme	Ref.	Evaluation Criteria											
		D1	D2	D3	D4	D5	D6	S1	S2	S3	S4	S5	S6
Yang et al.	[28]	×	√	√	√	√	√	√	√	×	√	×	×
Amin et al.	[6]	√	√	√	√	√	√	×	√	×	√	×	×
Wazid et al.	[13]	√	√	√	×	√	√	√	√	×	×	×	×
Wazid et al.	[26]	√	√	√	×	√	√	×	×	×	√	×	×
Sharif et al.	[52]	√	√	√	×	√	√	×	√	×	×	×	×
Das et al.	[16]	√	×	√	×	√	√	√	√	×	×	×	×
Srinivas et al.	[12]	√	√	√	×	√	√	×	√	√	√	×	×
Srinivas et al.	[53]	√	√	√	×	√	√	√	×	√	√	√	√
Li et al.	[25]	√	√	√	√	√	√	√	×	√	√	×	×
Jiang et al.	[7]	√	√	√	√	√	√	√	√	×	√	√	×
ours	-	√	√	√	√	√	√	√	√	√	√	√	√

an IoT device. In addition, the cloud center authenticates the gateway via X_{G_k}, and X_{G_k} cannot be acquired by the adversary as mentioned above. That is $\mathcal{A}$ cannot impersonate the gateway.

In short, our scheme is robust against impersonation attacks.

Proposition: Resistance to Resistant to De-synchronization Attacks.

Proof: We use the random number and the public key algorithm to achieve user anonymity and prevent replay attacks. The participants are not required to keep the consistency of the clock synchronized or some temporary certificate-related parameters. Therefore, our scheme can withstand de-synchronization attacks.

Proposition: Mutual Authentication. Each participant of the proposed scheme verifies the other one's identity.

Proof: The cloud center authenticates the user through $M_5 = h(k_i||ID_i||M_1||M_2||\ M_3||M_4)$, where k_i is their preset fixed shared secret and (M_1, M_2) is a pair of ciphertext and plaintext of public-key algorithm. k_i and M_1 are only known to the user and the cloud center, thus the authentication is effective. Similarly, the user authenticates the cloud center with the same key parameters; then the cloud center is authenticated by the gateway via M_8, which consists of their shared secret key X_{G_k}. The IoT device and the gateway authenticate each other via M_{12} and M_{10}, respectively. In consequence, the proposed scheme achieves mutual authentication.

VII. PERFORMANCE ANALYSIS

In this section, we compare our scheme with ten relevant state-of-the-art multi-factor user authentication schemes for cloud computing and IoT environment under the adversary model defined in Section III-A in security and performance, as shown in Table V, Table VI and Table VII. Note that, in order to show the comparison results more intuitively, we further present some important data in Table VI and Table VII in the form of two bar charts: Fig. 4 shows the computation costs on the user, gateway and IoT device side, and Fig. 5 shows their communication costs. We let T_C, T_P, T_B, T_H and T_S, T_e denote the operation time for the chebyshev chaotic-map, elliptic curve point multiplication, fuzzy extracting biometric data, hash, and symmetric encryption, bilinear pairing operation, respectively. According to [12] and [23], $T_H \approx 0.003$ms, $T_S \approx 0.02$ms, $T_C \approx T_P \approx T_B \approx 0.294$ms. These results are tested on Intel i7-4710HQ, 2.5GHZ CPU and 8G memory. Besides, according to [25] and [37], the identities, the tolerance error

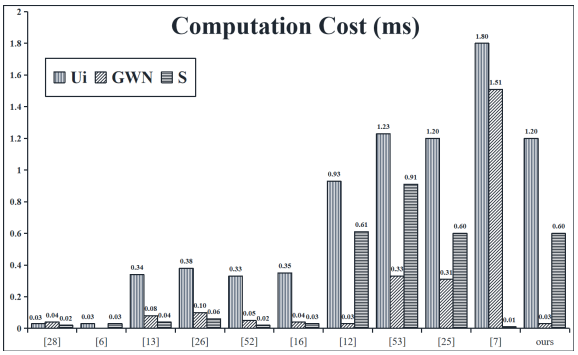

Fig. 4. Computation cost.

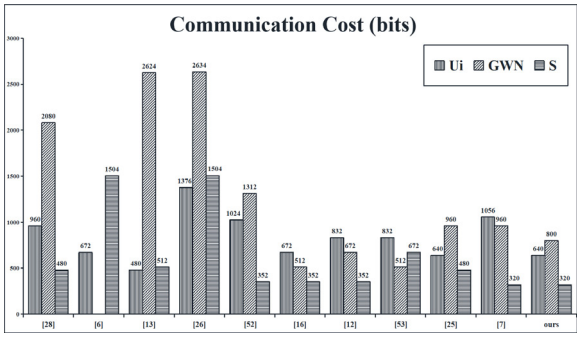

Fig. 5. Communication cost.

value and the public reproduction parameters are set to 128bits. ECC point, random numbers, and hash function outputs are set to 160bits. "n_0", the timestamp and the counter are 32bits. As the efficiency of an authentication scheme mainly depends on the cost of the login and authentication phase, we neglect the cost of the registration phase.

From a security point of view, as shown in Table V, our scheme is the only one that meets all twelve evaluation criteria. Most of the compared schemes cannot resist "S6 node capture attacks" (Only Srinivas et al.'s scheme [53] and ours meet this attribute). Besides, only three schemes (Jiang et al. [7], Srinivas et al. [53] and ours) are secure against "S5 smart mobile device loss attacks". And only the schemes of Srinivas et al. [12], Srinivas et al. [53], Li et al. [25] and ours provide "S3 forward secrecy".

It is worth noting that in Table V, the first six schemes only use symmetric algorithms, while the latter five schemes are based on asymmetric algorithms. It can be seen that the latter five schemes can generally meet more security goals: the first six schemes can only achieve seven security requirements on average; while the latter five schemes are able to achieve ten security requirements on average. In addition, as shown in Table V, the first six schemes based on symmetric algorithms are difficult to achieve "S5 Resistance to smart mobile device loss attacks" and "S6 Resistance to node capture attacks". Among the latter five asymmetric algorithms-based schemes, three schemes can realize "S5", and two schemes can realize "S6". From the comparison results, the schemes based on asymmetric algorithms can achieve better security than the schemes based on the symmetric cryptography algorithm. In short, the asymmetric-algorithm-based schemes achieve better security than symmetric-algorithm-based schemes, and the proposed scheme performs best in terms of security.

TABLE VI

COMPUTATION COST COMPARISON AMONG RELEVANT AUTHENTICATION SCHEMES

Scheme	Ref.	R.	Computational Cost (ms)			
			U_i	GWN	S	CloCen
Yang et al.	[28]	6	$8T_H$ ≈0.03	$14T_H$ ≈0.04	$7T_H$ ≈0.02	NI
Amin et al.	[6]	4	$9T_H$ ≈0.03	NI	$10T_H$ ≈0.03	$4T_H$ ≈0.01
Wazid et al.	[13]	4	$9T_H + T_S + T_B$ ≈0.34	$11T_H$+$2T_S$ ≈0.08	$7T_H + T_S$ ≈0.04	NI
Wazid et al.	[26]	4	$13T_H + 2T_S + T_B$ ≈0.38	$5T_H + 4T_S$ ≈0.10	$4T_H + 2T_S$ ≈0.06	NI
Sharif et al.	[52]	6	$11T_H + T_B$ ≈0.33	$17T_H$ ≈0.05	$5T_H$ ≈0.02	NI
Das et al.	[16]	3	$18T_H + T_B$ ≈0.35	$12T_H$ ≈0.04	$9T_H$ ≈0.03	NI
Srinivas et al.	[12]	3	$2T_C$+$15T_H$+T_B ≈0.93	$10T_H$ ≈0.03	$2T_C$+$6T_H$ ≈0.61	NI
Srinivas et al.	[53]	3	$3T_p$+$16T_H$+T_B ≈1.23	T_p+$11T_H$ ≈0.33	$3T_p$+$7T_H$ ≈0.91	NI
Li et al.	[25]	4	$3T_P$+T_B+$7T_H$ ≈1.20	T_P+$6T_H$ ≈0.31	$2T_P$+$4T_H$ ≈0.60	NI
Jiang et al.	[7]	4	$6T_P + 11T_H$ ≈1.80	$5T_P + 11T_H$ ≈1.51	$4T_H$ ≈0.01	NI
ours	-	6	$3T_P$+$8T_H$+T_B ≈1.20	$9T_H$ ≈0.03	$2T_P$+$4T_H$ ≈0.60	T_P+$10T_H$ ≈0.33

"R." denotes the number of the communication round.
"NI" denotes that the corresponding party is not involved.

TABLE VII

COMMUNICATION AND STORAGE COST COMPARISON AMONG RELEVANT AUTHENTICATION SCHEMES

Scheme	Ref.	Communication Cost (bits)				Storage Cost (bits)			
		U_i	GWN	S	CloCen	U_i	GWN	S	CloCen
Yang et al.	[28]	960	2080	480	NI	480	$320n_u + 338n_s$	288	NI
Amin et al.	[6]	672	NI	1504	640	640	NI	320	320
Wazid et al.	[13]	480	2624	512	NI	864	$960n_u$+$288n_s$	$704n_u$+$288n_s$	NI
Wazid et al.	[26]	1376	2624	1504	NI	1088	736	288	NI
Sharif et al.	[52]	1024	1312	352	NI	608	$288n_u + 128n_s$	288	NI
Das et al.	[16]	672	512	352	NI	1184	$256n_u + 736n_s$	704	NI
Srinivas et al.	[12]	832	672	352	NI	736	160+$160n_u$+$288n_s$	288	NI
Srinivas et al.	[53]	832	512	672	NI	1056	800+$128n_s$	288	NI
Li et al.	[25]	640	960	480	NI	768	$128n_s$	288	NI
Jiang et al.	[7]	1056	960	320	NI	960	$544n_u + 288n_s$	160	NI
ours	-	640	800	320	960	1120	320	160	320+$160n_u$+$128(n_g$+$n_s)$

Here, we do not additionally evaluate the storage costs for the following functions stored in the smart card: hash function, biometrics operations for Gen(·), Rep(·); "n_u", "n_g"and "n_s" denote the number of users, gateway node and device nodes, respectively;

From a computation cost point of view, as shown in Table VI, our scheme is also competitive. Firstly, among the compared schemes, three schemes' communication rounds are six, including Yang et al.'s scheme [28], Sharif et al.'s scheme [52] and the proposed scheme, which is the biggest among the eleven schemes. However, as shown in Table VI, our scheme involves four parties in the authentication phase, and the other schemes only involve three parties. Thus, in this premise, compared with the schemes having six communication rounds (Yang et al. [28] and Sharif et al. [52]), our scheme with four parties involved is competitive. Compared with the rest schemes with three parties involved in the authentication phase [6], [7], [12], [13], [16], [25], [26], [53], our scheme with four parties is acceptable.

Secondly, as shown in Fig. 4 and Table VI, the computation cost on each side of our scheme is still competitive. Especially, we achieve the minimum computation cost on the gateway node. Note that, in Table VI, the first six schemes are based on symmetric cryptographic algorithms; the latter five schemes are based on asymmetric cryptographic algorithms. As the security analysis above, the schemes based on asymmetric cryptographic algorithms achieve better security than the schemes based on the symmetric cryptography algorithm. As we all know, the computation complexity of asymmetric cryptographic algorithms is inherently greater than that of symmetric cryptographic algorithms. Besides, the user authentication schemes that set biometrics as one of the factors are bound to increase the time spent on the user side, while it improves security. Based on the above two points, our computation cost on the user side (1.2ms) is acceptable among these three-factor schemes using asymmetric cryptographic algorithms (0.93ms [12], 1.23ms [53], 1.20ms [25]), and better than Jiang et al.'s scheme [7] (1.8ms). In fact, the difference in time spent by these schemes is imperceptible to the user. Thus, our computation cost on the user side is acceptable and has little impact on the user experience.

Furthermore, our computation cost on IoT devices node (0.60ms) is also advantageous among these schemes using asymmetric cryptographic algorithms, whose computation costs on the IoT devices node are 0.61ms, 0.91ms, 0.60ms and 0.01ms respectively. Note that, as suggested by Ma et al. [39], there are two requirements to achieve "S3 forward secrecy": a public-key algorithm and at least two module exponentiation or point multiplication operations on the server side. In an IoT network, the server side refers to the IoT devices. Thus, from this design principle, if forward secrecy is to be achieved, at least two public-key cryptographic operations on the IoT device side are unavoidable. This conclusion is also consistent with the security analysis results of the schemes compared in Table V: among the schemes that are based on asymmetric cryptographic algorithms, only the scheme of Jiang et al. [7] does not deploy public-key operations on the IoT side, and thus the computation cost on the IoT side is only 0.01ms. Unsurprisingly, their scheme naturally fails to achieve forward secrecy; similarly, all schemes that are based on symmetric cryptographic algorithms cannot achieve forward secrecy. In summary, the computation cost of our scheme on the IoT device side is reasonable and competitive, achieving a better balance between security and performance.

For the communication cost and storage cost comparisons, as shown in Fig. 5 and Table VII, our scheme has the minimum storage cost on the gateway side (320bits) with competitive storage costs on the IoT device and the user side. Besides, our communication cost on the gateway side is 800 bits which ranks fourth among the compared schemes, and the top

three are Das et al.'s scheme [16] (512 bits), Srinivas et al.' scheme [53] (512 bits) and Srinivas et al.' scheme [12] (672 bits). Our communication costs on the IoT device and the user side rank second and first respectively. It can be seen that we move a large portion of storage costs to the cloud center to cut down the cost of the gateway.

In conclusion, the proposed scheme satisfies all twelve evaluation criteria. It provides the best security guarantee among all compared ten schemes. Besides, our scheme achieves the minimum computation and storage costs on the gateway side by utilizing the computing resource of the cloud center with acceptable computation costs on the user side and IoT devices. Thus, the proposed scheme is especially suitable for cloud-assisted IoT applications with massive IoT devices.

VIII. Conclusion

This paper designs a secure user authentication scheme for cloud-assisted IoT systems with lightweight computation on gateways. Firstly, we consider Wazid et al.'s scheme as a study case and identify the security weaknesses and unreasonableness of such schemes. Then, we propose a new secure user authentication scheme for cloud-assisted IoT environments and prove its security using provable security analysis, the Proverif tool, heuristic analysis, and BAN logic. In addition, we improve the efficiency of the proposed scheme by moving heavy computation and storage tasks to the cloud center. Finally, we demonstrate the superiority of the proposed scheme by comparing it with ten state-of-the-art authentication schemes. The comparative results show that our scheme achieves the best security with the minimum computation and storage costs on the gateway side, making it well-suited for situations where the resources of the gateway are limited relative to the connection of a large number of IoT devices.

References

[1] S. Li, K.-K.-R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "IoT forensics: Amazon echo as a use case," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6487–6497, Aug. 2019.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] A. Botta, W. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[4] N. Alhakbani, M. M. Hassan, M. A. Hossain, and M. Alnuem, "A framework of adaptive interaction support in cloud-based Internet of Things (IoT) environment," in *Proc. IDCS*, vol. 8729, 2014, pp. 136–146.

[5] M. S. Hossain and G. Muhammad, "Cloud-assisted Industrial Internet of Things (IIoT)—Enabled framework for health monitoring," *Comput. Netw.*, vol. 101, pp. 192–202, Jun. 2016.

[6] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018.

[7] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K.-R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9390–9401, Sep. 2020.

[8] J. Song, Q. Zhong, W. Wang, C. Su, Z. Tan, and Y. Liu, "FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17430–17438, Aug. 2021.

[9] W. Wang et al., "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8883–8891, Oct. 2021.

[10] S. A. Chaudhry, A. Irshad, K. Yahya, N. Kumar, M. Alazab, and Y. B. Zikria, "Rotating behind privacy: An improved lightweight authentication scheme for cloud-based IoT environment," *ACM Trans. Internet Technol.*, vol. 21, no. 3, pp. 1–19, Aug. 2021.

[11] P. Bhuarya, P. Chandrakar, R. Ali, and A. Sharaff, "An enhanced authentication scheme for Internet of Things and cloud based on elliptic curve cryptography," *Int. J. Commun. Syst.*, vol. 34, no. 10, pp. 4834–4853, Jul. 2021.

[12] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1133–1146, Nov. 2020.

[13] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, Dec. 2020.

[14] G. Sharma and S. Kalra, "Advanced lightweight multi-factor remote user authentication scheme for cloud-IoT applications," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 4, pp. 1771–1794, Apr. 2020.

[15] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and T. Yi, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 106, pp. 117–123, Mar. 2018.

[16] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[17] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.

[18] C. Wang, D. Wang, H. Wang, G. Xu, J. Sun, and H. Wang, "Cloud-aided privacy preserving user authentication and key agreement protocol for Internet of Things," in *Proc. Int. Symp. Secur. Privacy Social Netw. Big Data*, 2019, pp. 95–109.

[19] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.

[20] 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.

[21] K. Mahmood, J. Ferzund, M. A. Saleem, S. Shamshad, A. K. Das, and Y. Park, "A provably secure mobile user authentication scheme for big data collection in IoT-enabled maritime intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2411–2421, Feb. 2023.

[22] F. G. Darbandeh and M. Safkhani, "SAPWSN: A secure authentication protocol for wireless sensor networks," *Comput. Netw.*, vol. 220, Jan. 2023, Art. no. 109469.

[23] F. Wu, X. Li, L. Xu, P. Vijayakumar, and N. Kumar, "A novel three-factor authentication protocol for wireless sensor networks with IoT notion," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1120–1129, Mar. 2021.

[24] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 942–956, Sep. 2020.

[25] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2017.

[26] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[27] F. Wu et al., "An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment," *J. Netw. Comput. Appl.*, vol. 89, pp. 72–85, Jul. 2017.

[28] Z. Yang, J. He, Y. Tian, and J. Zhou, "Faster authenticated key agreement with perfect forward secrecy for industrial Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6584–6596, Oct. 2020.

[29] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–207, Mar. 1983.

[30] S. A. Sheik and A. P. Muniyandi, "Secure authentication schemes in cloud computing with glimpse of artificial neural networks: A review," *Cyber Secur. Appl.*, vol. 1, Dec. 2023, Art. no. 100002.

[31] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[32] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.

[33] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Trans. Depend. Secur. Comput.*, vol. 19, no. 1, pp. 507–523, Jan./Feb. 2022.

[34] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, Nov. 2015.

[35] P. Bagga, A. K. Das, and J. P. C. Rodrigues, "Bilinear pairing-based access control and key agreement scheme for smart transportation," *Cyber Secur. Appl.*, vol. 1, Dec. 2023, Art. no. 100001.

[36] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *J. Supercomput.*, vol. 74, no. 12, pp. 6428–6453, Dec. 2018.

[37] 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./Aug. 2018.

[38] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[39] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2215–2227, Oct. 2014.

[40] D. S. Gupta, S. H. Islam, M. S. Obaidat, P. Vijayakumar, N. Kumar, and Y. Park, "A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments," *IEEE Syst. J.*, vol. 15, no. 2, pp. 1732–1741, Jun. 2021.

[41] X. Li, T. Liu, M. S. Obaidat, F. Wu, and P. Vijayakumar, "A lightweight privacy-preserving authentication protocol for VANETs," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3547–3557, May 2020.

[42] L. Zhang, J. Xu, M. S. Obaidat, X. Li, and P. Vijayakumar, "A PUF-based lightweight authentication and key agreement protocol for smart UAV networks," *IET Commun.*, vol. 16, no. 10, pp. 1142–1159, Jun. 2022.

[43] M. Burrows and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1989.

[44] C. Boyd and W. Mao, "On a limitation of BAN logic," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 765. Berlin, Germany: Springer, 1993, pp. 240–247.

[45] D. M. Nessett, "A critique of the Burrows, Abadi and Needham logic," *ACM SIGOPS Operating Syst. Rev.*, vol. 24, no. 2, pp. 35–38, Apr. 1990.

[46] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, Mar. 2020.

[47] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107333.

[48] S. F. Aghili, H. Mala, M. Shojafar, and P. Peris-Lopez, "LACO: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT," *Future Gener. Comput. Syst.*, vol. 96, pp. 410–424, Jul. 2019.

[49] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1382–1392, Jun. 2017.

[50] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptogr. Public Key Cryptogr.*, vol. 3386, Jan. 2005, pp. 65–84.

[51] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 1807. Bruges, Belgium: Springer, 2000, pp. 139–155.

[52] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, and D. Abbasinezhad-Mood, "Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," *Future Gener. Comput. Syst.*, vol. 100, pp. 882–892, Nov. 2019.

[53] J. Srinivas, A. K. Das, M. Wazid, and A. V. Vasilakos, "Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7727–7744, May 2021.

Chenyu Wang received the Ph.D. degree in information security from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. She is currently a Special Associate Research Fellow with BUPT. She has received the Cyber Security Scholarship, China. She has published more than 20 papers at venues, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING and WWW 2022. Her research interests include applied cryptography and password-based authentication.

Ding Wang received the Ph.D. degree in information security from Peking University in 2017. He is currently a Full Professor with Nankai University. As the first author (or corresponding author), he has published more than 80 papers at venues, such as IEEE SECURITY & PRIVACY, ACM CCS, NDSS, USENIX Security, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND 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. His research interests include passwords, authentication, and provable security. He has been involved in the community as the PC Chair/TPC Member for over 60 international conferences, such as ACM CCS 2022, NDSS 2023/2024, PETS 2022–2024, ACSAC 2020–2023, ACM AsiaCCS 2022/2021, ICICS 2018–2023, and SPNCE 2020–2022. He has received the ACM China Outstanding Doctoral Dissertation Award, the Best Paper Award at INSCRYPT 2018, the Outstanding Youth Award of the China Association for Cryptologic Research, and the First Prize of the Natural Science Award of the Ministry of Education.

Yihe Duan received the B.S. degree from the College of Cyber Science, Nankai University, Tianjin, China, in June 2022, where he is currently pursuing the M.S. degree. His research interests include applied cryptography and password-based authentication.

Xiaofeng Tao (Senior Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1993, and the M.S. and Ph.D. degrees in telecommunication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999 and 2002, respectively. He is currently a Professor with BUPT. He has authored or coauthored over 200 papers and three books in wireless communication areas. He focuses on 5G/B5G research. He is a fellow of the Institution of Engineering and Technology and the Chair of the IEEE ComSoc Beijing Chapter.