Ad Hoc Networks 138 (2023) 102997

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Multi-factor based session secret key agreement for the Industrial Internet of Things

Jie Cui [a], Fangzheng Cheng [a], Hong Zhong [a,*], Qingyang Zhang [a], Chengjie Gu [b], Lu Liu [c]

[a] School of Computer Science and Technology, Anhui University, Hefei 230039, China
[b] Security Research Institute, New H3C Group, Hefei 230000, China
[c] School of Informatics, University of Leicester, Leicester LE1 7RH, UK

ARTICLE INFO

Keywords:
IIoT
Lightweight authentication
Key agreement
Provable security
Unlinkability

ABSTRACT

Industrial Internet of Things (IIoT) is the main field of application of the Internet of Things (IoT). The high degree of autonomy and resource constraints of the IIoT network poses challenges to the security of IIoT. Enabling the legitimate users to securely and remotely access resource-constrained intelligent terminal nodes in an open wireless channel has become a challenge in the IIoT environment. User authentication and key agreement schemes have been proposed for different IoT application scenarios to solve this problem. However, most existing three-party authentication and key agreement schemes for remote users have high computational costs and do not consider sufficient security attributes, such as unlinkability and anonymity. To address the aforementioned problems, the proposed scheme uses the IIoT as the application scenario and proposes a novel three-factor authentication and lightweight remote user identity authentication key agreement. This scheme is effective for devices with limited resources because it mainly uses a one-way hash function and a bitwise XOR operation. The security of the proposed scheme is demonstrated under the real-or-random model via a rigorous formal security analysis.

1. Introduction

The Internet of Things (IoT), that is, the Internet of Everything, refers to an extended and expanded network based on the Internet. It consists of many information-sensing devices that can be remotely accessed and controlled whenever desired through the Internet, in order to achieve an interconnection among users, machines, and objects regardless of their locations [1]. The Industrial Internet of Things (IIoT) [2,3] is one of the main applications of IoT. In the Internet environment, most IoT devices or nodes have the ability to process information and communication and possess a locatable Internet protocol address (IP address) but suffer from limited resources. For IoT devices in different IoT environments, users can access and control them through a network [4]. Industry 4.0 refer to the fourth industrial revolution led by intelligent manufacturing. The factory integrates production equipment, wireless signal connections, and sensors into an ecosystem to autonomously monitor the entire production process and execute decisions [5,6]. The high degree of autonomy and resource constraints of the IIoT network pose challenges to the security of IIoT [7,8]. Authentication and key establishment are important components of IIoT, and key agreement must consider factors such as security and performance [9,10]. Privacy, message integrity, and user authentication are vital in the IIoT environment because the adversary can eavesdrop, modify and forge communication messages [11]. As a result, it is necessary to adopt an appropriate security scheme to protect the communication link [12].

Existing schemes have proposed some security requirements to ensure the security and privacy of the system. First, the anonymity of users needs to be guaranteed. If it is not, the adversary may threaten the personal security of the user when unable to successfully attack the system. To protect user anonymity, Li et al. [13] sets up a pseudonym to protect the real identity of the user. Wazid et al. [14] encrypts the user's identity in the communication message, and the gateway calculates and fetches second user's identity. Second, unlinkability must be guaranteed. Otherwise, the adversary can trace the user by eavesdropping on the key agreement process and can also link to the user participating in the session from multiple session messages to identify the real legitimate user. This poses a threat to the privacy and security of users. Lyu et al. [15] proposed an anti-tracking remote user authentication and key agreement scheme. In this scheme, the user will directly use the pseudonym of the device in each session. However, after multiple sessions, an adversary can link to one or more remote

* Corresponding author.
E-mail address: zhongh@ahu.edu.cn (H. Zhong).

https://doi.org/10.1016/j.adhoc.2022.102997
Received 15 December 2021; Received in revised form 23 August 2022; Accepted 13 September 2022
Available online 22 September 2022

users who use the same device. Therefore, the user's unlinkability cannot be guaranteed. Zhang et al. [16] uses unique random numbers and each pseudo identity of each signature is unique in the authentication process. This scheme supports unlinkability. Apart from this, there are various attack possibilities in the IIoT environment [14,17]. Other attacks include internal privilege attacks, device capture attacks, and tracking and linking users.

1.1. Motivation

The communication and computational overheads of the session key scheme are used to measure whether the scheme is lightweight enough. In order to ensure that small devices with limited computing and storage resources can also securely interconnect with legitimate visitors, a sufficiently lightweight session key agreement scheme is required. To design a lightweight session key agreement scheme, it is necessary to reduce calculation and communication overheads, and improve the efficiency and versatility in practical applications. And the scheme should meet the necessary security attributes, ensure session security, and improve the security level. In terms of functionality, it meets more specific operations in accordance with actual operational feasibility and necessity, such as supporting cancellation of users, dynamically adding devices and improving the operability of the scheme.

1.2. Research contributions

The main contributions of the proposed scheme are as follows:

(1) We propose a novel session key agreement scheme for protecting remote user authentication of the IIoT. The scheme uses smart cards, passwords and biometrics so it is a three-factor authentication scheme. There will be three types of mutual authentication in this scheme: (a) between a user and the sever; (b) between the sever and the smart terminal node; (c) between the user and the smart terminal node. Finally, a symmetric session key is established between the user and the smart terminal node.
(2) The proposed scheme is lightweight by using fuzzy extractor, one-way hash function and simple bit operation XOR. This is suitable for terminal nodes with limited computing and storage resources.
(3) We perform a formal security analysis using the widely accepted real-or-random (ROR) model. Formal security analysis proves the semantic security of the adversary's acquisition of the session key between the user and the device in the proposed scheme.
(4) The proposed scheme use proxy re-encryption to reduce communication and computational overhead, and simplifies the key agreement process. The user's pseudo-name and private key are dynamically updated during each session key agreement to ensure one-time pad. Our scheme guarantee the anonymity and unlinkability of users.

1.3. Paper organization

The rest of this paper is structured as follows. Section 2 outlines the existing related schemes for different IoT application scenarios. Section 3 introduces the system model and threat model related to the session key agreement scheme. Section 4 discusses some preliminary knowledge. Section 5 describes in detail our scheme. In Section 6, we give a formal and informal security analysis. In Section 7, we compare the proposed scheme with other existing schemes to measure efficacy. Finally, we summarize the work of this paper.

2. Related work

In this section, we discuss lightweight authentication and key agreement schemes for two-party authentication and three-party authentication, as well as authentication schemes that use different cryptographic algorithms. In recent years, there have been many security authentication schemes for IoT environments, such as IIoT environments, smart home environments, and vehicular ad hoc networks (VANETs) environments.

2.1. Two-party authentication and key agreement

M2M (Machine to Machine) is often mentioned in the field of IoT. M2M is the transfer of data from one terminal to another, that is, a machine-to-machine dialogue. In the session key agreement scheme for two-party, Esfahani et al. [18] designed a lightweight authentication scheme for M2M for the IIoT based on the IoT. Because only the authentication of two parties is considered, in the IIoT environment, under the premise that the router is a trusted entity, the authentication of the device is definitely more lightweight than the multi-party authentication and key agreement scheme. In the IIoT environment, although the scheme is light enough, it does not consider other security threats, such as internal privilege attacks.

Kumar et al. [19] designed a two-party authentication scheme for smart home based on the IoT. Kumar et al. [19] is more computationally expensive than the scheme proposed by Esfahani et al. [18], because only lightweight cryptographic operations are used in [18]. And the security attributes of Kumar et al. [19] are not fully considered.

2.2. Three-party authentication and key agreement

In order to meet the requirements of remote access to the IoT devices, some three-party mutual authentication and session key agreement schemes have proposed. Remote users and devices can negotiate a session key through a trusted third party for secure communication. In 2018, Wazid et al. [14] proposed a three-party session key agreement scheme for the general IoT environment. This scheme through multiple communication messages encryption/decryption to obtain key information for authentication. However, multiple encryption/decryption operations bring more computational overhead compared to XOR, MAC, and hash operations. In 2020, Wazid et al. [20] proposed a three-party session key agreement scheme for a smart home environment based on the IoT. The user entity uses a mobile phone to save personal registration information. Different from Wazid et al. [14], this scheme updates the personal pseudonym for each session key agreement with the key agreement process, instead of using modified encrypted pseudonyms each time, which increases the communication overhead of this part. Banerjee et al. [21] proposed a lightweight session key agreement scheme for the IoT environment. In order to obtain more functionality, multiple judgments and calculations have been added at the gateway, including user revocation and device list update, which increases the cost of each session key agreement process. In this scheme, the user's revocation function is added in the key agreement phase. After the gateway performs the revocation calculation operation, the key agreement process continues. Only during the next key agreement, the gateway will verify that the user has been revoked. Therefore, the independent and real-time user revocation phase have practical significance. Shao et al. proposed an anonymous authentication scheme for VANETs [22]. In this scheme, the trusted organization will update the revocation list to the Road Side Unit (RSU) in real time.

Li et al. [13] proposed an authentication and session key agreement scheme based on elliptic curve cryptography (ECC). The scheme uses public key technology to enhance security, but the point multiplication operation increases computational overhead of terminal device. The scheme also do not consider more operability, such as user revocation and dynamic addition of device. Lyu et al. [15] proposed a session key

agreement scheme based on elliptic curve cryptography for the smart home environment. The overall cost of this scheme is relatively high, and the server node does not directly participate in the session key agreement. For the entire IoT system, this scheme assigns all smart devices under the smart home to the gateway for management. This scheme cannot achieve authentication between the user and the device.

2.3. More discussions on cryptographic algorithms

In the scheme of using other cryptographic methods, they used the Chinese remainder theorem (CRT), Shamir(k,n) threshold function, physical unclonable function (PUF), etc. Zhang et al. [16] proposed a vehicle authentication scheme based on the CRT under the VANETs. In this scheme, the vehicle does not need to preload the master private key. Trusted institutions can dynamically generate new group keys and spread them to vehicles in the network by using the CRT. This scheme focuses on the authentication and key distribution of the IoT devices (vehicles) in the area, and does not involve third-party users. Eldefrawy et al. [23] proposed an M2M mutual authentication scheme based on the CRT, and the device can restore the key based on the information. Same as other M2M schemes, because only the device-to-device secure communication is considered, the user level is not considered.

Aghili et al. [24] proposed a session key agreement scheme based on Shamir(k,n) threshold key sharing. In the session key agreement process, in order to security complete the mutual authentication between the gateway and the device and transmit the response message, the device have undergone multiple calculations and sending messages in a single key agreement process. Gope et al. [25] proposed a lightweight and physically secure anonymous mutual authentication scheme using PUF. In this scheme, during the registration phase, the user's personal device and IoT device use PUF to generate a unique challenge–response pair (CRP). The gateway saves the CRP of all parties as an authentication credential during the key agreement phase. In the key agreement process, the user needs to perform a round of verification, which increases the communication and computational overheads.

Chen et al. [26] proposed a symmetric key-based RFID device authentication scheme, which utilizes dynamically updating node identities to achieve the idea of anonymity and unlinkability. Arapinis et al. [27] proposed to use the public key of the home network to encrypt different message types of the user's response to the service network, making them indistinguishable, as a protection against misuse of messages. However, the unlinkability can be broken by an attacker with a simple replay attack on the identified message. An achieving better privacy authenticated key agreement was proposed by Fouque et al. [28]. This scheme is shown to be insecure, subject to desynchronization attacks and unlinkability attacks. Shin et al. [29] proposed an anonymous authenticated key agreement scheme between users and IoT gateways in IoT environments. However, if the server's secret key is compromised, the pseudonym of the same user is linkable, thus failing to satisfy the forward unlinkablely requirement. Li et al. [30] proposed an anonymous authenticated key agreement scheme between vehicles using homomorphic encryption. Vehicles first attempt to authenticate themselves anonymously to the RSU using a pseudonym and ticket, and then the vehicles attempt to anonymously authenticate to each other. However, it is easy to check if two tickets belong to the same car. Therefore, the scheme does not satisfy forward unlinkability. Khan et al. [31] proposed an anonymous authentication key agreement scheme between nodes and hub nodes in a wireless body area network. However, since a node's secret information (or identity) is made public, it is easy to tell if a scheme message was generated by that node. Therefore, this scheme does not satisfy forward unlinkability.

In summary, in most key agreement schemes, either the key agreement process is not lightweight enough to meet the requirements of more resource-constrained devices or sensor nodes, or it cannot meet the enough security in the IoT environment. In terms of functionality, most of the schemes do not meet the more ideal functional characteristics, such as revocation users, dynamically add devices, and dynamically change personal related keys information including biometrics and passwords. In order to better resolve these differences and adapt to IoT systems that require more lightweight agreement schemes, we are committed to designing a novel lightweight user authentication and session key agreement scheme suitable for the IIoT environment, using three factors authentication ensures the anonymity and unlinkability of users.

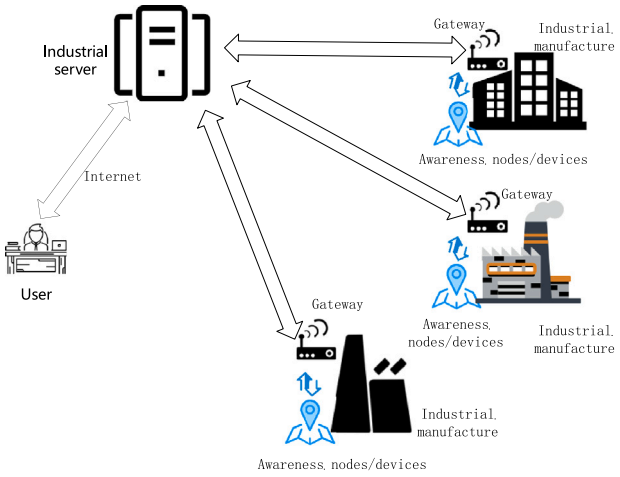


Fig. 1. Authentication model of the IIoT.

3. System models

In order to discuss and analyze our proposed scheme, we introduce the system model and threat model of the scheme in this section.

3.1. System model and network model

The system model shown in Fig. 1 is composed of industrial devices, industrial gateways, industrial servers and legitimate users. In an industrial environment, the underlying terminal device is used to collect real-time information, detect and monitor the real-time situation of the industry. Industrial devices are terminal devices in various IIoT working environments, such as sensor nodes and smart devices. These devices have low computing resources and storage resources. Because these devices are susceptible to physical capture, they are considered semi-trusted entities. The industrial gateway is a cluster head node in a specific IIoT area. The gateway is responsible for controlling network data, network interoperability and security management. The gateway and the terminal device in the area are only functionally different. The gateway is also a special terminal device. Industrial servers are fully trusted entities in industrial environments. The server is responsible for managing the relevant information of the devices and users in the entire industrial environment. A legitimate user refers to the management and operation personnel of a specific industrial area. Users are also semi-trusted entities. If a user needs to access a specific industrial device (including gateways and smart industrial devices) to obtain the status and data of the device, the user first needs to register himself as a legitimate user on a trusted industrial server. Similarly, all industrial devices and industrial gateways need to be registered on the industrial server before being deployed to the entire IIoT environment.

There are three types of communication in the network model of this scheme: (1) the communication between the user and the server; (2) the communication between the server and the device; (3) the communication between the device and the user. To achieve secure communication, secure key management between different devices is required. Only after the key management process has been successfully

executed can keys for secure communication be established between entities. For example the user and the accessed device should authenticate each other with the help of the server. After the mutual authentication between the user and the device is successful, the two parties will establish a session key for secure communication.

3.2. Threat model

In this section, we discuss the threat models that are widely used in existing schemes. We consider a recently described threat model for IoT security in [32].

Same as most authentication schemes, our scheme first considers the Dolev–Yao threat model [33]. In this threat model, the communications of the participants in the key agreement are all carried out on insecure channels. Adversary can eavesdrop on all messages in the communication, and can modify or delete them after intercepting them. The adversary can send the modified or directly forged message to the participants in order to obtain a favorable return. The adversary can forge himself as a legitimate participant in the agreement of the key and actively try to participate in the session key agreement.

The CK-adversary model [34] is considered to be the standard model of the key exchange scheme. The adversary under this model is more threatening than the adversary under the Dolev–Yao model. In the CK-adversary model, the adversary can not only fully control the communication link between the communication subjects and the scheduling of the scheme practice, but also obtain the private information of the participating subjects and related sessions through a series of queries. Because the adversary can damage the session information, including session state, session key and private key, it is necessary to ensure that when some forms of private information in the session are leaked, the security impact on other private information of the participating communicating entities is minimal. Therefore, the main consideration of the CK-adversary model is the security threat to other secret credentials of the communicating entity in the scheme caused by the leakage of some forms of private information.

In addition to the threat of communication messages, there are also threats to the privacy and security of user entity information. The adversary can make an educated guess about the user's identity and password. In the case where the user does not use a simple password or username, the adversary guesses the user's password and identity, and verifies his guess in polynomial time, from computationally speaking, it is difficult for an adversary to complete [35].

We also need to consider security threats from IoT devices. Similar to other scheme considerations, we assume that an adversary can use physical entities to capture some smart devices of the IoT, and then extract sensitive information from the device memory to gain the advantage of invading the entire system [36]. Adversary can use power analysis to obtain user-related information stored in the smart card [37].

In Section 6, we conduct the security analysis for the proposed scheme and prove that our scheme can resist security threats from the threat model.

4. Basic preliminaries

This section discusses the necessary prerequisite preparations required when designing and analyzing the scheme.

4.1. Fuzzy extractor and smart card

The smart card is a portable device and have certain computing capabilities. Its main function is data transmission, storage and processing. In processing data, the computing power of the smart card ensures that it runs a cryptographic algorithm, making the smart card a highly mobile and highly secure module. The novel smart cards are energy-efficient and security, and can even withstand differential power analysis (DPA) attacks [38]. The security of the smart card can be protected by the user's password and the biometric key extracted from the user's biometrics by the fuzzy extractor.

In the field of cryptography, the secret value of a cryptographic mechanism is generally a random string, which is required to be uniformly distributed, and the random string can be accurately restored when needed. In recent years of authentication schemes, fuzzy extractors are usually used to convert biometrics into uniformly distributed random numbers required in cryptographic systems [39].

Gen : Input w, output auxiliary data P and uniform random value R.

Rep : Given P, input w', and regenerate a uniform random value R.

Correctness: If $dis(w, w') \leq t$, an accurate R can be reconstructed; if $dis(w, w') > t$, no guarantee is provided for the output of Rep.

Security: The auxiliary data P will not reveal too much information about R; the distribution of R is close to even distribution.

Application: The R extracted from w can be used as a key, but it does not need to be stored. You can restore it from w' next time you use it. That w can be biological fingerprints, physical fingerprint PUF, or other cryptographic materials (such as passphrases that the user does not remember clearly), etc.

4.2. One-way hash function

The one-way hash function has been used in computer science for a long time. The following is the definition of the collision resistance of the hash function.

$Definition$: A one-way hash function: $h : \{1,0\}^* \rightarrow \{1,0\}^n$, which can accept input of any length, say $x \in \{1,0\}^*$ and output a fixed-length (n-bits) message digest $h(x) \in \{1,0\}^n$. $Adv_{\mathcal{A}}^{hash}(t)$ is defined as the advantage of the adversary $\mathcal{A}$ in detecting hash collisions within the run time t. $Pr[X]$ represents the probability of an event X and $(xp_1, xp_2) \in_R \mathcal{A}$ represents $\mathcal{A}$ randomly selected the pair (xp_1, xp_2). $Adv_{\mathcal{A}}^{hash}(t) = Pr[(xp_1, xp_2)] \in_R \mathcal{A} : xp_1 \neq xp_2$ and $h(xp_1) = h(xp_2)$. If an (ψ, t)-adversary $\mathcal{A}$ tries to find a hash collision of $h(\cdot)$, this means that the maximum running time of $\mathcal{A}$ is t and that $Adv_{\mathcal{A}}^{hash}(t) \leq \psi$.

5. Proposed scheme for IIoT

In this section, we give a detailed description of our proposed scheme. Our scheme is divided into the following steps: (1) server initialization phase; (2) smart device registration phase; (3) user registration phase; (4) user login phase; (5) identity authentication and key agreement phase; (6) password and biometrics update phase; (7) dynamically add smart device phase; (8) user revocation phase. We can see a schematic diagram of these phases in Fig. 2. In our scheme, the server deployed in the smart industry is initialized and the necessary information settings are completed. Then, each smart IoT device deployed in the IIoT environment is registered through the industrial server before deployment, and the necessary information is preloaded at the same time. In the device registration phase, the server generates an identity RID_j, device private key Ksd_j for each device SD_j. This information is required during the certification phase. The server will generate a device list $SDList$ for each IIoT smart device in the working environment of the workshop or the same area, and save it.

When a legitimate user or employee U_i of a factory needs to communicate with a smart device SD_j in a specific workshop, U_i must first register himself as a legitimate and valid user with the server through the registration phase. Store information in its own smart card on a security, operable and perceptible platform. After the user U_i successfully logs in to the security operation platform, the server sends an authentication request to the corresponding device through its own gateway in the authentication and key agreement phase, and the user obtains an identity verification response from the corresponding device. After this phase is over, a shared SK is generated between the user and the device, which is convenient for future secure communication. For

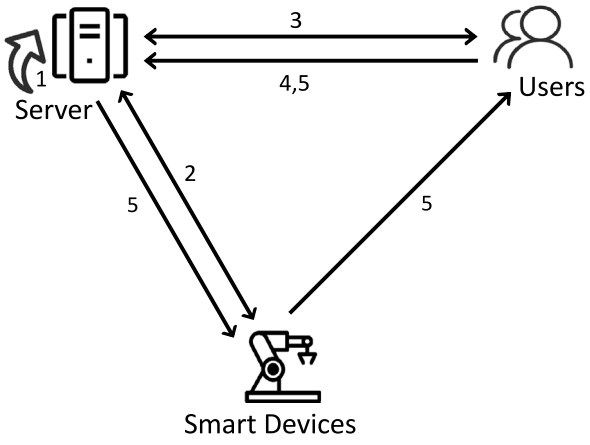


Fig. 2. The authentication and key agreement process of our scheme.

Table 1
Symbolic identification of the scheme.

Symbol	Description
$Server$	Industrial Internet of Things Central Server
U_i, SC_i	ith user,i-th user's smart card
ID_i, ID_j	The identity of the ith user/j-th device
RID_i, RID_j	ith user's/j-th device's pseudo-identity
EID_i	ith user's revoke-identity
$h(\cdot)$	Collision-resistant one-way hash function
$Gen(\cdot)$	Fuzzy extractor probabilistic generation function
$Rep(\cdot)$	Fuzzy extractor deterministic reproduction function
σ_i	Biometric secret key of U_i
τ_i	Public reproduction parameter of U_i
S	Server private key for a specific IoT area
Ksu_i, Ksd_j	The private key of the ith user/j-th device
t	Error tolerance threshold used in fuzzy extractor
T_i	Timestamp used in sequence during key agreement
$\triangle T$	Maximum transmission delay in communication
PW_i, BIO_i	ith user's password and biometric information
$\|\|, \oplus$	Concatenation,bitwise XOR
r_u, r_d, r_s	Random secrets generated by user/device/server
SK	Negotiated session key
Ksu_i^{new}	New ith user's private key
Ksu_i^{TC}	ith user's temporary authentication credential
$SDList$	List of device information in designated IoT areas
Mi, Mui	ith communication message in the key agreement process
MQi	ith message queue of the key agreement process
r_i, r_j	Random secrets generated by user and server respectively
TPW_i	ith user authentication signature stored in the smart card

security and other considerations, users may update their passwords and personal biometrics. In the password and biometric update phase, users can operate on a security and operable perception platform. For normal device replacement, or other device replacement and addition operations, it can be completed in the dynamic device update phase.

The symbols in the scheme and their descriptions are shown in the Table 1. We will use these symbols to describe the scheme in detail.

5.1. Server initialization phase

In the server initial setup phase, the $Server$ sets up public parameters. $Server$ selects the probabilistic generation function $Gen(\cdot)$ and deterministic reproduction function $Rep(\cdot)$ for biometric extraction. And the $Server$ also selects the one-way hash function $h(\cdot)$. $Server$ establishes various data tables for subsequent phases. The $Server$ also selects a 160-bit long-term secret key S as private key for this area for each independent IIoT area, which is only known to the server.

5.2. Smart device registration phase

After the main server has been initialized, the IoT device can register itself at any time and deploy it in its own work area.

In a security offline state, the device registers with the $Server$. The $Server$ generates a random value r_j, and then uses the random value to calculate the pseudonym $RID_j = h(ID_j \| r_j)$. The $Server$ uses the private key (S) of the IoT network area in the server memory to calculate the device private key $Ksd_j = h(RID_j \| S)$. The $Server$ saves the device pseudo-name and device private key in the memory of the smart device or sensor node. The $Server$ creates a device list $SDList$ for each IoT smart device and sensor node working in the same IIoT network environment.

5.3. User registration phase

At this phase, legitimate users register through the central server. A legitimate user U_i gets the legitimate use right of a specific IIoT working area. In a secure registration environment, U_i generates a random number r_i and calculates his pseudonym $RID_i = h(ID_i \| r_i)$. The $Server$ recognizes the legality of the pseudo-identity, and uses the private key S of the area legally used by the user to calculate the U_i's private key $Ksu_i = h(RID_i \| S)$. The private key and the list of devices $SDList$ in the area are distributes to U_i through smart card SC_i.

The U_i related information such as ID_i, RID_i and Ksu_i is stored securely in the $Server$'s memory. After the user U_i obtains the registration information and the smart card SC_i, U_i collects personal biometric information BIO_i from a specific terminal sensor, and uses the fuzzy extractor probabilistic generation function $Gen(\cdot)$ as $(\sigma_i, \tau_i) = Gen(BIO_i)$ to obtain σ_i and τ_i. Then U_i sets the password (PW_i) and calculates the authentication token $TPW_i = h(ID_i \| PW_i \| \sigma_i)$. U_i stores all kinds of information in the smart card after processing, including $Ksu_i^* = Ksu_i \oplus h(PW_i \| ID_i \| \sigma_i)$; $RID_i^* = RID_i \oplus h(PW_i \| \sigma_i \| ID_i)$; $SDList^* = SDList \oplus h(\sigma_i \| ID_i \| PW_i)$; TPW_i; τ_i.

5.4. User login phase

In order to security access and control devices and sensor nodes, registered legitimate users must log in and perform identity verification on the device point or remote security operation platform. After the verification is completed, the user will negotiate with the device secret key. U_i enters identity ID_i, password PW_i and collects biometrics. The biometrics collected again can be recovered by the $Rep(\cdot)$ function within the threshold t. SC_i calculates $TPW_i' = h(ID_i \| PW_i \| \sigma_i)$ and compares it with the TPW_i stored on the smart card to determine whether it is the owner of the smart card SC_i. If the identity is confirmed, U_i can access the information stored in the smart card and can proceed to the next step to negotiate the key. We can see the main process of authentication and key agreement in Fig. 3. In Fig. 3, as indicated by the arrow, the user sends the message queue MQ1 to the server. The server sends the response message queue MQ2 to the device. The device sends the reply message queue MQ3 to the user.

5.5. Identity authentication and key agreement phase

After U_i logs in, U_i can secure proceed to all the next phases. In this phase, the RID_i, PW_i and σ_i provided by the U_i are confirmed to

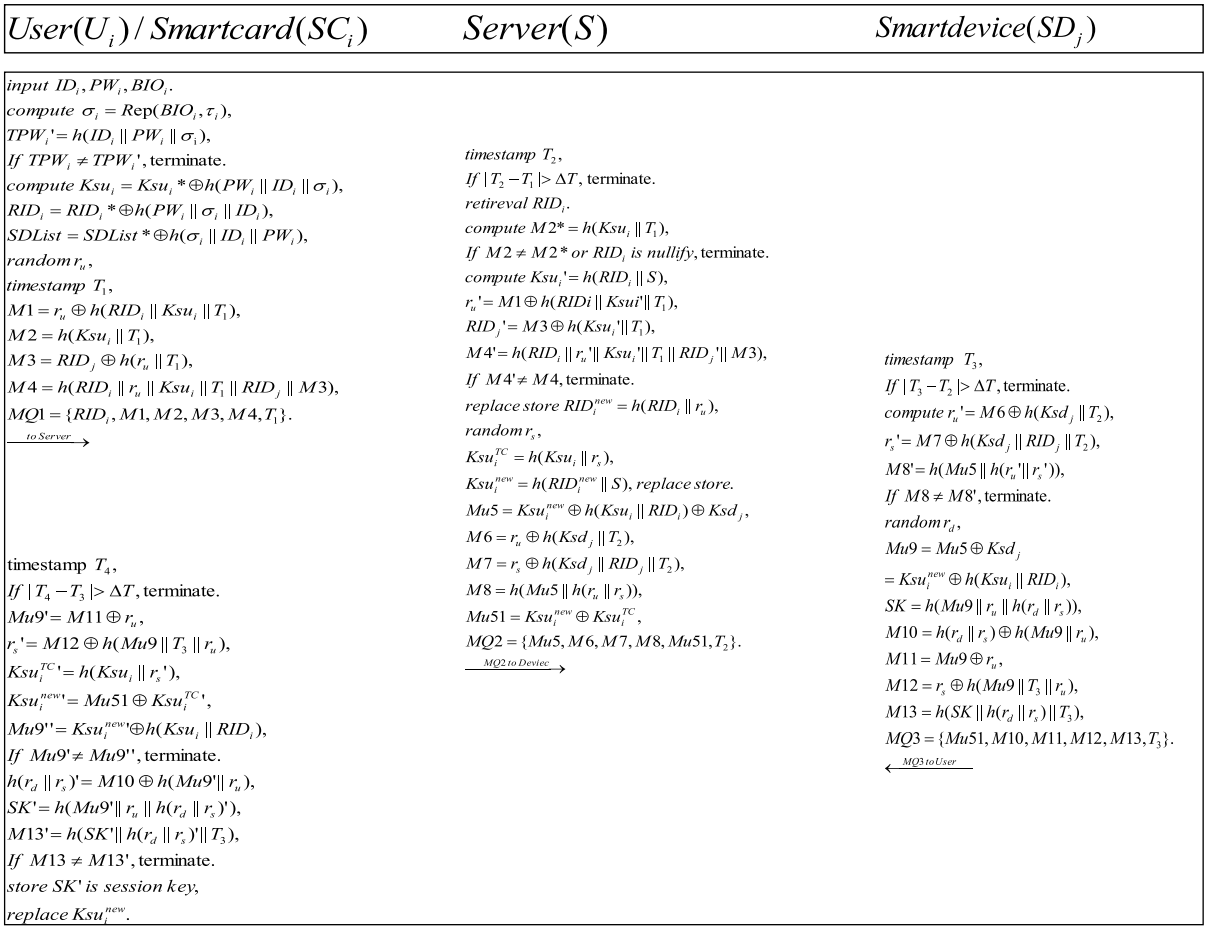


Fig. 3. Identity Authentication and Key agreement Phase.

be correct. SC_i then calculates $Ksu_i = Ksu_i^* \oplus h(PW_i \| ID_i \sigma_i)$; $RID_i = RID_i^* \oplus h(PW_i \| \sigma_i \| ID_i)$; $SDList = SDList^* \oplus \mathrm{h}(\sigma_i \| ID_i \| PW_i)$. Then SC_i generates a random nonce r_u and the current timestamp T_1. SC_i selects device pseudonym RID_j and calculates parameters $M1 = r_u \oplus h(RID_i \| Ksu_i \| T_1)$, $M2 = h(Ksu_i \| T_1)$, $M3 = RID_j \oplus h(r_u \| T_1)$, $M4 = h(RID_i \| r_u \| Ksu_i \| T_1 \| \ RID_j \ \| \ M3)$. Finally, U_i sends the request message $MQ1 = \{RID_i, M1, M2, M3, M4, T_1\}$ to *Server* via an open channel.

When the *Server* receives the request message $MQ1$, *Server* first uses the current timestamp T_2 to detect the freshness of the message. Only when $|T_2 - T_1| \leq \triangle T$ is satisfied, *Server* will perform the following operations, $\triangle T$ is the maximum allowable transmission delay. If it passes, *Server* retrieves the U_i's information and uses the stored user private key to calculate $M2' = h(Ksu_i \| S)$. When the calculated $M2' \neq M2$ or the user information is not retrieved, the *Server* will terminate. After passing the verification, *Server* calculates r_u' and RID_j' from M1 and M3, and further calculates $M4' = h(RID_i \| r_u' \| Ksu_i \| T_1 \| RID_j' \| M3)$. If $M4' = M4$, it is proved that the parsed r_u' and RID_j' have not been tampered with and are correct. Then the *Server* generates a random value r_s, and calculates the temporary user private key $Ksu_i^{TC} = h(Ksu_i \| r_s)$, $Mu5 = Ksu_i^{new} \oplus h(Ksu_i \| RID_i) \oplus Ksd_j$, $M6 = r_u \oplus h(Ksd_j \| T_2)$, $M7 = r_s \oplus h(Ksd_j \| RID_j \| T_2)$, $M8 = h(Mu5 \| h(r_u \| r_s))$, $Mu51 = Ksu_i^{new} \oplus Ksui^{TC}$. And *Server* packs all the information into $MQ2 = \{Mu5, M6, M7, M8, Mu51, T_2\}$ and sends it to the device RID_j. $Mu5$ is a message that needs to be re-encrypted by the device, and the decrypted message by the user will be verified to see if it has been modified.

After the device SD_j receives the message queue $MQ2$, SD_j first uses the current timestamp T_3 to detect the freshness of the message through $|T3 - T2| \leq \triangle T$. If it passes the test, SD_j uses the device private key to get r_u' and r_s' from the messages $M6$ and $M7$. SD_j calculates $r_u' = M6 \oplus h(Ksd_j \| T_2)$, $r_s' = M7 \oplus h(Ksd_j \| RID_j \| T_2)$. Then SD_j uses the calculated value to calculate $M8' = h(Mu5 \| h(r_u' \| r_s'))$, and pass the judgment $M8' \neq M8$, verify whether the message has been tampered with. Then SD_j generates a random value r_d, and calculates $Mu9 = Mu5 \oplus Ksd_j = Ksu_i^{new} \oplus h(Ksu_i \| RID_i)$, uses the random value generated by the three parties and $Mu9$ together to calculate the session key $SK = h(Mu9 \| r_u \| h(r_d \| r_s))$, and saves it. SD_j calculates the return message and sends it to the user U_i. SD_j calculates $M10 = h(r_d \| r_s) \oplus h(Mu9 \| r_u)$, $M11 = Mu9 \oplus r_u$, $M12 = r_s \oplus h(Mu9 \| T_3 \| r_u)$, $M13 = h(SK \| h(r_d \| r_s) \| T_3)$. SD_j sends message queue $MQ3 = \{M10, M11, M12, M13, T_3\}$ to U_i.

After receiving the message queue $MQ3$ from the device SD_j, SC_i uses the current timestamp T_4 to detect the freshness of the message through $|T4 - T3| \leq \triangle T$. After the verification is passed, SC_i calculates $Mu9' = M11 \oplus r_u$, $rs' = M12 \oplus h(Mu9' \| T_3 \| r_u)$, $Ksu_i^{TC'} = h(Ksu_i \| r_s')$, $Ksu_i^{new'} = Mu51 \oplus Ksu_i^{TC'}$, $Mu9'' = Ksu_i^{new'} \oplus h(Ksu_i \| RID_i)$. By judging $Mu9'' = Mu9'$ to verify that the messages in the message queue are not tampered with.

5.6. Password and biometrics update phase

In order to simplify user operations and reduce server usage, legitimate users can update their passwords and biometrics locally at any time. In a secure operating environment, U_i reads the smart card SC_i through a card reader and provides his own ID_i, old password PW_i^{old} and old biometrics BIO_i^{old}. SC_i calculates $\sigma_i^{old} = Rep(BIO_i^{old}, \tau_i)$, further calculates $TPW_i^{old''} = h(ID_i \| PW_i^{old} \| \sigma_i^{old})$. By verifying whether $TPW_i^{old'}$ is equal to the TPW_i on the smart card, determine whether to

perform the following operations. When the verification is completed, SC_i calculates $Ksu_i' = Ksu_i^* \oplus h(PW_i^{old}\|ID_i\|\sigma_i^{old})$, $RID_i' = RID_i^* \oplus h(PW_i^{old}\|\sigma_i^{old}\|ID_i)$, $SDList' = SDList^* \oplus h(\sigma_i^{old}\|ID_i\|PW_i^{old})$.

After U_i gets the next instruction of SC_i, U_i enters the new password PW_i^{new}, and enters the new biometric information BIO_i^{new} to calculate $Gen\ (BIO_i^{new})$ get $(\sigma_i^{new}, \tau_i^{new})$. SC_i uses the new password PW_i^{new} and biometrics σ_i^{new} to calculate separately $Ksu_i^{*new} = Ksu_i \oplus h(PW_i^{new}\|ID_i\|\sigma_i^{new})$; $RID_i^{*new} = RID_i \oplus h(PW_i^{new}\|\sigma_i^{new}\|ID_i)$; $SDList^{*new} = SDList \oplus h(\sigma_i^{new}\|\ ID_i\ \|\ PW_i^{new})$; $TPW_i^{new} = h(ID_i\|PW_i^{new}\|\sigma_i^{new})$.

Finally, the smart card SC_i replaces the Ksu_i; RID_i; $SDList$; TPW_i; τ_i in the memory with Ksu_i^{*new}; RID_i^{*new}; $SDList^{*new}$; TPW_i^{new}; τ_i^{new}. At this point, the password and biometric update are complete. U_i can also only update the password or biometrics, but for security and biometric accuracy considerations, it is recommended to update the password and biometrics regularly.

5.7. Dynamically add smart device phase

In order to conveniently and directly deploy new smart devices into the work area of a specific IoT environment, we only need to perform the following two steps each time:

$Step1$: First, register IoT devices from different manufacturers on the $Server$. $Server$ uses the random secret value r_j^{new} generated by itself to encapsulate the SD's unchangeable identity ID_j^{new}, gets a pseudo-name identifier $RID_j^{new} = h(ID_j^{new}\|r_j^{new})$ that is different from the existing node. Then $Server$ uses the private key S of the work area where the IoT device will be deployed, calculates the private key $Ksd_j^{new} = h(RID_j^{new}\|S)$ of the IoT device. $Server$ stores the registration information in the memory of the new IoT device. And $Server$ updates the device information into the device list in this area.

$Step2$: Deploy IoT devices in their specific work area, notify legitimate users in the area about the deployment of new devices, and user updates the device list security so that legitimate users can communicate with new devices to obtain access control and services.

5.8. User revocation phase

In a large-scale industrial environment in actual applications, in order to ensure that the specific implementation steps are traceable, and for the record of the operation, the industrial server will register and record the legitimate users who have participated in the session communication and authorized. For all registered executable users, $Server$ can modify their legality and revoke their executable authorization. $Server$ can use the direct long-term key to re-encrypt and encapsulate the revoked user's name $EID_i = DID_i \oplus S$, and the user's ID after encapsulation is still stored in the authorization list of the IoT area as a record and certificate. In the session key agreement phase, when the user sends a request message $MQ1 = \{RID_i, M1, M2, M3, M4, T_1\}$ to the $Server$, $Server$ will verify whether the user still has legitimate authorization and retrieve whether the user is authorized User list. If the user has been revoked, $Server$ will not retrieve the pseudonym information of the user in the list at this time, and the request information sent by the revoked user will not receive a response.

6. Security analysis

In this section, we use formal and informal methods to analyze the security robustness of the proposed scheme. We use the ROR model to conduct a mathematical-based formal security analysis of the scheme, and prove the strong forward security of the scheme, resisting password guessing attacks, resisting message forgery attacks. In Section 6.2, we will demonstrate the security of the scheme under the threat model of Section 3.2.

6.1. Formal security using ROR model

In this section, we will formally prove the security of our proposed scheme under the ROR model [40]. It proves that our scheme protects the session key security (SK security).

6.1.1. ROR model

The main participants in this model are: user U_i, central server $Server$ and device node SD_j.

$Participants$: We use $\Pi_{U_i}^u$, Π_{Server}^v, $\Pi_{SD_j}^t$ to represent instances u, v, t of U_i, $Server$ and SD_j. They are also called oracles.

$Partnering$: When $\Pi_{SD_j}^t$ and $\Pi_{U_i}^u$ are in mutual authentication, $\Pi_{SD_j}^t$ of SD_j is the partner of $\Pi_{U_i}^u$ of U_i and vice-versa.

$Freshness$: Under the given reveal (Π^t) query, if the adversary cannot know the session key SK between U_i and SD_j, we call $\Pi_{U_i}^u$ or $\Pi_{SD_j}^t$ is fresh.

$RandomOracle$: All communication participants and adversaries $\mathcal{A}$ can access the one-way hash function $h(\cdot)$. And $h(\cdot)$ is modeled as a random oracle, say $\mathcal{HS}$.

$Adversary$: Under the ROR model, the adversary $\mathcal{A}$ can control the communication messages of all participants, such as modifying and deleting all information in transmission.

$Execute(\Pi_{U_i}^u, \Pi_{Server}^v, \Pi_{SD_j}^t)$: $\mathcal{A}$ can obtain the information transmitted between the participants U_i, $Server$, and SD_j in the three-party communication by performing this operation, and further model it as an eavesdropping attack.

$Reveal(\Pi^t)$: By performing this operation, $\mathcal{A}$ can know the current session key SK jointly generated by Π^t and its partners.

$Send(\Pi^t, msg)$: Through this operation, $\mathcal{A}$ can send a message to the instance Π^t and can receive a response message from the receiver. And it can be modeled as an active attack.

$CorruptSC(\Pi_{U_i}^u)$: This query indicates that the smart card is lost/stolen, the information stored in the smart card is exported, and the query is modeled as a smart card lost/stolen attack.

$CorruptSD(\Pi_{SD_j}^t)$: The query simulates the corruption attack of the smart terminal node. The smart terminal node is captured and corrupted, and various information in it is disclosed to the adversary $\mathcal{A}$ to verify the security of the scheme. In the scheme [41], the weak damage model is mentioned, which means that the query of the corrupted smart card and smart terminal does not damage the relevant information of the participant and the session key.

$Test(\Pi^t)$: This query model represents the semantic security of the session key SK generated between U_i and SD_j due to indistinguishability in ROR [42]. When an unbiased coin experiment begins, only the adversary $\mathcal{A}$ knows the flipping result of the coin r, and the adversary uses this to determine the result of the test query. If the adversary $\mathcal{A}$ executes the query and the session key is also new, then Π^t returns SK in case r=1 or a random number for r=0; otherwise, it outputs $\perp$(null value).

6.1.2. Security proof

The security proof given by $Theorem$ 1 is similar to the scheme [41, 43]. $Theorem$ 1: Suppose $\mathcal{A}$ is an adversary active in polynomial time t against our scheme LS in the random oracle. $\mathcal{PD}$ is a uniform distribution of password dictionary, $|\mathcal{PD}|$ is the size of $\mathcal{PD}$, l is number of bits in the biological key σ_i. And q_h, q_{send} is the number of H queries, $Send$ queries. $\mathcal{HASH}$ is the range space of $h(\cdot)$. $Adv_{\Omega,SE}^{IND-CPA}(n)/Adv_{\Omega,ME}^{IND-CPA}(n)$ is the advantage of $\mathcal{A}$ of breaking the IND-CPA secure cipher Ω. And $Adv_{\Omega}^{IND-CPA}(n) = Adv_{\Omega,SE}^{IND-CPA}(n)$ or $Adv_{\Omega,ME}^{IND-CPA}(n)$.

$Proof$: Next, we will use five game completion proofs say $Game_i$ $(i = 0, 1, 2, 3, 4)$. Assume that PS_i is an event in which the adversary $\mathcal{A}$ can correctly guess the random bit c in $Game_i$.

$Game_0$: In $Game_0$, the adversary $\mathcal{A}$ directly attacks our scheme LS in the ROR model. At this point, the adversary $\mathcal{A}$ guesses that bit c is reluctant. It is then clear that

$$Adv_{LS} = |2.Pr[PS_0] - 1|. \tag{1}$$

$Game_1$: After transferring to $Game_1$, by executing (Π^t, Π^u) query to simulate the eavesdropping attack of the adversary. Adversary $\mathcal{A}$ queries the result at the end of the game and confirms whether the result of the query is SK or a random value. $SK = h(Mu9\|r_u\|h(r_d\|r_s))$, among them $Mu9 = Ksu_i^{new} \oplus h(Ksu_i\|RID_i)$. If you want to calculate SK, you need to know r_u, r_d, r_s, Ksu_i, Ksu_i^{new}. Without these secrets, the authentication message queue $MQ1 = \{RID_i, M1, M2, M3, M4, T_1\}$, $MQ2 = \{Mu5, M6, M7, M8, Mu51, T_2\}$, $MQ3 = \{M10, M11, M12, M13, T3\}$ intercepted by the adversary will not increase his chance of winning in this game. It is then clear that

$$Pr[LS_0] = Pr[LS_1]. \tag{2}$$

$Game_2$: In this round of games, $Send$ and H oracles will be added to $Game_1$. In this round of the game, the ultimate goal of the adversary is to make a participant to accept a message queue that has been changed. The adversary has the right to perform various H queries to obtain hash collisions. However, all message queues in the agreement process, including $MQ1$, $MQ2$, and $MQ3$, contain participant identities, short-term/long-term secrets, random nonces and timestamps. Therefore, if an adversary makes a $Send$ query, there will be no conflicts. Derived from the birthday paradox:

$$|Pr[LS_1] - Pr[LS_2]| \le q_h^2/(2|\mathcal{HASH}|). \tag{3}$$

$Game_3$: This round of games simulates the compromised SC Oracle. If the user of the smart card chooses a low-entropy password and stores it in the smart card SC_i. Then the adversary can use the password dictionary to guess the user's password. An adversary can use a strong fuzzy extractor to extract random bits about m of σ_i. The adversary guesses that the probability of $\sigma_i \in \{0,1\}^m$ is approximately $(1/[2^m])$. Define that the security system only allows a limited number of incorrect password entries. It is clear that

$$|Pr[LS_2] - Pr[LS_3]| \le q_{send}/(2^m.|\mathcal{PD}|). \tag{4}$$

$Game_4$: In the final game, the adversary will simulate the damaged smart device SD oracle, and get one or more compromised terminal nodes. The adversary can get the RID_j and the device private key Ksd_j in the memory of the compromised node, but the adversary $\mathcal{A}$ cannot use these to calculate the session key between the user and other non-compromise nodes. In the process of session key agreement, the form of r_u, r_s, r_d, $Mu9$ that synthesizes the final session key and stored in the message queue is all encrypted using the password Ω. Well known that Ω is IND-CPA secure. It is clear that

$$|Pr[LS_3] - Pr[LS_4]| \le Adv_{\Omega}^{IND-CPA}(n). \tag{5}$$

After completing the query $Test$ oracle, if the adversary $\mathcal{A}$ successfully guesses the bit c, he will win this round of the game. It is clear that $Pr[LS_4] = 1/2$.

From Eq. (1), extrapolate that

$$\frac{1}{2} \cdot Adv_{LS} = |Pr[PS_0] - \frac{1}{2}|. \tag{6}$$

Use the triangle inequality relationship to get that: $|Pr[PS_1] - Pr[PS_4]| \le |Pr[PS_1] - Pr[PS_2]| + |Pr[PS_2] - Pr[PS_4]| \le |Pr[PS_1] - Pr[PS_2]| + |Pr[PS_2] - Pr[PS_3]| + |Pr[PS_3] - Pr[PS_4]| \le q_h^2/(2 \cdot |\mathcal{HASH}|) + q_{send}/(2^l \cdot |PD|) + Adv_{\Omega}^{IND-CPA}(n)$.

At last, from (2) to (6), we get the final result

$$Adv_{LS} \le \frac{q_h^2}{|\mathcal{HASH}|} + \frac{q_{send}}{2^{l-1} \cdot |\mathcal{PD}|} + 2Adv_{\Omega}^{IND-CPA}(n). \tag{7}$$

6.2. Informal security analysis

The informal security analysis shows that our proposed scheme can resist known attacks under the threat model. Our scheme guarantees the anonymity and unlinkability of users. Attacks under Dolev–Yao threat model are camouflage server attack, camouflage device attack, skip the server negotiation key attack. Attacks under CK-adversary model are session key security, internal privilege attack. Password guessing threats can cause user disguised attack. Physical capture and power analysis lead to smart card loss attack, edge node compromise attack.

6.2.1. Anonymity

Anonymity is aimed at adversary. On the trusted server side, the true identities of all parties can still be tracked and verified. When the user logs in securely and sends a session key agreement request message, in $MQ1 = \{RID_i, M1, M2, M3, M4, T_1\}$, the user uses the generated random secret value r_u and the current timestamp to further encrypt the device RID_j, and the user's own ID_i has been set with a pseudonym by the server during the registration phase. It is worth noting that after each key agreement is completed, the user side and the server side will update the user's pseudonym, which means that the user's pseudonym will always be dynamically updated, and the user's pseudonym will be used in each session key. The pseudonym is one-time, although the pseudo-name is not encrypted again, it will not affect the anonymity between users. Therefore, the adversary cannot identify the pseudonym of the user and the device from these messages.

6.2.2. Unlinkability and untraceability

In many applications, in order to protect the user's personal privacy, in addition to ensuring anonymity, it is generally set not to allow linking and tracking users. When a user with the same identity or role participates in the session key agreement scheme for two or more times, the adversary cannot tell from the outside that the messages in the two or more sessions come from the user with the same role or identity. During each key agreement, the request message $MQ1$ sent by the same user is different, because the user's pseudonym and private key are completing each session key agreement It will be updated dynamically. In addition, the temporary secret value r_u and the time stamp generated during each session are different, which ensures that each message queue does not have the same message as the previous key agreement. Similarly, the adversary cannot track users and devices, and cannot link multiple messages to the same sender.

6.2.3. Camouflage server attack

When an adversary tries to pretend to be a server to participate in the session key agreement process and interferes with the normal session key agreement, the adversary intercepts the message queue $MQ1$ from the user and tries to crack the message, forging the response message $MQ2'$. The adversary attempt to use the feedback message of the device to crack the entire session key process. However, the adversary cannot calculate the user's private key Ksu_i and the secret value r_u generated by the user. Therefore, the adversary cannot successfully complete the reasonable modification of $MQ2$. In the device's response message $MQ3$, for the same reason, the adversary could not calculate the temporary device private key Ksu_i^{TC} and the updated Ksu_i^{new}, so useful messages could not be derived from $MQ3$, and $MQ3$ could not be performed. The user can know whether $Mu9$ has been modified by comparing $Mu9'$ and $Mu9''$ calculated in $M11$ and $Mu51$ respectively. By comparing $M13'$ with $M13$, it can be known whether there is an error in the operation of the entire message $MQ3$. Therefore, the masquerading server attack is not a threat to this scheme.

6.2.4. Camouflage device attack

Suppose the adversary hopes to pretend to be a device, participate in the session key agreement with the user, and use the negotiated session key to grasp the user's operation on the device in real time, and pretend to be a legitimate user to control the device. The adversary intercepted the message queue $MQ2$ sent by the server to the device, tried to derive valuable information from it, and forged $MQ2'$. But without knowing the device private key Ksd_j, the adversary cannot obtain valuable information from the message queue $MQ2$, and combined message about the user's new private key $Mu9$. Without knowing these key information, the adversary has no way to complete a reasonable modification to the message queue $MQ3$. In other words, the server cannot pass the $MQ3'$ forged by the adversary. Therefore, our scheme can resist the camouflage attack of the device.

6.2.5. Skip the server negotiation key attack

For security reasons, each session key must be known by the server, and if there is a trace to follow, it can be tracked as a secure record. In our scheme, if the user and the device try to bypass the server, or one of the endpoints simulates the server, this is impossible. Because when the key is negotiated, the server will update the user's private key. More importantly, in $Mu9 = Ksu_i^{new} \oplus h(Ksu_i \| RID_i)$ and $Mu51 = Ksu_i^{new} \oplus Ksu_i^{TC}$, the user cannot know the private key of the device, and the device wants to forge $Mu9'$ also must know the private key of the user. Because the user restored Ksu_i^{new} from $Mu51$ and calculated $Mu9''$. User can check whether $Mu9'$ is forged by using $Mu9''$. It is impossible for the device to skip the server and negotiate the session key with the user. Therefore, our scheme can resist the skip the server negotiation the session key attack.

6.2.6. Session key security

The finally negotiated session key SK can be calculated by the user and the device respectively. The response message sent by the device to the user does not contain the complete form of SK. If the adversary intercepts the message $MQ3$, then try to get the corresponding information $\{r_s, r_d, Mu9\}$ from $MQ3$, the adversary need to get the user's private key Ksu_i, the updated private key Ksu_i^{new}, and the random secret value r_u generated by the user. However, the adversary has no way of knowing these key information. Therefore, the adversary cannot obtain the key information of the synthesis key from the message $MQ3$, and thus cannot synthesize the session key SK.

6.2.7. Internal privilege attack

In this attack threat, we assume that the internal adversary can obtain the information of the user during the registration phase and obtain the user's smart card after the registration is completed. Then the adversary can obtain the information in the smart card through power analysis, including $\{RID_i^*, Ksu_i^*, SDList^*, TPW_i, \tau_i, Gen(\cdot), Rep(\cdot)\}$. In the previous smart card loss analysis, we know that the adversary cannot know the key user information, including $\{ID_i, PW_i, \sigma_i\}$. Although the internal adversary can learn the pseudonym information of the user, and link the pseudonym with the smart card, the user completes the collection of passwords and biometric information security locally, these informations can be changed dynamically. The adversary cannot guess the user's password PW_i offline, nor can it derive the user's biometric information computationally feasible. Therefore, the internal adversary cannot decompose the corresponding information in the smart card from the information he has obtained without knowing the password PW_i and σ_i. Therefore, our scheme is still security for users with internal privileges.

6.2.8. User disguised attack

Suppose an adversary obtains the user's lost or stolen smart card, and uses power analysis to derive all user information stored on the smart card, including $\{RID_i^*, Ksu_i^*, SDList^*, TPW_i, \tau_i, Gen(\cdot), Rep(\cdot)\}$. When the adversary waits for the user to send a request message $MQ1 = \{RID_i, M1, M2, M3, M4, T_1\}$, intercept $MQ1$, and try to use known information to forge a request message, such as $MQ1' = \{RID_i', M1', M2', M3', M4', T_1'\}$. In order to forge $M1', M2', M3', M4'$, and falsified information can be verified by the server, the adversary must be able to calculate the user's private key Ksu_i, otherwise the legitimate $M2'$ cannot be calculated. As in the previous security analysis, the adversary cannot calculate the user's private key Ksu_i. In the adversary's forged message queue $MQ1$, the adversary cannot know the device RID_j and the secret value r_u generated by the user. The adversary's forged request message cannot complete a reasonable change to the entire message queue, so when the server responds, it cannot pass two checks and authentications. Therefore, even if the opponent obtains the user's information in the user's lost or stolen smart card, he cannot pretend to be a legitimate user to participate in the session key agreement process. Our scheme can resist the user disguised attack.

6.2.9. Smart card loss attack

When the user accidentally loses the smart card or is stolen by the adversary, the adversary can use the power analysis to obtain the user information stored on the smart card, including $\{RID_i^*, Ksu_i^*, SDList^*, TPW_i, \tau_i, Gen(\cdot), Rep(\cdot)\}$. If an adversary wants to impersonate a legitimate user, he must pass the TPW_i test, because the adversary cannot provide the corresponding user information including $\{ID_i, PW_i, \sigma_i\}$, so it is impossible to log in using a smart card by directly impersonating a legitimate user. In order to extract the user's pseudonym, user's private key and device list information from the information $\{RID_i^*, Ksu_i^*, SDList^*\}$ obtained from the power analysis, the adversary must be obtained that the user's ID_i, PWi and σ_i. This information is forged by an adversary's inability to guess and is stored in the verification message using a hash function. Therefore, the scheme resists the loss of smart card attacks and ensures the local security of user messages.

6.2.10. Edge node compromise attack

In the IIoT environment, suppose a smart device or an edge sensor node is captured by an adversary, and the adversary derives the device information which stored in device's memory. Each device or edge node stores device's information about the node, including $\{RID_j, Ksd_j\}$. It is impossible for the adversary to use this information to calculate the server private key of the IIoT area. Because the server private key in this area is stored in the private key of the device using a hash function, and it is difficult to calculate the server private key by using the device's private key Ksd_j. For different IIoT working areas, the server private key S in this area is also different. Although the adversary can derive the session key SK completed with the legitimate user from the device memory, in the entire IIoT environment, the compromise of a device or sensor node will not threaten the security of legitimate users and other uncompromising device nodes. Therefore, for the compromise attack of the edge node, this scheme can well resist the influence of the compromise node on the normal operation of the area.

7. Performance analysis

In this section, we analyze our proposed scheme from three aspects: (a) comparison of security features and functional features; (b) communication cost comparison; (c) computational cost comparison.

Table 2
Security and functional features comparison.

Features↓	Schemes↓					
	[21]	[14]	[13]	[18]	[41]	Ours
FN_1	✔	✔	✔	N/A	✔	✔
FN_2	✔	✔	✔	✔	✗	✔
FN_3	✔	✔	✔	✔	✔	✔
FN_4	✔	✔	✔	✔	✔	✔
FN_5	✔	✔	✗	✗	✔	✔
FN_6	✔	✔	✔	✗	✔	✔
FN_7	✔	✔	✔	✔	✔	✔
FN_8	✔	✔	✔	✔	✗	✔
FN_9	✔	✔	✔	N/A	✗	✔
FN_{10}	✔	✔	✔	✗	✗	✔
FN_{11}	✔	✔	✔	✔	✔	✔
FN_{12}	✔	✔	✔	N/A	N/A	✔
FN_{13}	✔	✔	✔	N/A	✗	✔
FN_{14}	✗	✗	✗	N/A	✗	✔
FN_{15}	3	3	3	N/A	2	3
FN_{16}	✔	✔	✔	✔	✔	✔
FN_{17}	✔	✔	✔	✔	✗	✔
FN_{18}	✔	✔	✔	N/A	✗	✔
FN_{19}	✔	✔	✔	✔	✗	✔
FN_{20}	✔	✔	✔	✗	✗	✔

Note: ✔ :the security and functional features supported or satisfied by the scheme; ✗:the security and functional features that the scheme does not support or cannot meet. FN_1: user anonymity; FN_2: untraceability; FN_3: resist counterfeiting attacks; FN_4: resist replay attacks; FN_5: clock synchronization; FN_6: resist the loss of smart cards/mobile devices; FN_7: mutual authentication; FN_8: dynamic addition of IoT devices; FN_9: lost temporary session secret; FN_{10}: unlinkability; FN_{11}: man-in-the-middle attack; FN_{12}: local biometric update; FN_{13}: local password update; FN_{14}: independent and real-time user revocation; FN_{15}: two/three factor authentication; FN_{16}: smart device captures attack resilience; FN_{17}: device anonymity; FN_{18}: offline password guessing attack; FN_{19}: fast error detection; FN_{20}: internal privilege attack.

7.1. Comparison of security features and functional features

We compare the security features and functional features of the existing scheme with the one we have drawn up. The details are shown in the Table 2. In terms of security features, we consider the security of the scheme from a more detailed perspective. In a scheme involving user authentication, if a password and a smart card or smart mobile device are used, the scheme is a two-factor authentication; if the scheme also uses other factors, such as biometrics extracted by a fuzzy extractor, then it is called three-factor authentication. Compared with two-factor authentication, three-factor authentication is firstly resistant to device capture attacks in the threat model, including the loss and theft of smart card. Secondly, the attacker cannot recover the biometrics, which is better resistant to password guessing attacks and user impersonation attacks.

Chang et al. [41] is a two-factor authentication scheme. Compared with two-factor authentication, three-factor authentication can better protect user privacy and security. At the same time, this scheme does not achieve unlinkability and untraceability. During the multiple login and authentication process of the user, the adversary can link to the same user, thereby exposing user privacy. Although this scheme guarantees anonymity during authentication and key agreement, it exposes users' private information in the process of tracking and linking users. Wazid et al. [14] and Banerjee et al. [21] use a dynamic revocation mechanism, which increases computational and communication costs in authentication, with a corresponding increase in response time. Compared with the three-party authentication scheme, Esfahani et al. [18] for M2M has more security threats. Li et al. [13] do not adopt the mechanism of clock synchronization, which have the problem of asynchronous communication, and after the adversary steals the message, it may carry out DDoS attack on the system. It can be seen from Table 2 that the proposed scheme not only meets the security requirements from more angles, but also provides more functional features suitable for the real environment.

Table 3
Communication cost comparison.

Scheme/Cost(bits)	User	Server	Device	Total coat
Banerjee et al. [21]	544	928	1088	2560
Wazid et al. [14]	736	1472	640	2848
Li et al. [13]	800	1280	640	2720
Esfahani et al. [18]	–	480	800	1280
Chang et al. [41]	672	1216	512	2400
Ours	832	832	832	2496

Table 4
Computational cost comparison.

Scheme	Total cost	Estimate (ms)
Banerjee et al. [21]	$19T_h + 10T_{E/D} + T_f$	0.4699
Wazid et al. [14]	$22T_h + 8T_{E/D} + T_f$	0.465
Li et al. [13]	$19T_h + 6T_m$	2.6539
Esfahani et al. [18]	$13T_h$	0.0013
Chang et al. [41]	$21T_h + 4T_m$	1.7701
Ours	$35T_h + T_f$	0.4455

7.2. Communication cost comparison

The Table 3 and Fig. 4 shows the communication cost between our proposed scheme and other schemes, mainly considering those communication steps that are often used throughout the key agreement phase. In order to be able to compare these schemes fairly, we calculate the respective communication overheads of all schemes on the basis of making a unified assumption for some parameters in the scheme. We assume that in the clock synchronization scheme, the size of the time stamp is 32 bits, and the identities of all users and devices or nodes are 160 bits. The symmetric encryption/decryption is unified assuming that the AES-128 algorithm is used, and the block size is 128 bits. The size of all random secret values generated in is 160 bits. In the scheme using the ECC algorithm, the subgroup G of ECC is 160 bits, and the size of the element is 320 bits. In addition, the output of the most used hash function is assumed to be 160 bits uniformly. The assumption of ECC, symmetric encryption and hash function bit size is based on the same key strength as the comparison scheme [20]. The timestamp size is 4 bytes (32 bits) according to the standard size of Real Time Messaging Protocol (RTMP).

In our proposed scheme, there are three communication messages during the process of logging in, mutual authentication and completing key agreement $MQ1 = \{RID_i, M1, M2, M3, M4, T_1\}$, $MQ2 = \{Mu5, M6, M7, M8, Mu51, T_2\}$, $MQ3 = \{M10, M11, M12, M13, T_3\}$ respectively need $(160 + 160 + 160 + 160 + 160 + 32) = 832$ bits, $(160 + 160 + 160 + 160 + 160 + 32) = 832$ bits, $(160 + 160 + 160 + 160 + 160 + 32) = 832$ bits. Therefore, our total communication cost is $832 + 832 + 832 = 2496$ bits. From Table 3, we can see that the scheme of Esfahani et al. [18], which only realizes mutual authentication between the device and the gateway and completes key agreement, reduces communication parties, communication times and communication cost because of the reduction of communication parties. In the scheme that realizes mutual authentication and key agreement among users, gateways and devices, our scheme have relatively lightweight communication overhead.

7.3. Computational cost comparison

In the comparison of computational cost, we calculate the computational cost used in the main key agreement phase, including the phase of login and mutual authentication and key agreement. We list the calculation operations used in each phase in each scheme. The symbols T_h, $T_{E/D}$, T_f, and T_m used in Table 4 and Fig. 5 respectively represent the use of one-way function calculations and encryption/decryption operations using symmetric encryption technology, the calculation time

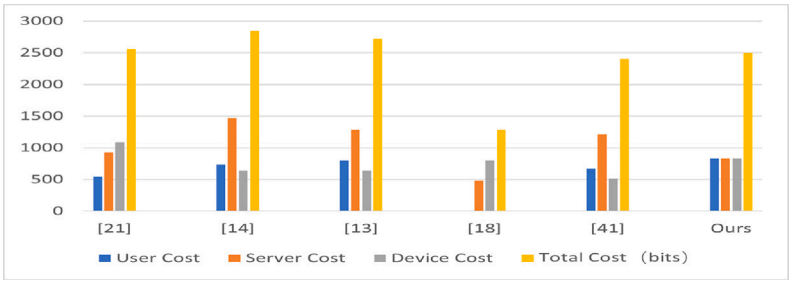


Fig. 4. Communication Cost Comparison.

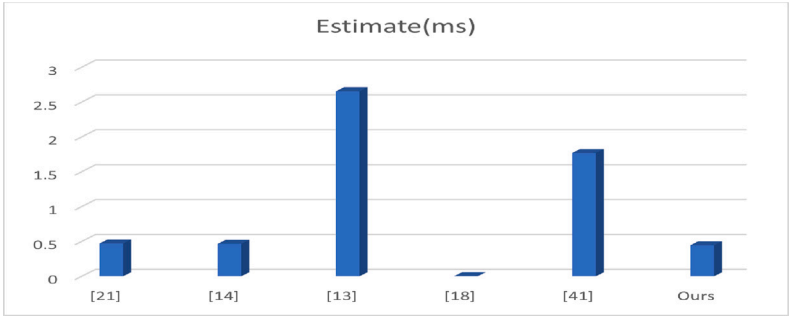


Fig. 5. Computational Cost Comparison.

Table 5
Execution time of cryptographic operations.

Cryptographic operation	Execution time (ms)
T_m	0.442
$T_{E/D}$	0.0026
T_h	0.0001
T_f	0.442

Note: $T_f \approx T_m$.

required for the fuzzy extraction operation by using the fuzzy extractor and the ECC point multiplication. Although the exclusive XOR operation is commonly used in various schemes, compared with the calculation operations mentioned in the table, the bitwise calculation of the XOR operation requires little computational overhead. Therefore, in the performance evaluation of the computational overhead, we do not consider the computational cost of the XOR operation.

The computational cost of the arithmetic operations used in Table 5. Based on the MIRACL library [44], we test the 10,000 operation times of various encryption operations on a personal computer (processor: Intel(R) Core(TM) i7-6700@ 3.4 GHz, main memory: 8 GB; operation system: Ubuntu 14.04), and finally obtain the average time of various encryption operations. We use the approximate time required for each operation to calculate the computational cost of each scheme. From the calculation cost estimation, we can see that the use of a scheme based on the asymmetric encryption algorithm ECC generally requires a higher calculation cost. Although the scheme of Esfahani et al. [18] have lightweight computational overhead, this scheme only considers the mutual authentication between the device and the gateway node. Three-factor authentication increase the time of fuzzy extraction. The computing overhead of the smart card is increased when the user log in to the smart card. Fuzzy extractor can be constructed by general hash functions or error correction codes that only require lightweight operations [45]. In general, we assume that the time to execute the fuzzy extractor is the same as the time to execute the elliptic curve point multiplication. It is worth noting that elliptic curve point multiplication is a complex and time-consuming encryption operation. Our scheme have a relatively lightweight computational cost on the basis of meeting sufficient functional requirements and security requirements.

8. Conclusion

In this paper, we discuss the security issues of IIOT, and on this basis, we propose a novel lightweight authentication and session key agreement scheme. Our scheme guarantees the anonymity and unlinkability of users. Under the widely accepted ROR model, a rigorous formal security analysis is performed on the scheme, and non-formal security analyses are performed on various attack threats, thus proving its security robustness. In addition, we compare its security and performance with previous studies. The comparison results show that our scheme provides a better trade-off between security and functionality, ensuring sufficient security attributes and functional characteristics. Moreover, it also reduces the computational and communication overheads.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The work was supported in part by the National Natural Science Foundation of China under Grant 62272002 and Grant U1936220, in part by the Excellent Youth Foundation of Anhui Scientific Committee under Grant 2108085J31, and in part by the Special Fund for Key Program of Science and Technology of Anhui Province, China under Grant 202003A05020043.

References

[1] S. Vashi, J. Ram, J. Modi, S. Verma, C. Prakash, Internet of things (IoT): A vision, architectural elements, and security issues, in: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC, 2017, pp. 492–496.

[2] M. Nkomo, G.P. Hancke, A.M. Abu-Mahfouz, S. Sinha, A. Onumanyi, et al., Overlay virtualized wireless sensor networks for application in industrial internet of things: A review, Sensors 18 (10) (2018) 3215.

[3] C. Arnold, et al., The industrial internet of things from a management perspective: A systematic review of current literature, J. Emerg. Trends Mark. Manag. 1 (1) (2017) 8–21.

[4] P. Vijayakumar, M.S. Obaidat, M. Azees, S.H. Islam, N. Kumar, Efficient and secure anonymous authentication with location privacy for IoT-based WBANs, IEEE Trans. Ind. Inf. 16 (4) (2019) 2603–2611.

[5] H. Zhang, J. Yu, M.S. Obaidat, P. Vijayakumar, L. Ge, J. Lin, J. Fan, R. Hao, Secure edge-aided computations for social internet-of-things systems, IEEE Trans. Comput. Soc. Syst. (2020).

[6] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. Tsang, J. Rodriguez, Massive internet of things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation, IEEE Ind. Electron. Mag. 11 (2017) 28–33.

[7] S. Vitturi, C. Zunino, T. Sauter, Industrial communication systems and their future challenges: Next-generation ethernet, IIoT, and 5G, Proc. IEEE 107 (2019) 944–961.

[8] J. Cui, F. qun Wang, Z. Qing-yang, Y. Xu, Z. Hong, An anonymous message authentication scheme for semi-trusted edge-enabled IIoT, IEEE Trans. Ind. Electron. (2020) 1.

[9] R. Vinoth, L.J. Deborah, P. Vijayakumar, N. Kumar, Secure multifactor authenticated key agreement scheme for industrial IoT, IEEE Internet Things J. 8 (5) (2020) 3801–3811.

[10] M. Azees, P. Vijayakumar, M. Karuppiah, A. Nayyar, An efficient anonymous authentication and confidentiality preservation schemes for secure communications in wireless body area networks, Wirel. Netw. 27 (3) (2021) 2119–2130.

[11] D. Zhang, Y. Qian, J. Wan, S. Zhao, An efficient RFID search protocol based on clouds, Mob. Netw. Appl. 20 (2015) 356–362.

[12] B. Gupta, D. Agrawal, S. Yamaguchi, Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security, 2016.

[13] X. Li, J. Niu, M.Z.A. Bhuiyan, F. Wu, M. Karuppiah, S. Kumari, A robust ECC-based provable secure authentication protocol with privacy preserving for industrial internet of things, IEEE Trans. Ind. Inf. 14 (2018) 3599–3609.

[14] M. Wazid, A.K. Das, V. Odelu, N. Kumar, M. Conti, M. Jo, Design of secure user authenticated key management protocol for generic IoT networks, IEEE Internet Things J. 5 (2018) 269–282.

[15] Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen, J. Liu, Remotely access "my" smart home in private: An anti-tracking authentication and key agreement scheme, IEEE Access 7 (2019) 41835–41851.

[16] J. Zhang, J. Cui, H. Zhong, Z. Chen, L. Liu, PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks, IEEE Trans. Dependable Secure Comput. 18 (2021) 722–735.

[17] A.K. Das, M. Wazid, N. Kumar, A. Vasilakos, J. Rodrigues, Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment, IEEE Internet Things J. 5 (2018) 4900–4913.

[18] A. Esfahani, G. Mantas, R. Matischek, F.B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. Tauber, C. Schmittner, J. Bastos, A lightweight authentication mechanism for M2M communications in industrial IoT environment, IEEE Internet Things J. 6 (2019) 288–296.

[19] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, M. Sain, Lightweight and secure session-key establishment scheme in smart home environments, IEEE Sens. J. 16 (2016) 254–264.

[20] M. Wazid, A.K. Das, V. Odelu, N. Kumar, W. Susilo, Secure remote user authenticated key establishment protocol for smart home environment, IEEE Trans. Dependable Secure Comput. 17 (2020) 391–406.

[21] S. Banerjee, V. Odelu, A.K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, K. Choo, A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment, IEEE Internet Things J. 6 (2019) 8739–8752.

[22] J. Shao, X. Lin, R. Lu, C. Zuo, A threshold anonymous authentication protocol for VANETs, IEEE Trans. Veh. Technol. 65 (2016) 1711–1720.

[23] M.H. Eldefrawy, N. Pereira, M. Gidlund, Key distribution protocol for industrial internet of things without implicit certificates, IEEE Internet Things J. 6 (2019) 906–917.

[24] S.F. Aghili, H. Mala, M. Shojafar, M. Conti, PAKIT: Proactive authentication and key agreement protocol for internet of things, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, 2019, pp. 348–353.

[25] P. Gope, A.K. Das, N. Kumar, Y. Cheng, Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks, IEEE Trans. Ind. Inf. 15 (2019) 4957–4968.

[26] C.-M. Chen, B. Xiang, T.-Y. Wu, K.-H. Wang, An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks, Appl. Sci. 8 (7) (2018) 1074.

[27] M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, R. Borgaonkar, New privacy issues in mobile telephony: Fix and verification, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, 2012, pp. 205–216.

[28] P.-A. Fouque, C. Onete, B. Richard, Achieving better privacy for the 3GPP AKA protocol, Cryptol. EPrint Arch. (2016).

[29] S. Shin, T. Kwon, Two-factor authenticated key agreement supporting unlinkability in 5G-integrated wireless sensor networks, IEEE Access 6 (2018) 11229–11241.

[30] X. Li, J. Liu, M.S. Obaidat, P. Vijayakumar, Q. Jiang, R. Amin, An unlinkable authenticated key agreement with collusion resistant for VANETs, IEEE Trans. Veh. Technol. 70 (8) (2021) 7992–8006.

[31] H. Khan, B. Dowling, K.M. Martin, Pragmatic authenticated key agreement for IEEE Std 802.15. 6, Int. J. Inf. Secur. (2021) 1–19.

[32] W. Hsieh, J.-S. Leu, Anonymous authentication protocol based on elliptic curve Diffie-Hellman for wireless access networks, Wirel. Commun. Mob. Comput. 14 (2014) 995–1006.

[33] D. Dolev, A. Yao, On the security of public key protocols, in: 22nd Annual Symposium on Foundations of Computer Science, Sfcs 1981, 1981, pp. 350–357.

[34] R. Canetti, H. Krawczyk, Universally composable notions of key exchange and secure channels, IACR Cryptol. EPrint Arch. 2002 (2002) 59.

[35] R. Amin, S.H. Islam, N. Kumar, K.-K.R. Choo, An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks, J. Netw. Comput. Appl. 104 (2018) 133–144.

[36] E. Bertino, N. Shang, S. Wagstaff, An efficient time-bound hierarchical key management scheme for secure broadcasting, IEEE Trans. Dependable Secure Comput. 5 (2008) 65–70.

[37] T.S. Messerges, E.A. Dabbish, R. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (2002) 541–552.

[38] S. Kumar, H. Thapliyal, A. Mohammad, EE-SPFAL: A novel energy-efficient secure positive feedback adiabatic logic for DPA resistant RFID and smart card, IEEE Trans. Emerg. Top. Comput. 7 (2019) 281–293.

[39] A.K. Das, A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor, Int. J. Commun. Syst. 30 (2017).

[40] M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, IACR Cryptol. EPrint Arch. 2004 (2004) 233.

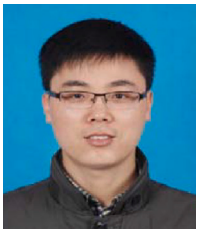
[41] C. Chang, H.-D. Le, A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks, IEEE Trans. Wireless Commun. 15 (2016) 357–366.

[42] P. Sarkar, A simple and generic construction of authenticated encryption with associated data, IACR Cryptol. EPrint Arch. 2009 (2010) 215.

[43] S. Chatterjee, S. Roy, A.K. Das, S. Chattopadhyay, N. Kumar, A. Vasilakos, Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment, IEEE Trans. Dependable Secure Comput. 15 (2018) 824–839.

[44] Miracl cryptographic sdk, 2021, https://github.com/miracl/MIRACL/. (Accessed 9 May 2021).

[45] Y. Dodis, R. Ostrovsky, L. Reyzin, A.D. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, SIAM J. Comput. 38 (2008) 97–139.



Jie Cui was born in Henan Province, China, in 1980. He received his Ph.D. degree in University of Science and Technology of China in 2012. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). He has over 120 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing,

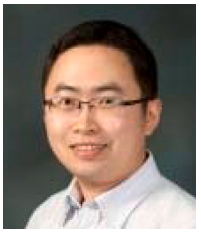
IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Network and Service Management, IEEE Transactions on Emerging Topics in Computing, IEEE Transactions on Cloud Computing and IEEE Transactions on Multimedia), academic books and international conferences.

Fangzheng Cheng is now a research student in the School of Computer Science and Technology, Anhui University. His research focuses on the security of Industrial Internet of Things (IIoT).

Hong Zhong was born in Anhui Province, China, in 1965. She received her Ph.D. degree in computer science from University of Science and Technology of China in 2005. She is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). She has over 150 scientific publications in reputable journals (e.g. IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Multimedia, IEEE Transactions on Vehicular Technology, IEEE Transactions on Network and Service Management, IEEE Transactions on Cloud Computing and IEEE Transactions on Big Data), academic books and international conferences.

Qingyang Zhang is currently a Ph.D. student in the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests include vehicular ad hoc network and IoT security.

Chengjie Gu received his Ph.D. degree in Nanjing University of Posts and Telecommunications in 2012. From 2012 to 2017, he was an innovation team leader in the 38th Research Institute of CETC and conducted research and development in the communication and networking sector. Currently he is a president of security research institute in new H3C group. He is also studying for postdoctoral fellowship at the USTC. He is a high-level innovation leader of Anhui province and a cybersecurity expert of Zhejiang province in China. His research interest includes network security and trusted network architecture, etc.



Lu Liu is the Professor of Informatics and Head of School of Informatics in the University of Leicester, UK. Prof Liu received the Ph.D. degree from University of Surrey, UK and M.Sc. in Data Communication Systems from Brunel University, UK. Prof Liu's research interests are in areas of cloud computing, service computing, computer networks and peer-to-peer networking. He is a Fellow of British Computer Society (BCS).