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

Chenyu Wang, Ding Wang, Yihe Duan, and Xiaofeng Tao

***Abstract*—The cloud-aided Internet of Things (IoT) overcomes the resource-constrained nature of the traditional IoT and develops rapidly in such fields as smart grid and intelligent transportation. In a cloud-aided IoT system, users can remotely control the IoT devices or send specific instructions to them. When the user's identity is not verified, and an adversary delivers malicious instructions to IoT devices, the system's security may be compromised. Besides, the real-time data stored in IoT devices can also be exposed to illegal users, causing security issues. Thus, the authentication mechanism is indispensable. Furthermore, with the exponential growth of interconnected devices, a gateway may connect to mass IoT devices. The efficiency of authentication schemes is easily affected by the computation power of the gateway. Although recent research has proposed many user authentication schemes for IoT, only a dozen schemes are designed for cloud-aided IoT. Therefore, we take a typical scheme (presented at IEEE TDSC 2020) as an example to capture user authentication schemes' common weaknesses and design challenges for cloud-aided IoT. Then, we propose a new secure user authentication scheme for cloud-aided IoT with lightweight computation on gateways. The proposed scheme provides secure access between the remote user 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. Finally, we compare the proposed scheme with eleven state-of-the-art schemes in security and performance. The results show that the proposed scheme achieves all listed twelve security requirements with minimum computation and storage costs on gateways.**

***Index Terms*—User authentication; Internet of Things; Cloud computing; Offline dictionary attack; Forward secrecy.**

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 even exceeds the number of users in 2011, and is projected at 24 billion by 2020 [3]. Such a large number of interconnected devices put great computation and storage pressure on IoT networks. Under this situation, numerous researchers [3]–[5] are pursuing the issue 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 to the real world by offering a mass of environment data. The integration of IoT and cloud computing maximizes mutual benefits [6]. It greatly improves applications in the smart city, the smart transportation, and the smart grid, where large quantities of data and numbers of devices are involved, and complex computation and analysis are required [3], [7]–[9].

However, the benefits of integrating cloud computing with IoT techniques are accompanied by new security challenges.

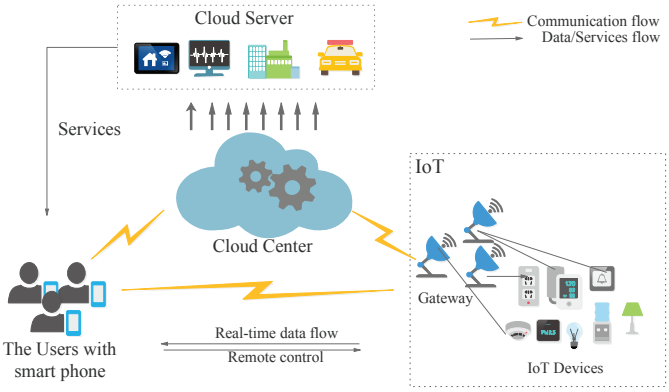

Fig. 1. Architecture of Cloud-aided Internet of Things.

Concerns have been expressed about the personal information being acquired in the cloud computing environment by adversaries with ulterior motives. Integrating cloud computing with IoT systems will make these concerns about privacy protection more prominent, because IoT networks bring real-world data to the cloud center, and cloud computing increases the number of actions that can be conducted in the real world. Therefore, it is significant to prevent unauthorized access to these sensitive data. As the first line of defense for system security, user authentication has received extensive attention. With a sound user authentication method, the access to the data can be controlled efficiently.

Figure 1 shows the architecture of a cloud-aided 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 gateways and the IoT devices consist of the IoT networks where quantities of IoT devices and a limited number of gateways are involved. The IoT devices collect real-time data from environments and send the data to their connected gateways. The gateways then upload the collected data onto the cloud center. Then the cloud center, armed with powerful computation and storage capabilities, processes the environment data, realizes intelligent services, and provides better services for users.

In a cloud-aided 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, it involves the users and cloud center; Auth-Sce II, where the users want to access the real-time data from the IoT device, or send instructions to IoT devices via the cloud center (such as via the application program installed on their phones), it involves three or four participants. In the latter authentication scenario, if there is a limited number of IoT devices and they are directly connected to the cloud center, the authentication then involves the users, the cloud center, and IoT devices; if the number of IoT devices is extensive and they are connected to the cloud center via the gateway nodes, the authentication then involves the users, the

cloud center, the gateway, and the IoT devices.

Usually, the Auth-Sce II containing four participants has various real-life applications. For example, in an industrial predictive maintenance system, the IoT devices are deployed to continuously monitor and update the real-time status of the critical industrial machines. Once abnormal data are found, the gateway can send them to the cloud center; the cloud center can make a comprehensive diagnosis based on the data submitted by each gateway. On this occasion, the responsible persons (the users) may need to access the real-time data directly from the IoT devices for further check and send the instructions to a specific device to handle the exception. Under this case, users and target IoT devices should mutually verify their identities first 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 gateway needs to store user-related data and do some computation to authenticate the users. When thousands of users and IoT devices are connected to one gateway, the efficiency of the authentication scheme is primarily affected by the performance of the gateway. Thus, the participation of the cloud center in the authentication phase can greatly alleviate the computation complexity at the gateway.

A. Motivations and Contributions

With the exponential growth of interconnected IoT devices, a gateway may need to simultaneously connect to thousands of IoT devices [18]. One gateway node may simultaneously perform mutual authentication with thousands of IoT devices. As such, the efficiency of authentication schemes is greatly affected by the computing and storage bottleneck of the gateway. Thus, finding a way to share the load of the gateway is essential to improve the efficiency of authentication schemes. Cloud computing technology is regarded as a promising way to solve this issue [3], [6]. Properly using the cloud center's computing power and storage capabilities to alleviate the gateway's load can significantly improve the efficiency of the authentication scheme. Nevertheless, from the history of user authentication schemes for cloud-aided IoT, most schemes are not designed for users to remotely control and access real-time data on IoT devices. Alternatively, the impact of growing sensor nodes on the performance of the gateway is not fully considered , causing the cloud center to only participate in the registration process but not be incorporated in the authentication process. As such, with the increasing number of connected IoT devices, the gateway has to deal with a large number of concurrent user requests, putting tremendous computation pressure on it.

In this paper, we aim to design a user authentication scheme for the Auth-Sce II with four parities, considering the computation and storage pressures on gateway nodes due to the explosion of IoT devices. Contributions are summarized below.

1) We improve the adversary model and evaluation criteria based on existing ones for wireless sensor networks.
2) We take a state-of-the-art authentication scheme (published by Waizd et al. at IEEE TDSC'20 [13]) as a case study to reveal the challenges and subtleties in designing a practical authentication scheme for cloud-aided IoT. We identify the security weaknesses in Wazid et al.'s scheme, and discuss the unreasonableness of the scheme in terms of the way to integrate cloud computing technology and IoT network.
3) We propose a secure and efficient user authentication scheme for remote control and real-time data access in cloud-aided IoT. Armed with the elliptic curve cryptography algorithm, fuzzy-verifier, and honey-words technique, our scheme provides many ideal attributes, such as user anonymity and multi-factor security. Furthermore, the proposed scheme uses the capabilities of the cloud center to reduce the computational burden of the gateway. So it is especially suitable for cloud-aided IoT applications with massive IoT devices.
4) We analyze the security of the proposed scheme using provable security analysis, the ProVerif tool, heuristic analysis, and BAN logic, and then compare it with eleven 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 gateway nodes.

Note that this paper expands upon an earlier conference paper [19], 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-aided IoT authentication schemes. 2) The extended version provides formal security proof for the proposed scheme. 3) The extended version improves the original scheme in [19] to achieve better security (i.e., resistance to DDoS attacks). 4) The extended version describes the adversary model and evaluation criteria for cloud-aided IoT authentication schemes.

II. Related Work

In 2009, to support the user in securely accessing the real-time data stored in sensor nodes, Das et al. [20] firstly proposed a two-factor user authentication scheme for wireless sensor networks (WSNs, one of the important infrastructure of IoT). Since then, numerous authentication schemes for WSNs have been proposed [7], [21], but most of them are unsatisfactory in some way. For example, they are identified as being subject to offline dictionary attacks, insider attacks, and impersonation attacks. Recently, with the prevalence of IoT techniques, an increasing number of user authentication schemes for IoT systems have been developed. There are some notable schemes like [12], [22], [23]. However, these schemes still suffer from various attacks. For example, Wazid et al.'s scheme [23] is vulnerable to offline dictionary attacks and cannot achieve user anonymity and forward secrecy; Wu et al.'s scheme [24] 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-aided IoT environments. Their scheme contains two parties, i.e., the cloud center and the users. That is, this scheme is for the Auth-Sce I, not the Auth-Sce II where the user requires to know the real-time data of IoT devices. In addition, this scheme is susceptible to various security threats. As shown in Tab. I, similar considerations apply to the schemes of Shen et al. [10], Das et al. [11], Sharma et al. [14] and Bhuarya et al. [16]. These schemes are all not designed for real-time data access to IoT devices.

In 2020, Jiang et al. [7] presented a user authentication scheme for cloud-aided autonomous vehicles (an IoT application). Their scheme finishes the authentication among the users, the cloud

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

Schemes	Years	Auth-Scenarios*	Participants	Main limitations
Amin et al. [6]	2018	Auth-Sce I	User, cloud center	Cannot achieve forward secrecy and multi-factor security
Shen et al. [10]	2018	Auth-Sce I	User, cloud center	Using timestamps, cannot achieve multi-factor security
Das et al. [11]	2018	Auth-Sce I	User, cloud center	Cannot achieve forward secrecy and 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
Deebak et al. [15]	2021	Auth-Sce II	User, gateway, cloud center, IoT devices	Cannot achieve forward secrecy
Bhuarya et al. [16]	2021	Auth-Sce I	User, cloud centers	Using timestamps, cannot achieve multi-factor security
Chaudhry et al. [17]	2021	Auth-Sce II	User, gateway, IoT devices	Cannot achieve multi-factor security

Auth-Scenarios*: Authentication Scenarios. In Section I, we mentioned, that there are usually two typical authentication scenarios. Auth-Sce I: the users want to access the services provided by the cloud center; Auth-Sce II: the users want to access the real-time data from the IoT device or send command instructions to the IoT device via the cloud center.

center, and the IoT devices. In 2021, Chaudhry et al. [17] also proposed a lightweight scheme for cloud-aided 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 applied to the Auth-Sce II with three parties. But the security of both schemes is not guaranteed. From the protocol design, there is no essential difference in whether the set of participants is (the users, the IoT devices, the gateways) or (the users, the IoT devices, the cloud center). Thus, we view these two schemes as one class. Three-parities authentication schemes of this kind are developed very well and there are many proposed schemes [22], [23], [25]. Thus, the design of a scheme for the Auth-Sce II with three parities is not the focus of our research.

In 2020, Wazid et al. [13] presented a three-factor user authentication scheme and proved its security using formal security proof in a smart home environment (an IoT application). In this scheme, the registration server responsible for the key distribution and registration of participants, can be regarded as a cloud center in cloud-aided IoT environments. This scheme supports the authentication among users, the gateway, IoT devices, and the cloud center. It is suited to 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, and so do the schemes of Srinivas et al. [12], and Deebak et al. [15].

From the history of user authentication for cloud-aided IoT, little attention is paid to the Auth-Sce II with four parities. Besides, the current research pays more attention to the computation complexity at the IoT devices, rather than at the gateway. 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, there will be more and more IoT devices connected by a single gateway [3], [18]. In this way, a single gateway may need to process a large number of authentication requests simultaneously, and the computation complexity of the authentication scheme at the gateway will have a significant impact on the system efficiency.

Therefore, this paper aims to provide a secure authentication scheme for the Auth-Sce II with four parities. In addition to focusing on the properties of the scheme to resist various attacks, we also focus on the properties to reduce the computation complexity of the gateways through the cloud center, so that it can be applied to a network environment with massive IoT devices.

TABLE II
NOTATIONS AND ABBREVIATIONS.

Symbol	Description	Symbol	Description
U_i	i^{th} user	S_j	j^{th} sensor node
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	password and biometrics of U_i
Gen/Rep	fuzzy extractor	x/y	$CloCen$'s long term secret key
X_{S_j}	secret key of S_j	x_{G_k}	GWN's long term secret key
X_{G_k}	secret key of GWN_k	$K_{\mathrm{GWN}-U_i}$	secret key of GWN for U_i
$\oplus$	bitwise XOR operation	$K_{\mathrm{GWN}-S_j}$	secret key of GWN for S_j
$\rightarrow$	an insecure channel	$\Longrightarrow$	a secure channel
$h(\cdot)$	one-way hash function	$\|$	concatenation operation

III. ADVERSARY MODEL AND EVALUATION CRITERIA

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

A. Adversary Model

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

C-1. According to the Dolev-Yao model [26], $\mathcal{A}$ can fully control the messages transmitted among users, the cloud center, the gateway, and IoT devices in an insecure channel.

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 users' identities and passwords, respectively; $\mathcal{A}$ also could obtain users' identities when assessing the security of authentication schemes. The two capabilities are given from the facts: 1) users' passwords are usually memorable strings and follow a Zipf distribution [27], resulting in the limited space of passwords. 2) Users' identities are usually static and can be harvested from popular forums. Besides, people normally do not keep their identities secret, thus increasing the risk of leakage [28].

C-3. $\mathcal{A}$ can obtain n-1 (n=2,3) factors in n-factor user authentication schemes [29], [30]. The factors include: 1) passwords; 2) data in smart mobile devices; and 3) biometrics.

C-4. $\mathcal{A}$ can obtain the secret key of the cloud center or 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 [13], [31].

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 [23], [29].

C-6. $\mathcal{A}$ can obtain previous session keys between users and IoT devices [22], [31].

C-7. When determining the security of the registration phase, $\mathcal{A}$ is able to be the administrator of the cloud center [7], [22]. This capability is allowed to capture insider attacks in the registration phase. In this attack, the administrator is able to obtain/guess users' passwords.

B. Evaluation Criteria

As shown in Tab. III, we construct our evaluation criteria based on a widely accepted criteria framework [29]. 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 attacking the scheme.

IV. Cryptoanalysis of Wazid et al.'s Scheme

In 2020, Wazid et al. [13] presented a user authentication protocol for a smart home environment (a typical IoT application). The cloud center of the scheme is served as a registration center to distribute the three parties' secret keys. In this section, we use this typical scheme as an example to show the common security threats and the design challenges of an authentication scheme for the Auth-Sec II in cloud-aided IoT. The review of Wazid et al.'s scheme is shown in Appendix A.

Note that, $Gen(\cdot)$/$Rep(\cdot)$ is a fuzzy extractor algorithm [32]:

- $Gen(\cdot)$: $Gen(Bio_i)$=(δ_i, τ_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.
- $Rep(\cdot)$: $Rep(Bio_i', \tau_i) = \delta_i$. 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 fundamental requirement of a multi-factor user authentication scheme. It ensures that even if an adversary has compromised any two of the three factors, the rest factor remains secure. However, we find that if an adversary in Wazid et al.'s scheme compromises a victim's smart mobile device (to get $\{\tau_i, B_i\}$) and biometrics, then he can conduct an offline dictionary attack to get U_i's password as 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 δ_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 step 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 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 attack, $\mathcal{A}$ needs to be armed with three capabilities: get $\{\tau_i, B_i\}$ from U_i's smart mobile device; obtain U_i's biometric; and eavesdrop $\{TID_i, M_2, M_3, T_1\}$ transmitted between the user and 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 δ_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 step 1-8 until the correct value is found.

This attack time complexity is also $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (3T_H + T_B))$. The inherent reasons for the above attacks are similar: $\mathcal{A}$ can construct and obtain a verification parameter using the victim's biometric, 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 [31]. 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 $|\mathcal{D}_{pw}| = |\mathcal{D}_{id}| = 2^6$. To verify the correctness of the guessed password, $\mathcal{A}$ has to interact with the cloud center online. However, since the honey-words will record the number of failures of authentication, $\mathcal{A}$ can only conduct a limited number of online queries. Thus, the probability that $\mathcal{A}$ obtains the correct password is very small.

For the latter attack, Ma et al. [33] have proved that a public-key algorithm is needed. 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 using the secret key of the cloud center. As such, $\mathcal{A}$ cannot construct such a M_5^* to verify the guessed password as above without Pub, and 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 and is hard to avoid by making minor changes. 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 the desynchronization issue. The attack steps are shown below:

Step 1. Intercept $\{M_{14}, M_{15}, M_{16}, T_3, T_4\}$.

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 will build a session key.
	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.
	S5	Resistance to Smart Mobile Device Loss Attacks	$\mathcal{A}$ failed to attack the scheme via a user's mobile device.
	S6	Resistance to Node Capture Attacks	$\mathcal{A}$ cannot attack the scheme via the IoT devices.

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.

This attack time complexity is very small. Note that, according to Wazid et al.'s scheme, once 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 $\mathrm{GWN_k}$ cannot authenticate each other successfully.

C. No Forward Secrecy

With the increasing attacks on servers, forward secrecy has become the final significant defense to protect the security of a system. Forward secrecy guarantees that even if the entire system is compromised, the previous conversations are still secure. 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 $K_{\mathrm{GWN}-S_j}$ stored in the gateway and controls the open channel, he can compute the previous session keys between users and IoT devices as below:

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

This attack time complexity is $\mathcal{O}(2T_H)$. From the attack mentioned above, we can see that it is not trivial to provide forward secrecy: Ma et al. [33] show that there are two requirements to achieve forward secrecy, i.e., a public-key algorithm and at least two module exponentiation or point multiplication operations on the server side (i.e., the IoT devices).

V. PROPOSED SCHEME

In Wazid et al.'s scheme [13], the cloud center (the registration center) simply assigns some parameters to other participants and does not join the authentication phase. As such, when the number of IoT devices becomes large, one gateway may execute thousands of authentication sessions concurrently, and the efficiency of the protocol is easily affected by the capabilities of the gateway. Therefore, Wazid et al.'s scheme is not suited for cloud-aided IoT environments where large numbers of IoT devices are involved. To overcome the identified weaknesses, we design an enhanced and efficient three-factor user authentication scheme. In the proposed scheme, the cloud center is involved in the authentication phase and shares part of the computation and storage stress of the gateway.

A. IoT device and Gateway Registration Phase

In the proposed scheme, to provide better security, the cloud center CloCen owns two secret long-term keys x and y, related to users and the gateway, respectively. The cloud center will distribute the gateway a secret key X_{G_k} (=$h(x||GID_k)$) to serve as an authenticated credential. The gateway and IoT devices will 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 a wider space for the application of the protocol 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 users register to the cloud center, and the IoT devices register to the gateway. After the registration, the cloud center will build a secret key with the gateway and users, 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 firstly sends the registration request to the cloud center, and then the cloud center returns a secret key X_{G_k}. The whole steps are given below.

R1. $\mathrm{GWN}_k \Longrightarrow \mathrm{CloCen}$: registration request (including the identity GID_k of the gateway GWN_k).
R2. $\mathrm{CloCen} \Longrightarrow \mathrm{GWN_k}$: $\{\mathrm{GWN}_k, X_{G_k}\}$. 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_{G_k} is the secret key of the cloud center, and finally sends $\{\mathrm{GWN}_k, X_{G_k}\}$ to GWN_k.
R3. GWN_k keeps X_{G_k}.

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

R1. $S_j \Longrightarrow \mathrm{GWN_k}$: registration request (including the identity SID_j of the IoT device S_j).
R2. $\mathrm{GWN_k} \Longrightarrow S_j$: $\{SID_j, X_{S_j}\}$. $\mathrm{GWN_k}$ firstly checks the validity of SID_j, then computers $X_{S_j} = h(SID_j||\ x_{G_k})$, where x_{G_k} is the secret key of $\mathrm{GWN_k}$.
R3. S_j keeps X_{S_j} as its private key.

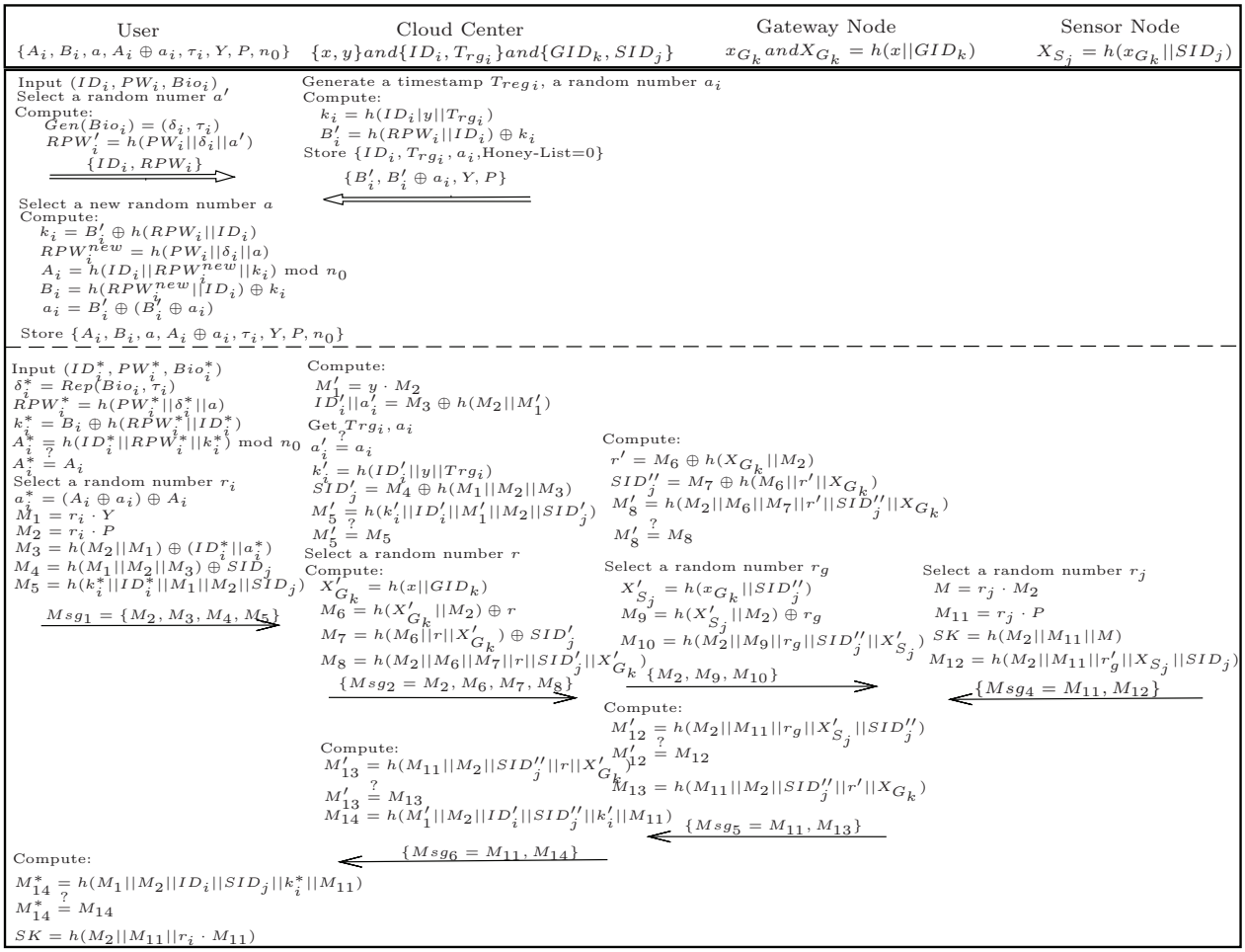

Fig. 2. The Proposed Scheme.

B. User Registration Phase

In the user registration phase, the user U_i firstly 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 as below.

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. Then, 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 avoid privilege 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 as 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 step 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 biometrics 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^* according to 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 resist this insider attack is to make some changes to the RPW_i to make it RPW_i'. Note that the user logs in with RPW_i'. 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

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: δ_i^* = $Rep(Bio_i^*, \tau_i)$, RPW_i^* = $h(PW_i^*||\delta_i^*||a)$, k_i^* = $B_i \oplus RPW_i^*$, A_i^* = $h(ID_i^*||\ RPW_i^*||k_i^*)$ mod n_0, then it compares A_i^* with A_i to verifies the authenticity of U_i.
If $A_i^* \neq A_i$, the user's request will be 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

The authentication phase is the core step of the scheme. It consists of six message flows and involves authentication among four participants. The authentication steps are given below:

V1. CloCen $\longrightarrow$ $\mathrm{GWN_k}$: $\{M_2, M_6, M_7, M_8\}$.
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 rejects the session. Then, CloCen sets Honey-list = Honey-list+1. Once the Honey-list exceeds a preset value (such as 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 the gateway $\mathrm{GWN_k}$ to which S_j belongs, 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 $\mathrm{GWN_k}$.

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

V3. S_j $\longrightarrow$ $\mathrm{GWN_k}$: $\{M_{11}, M_{12}\}$.
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, and 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 then responds $\{M_{11}, M_{12}\}$ to the gateway $\mathrm{GWN_k}$.

V4. $\mathrm{GWN_k}$ $\longrightarrow$ CloCen: $\{M_{11}, M_{13}\}$.
On receiving S_j's respond, the gateway $\mathrm{GWN_k}$ 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 $M_{12}' \neq M_{12}$, $\mathrm{GWN_k}$ ends the session. Otherwise, $\mathrm{GWN_k}$ 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 $\mathrm{GWN_k}$'s respond, CloCen computes M_{13}' = $h(M_{11}||M_2||SID_j''||r||X_{G_k}')$ to test the identity of $\mathrm{GWN_k}$. 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 returns $\{M_{11}, M_{14}\}$ to U_i.

V6. Once obtaining CloCen's reply $\{M_{11}, M_{14}\}$, the smart mobile device computes M_{14}^* = $h(M_1||M_2||ID_i||SID_j||\ k_i^*||M_{11})$, 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, a_i is applied to test whether the smart mobile device has been compromised, further preventing DDoS attacks. Review the scheme in [19], no such a parameter a_i is involved, the cloud center simply uses M_5 to judge whether the smart mobile device is compromised. As such, $\mathcal{A}$ can construct a login request arbitrarily and send it to the cloud center. Then according to the protocol's procedures, the cloud center will set Honey-list = Honey-list+1. With a limited number of malicious incorrect login requests, U_i's account will be locked, and a DDoS attack will succeed. To avoid this attack, we set an additional parameter a_i. Besides, there is an additional step for the cloud center to check whether the smart device is compromised: check the value of a_i' and 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 the smart mobile device is compromised.

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 δ_i^* = $Rep(Bio_i^*, \tau_i)$, RPW_i^* = $h(PW_i^*||\delta_i^*||a)$, k_i^* = $B_i \oplus RPW_i^*$, A_i^* = $h(ID_i^*||\ RPW_i^*||k_i^*)$ mod 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} = $h(ID_i^*||\ RPW_i^{new}||k_i^*)$ mod n_0, B_i^{new} = $h(RPW_i^{new}||\ ID_i) \oplus k_i$, and 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: $\{ID_i, RPW_i, revoke - requst\}$, where $Gen(Bio_i)$ = (δ_i, τ_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 session keys via provable security analysis, heuristic analysis and BAN logic, and simulate the protocol by using the ProVerif tool. These methods analyze the security of the proposed scheme from different angles and are widely accepted in the analysis of user authentication schemes [6], [22], [29], [31]. The focus of these methods is different, and they all have their own limitations.

In general, BAN logic [34] 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 [34]. However, the security goal of BAN logic analysis is generally to prove the authenticity of the identity of the two communicating parties and the parameters or keys shared between them. Furthermore, BAN logic has been recognized by many researchers that there are limitations to its power [35], [36], 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 [24], [37]–[39]. Usually, the researchers apply this tool to prove the three properties of multi-factor authentication schemes: the adversary cannot obtain users' passwords or session keys, and the authentication. As far as we know, other properties of the user authentication schemes, such as resistance to node capture attacks, are not analyzed by the ProVerif tool at present. Furthermore, the adversary model of multi-factor authentication schemes is not fully characterized in ProVerif at present.

Provable security analysis is based on computational models and has become an indispensable tool in analyzing and evaluating new cryptographic schemes [31], [40], [41]. However, according to Wang et al. [28], provable security analysis fails to capture some attacking scenarios and security properties.

Heuristic analysis is a crucial way to evaluate the security of a scheme [12], [13], [40]. In the heuristic analysis, the adversary capabilities and various security properties of the multi-factor protocol are fully considered. However, it relies heavily on analyst's experience, and there may be negligence in the analysis process, resulting in inaccurate analysis results.

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 heuristic analysis. However, they have some limitations in characterizing the security properties and adversary capabilities of multi-factor authentication protocols, and the heuristic analysis can ideally make up for this deficiency.

It is well known that the design of security protocols is notoriously hard. Therefore, we adopt these four methods to evaluate our scheme and hope that this will help scrutinize the security of our protocol comprehensively and accurately. Our scheme is proven secure under these four security analysis methods. This provides us with an adequate level of confidence about the security of our protocol.

A. Provable Security Analysis

This section formally analyzes the proposed scheme under the random oracle model. Specifically, we extend the BPR00 model [42] from the following two aspects: according to [31], we extend the $Corrupt()$ query to capture smart-devices-loss attacks; like [12], [22], [25], we build a multi-party password authentication model based on [41]. As shown below, the adversary's capabilities and behavior can be modeled via a series of notations and queries.

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 that supports cryptographic operations and biometric inputting storing $\{A_i, B_i, a, A_i \oplus a_i, \tau_i, Y, P, n_0\}$
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)

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 Tab. IV. In the execution of the protocol, U, CloCen, GWN and S is instantiated as U_i, CloCen_m, GWN, and S_j, respectively. We let I be the set of instances, and I^s be the s-th instance of I.

Queries. We define the queries depicting adversary's behavior in a real attack as below:

- $Execute(U_i^r, \text{CloCen}, \text{GWN}_k^s, S_j^t)$: it models the entire protocol flow, and outputs the messages transmitted among the participants $\{U_i, \text{CloCen}, \text{GWN}_\text{k}, S_j\}$.
- $Send(I, I^s, m)$: 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 biometric and the data in SD;
  - For a=3, output U_i's biometric and password.
- $Corrupt(I)$: 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 is instantiated to different objects, the output of the query is also different.
  - When I==CloCen_m, output the long-term secret keys $\{x, y\}$, $\{ID_i, T_{rg_i}, a_i\}$ and $\{GID_k, SID_j\}$.
  - When I==GWN_k, output the secret keys $\{x_{G_k}, X_{G_k}\}$.
  - When I==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) $Corrupt(U)$-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 a $Test$-query to a fresh instance. With these information from the queries, $\mathcal{A}$ attempts to output a guessed bit b' for b in $Test$-query. Let $Succ$ be the event that $\mathcal{A}$ guesses b correctly. Then the advantage of $\mathcal{A}$ in breaking the semantic security of $\mathcal{P}$ is:

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

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 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:

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

where l is a system security parameter, $\mathcal{D}$ is the password space whose frequency distribution satisfy a Zipf's law [27], C' and s' are Zipf parameters [27].

Elliptic Curve Gap Diffie-Hellman (ECGDH) problem: given $a \cdot P$ 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 $mathcalF_p$, p and q are two large primes, and $|q| = l$. $|\mathcal{D}|$ is a password space following Zipf's law [27]. 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:

$$\begin{aligned}\mathbf{Adv}^{ake}_{\mathcal{P}}(\mathcal{A}) \leq C' q^{s'}_s + \frac{2q^2_h + 3q^2_s + 3(2q_h + q_s)^2}{2^l} \\ + \frac{(q_s + q_e)^2}{2(p-1)} + 2q_h((q_s + q_e)^2 + 1)Adv_{\mathcal{A}(t')},\end{aligned} \tag{3}$$

where C' and s' are the Zipf parameter [27], T_m is time for scaler multiplication in $\mathcal{G}$, and $t' \leq t + (2q_s + 6q_e + 1) \cdot T_m$. Please refer to Appendix B for details of the proof.

B. The Proverif Tool

We demonstrate the security of our scheme in the Proverif framework. Please refer to Appendix C for details.

C. Heuristic Analysis

We provide a heuristic analysis of our scheme from the perspective of a real adversary in Appendix D.

D. BAN Logic Analysis

We prove the security of the scheme using BAN logic analysis [34]. The proof details can be referred to Appendix E.

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.	[25]	√	√	√	√	√	√	×	×	×	√	×	×
Amin et al.	[6]	√	×	√	×	√	√	√	×	×	×	×	×
Wazid et al.	[13]	√	√	√	×	√	×	√	×	×	×	×	×
Wazid et al.	[23]	√	√	√	×	√	×	√	√	×	√	×	×
Sharif et al.	[43]	√	√	√	√	×	√	√	√	×	×	×	×
Das et al.	[11]	√	√	√	×	√	√	×	√	×	×	×	×
Srinivas et al.	[12]	√	√	√	×	√	√	√	×	×	×	×	×
Srinivas et al.	[44]	√	√	√	×	√	√	×	×	√	×	×	√
Li et al.	[22]	√	√	√	√	√	√	√	×	√	√	×	×
Jiang et al.	[7]	√	√	√	√	√	√	×	√	×	√	√	×
Deebak et al.	[15]	√	√	√	√	√	√	√	√	×	√	√	×
ours	-	√	√	√	√	√	√	√	√	√	√	√	√

VII. PERFORMANCE ANALYSIS

In this section, we compare our scheme with eleven 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 Tab. V, Tab. VI and Tab. VII. Note that, we let T_C, T_P, T_B, T_H and T_S denote the operation time for chebyshev chaotic-map, elliptic curve point multiplication, fuzzy extracting biometric data, hash, and symmetric encryption, respectively. According to [12], [21], $T_H \approx 0.003$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, 2.5 GHz CPU and 8 G memory, with miracle library and gcc 7.5.0 under Ubuntu 18.04. 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. Besides, according to [22], [31], the identities, the tolerance error 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. The secret key of symmetric encryption is 160bits, and the ciphertext is 1024bits.

From a security point of view, as shown in Tab. V, our scheme is the only one that meets all twelve evaluation criteria. As for other compared schemes, most of them cannot resist "S6 node capture attacks" (Only Srinivas et al.'s scheme [44] meets this attribute). Only two schemes (Jiang et al. [7], Deebak et al. [15]) are secure against "S5 smart mobile device loss attacks". And only two schemes (Srinivas et al. [44], Li et al. [22]) provide "S3 forward secrecy". Also only five schemes ([7], [15], [22], [23], [25]) meet the security requirement "S4 resistance to known attacks". In short, other compared schemes all have security issues. Compared with these schemes, the proposed scheme performs best in terms of security.

From a computation cost point of view, as shown in Tab. VI, our scheme is also competitive. Firstly, the number of our communication round is 6, which is the biggest among the eleven schemes. However, as shown in Tab. VI, our scheme involves four parties in the authentication phase, and other schemes only involve three parties. Thus, in this premise, compared with the schemes having six communication rounds (Yang et al. [25], Sharif et al. [43], and Deebak et al. [15]), our scheme with four parties involved is competitive. Compared with the rest schemes [6], [7], [11]–[13], [22], [23], [44], our scheme is acceptable.

Secondly, the computation cost on each side of our scheme is still competitive. Especially, we achieve the minimum compu-

TABLE VI
COMPUTATION COST COMPARISON AMONG RELEVANT AUTHENTICATION SCHEMES.

Scheme	Ref.	R.	Computational Cost (ms)			
			U_i	GWN	S	CloCen
Yang et al.	[25]	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.	[23]	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.	[43]	6	$11T_H + T_B$ ≈0.33	$17T_H$ ≈0.05	$5T_H$ ≈0.02	NI
Das et al.	[11]	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.	[44]	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.	[22]	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
Deebak et al.	[15]	6	$3T_P$+$17T_H$+$3T_e$+T_s ≈ 320.73	$2T_P$+$6T_H$+ T_e+T_s ≈107.13	$2T_P$+$3T_H$+$2T_e$ ≈213.60	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.	[25]	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.	[23]	1376	2624	1504	NI	1088	736	288	NI
Sharif et al.	[43]	1024	1312	352	NI	608	$288n_u + 128n_s$	288	NI
Das et al.	[11]	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.	[44]	832	512	672	NI	1056	800+$128n_s$	288	NI
Li et al.	[22]	640	960	480	NI	768	$128n_s$	288	NI
Jiang et al.	[7]	1056	960	320	NI	960	$544n_u + 288n_s$	160	NI
Deebak et al.	[15]	992	1184	320	NI	1152	480+$576n_u$+$160n_s$	160	$768n_u$
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 smart card: hash function, biometrics operations for Gen$(\cdot)$, $Rep(\cdot)$;
"n_u", "n_g"and "n_s" denote the number of users, gateway node and device nodes, respectively;

tation cost on the gateway. Note that, in Tab. VI, the first six schemes are based on symmetric cryptographic algorithms; the rest six schemes are based on asymmetric cryptographic algorithms. As we know, the computation complexity of asymmetric cryptographic algorithms must be larger than that of symmetric cryptographic algorithms. Besides, biology-based authentication is certainly slower than password-based authentication. 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 [44], 1.20ms [22]), and better than Jiang et al. [7] (1.8ms) and Deebak et al. [15] (320.73ms). In fact, except for the cost of Deebak et al.'s, the difference among the computation cost of other schemes is less likely to be perceived by users. Furthermore, our computation cost on IoT devices is moderate among these schemes using asymmetric cryptographic algorithms.

For communication cost and storage cost comparisons, as shown in Tab. VII, our scheme has the minimum storage cost on the gateway side (320bits) with competitive communication and storage cost on IoT devices and users' side. 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 eleven schemes. Besides, our scheme achieves the minimum computation and storage costs on the gateway side by using the computing resource of the cloud center with acceptable computational cost on the user side and IoT devices. Thus, the proposed scheme is especially suitable for cloud-aided IoT applications with massive IoT devices.

VIII. CONCLUSION

This paper aims to design a secure user authentication scheme for cloud-aided IoT systems with lightweight computation on gateways. Firstly, we consider Wazid et al.'s scheme as a case study and identify the security weaknesses and unreasonableness in such schemes. Then, we propose a new secure user authentication scheme for cloud-aided IoT environments and verify it with 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 eleven state-of-the-art authentication schemes. The results show that our scheme achieves the best security with the minimum computation and storage costs on the gateway side.

In this paper, we do not implement our scheme in real-world scenarios to test its efficiency. Also, the scheme is designed under the random oracle model. In our future work, we will improve the communication performance of the proposed scheme, and deploy it in real-world scenarios to test its security and efficiency. Furthermore, we will explore ways to design a secure four-party authentication scheme for cloud-aided IoT under other security models, such as the universal composability model.

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, 2019.

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

[3] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Gener. Comput. Sys.*, vol. 56, pp. 684–700, 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 (Internet and Distributed Computing System)*, 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, 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. Sys.*, vol. 78, pp. 1005–1019, 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. Tech.*, vol. 69, no. 9, pp. 9390–9401, 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. 17 430–17 438, 2021.

[9] W. Wang, C. Qiu, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, "Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet Things J.*, pp. 1–9, 2021, doi:10.1007/s12652-017-0474-8.

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

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

[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. Depend. Sec. Comput.*, vol. 17, no. 6, pp. 1133–1146, 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. Depend. Secur. Comput.*, vol. 17, no. 2, pp. 391–406, 2020.

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

[15] B. D. Deebak and F. Al-Turjman, "Lightweight authentication for iot/cloud-based forensics in intelligent data computing - sciencedirect," *Future Gener. Comput. Sys.*, vol. 116, pp. 406–425, 2021.

[16] 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, 2021.

[17] 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. Int. Tech.*, vol. 21, no. 3, pp. 1–19, 2021.

[18] S. Li, L. Da Xu, and S. Zhao, "5g internet of things: A survey," *J. Industrial Information Integration*, vol. 10, no. 1–9, 2018.

[19] XX, "Cloud-aided privacy preserving user authentication and key agreement protocol for internet of things," in *Pro. SocialSec 2019*, pp. 95–109.

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

[21] 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 Systems Journal*, vol. 15, no. 1, pp. 1120–1129, 2020.

[22] 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. Inform.*, vol. 14, no. 8, pp. 3599–3609, 2018.

[23] M. Wazid, A. K. Das, M. K. Khan, A. D. Al-Ghaiheb, N. Kumar, and A. Vasilakos, "Design of secure user authenticated key management protocol for generic iot networks," *IEEE Internet of Things J.*, vol. 5, no. 1, pp. 269–282, 2018.

[24] F. Wu, L. Xu, S. Kumari, X. Li, J. Shen, K. K. R. Choo, M. Wazid, and A. K. Das, "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, 2017.

[25] 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. Inform.*, vol. 16, no. 10, pp. 6584–6596, 2020.

[26] D. Dolev and A. Yao, "on the security of public key protocols," *IEEE Trans. Inform. Theor*, vol. 29, no. 2, pp. 198–208, 1983.

[27] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inform. Foren. Secur.*, vol. 12, no. 11, pp. 2776–2791, 2017.

[28] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Secur. Comput.*, vol. 12, no. 4, pp. 428–442, 2015.

[29] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Evaluating the node capture attack in user authentication scheme of wireless sensor networks," *IEEE Trans. Depend. Secur. Comput.*, vol. 19, no. 1, pp. 507–523, 2022.

[30] 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," *Inform. Sci.*, vol. 321, pp. 263–277, 2015.

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

[32] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *Siam Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2006.

[33] C. Ma, D. Wang, and S. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2215–2227, 2012.

[34] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *IEEE Trans. Comput.*, vol. 8, no. 1, pp. 18–36, 1990.

[35] C. Boyd and W. Mao, "On a limitation of ban logic," in *Proc. EUROCRYPT*, 1993, pp. 240–247.

[36] D. M. Nessett, "A critique of the burrows, abadi and needham logic," *ACM SIGOPS Operating Systems Review*, vol. 24, no. 2, pp. 35–38, 1990.

[37] 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 Sys. J.*, vol. 14, no. 1, pp. 39–50, 2020.

[38] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care iot," *Comput. Netw.*, vol. 177, pp. 107 333–107 349, 2020.

[39] 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. Sys.*, vol. 96, pp. 410–424, 2019.

[40] 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. Inform. Foren. Secur.*, vol. 12, no. 6, pp. 1382–1392, 2017.

[41] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. PKC*, 2005, pp. 65–84.

[42] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT 2000*, vol. 1807, pp. 139–155.

[43] 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. Sys.*, vol. 100, pp. 882–892, 2019.

[44] 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 of Things Journal*, vol. 8, no. 9, pp. 7727–7744, 2021.

Secure and Lightweight User Authentication Scheme for Cloud-Aided Internet of Things (Appendix File)

Abstract—This appendix file consists of five parts. Appendix A reviews the scheme of Wazid et al.. Appendix B shows the provable security proof process of the proposed scheme. Appendix C describes the security verification using the ProVerif tool, and Appendix D presents the heuristic analysis. Finally, Appendix E shows the BAN logic analysis.

Index Terms—Appendix file.

APPENDIX A:
REVIEW OF WAZID ET AL.'S SCHEME

In this section, the Wazid et al.'s scheme [1] is reviewed briefly. Their scheme consists of four phases, the IoT device and gateway registration phase, the user registration phase, the login phase, and the authentication phase.

The IoT device and the gateway can register to the networks via the cloud center (registration center) as below:

R1. $S_j \Longrightarrow RA$: registration request.
R2. $RA \Longrightarrow S_j$: $\{SID_j, X_{S_j} = h(SID_j||K_{GWN-S_j})\}$.
R3. $GWN_k \Longrightarrow RA$: registration request.
R4. $RA \Longrightarrow GWN_k$: $\{TID_i, ID_i, K_{GWN-U_i}\}$ and $\{SID_j, K_{GWN-S_j}\}$.

1 USER REGISTRATION PHASE

R1. $U_i \Longrightarrow RA$: $\{ID_i, RPW_i\}$, where $Gen(Bio_i) = (\delta_i, \gamma_i)$, $RPW_i = h(PW_i||\delta_i||a) \oplus r$, where a and r are two random numbers.
R2. $RA \Longrightarrow U_i$: $\{A_i, TID_i\}$, where $A_i = h(ID_i||K_{GWN-U_i}) \oplus RPW_i$, and TID_i is a temporary identity for U_i generated by RA. Furthermore, RA will store $\{ID_i, TID_i\}$ in GWN_k's database, and delete A_i and RPW_i from its database.
R3. The device computes $B_i = h(ID_i||\delta_i) \oplus a$, $RPW_i' = RPW_i \oplus r = h(PW_i)$, $C_i = h(ID_i \ ||RPW_i'||\delta_i)$, $A_i^* = A_i \oplus r = h(ID_i||K_{GWN-U_i}) \oplus RPW_i'$, then deletes A_i, stores $\{TID_i, \ A_i^*, B_i, C_i, \tau_i, h(\cdot), \ Gen(\cdot), Rep(\cdot), t\}$, where t is the error tolerance parameter of $Rep(\cdot)$.

2 LOGIN PHASE

L1. $U_i \Longrightarrow GWN_k$: $\{TID_i, M_2, M_3, T_1\}$.
U_i inputs $\{ID_i, PW_i, Bio_i\}$, and the smart mobile device computes: δ_i^* = $Rep(Bio_i^*, \tau_i)$, a^* = $B_i \oplus h(ID_i||\ \delta_i^*)$, RPW_i^* = $h(PW_i^*||\delta_i^*||a^*)$, C_i^* = $h(ID_i||\ RPW_i^*||\delta_i^*)$.
If $C_i^* \neq C_i$, the smart mobile device exits the session. Otherwise, the smart mobile device computes: M_1 = $A_i^* \oplus RPW_i^*$ = $h(ID_i||K_{GWN-U_i})$, M_2 = $M_1 \oplus r_{U_i}$, M_3 = $h(M_2||T_1 \ ||ID_i||TID_i||r_{U_i})$, where r_{U_i} is a random number and T_1 is a timestamp.

3 AUTHENTICATION PHASE

V1. $GWN_k \Longrightarrow S_j$: $\{M_7, M_8, T_2\}$.
After checking the valid of T_1, GWN_k gets ID_i and K_{GWN-U_i} from the database, then computes M_4 = $h(ID_i||K_{GWN-U_i})$ = M_1, $r_{U_i}^*$ = $M_2 \oplus M_4$, M_5 = $h(M_2||T_1||ID_i||TID_i||r_{U_i}^*)$.
If $M_5 \neq M_3$, GWN_k exits the session. Otherwise, GWN_k computes M_6 = $h(SID_j||K_{GWN-S_j})$, M_7 = $E_{M_6}[\ ID_i, GID_k, r_{U_i}^*, r_{GWN}, h(M_4)]$, M_8 = $h(M_6||T_2||\ ID_i||SID_j||GID_k||r_{GWN})$.
V2. $S_j \Longrightarrow GWN_k$: $\{M_{10}, M_{11}, M_{12}, T_3\}$. The IoT device first checks T_2, and then decrypts M_7 with X_{S_j} to get $(ID_i, GID_k, r_{U_i^*}, r_{GWN}, h(M_4))$, and computes M_9 = $h(h(SID_j||K_{GWN-S_j})||T_2||ID_i||SID_j||GID_k||r_{GWN})$.
If $M_9 \neq M_8$, S_j ends the session. Otherwise, S_j computes: SK = $h(ID_i||SID_j||GID_k||r_{U_i}^*||r_{GWN}||r_{S_j}$ $||h(M_4)||h(h(SID_j||K_{GWN-S_j})))$, M_{10} = $h(h(SID_j||\ K_{GWN-S_j})\ ||T_3) \oplus r_{S_j}$, M_{11} = $h(SK||T_3)$, M_{12} = $h(r_{S_j}||r_{GWN}||\ SID_j||GID_k||T_3)$.
V3. $GWN_k \Longrightarrow U_i$: $\{M_{14}, M_{15}, M_{16}, T_3, T_4\}$.
The gataway first checks the valid of T_3, then computes $r_{S_j}^*$ = $M_{10} \oplus h(h(SID_j||K_{GWN-S_j})\ ||T_3)$, M_{13} = $h(r_{S_j}^*||r_{GWN}||SID_j||GID_k||T_3)$.
If $M_{13} \neq M_{12}$, GWN_k ends the session. Otherwise, GWN_k computes: M_{14} = $E_{M_{14}}[r_{U_i}^*, r_{GWN}, r_{S_i}^*, SID_j, GID_k, h(M_6)]$, M_{15} = $TID_i^{new} \oplus h(\ TID_i||M_4||T_3||T_4)$, M_{16} = $h(M_{11}||T_4||\ r_{U_i}^*)$, where T_4 is a timestamp and TID_i^{new} is a new unique identity generated by GWN_k.
V4. After checking T_4, the smart mobile device decrypts M_{14} with M_1, then compares r_{U_i} with $r_{U_i}^*$. If they are not equal, the smart mobile device ends the conversation; otherwise, computes SK' = $h(ID_i||\ SID_j||GID_k||r_{U_i}||r_{GWN_k}^*||\ r_{S_j}^*||h(M_1)||h(M_6))$, M_{17} = $h(h(SK'||T_3)||T_4||r_{U_i})$. If $M_{17} == M_{16}$, both U_i and S_j accept the session key, and then the smart mobile device replaces TID_i with TID_i^{new}, where TID_i^{new} = $M_{15} \oplus h(TID_i||M_1||T_3||T_4)$.

APPENDIX B: THE FORMAL PROOF OF THE PROPOSED SCHEME UNDER THE RANDOM ORACLE MODEL

In section V-A, we mention the semantic security of $\mathcal{P}$ that the adversary cracks for a probabilistic polynomial time. For q_s times $Send$-query, q_e times $Execute$-query and q_h times $Hash$-query within time t, there is an inequality:

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

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

This theorem is proved via a sequence of games which model the attack process 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, we have

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

where $Succ_n$ denotes $\mathcal{A}$ in Game 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 input and output of hash-query; $\Lambda_{\mathcal{M}}$ which keeps the input and output of $Execute$-query; $\Lambda_{\mathcal{H}^{\mathcal{A}}}$ which keeps hash-query asked by the adversary $\mathcal{A}$. In this game, the protocol is conducted as Sec. V. Then the adversary $\mathcal{A}$ intercepts message among the four participants via $Execute$-query, and finally executes $Test$-query to guess b. Obviously, $\mathcal{A}$ 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 $\mathcal{A}$ is not increase:

$$|Pr[Succ_1] - Pr[Succ_0]| = 0 \tag{2}$$

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 making valid messages correctly, will the semantic security of the protocol be compromised. In our protocol, there are two kinds of collisions as follows:

- The collisions of the output 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:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2(p-1)} \tag{3}$$

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

- 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 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 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 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 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 is: $\frac{q_s^2}{2^l}$;

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

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{3(2q_h + q_s)^2 + 3q_s^2}{2^{l_s}} \tag{4}$$

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, $\mathcal{A}$ needs to get the information of the victim's biometrics via 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 to replace the victim's, and its probability is $q_s \cdot \varepsilon_b$ (ε_b is the probability that two persons' biometric are similar 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 biometric and data in the device. Then $\mathcal{A}$ needs to guess the victim's password correctly in q_s queries. Since the distribution of passwords follows Zip'f law, the probability of this is: $C' q_s^{s'}$, where C' and s' are Zip'f parameters [2].
- $\mathcal{A}$ queries $Corrupt(U_i, 3)$ to get the victim's password and biometric. Then $\mathcal{A}$ needs 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:

$$\begin{aligned} |Pr[Succ_4] - Pr[Succ_3]| &\leq max\{\frac{q_s}{2^{l_r}} + q_s\varepsilon_b, C' q_s^{s'}, \frac{q_s}{2^{l_s}}\} \\ &= C' q_s^{s'} \end{aligned} \tag{5}$$

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:

$$Pr[AskH_5] = Pr[AskH_5^0] \leq 2q_h \cdot Adv_p^{ECGDH}(t') \tag{6}$$

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_iP and r_jP in a session is $\frac{1}{(q_s+q_e)^2}$, then we have:

$$Pr[AskH_5] = Pr[AskH_5^0] \leq 2q_h(q_s+q_e)^2 \cdot Adv_p^{ECGDH}(t') \tag{7}$$

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 $G_0 \sim G_6$:

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

Appendix C: Security Verification using the ProVerif Tool - Simulation Study

The ProVerif tool is a mature formal security verification tool to analyze the security of cryptographic schemes [3], which has been widely used in analyzing multi-factor user authentication schemes. It mainly assesses the security of session keys.

The first step of validation is to define the premises. As shown in Tab. 1, we define two kinds of channels: ch1/2/3 and sch1/2/3. The notations ch and sch denote insecure and secure channels, respectively. The numbers 1, 2, and 3 denote the channel between users and cloud center, cloud center and gateway, gateway and IoT devices, respectively. sku and sks represent the session key computed by users and IoT devices, respectively. Furthermore, some secret keys, constants, and functions are defined. The query attacker(sku/sks) is to find whether the session key sku/sks is resistant to the attacker. With the premises, we can simulate the process user, cloud center, gateway and IoT device as shown in Tab. 2, Tab. 3, Tab. 4. The result is shown in Tab. 5. Its first two lines demonstrate that our weak strings password and identity are secure against offline guessing attacks. Its last two lines show that our session key is resistant to known attacks. Therefore, our scheme is secure under this ProVerif framework.

Appendix D: 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 with x. 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.

TABLE 1
Premises for the code Process

```
(* channels *)
free ch1: channel.
free ch2: channel.
free ch3: channel.
free sch1: channel [private].
free sch2: channel [private].
free sch3: channel [private].
(* secret key *)
free x: bitstring [private].
free y: bitstring [private].
free GIDk: bitstring [private].
free XGk: bitstring [private].
free Xsj: bitstring [private].
free PWi:[private].
free BIOi: bitstring [private].
(* shared keys *)
free sku: bitstring [private].
free sks: bitstring [private].
(* constants *)
free IDi: bitstring [private].
const n0: bitstring.
const P: bitstring.
const SIDj: bitstring.
table ud(bitstring,bitstring).
(* weak elements *)
weaksecret IDi.
weaksecret PWi.
(* functions *)
fun h(bitstring):bitstring. (* hash functions *)
fun con(bitstring,bitstring): bitstring. (* string concatenation *)
fun xor(bitstring,bitstring): bitstring. (* X-or functions *)
fun mul(bitstring,bitstring): bitstring. (* scalar multiplication *)
fun Gen(bitstring): bitstring. (* biometric fuzzy extract *)
fun Rep(bitstring, bio): bitstring. (* biometric fuzzy extract *)
fun Mod(bitstring,bitstring): bitstring. (* module functions *)
(* equations *)
equation forall m:bitstring,n:bitstring; xor(xor(m,n),n)=m.
equation forall m:bitstring,n:bitstring; mul(mul(P,m),n)=
mul(mul(P,n),m).
(* queries *)
query attacker(sku)
query attacker(sks)
```

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} consisting of the session key. According to our scheme, the session key is computed as $SK = h(M_2||M_{11}||r_iM_{11})$, thus the adversary still needs to obtain the parameter $M = r_jM_2 = r_iM_{11}$. 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 is bound to fail to compute M. Thus, our scheme achieves forward secrecy.

Proposition: 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 let the user send $\{ID_i, RPW_i = h(PW_i||\delta_i||a)\}$ to the cloud center when he registers to avoid exposing sensitive information. Under this circumstance, the administrator of the cloud center

TABLE 2
Cloud Center Process

```
let CloCReg1=
in(sch1,(cIDi:bitstring,cRPWi:bitstring));
new cTrgi:bitstring
let cki=h(con(con(cIDi,y),cTrgi)) in
let cBi=xor(h(con(cRPWi,cIDi)),cki) in
let cY=mul(y,P) in
insert ud(cIDi,cTrgi)
let m = xor(cBi,ai) in
out(sch1,(cBi,m,cY)).
let CloCReg2=
let cxGk=h(con(GIDk,x)) in
out(sch2,cxGk).
let CloCAuth=
in(ch1,(cM2:bitstring,cM3:bitstring,cM4:bitstring,cM5:bitstring));
let cM1=mul(y,cM2) in
let cIDi=xor(cM3,h(con(cM2,cM1))) in
get ud(=cIDi,cTrgi) in
let cki=h(con(con(cIDi,y),cTrgi)) in
let cSIDj=xor(cM4,h(con(con(cM1,cM2),cM3))) in
if cM5=h(con(con(con(con(con(cki,y),cIDi),cM1),cM2),cSIDj)) then
new cr:bitstring;
let cXGk=h(con(GIDk,x)) in
let cM6=xor(h(con(cXGk,cM2)),cr) in
let cM7=xor(h(con(con(cM6,cr),cXGk)),cSIDj) in
let cM8=h(con(con(con(con(con(cM2,cM6),cM7),cr),cSIDj),cXGk)) in
out(ch2,(cM2,cM6,cM7,cM8));
in(ch2,(cM11:bitstring,cM13:bitstring));
if cM13=h(con(con(con(con(cM11,cM2),cSIDj),cr),cXGk)) then
let cM14=h(con(con(con(con(con(cM1,cM2),cIDi),cSIDj),cki),cM11)) in
out(ch1,(cM11,cM14)).
let CloC=CloCReg1|CloCReg2|CloCAuth
```

TABLE 3
User Process

```
let User=
new a1:bitstring;
let (Pi:bitstring,Ri:bitstring)=Gen(BIOi) in
let RPWi1=h(con(con(PWi,Ri),a1)) in
out(sch1,(IDi,RPWi1));
in(sch1,(ugBi:bitstring,m:bitstring,ugY:bitstring));
new a:bitstring;
let ki=xor(ugBi,h(con(RPWi1,IDi))) in
let RPWi=h(con(con(PWi,Ri),a)) in
let uAi=Mod(n0,h(con(con(IDi,RPWi),ki))) in
let uBi=xor(h(con(RPWi,IDi)),ki) in
let ai=xor(ugBi,m) in
!
(
let uRi=Rep(BIOi,Pi) in
let uRPWi= h(con(con(PWi,uRi),a)) in
let uki= xor(uBi,h(con(uRPWi,IDi))) in
if uAi=Mod(n0,h(con(con(IDi,uRPWi),uki))) then
new uri: bitstring;
let uM1=mul(y,uri) in
let uM2=mul(P,uri) in
let uM3=xor(h(con(uM2,uM1)),IDi) in
let uM4=xor(h(con(con(uM1,uM2),uM3)),SIDj) in
let uM5=h(con(con(con(con(uki,IDi),uM1),uM2),SIDj)) in
out(ch1,(uM2,uM3,uM4,uM5));
in (ch1,(uM11:bitstring,uM14:bitstring));
if uM14=h(con(con(con(con(con(uM1,uM2),IDi),SIDj),ki),uM11)) then
let sku=h(con(con(uM2,uM11),mul(uri,uM11))) in
0
).
```

TABLE 4
Gateway Process & IoT Device Process

```
let GatewayReg=
let gXsj=h(con(SIDj,XGk)) in
out(sch3,gXsj).
let GatewayAuth=
in(sch2,XGk:bitstring);
in(ch2,(gM2:bitstring,gM6:bitstring,gM7:bitstring,gM8:bitstring));
let gr=xor(gM6,h(con(XGk,gM2))) in
let gSIDj=xor(gM7,h(con(con(gM6,gr),XGk))) in
if gM8=h(con(con(con(con(gM2,gM7),gr),gSIDj),XGk)) then
new grg:bitstring;
let gXsj=h(con(gSIDj,XGk)) in
let gM9=xor(h(con(gXsj,gM2)),grg) in
let gM10=h(con(con(con(con(gM2,gM9),grg),gSIDj),gXsj)) in
out(ch3,(gM2,gM9,gM10));
in(ch3,(gM11:bitstring,gM12:bitstring));
if gM12=h(con(con(con(con(gM2,gM11),grg),gXsj),gSIDj)) then
let gM13=h(con(con(con(con(gM11,gM2),gSIDj),gr),XGk)) in
out(ch2,(gM12,gM13));
let Gateway=GatewayReg|GatewayAuth.
let Device=
in(sh3,Xsj:bitstring);
!
(
in(ch3,(sM2:bitstring,sM9:bitstring,sM10:bitstring));
let srg=xor(sM9,(h(con(Xsj,sM2)))) in
if sM10=h(con(con(con(con(sM2,sM9),srg),SIDj),Xsj)) then
new srj:bitstring;
let sM = mul(srj,sM2) in
let sM11 = mul(srj,P) in
let sks=h(con(con(sM2,sM11),sM)) in
let sM12=h(con(con(con(con(sM2,sM11),srg),Xsj),SIDj)) in
out(ch3,(sM11,sM12));
0
).
```

TABLE 5
Result of the Verification

RESULT Weak secret IDi is true (bad not derivable).
RESULT Weak secret PWi is true (bad not derivable).
RESULT not attacker(sku[]) is true.
RESULT not attacker(sks[]) is true.

cannot gain any useful information since the password PW_i is protected by two parameters.

Proposition: Resistant to Smart Devices Loss Attacks. In the proposed scheme, an adversary cannot conduct an attack by using 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 only 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 phone or wants to change the password without being noticed by the smart mobile device, he has to construct correct $A_i = h(ID_i||RPW_i||k_i)$ mod 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 can find such a password and identity of the user that satisfy $h(ID_i^*||RPW_i^*||k_i)$ mod n_0 == A_i, he still is not sure whether the password is correct,
thirdcheckfor there are $|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| \setminus n_0 \approx 2^{32}$ candidates of $\{ID_i, PW_i\}$ pair when $n_0 = 2^8$ and $|\mathcal{D}_{pw}| = |\mathcal{D}_{id}| = 2^6$ according to Wang et al. [4]. 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 biometric, 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, though, 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: Resistant to Impersonation Attacks. The proposed scheme can resist impersonation attacks.

Proof: Firstly, we consider a user impersonation attack. As such, the adversary does not acquire information from a smart card. On the one hand, the adversary cannot obtain a user's password via offline dictionary attacks, according to our analysis of "smart card loss attack". On the other hand, the adversary cannot directly construct such a valid access request $\{M_2, M_3, M_4, M_5\}$, where M_5 consists of k_i that can only be computed via user-sensitive information, such as a password, a biometric and a smart mobile device, or the long-term secret y and verifier table. That is, the adversary cannot impersonate the user.

Next, we discuss a cloud center impersonation attack. According to our protocol, both the user and the gateway must 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} that 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 an IoT device. In addition, the cloud center also authenticates the gateway via X_{G_k}, which cannot be acquired by the adversary as mentioned above. That is $\mathcal{A}$ cannot impersonate the gateway.

In conclusion, our scheme can well withstand against impersonation attack.

Proposition: 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 against de-synchronization attack.

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 in 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 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.

APPENDIX E: BAN LOGIC ANALYSIS

BAN logic [5] is a popular method to analyze the security of the session keys in an authentication scheme [1], [6], [7]. It uses a set of notations to idealize the original scheme and define the goals of the scheme, then depicts and reduces the process of the scheme with a series of rules. Note that the notations in BAN logic are described in Table 6.

Firstly, we define our goals of the scheme as follows:

- Goal 1: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- Goal 2: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- Goal 3: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- Goal 4: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.

Secondly, we transform the scheme into an idealized form:

- Mes_1: $\langle M_1, M_2, ID_i, SID_j\rangle_{U_i \overset{k_i}{\longleftrightarrow} CloCen}$.
- Mes_2: $\langle M_2, r, SID_j\rangle_{CloCen \overset{X_{G_k}}{\longleftrightarrow} GWN_k}$.
- Mes_3: $\langle M_2, r_g, SID_j\rangle_{GWN_k \overset{X_{S_j}}{\longleftrightarrow} S_j}$.
- Mes_4: $\langle M_{11}, r_g, SID_j\rangle_{GWN_k \overset{X_{S_j}}{\longleftrightarrow} S_j}$.
- Mes_5: $\langle M_{11}, r, SID_j\rangle_{CloCen \overset{X_{G_k}}{\longleftrightarrow} GWN_k}$.
- Mes_6: $\langle M_{11}, M_1, ID_i, SID_j\rangle_{U_i \overset{k_i}{\longleftrightarrow} CloCen}$.

Thirdly, we make some assumptions:

- H_1: $CloCen \mid\equiv \sharp(M_2)$.
- H_2: $GWN_k \mid\equiv \sharp(r)$.
- H_3: $S_j \mid\equiv \sharp(r_g)$.
- H_4: $GWN_k \mid\equiv \sharp(r_g)$.
- H_5: $CloCen \mid\equiv \sharp(r)$.
- H_6: $U_i \mid\equiv \sharp(M_1)$.
- H_7: $CloCen \mid\equiv CloCen \overset{k_i}{\longleftrightarrow} U_i$.
- H_8: $GWN_k \mid\equiv CloCen \overset{X_{G_k}}{\longleftrightarrow} GWN_k$.
- H_9: $S_j \mid\equiv GWN_k \overset{X_{S_j}}{\longleftrightarrow} S_j$.
- H_{10}: $GWN_k \mid\equiv GWN_k \overset{X_{S_j}}{\longleftrightarrow} S_j$.
- H_{11}: $CloCen \mid\equiv GWN_k \overset{X_{S_j}}{\longleftrightarrow} S_j$.
- H_{12}: $U_i \mid\equiv CloCen \overset{k_i}{\longleftrightarrow} U_i$.
- H_{13}: $U_i \mid\equiv S_j \mid\Rightarrow M_{11}$.
- H_{14}: $S_j \mid\equiv U_i \mid\Rightarrow M_2$.

Now we can analyze our scheme with BAN logic:

From Mes_1, it is easy to get S_1: $CloCen \triangleleft \langle M_1, M_2\rangle_{k_i}$

Then according to H_7, S_1, $RULE(1)$, we get S_2: $CloCen \mid\equiv U_i \mid\sim M_2$

According to H_1, S_2 and $RULE(2)$, we get S_3: $CloCen \mid\equiv U_i \mid\equiv M_2$

From Mes_2, it is easy to get S_4: $GWN_k \triangleleft \langle M_2, r, SID_j\rangle_{X_{G_k}}$

Then according to H_8, S_4, $RULE(1)$, we get S_5: $GWN_k \mid\equiv CloCen \mid\sim \langle M_2, r\rangle$

TABLE 6
Notations in BAN logic

$P \mid\equiv X$	P believes X, ie. principal P believes statement X is true.
$P \triangleleft X$	P sees X, ie. principal P receives a message that contains X.
$P \mid\Rightarrow X$	P has jurisdiction over X, ie. P generates or computes X.
$P \mid\sim X$	P said X, ie. the principal P has sent a message containing X.
$\sharp(X)$	X is fresh, ie. X is sent in a message only at the current run of the protocol, it is usually a timestamp or a random number.
$P \overset{K}{\longleftrightarrow} Q$	K is the shared key for P and Q.
$P \overset{Y}{=\!=} Q$	Y is a secret known to P and Q or principals trusted by them.
$\langle X \rangle_Y$	X combined with Y, and Y usually is a secret.
$\{X\}_K$	X encrypted with K.
$\frac{P\mid\equiv P\overset{K}{\longleftrightarrow}Q, P\triangleleft\{X\}_K}{P\mid\equiv Q\mid\sim X}$ or $\frac{P\mid\equiv P\overset{Y}{=\!=}Q, P\triangleleft\langle X\rangle_Y}{P\mid\equiv Q\mid\sim X}$	$RULE(1)$: the message-meaning rule.
$\frac{P\mid\equiv\sharp(X), P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$	$RULE(2)$: the nonce-verification rule.
$\frac{P\mid\equiv Q\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$	$RULE(3)$: the jurisdiction rule.
$\frac{P\mid\equiv\sharp(X)}{P\mid\equiv\sharp(X,Y)}$	$RULE(4)$: the freshness-conjuncatenation rule.

According to H_2, S_5, $RULE(4)$, and $RULE(2)$ we get S_6: $GWN_k \mid\equiv CloCen \mid\equiv \langle M_2, r\rangle$

From Mes_3, it is easy to get S_7: $S_j \triangleleft \langle M_2, r_g, SID_j\rangle_{X_{S_j}}$

Then according to H_9, S_7, $RULE(1)$, we get S_8: $S_j \mid\equiv GWN_k \mid\sim \langle M_2, r_g\rangle$

According to H_3, S_7, $RULE(4)$ and $RULE(2)$ we get S_9: $S_j \mid\equiv GWN_k \mid\equiv \langle M_2, r_g\rangle$

From Mes_4, it is easy to get S_{10}: $GWN_k \triangleleft \langle M_{11}, r_g, SID_j\rangle_{X_{S_j}}$

Then according to H_{10}, S_{10}, $RULE(1)$, we get S_{11}: $GWN_k \mid\equiv S_j \mid\sim \langle M_{11}, r_g\rangle$

According to H_4, S_{11}, $RULE(4)$ and $RULE(2)$ we get S_{12}: $GWN_k \mid\equiv S_j \mid\equiv \langle M_{11}, r_g\rangle$

From Mes_5, it is easy to get S_{13}: $CloCen \triangleleft \langle M_{11}, r\rangle_{X_{G_k}}$

Then according to H_{11}, S_{14}, $RULE(1)$, we get S_{14}: $CloCen \mid\equiv GWN_k \mid\sim \langle M_{11}, r\rangle$

According to H_5, S_{15}, $RULE(4)$ and $RULE(2)$ we get S_{15}: $CloCen \mid\equiv GWN_k \mid\equiv \langle M_{11}, r\rangle$

From Mes_6, it is easy to get S_{16}: $U_i \triangleleft \langle M_{11}, M_1\rangle_{k_i}$

Then according to H_{12}, S_{16}, $RULE(1)$, we get S_{17}: $U_i \mid\equiv CloCen \mid\sim \langle M_{11}, M_1\rangle$

According to H_6, S_{17}, $RULE(4)$ and $RULE(2)$ we get S_{18}: $U_i \mid\equiv CloCen \mid\equiv \langle M_{11}, M_1\rangle$

According to S_{12}, S_{15}, S_{18}, we have S_{19}: $U_i \mid\equiv CloCen \mid\equiv GWN_k \mid\equiv S_j \mid\equiv M_{11}$

Then, S_{20}: $U_i \mid\equiv S_j \mid\equiv M_{11}$, as SK = $h(M_2||M_{11}||\ r_i M_{11})$, we have

S_{21}: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ **(Goal 1)**

According to H_{13} and $RULE(3)$, we have S_{22}: $U_i \mid\equiv M_{11}$, that is:

S_{23}: $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ **(Goal 2)**

According to S_3, S_6, S_9, we have S_{24}: $S_j \mid\equiv GWN_k \mid\equiv CloCen \mid\equiv U_i \mid\equiv M_2$

Then, S_{25}: $S_j \mid\equiv U_i \mid\equiv M_2\rangle$, as SK = $h(M_2||M_{11}||\ r_j M_2)$, we have

S_{26}: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ **(Goal 3)**

According to H_{14} and $RULE(3)$, S_{27}: $S_j \mid\equiv M_2$, that is:

S_{28}: $S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ **(Goal 4)**

Till now, we have finished our BAN logic analysis. The result shows that our scheme achieves the four goals, meaning that the user and the IoT device securely build a session key.

REFERENCES

[1] 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. Depend. Secur. Comput.*, vol. 17, no. 2, pp. 391–406, 2020.

[2] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inform. Foren. Secur.*, vol. 12, no. 11, pp. 2776–2791, 2017.

[3] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules." in *Pro. IEEE Compuit. Soc. Found. (CSFW)*, vol. 1, 2001, pp. 82–96.

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

[5] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *IEEE Trans. Comput.*, vol. 8, no. 1, pp. 18–36, 1990.

[6] 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 Sys. J.*, vol. 14, no. 1, pp. 39–50, 2020.

[7] R. Ali, A. K. Pal, S. Kumari, M. Karuppiah, and M. Conti, "A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring," *Future Gener. Comput. Sys.*, vol. 84, pp. 200–215, 2018.