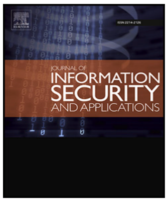
Contents lists available at ScienceDirect

Journal of Information Security and Applications

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



BP-AKAA: Blockchain-enforced Privacy-preserving Authentication and Key Agreement and Access Control for IIoT

Suhui Liu [a], Liquan Chen [a,b,*], Hongtao Yu [a], Shang Gao [c], Huiyu Fang [a]

[a] School of Cyber Science and Engineering, Southeast University, Nanjing, 211102, China
[b] Purple Mountain Laboratories, Nanjing, 211100, China
[c] Department of Computing, the Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:
Blockchain
Authentication and key agreement
Non-interactive zero-knowledge proof
Industrial Internet of Things
Access control

ABSTRACT

The Industrial Internet of Things (IIoT) links multiple subnets to accomplish more real-time, efficient, and high-class production. Authentication is an essential prerequisite for secure communication and data sharing between mutually untrusted subdomains.However, solving trust issues between subnets through third-party trusted servers inevitably introduces security and efficiency bottlenecks. In addition, the issue of not compromising the privacy of mutual authentication remains a challenge. Furthermore, key agreement and access control, as two follow-up steps of authentication, is non-negligible for achieving secure and efficient data sharing. Existing authentication works either require heavy computational overhead or lack necessary features for data sharing. Therefore, this paper proposed a **b**lockchain-enforced cross-domain **p**rivate-protected **a**uthentication and **k**ey **a**greement scheme supporting attribute-based **a**ccess control, named **BP-AKAA**. To the best of our knowledge, this is the first scheme that simultaneously supports privacy authentication, key agreement, and access control. Non-interactive zero-knowledge proof technology is adopted to protect the identities of devices. In addition, with the assistance of distributed blockchain, the untrust issue of cross-domain authentication is solved.Performance analysis demonstrates that our scheme satisfies multiple functions, including cross-domain, privacy-preserving, and mutual authentication, and outperforms existing schemes in terms of key generation, authentication, and access control.

1. Introduction

The Industrial Internet of Things (IIoT) is emerging as a universal paradigm due to the ubiquitous application of high-performance computers, smart embedded devices, and 5G communications [1]. IIoT contains devices with different resources from different trust domains, such as servers, gateways, sensors, etc., which leads to various communication types in IIoT, as depicted in Fig. 1. The complex network environment allows some attacks to take advantage of, and intrusion detection [2,3] is an advantageous technique to defend against attacks. In addition, many specialized communication protocols have been proposed for different types of communication. For example, an Internet protocol version 6 (IPv6) is used for low-power wireless personal area networks, message queue telemetry transport (MQTT) is a standardized protocol for lightweight D2D applications, and extensible messaging and presence protocol (XMPP) is used for real-time messaging, online presence, and request–response services [4].

Device-to-device (D2D) communication is the core technology to promote a new era of the industrial revolution [5]. Different from server-to-device communication which usually transmits big data, D2D communication focuses on small data transmission. Unfortunately, this small but frequent and wireless-depended communication may cause serious security concerns. Furthermore, compared with other types of internet of things (IoT) systems, the IIoT not only needs to link multiple subnets for communication to complete efficient production but more importantly, its sensitive data leakage and improper data use may cause life-threatening consequences in addition to economic losses. Some papers, e.g. [6], try to design efficient data clustering algorithms to analyze big data of IIoT. However, they lost sight of an essential precondition-effective authentication and access control (AKA) for IIoT systems.

Considering the cross-domain D2D authentication of the IIoT (Type 3 in Fig. 1), designing a practical and efficient authentication scheme to ensure communication security faces the following difficulties. First, device resources are limited. Since most underlying IoT devices do not

[*] Corresponding author at: School of Cyber Science and Engineering, Southeast University, Nanjing, 211102, China.
E-mail addresses: suhuiliu@seu.edu.cn (S. Liu), Lqchen@seu.edu.cn (L. Chen), seu_yht@seu.edu.cn (H. Yu), shanggao@polyu.edu.hk (S. Gao), fanghuiyu@seu.edu.cn (H. Fang).

https://doi.org/10.1016/j.jisa.2023.103443

Available online 8 February 2023

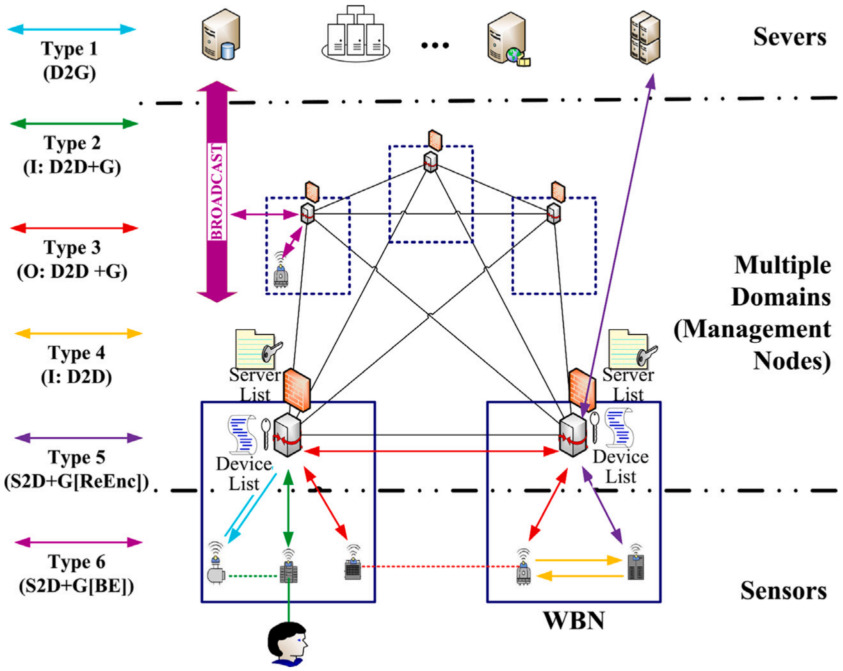


Fig. 1. IIoT Communication Types. Type 1 is device-to-gateway (D2G). Type 2 is device-to-device (D2D) communication within one domain with the assistance of gateways, and Type 4 is direct D2D communication. Type 3 is cross-domain communication. Type 5 and Type 6 are device-to-server (D2S) communication using re-encryption technology and broadcast technology.

have sufficient computing and storage resources, complex authentication protocols cannot be applied directly on feasible authentication even with a high level of security. Second, devices are in different trust domains, and inter-domain communication cannot rely on a centralized trusted server. Finally, device privacy is an essential prerequisite for defending against attacks and enabling secure data sharing.

In IIoT, each device has a unique device number when it was produced, which naturally can be used for effective identity authentication. Nevertheless, identity-based authentication (IBA) faces the most serious security threat: identity privacy leakage. When the device number or identity of a device is exposed, the probability of this device being the target of different attacks will increase greatly. Non-interactive zero-knowledge proofs (NIZKP) or zero-knowledge proofs (ZKP) [7] enable privacy-preserving identity authentication. Some works have adopted ZKP to complete the authentication between the device and the management node, but there are still some difficulties in implementing D2D authentication using ZKP, considering limited device resources and cross-domain trust issues.

Blockchain [8], designed as a distributed, traceable and tamper-resistant technology, has been used to realize trustless authentication between different domains. Furthermore, authentication is only a part of the preparation for secure communication and data exchange, more features like access control and key agreement issues need to be considered, as well. A new primitive, authentication and key agreement and access control (AKAA) were proposed by combining identity-based AKA and attribute-based access control (ABAC) to achieve efficient authentication and access control functions for IoT. Unfortunately, the use of blockchain for efficient and secure cross-domain authentication or the design of authentication schemes that support effective access control and key agreement have received little attention.

To sum up, designing a private-protected secure communication scheme for IIoT D2D is challenging on account of functionality and utility. In this paper, considering the IIoT scenario where end devices (smart sensors) are equipped with secure elements (SE) and in-domain management nodes (registration server or attribute server) are equipped with a trusted platform module (TPM), we design a blockchain-enforced private-preserving authentication, key management, and access control scheme for IIoT D2D communication, named BP-AKAA. The main contributions are listed here:

- This paper proposes a blockchain-enforced authentication framework for cross-trust-domain communication in IIoT systems. Benefiting from the blockchain advantages such as traceability, distribution, and non-tamperability, devices from different domains can realize authentication and construct secure communication channels.
- Our BP-AKAA scheme utilizes two NIZKP schemes, proof of knowledge of discrete logarithms (also known as Schnorr Protocol) and proof of the equality of two discrete logarithms, to protect the authentication privacy of IIoT devices. Specifically, the first NIZKP scheme is used for non-key-escrow registration of devices, and the second one is used for D2D authentication.
- The attribute-based access control is incorporated in our scheme to implement secure data sharing. Besides, the key agreement process is included for generating session keys and accomplishing symmetric-key-based encrypted data exchange.
- The scheme is feasible and advisable for IIoT applications as the security proof, theoretical analysis, and simulation results demonstrate that the BP-AKAA scheme has a high-security level and advisable performance.

The rest of the paper is organized as follows. Section 2 reports the related work and we describe preliminaries in Section 3. The system model and system procedures, and security assumption of the BP-AKAA scheme for IIoT are defined in Section 4. Section 5 presents the concrete construction of the proposed BP-AKAA scheme. The security analysis and the performance analysis are provided in Section 6 and Section 7, respectively. Finally, we conclude in Section 8.

2. Related work

ID-based authentication is widely applied in IoT applications due to its lightweight and other advantages. Zhou et al. [9] proposed a lightweight ID-based authentication for cloud-IoT, which uses smart cards and pseudo-identity. Lopes et al. [10] extended ID-based AKA to a group-signature-based AKA scheme, where a set of devices that are locationally close are constituted to a group, and a group leader is responsible for secure direct D2D communication within this group. Xiong et al. [11] proposed a privacy-preserving authentication protocol for heterogeneous systems, which utilizes proxy re-signature technology to realize communications between ID-based systems and certificate-less-based systems. But the cross-system privacy of Xiong's scheme relies on the Cloud server. Kumar et al. [12] presented an ECC-based ID-based AKA scheme for authentication between devices and the trusted server. However, identity-based authentication and AKA schemes usually face privacy leakage issues or do not support access control, and most abovementioned schemes are not designed for D2D communications.

Attribute-based authentication can protect the privacy and realize access control simultaneously. Liu et al. [13] brought a new primitive, attribute-based handshake for medical IoT to protect identity privacy and to realize fine-grained access control. Ibrahim et al. [14] presented a lightweight authentication scheme that supports attribute-based access control. Ibrahim's scheme depends on a trusted registration server and did not consider the key agreement issue. To protect identity privacy, Zhang et al. [15] designed an attribute-based group key agreement protocol to achieve secure and efficient in-group communication. Still, this scheme did not consider the cross-domain problem. Lin et al. [16] gave an attribute-based mutual authentication scheme, yet this scheme did not protect the privacy of identity. Recently, Sun et al. [17] improved an outsourced attribute-based signature scheme for authentication in IoT. Sucasas et al. [18] proposed a privacy-preserving pseudonymity authentication scheme for cloud servers. Zhang et al. [19] designed an attribute-based encryption (ABE) scheme and gave a progressive authentication model. Yet, compared with id-based schemes, realizing authentication solely depends on attribute structure may lead to heavy computation overhead, which means they may be impractical for resource-constrained IoT devices such as sensors.

Despite message authentication [20], several device authentication/ identity authentication schemes were proposed specifically for IIoT [11] and D2D communication (not direct handshake, but with gateways). Esfahani et al. [21] designed a lightweight authentication mechanism for machine-to-machine communications in IIoT only based on hash and XOR operations. Esfahani's scheme requires a secure channel between the device and the registration center and a pre-shared secret key between the registration center and the authentication center (router). Gupta et al. [22] designed a lightweight device authentication scheme, yet it did not consider the privacy of device identity. Abdi et al. [23] proposed an anonymous ID authentication scheme with the assistance of password and biometric technology. Lately, Xu et al. [24] designed a cross-layer device authentication scheme based on quantum walks on circles, which can resist attacks from quantum computers. Some of the abovementioned schemes for IIoT are private-preserving, but none of them considered a critical function for data sharing between devices, the access control.

NIZKP is another technology to protect the privacy of authentication. Martin et al. [25] designed a NIZKP authentication protocol for IoT networks based on the graph isomorphism problem. Walshe et al. [26] constructed a NIZKP protocol for IoT using the Merkle tree structure to create authentication challenges. Rasheed et al. [27] proposed an adaptive ZKP authentication protocol for ad hoc networks, which achieves various levels of privacy and can resist inside attacks caused by parameter leakage. Soewito and Marcellinus [28] gave a modified ZKP algorithm for IoT device authentication by combining the most popularly adopted encryption algorithm, AES. Gaba et al. [29] presented a mutual AKA protocol using ZKP for sustainable healthcare applications, where the authentication between data user and sensors requires the middle node, gateways. Nevertheless, none of the abovementioned (NI)ZKP protocols considers distributed cross-domain and access control.

Owing to the advantages of blockchain, such as distribution, immutability, and traceability, exploiting blockchain for authentication has attracted increasing interest from academics [30]. For example, considering the complex network architecture of IoT, Wang et al. [31] proposed a cross-domain dynamic authentication with blockchain assistance, which incorporates accumulator knowledge and signature technology. An identity-based aggregate signcryption scheme with blockchain was proposed by Yang et al. [32] for IoT-enabled maritime transportation systems. Furthermore, several researchers cooperated (NI)ZKP to enhance privacy in blockchain-assisted authentication [33]. For unmanned aerial vehicles (UAV) networks, Andola et al. [34] proposed a NIZKP-based authentication and authorization scheme with blockchain assistance to realize distributed authentication between vehicles, where four approaches were designed for progressive functions, such as unlinkability, non-malleability, and trackability. Gabay et al. [35] integrated blockchain and ZKP for authentication between electric vehicles, which avoids the need for a central authentication server. Feng et al. [36] utilized ZKP, smart contract, and re-encryption to protect id privacy and data confidentiality, and ensure data availability, respectively. Kumar et al. [37] integrated permissioned blockchain and ZKP with deep learning to present secure and private data for industrial healthcare systems. However, existing blockchain-assisted authentication schemes with (NI)ZKP did not direct IIoT D2D communications.

Table 1
Notation I.

Notation	Meaning
κ	A security parameter
$\mathrm{MPK}, \mathrm{ASK}$	The master public key, the attribute secret key
$\mathrm{RPK}, \mathrm{RSK}$	The public and secret key for the AS
$\mathrm{IPK}', \mathrm{ISK}$	The incomplete public id key, the secret id key
IPK	The public identity key of a DU
$\mathrm{AK}, \mathrm{AUK}$	The transformed and the user attribute key
AuK	The authentication key of a DU
Ψ_{PoK}	A ZKP of knowledge of a discrete logarithm
Ψ_{PoE}	A ZKP of equality of two discrete logarithms
Ack	An acknowledgment of successful authentication
ATok	An attribute token for access
$\mathrm{SeK}', \mathrm{SeKey}$	A key agreement parameter, the session key
$\mathbb{G}, \mathbb{G}_T$	Two multiplicative groups of prime order p
g	A generator of $\mathbb{G}$
$\mathsf{H}_1, \mathsf{H}_2$	Five collision-resisted hash functions
KDF	A key Derivation function with l output
$\leftarrow \$$	randomly choose an element from a group
N	The number of attributes in attribute Universe
N_{AA}	The number of attributes in an AA's attribute set
N_{s}	The number of RS registered in the system
N_{u}	The number of users registered in the system
N_{a}	The number of attributes in a user set
$\lvert\mathrm{UAtS}\rvert$	The cost for storing attribute set
$\lvert\mathbb{G}\rvert, \lvert\mathbb{G}\rvert_{\mathrm{T}}, \lvert\mathbb{Z}_{\mathrm{p}}^*\rvert$	The size of an element in group $\mathbb{G}$, $\mathbb{G}_{\mathrm{T}}$ and $\mathbb{Z}_p^*$
$\mathbb{D}$	A set of $(\mathbb{G}, \mathbb{G}_T, p, g, e)$
$\lvert\mathrm{H}\rvert$	One hash function
$\lvert\Psi\rvert$	Communication overhead for a NIZKP
$\lvert\mathrm{Txt}\rvert$	Communication overhead for a BC transaction
$\mathrm{E}, \mathrm{E}_{\mathrm{T}}$	One exponentiation in group $\mathbb{G}, \mathbb{G}_T$
P, H	One pairing operation, one hash operation
BC_{c}	Overhead for consensus

Based on the above analysis, it can be obtained that there is no efficient and practical AKA scheme designed for IIoT D2D scenarios that have blockchain-supported authentication, NIZKP-based privacy preserve, and attribute-based access control simultaneously.

3. Preliminary

3.1. Notaions

The notations used in this article are summarized in Table 1.

3.2. ZKP

The concept of ZKP was introduced by Goldwasser et al. in 1989 [38], which generally was formulated as a decision problem. A ZKP

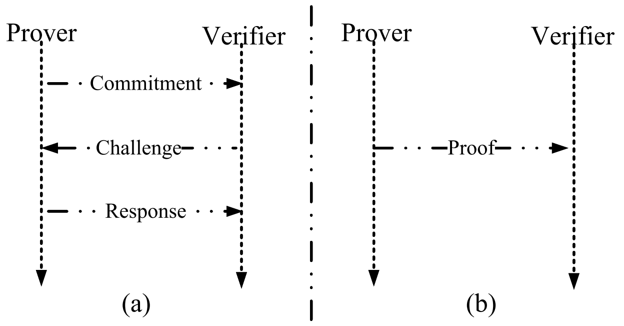


Fig. 2. (NI)ZKP model.

protocol involves two participants, a prover $\mathcal{P}$ and a verifier V. The prover P tries to prove a statement x is true to the V without revealing anything but the truth of the statement, where the statement is associated with a language of the NP (nondeterministic polynomially) class. The classical ZKP protocol, Sigma protocol, contains three rounds of message exchange between P and V, which are called commitment, challenge, and response (as depicted in Fig. 2(a)). However, interactions are infeasible in some situations, such as unstable wireless networks. Non-interactive ZKP was achieved by the Fiat–Shamir heuristic transformation [39], where a hash function is used to generate the challenge. In this way, only a single message is sent from V to P, as depicted in Fig. 2(b). The mathematical problem that forms the basis of a ZKP is essential, and paper [40] indicates that, under certain complexity assumptions, any NP problem can be used to define a ZKP and only problems in BPP ((bounded-error probabilistic polynomial-time) can be used to describe NIZKP.

Take Schnorr's ZKP protocol as an example, which has the following three properties:

(a) Perfect Completeness: If the statement provided by the prover is true, the prover can always convince the verifier that the statement is true.

(b) Special Soundness: If the provided by the prover is wrong, no prover can convince the verifier that the statement is true, except with a very small probability.

(c) Honest-verifier Perfect Zero-knowledge: If the statement provided by the prover is true, nothing can be gained by the honest verifier in addition to the facts during the verification. Given an honest-verifier simulator that runs in probabilistic polynomial time (PPT), zero-knowledge means that the results produced by a real proof are computationally indistinguishable from the ones produced by the simulator.

In this paper, we altered two discrete-logarithm-based NIZKP protocols to realize anti-key-escrow (certificateless) key generation and privacy-protected authentication. The underlying protocols are also known as Schnorr's Protocol and Chaum-Pedersen's protocol.

Definition 1. Proof of Knowledge of a Discrete Logarithm (PoK).

Given public parameters $(\mathbb{G}, g, p, \mathtt{H})$, where $\mathbb{G}$ is a multiplicative cyclic group of prime order p, g is a generator of $\mathbb{G}$, and $\mathtt{H}$ is a cryptographically-secure one-way hash function. For $\mathrm{Y} \in \mathbb{G}$, a representation of Y that related to g is an element $x \in \mathbb{Z}_p$ that satisfies the relation $\mathcal{R} = \{(x, \mathrm{Y}) \in \mathbb{Z}_p \times \mathbb{G} : g^x = \mathrm{Y}\}$. The prover $\mathcal{P}$ tries to convince a skeptical but honest verifier $\mathcal{V}$ that he/she knows a representation of a given Y without revealing anything about the secret x.

- $\mathcal{P}$ chooses $v \leftarrow \mathbb{Z}_p^*$ to compute $\mathrm{V} = g^v$, $c = \mathtt{H}(g, \mathrm{Y}, \mathrm{V})$ and $r = v - cx \ (mod p)$. $\mathcal{P}$ sends the proof $\Psi_{\mathrm{PoK}} = \langle \mathrm{Y}, \mathrm{V}, r \rangle$ to the verifier.
- $\mathcal{V}$ computes c first. If the condition $\mathrm{V} = g^r \cdot \mathrm{Y}^c$ holds, $\mathcal{V}$ accepts this proof, otherwise rejects.

Definition 2. Proof of Equality of two Discrete Logarithms (PoE).

Given public parameters $(\mathbb{G}, g, p, \mathtt{H}, \mathtt{H}')$, where $\mathbb{G}$ is a multiplicative cyclic group of prime order p, g is a generator of $\mathbb{G}$, and $\mathtt{H}$ and $\mathtt{H}'$ are two cryptographically-secure one-way hash functions. Given another parameter $h \in \mathbb{G}$, for $(\mathrm{Y}, \mathrm{Z}) \in \mathbb{G}^2$, a representation of (Y, Z) that related to (g, h) is an element $x \in \mathbb{Z}_p$ that satisfies the relation $\mathcal{R} = \{(x, (\mathrm{Y}, \mathrm{Z})) \in \mathbb{Z}_p \times \mathbb{G}^2 : log_g \mathrm{Y} = log_h \mathrm{Z}\}$. The prover $\mathcal{P}$ tries to convince a skeptical but honest verifier $\mathcal{V}$ that he/she knows a representation of a given Y without revealing anything about the secret x.

- $\mathcal{P}$ chooses $r \leftarrow \mathbb{Z}_p^*, \mathrm{R} \leftarrow \{0,1\}^*$ to compute $h = \mathtt{H}'(*, *, \mathrm{R})$, $p_1 = g^r$, $p_2 = h^r$, $c = \mathtt{H}(g, h, \mathrm{Y}, \mathrm{Z}, p_1, p_2)$ and $y = r + cx \ (mod p)$. Then, $\mathcal{P}$ sends the proof $\Psi_{\mathrm{PoE}} = \langle \mathrm{Y}, \mathrm{Z}, p_1, p_2, y \rangle$ along with R to the verifier.
- $\mathcal{V}$ computes h and c first. Then, it verifies if the following two conditions follow simultaneously. If yes, $\mathcal{V}$ accepts this proof, otherwise rejects it.

$$g^y = p_1 \cdot \mathrm{Y}^c, \quad h^y = p_2 \cdot \mathrm{Z}^c.$$

3.3. Permissioned blockchain

In blockchain systems, the security of a ledger is assured by the hash-value-linked structure and the consensus algorithm. The latest block contains the hash value of the previous block header. The right to generate a new block is decided by the consensus algorithm. After the ledger is updated, the new ledger will be broadcasted to the whole system, and honest nodes will update their local ledgers to accomplish global consistency. In this way, no change can be accomplished unless the adversary can control more than 51% power of the whole system.

According to ownership, blockchain can be divided into two categories, permissioned and permissionless. In a permissionless one, also known as a public chain, such as Ethereum, any user can join the blockchain and participant in the consensus process to compete for bookkeeping rights. On the contrary, nodes in a permissioned blockchain, such as Hyperledger Fabric, require legal authorization from a membership management server. In other words, all users can read the ledger while only permissioned nodes can write the ledger. Considering the characteristics of these two blockchains, most believe that permissioned chains are more suitable for most practical application scenarios [41].

3.4. Attribute-based encryption

Waters [42] proposed an expressive, efficient, and provably secured ciphertext-policy attribute-based encryption scheme, which contains the following algorithms:

- $Setup(\lambda, \mathcal{U}) \rightarrow (\mathrm{MPK}, \mathrm{MSK})$: a center authorization takes as input a security parameter λ and an attribute universe $\mathcal{U}$ to run this algorithm to output system public and private key $(\mathrm{MPK}, \mathrm{MSK})$.
- $Enc(\mathrm{MPK}, \mathrm{MSG}, \phi) \rightarrow \mathrm{CT}$: this encryption algorithm takes as input the system public key, the to-be-encrypted message MSG, and an access structure ϕ to generate ciphertext CT such that only a user whose attribute set satisfies the access structure can decrypt successfully.
- $KeyGen(\mathrm{MPK}, \mathrm{MSK}, \mathcal{S}) \rightarrow \mathrm{DK}$: this key generation algorithm takes as input the system public and private keys, and a user attribute set $\mathcal{S}$ to output a decryption (private) key DK for the user.
- $Dec(\mathrm{MPK}, \mathrm{CT}, \mathrm{DK}) \rightarrow \mathrm{MSG}$: this decryption algorithm takes as input the system public key, a ciphertext (related to an access structure), and a decryption key (related to a set of attributes). If the attribute set satisfies the access structure, then the algorithm will output valid plaintext MSG.

This paper incorporates an attribute-based encryption scheme to implement fine-grained attribute-based access control.

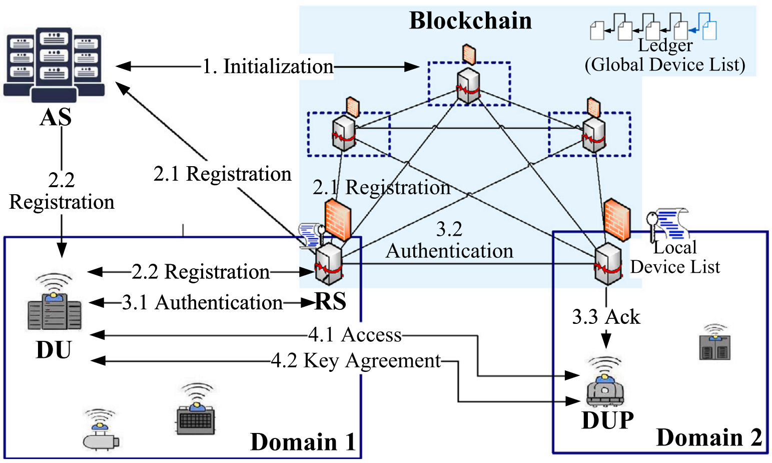


Fig. 3. System procedure.

4. Model definition

4.1. System model

The system model of our blockchain-enforced cross-domain authentication and key agreement and access control scheme contains the following four participants:

(**1**) **RS** (**Registration Sever**) : Devices within one trust domain consist of a subnet, which has a management node/ gateway node that is responsible for device registration. Assuming each RS is equipped with a security component, TPM.

(**2**) **Blockchain** (**BC**) : There are two ways to construct a blockchain network, using those registration servers or using non-related servers. Assuming the registration servers are resource-adequate in our scheme, the second way is adopted.

(**3**) **AS** (**Attribute Server Cluster**) : AS usually consists of several resource-adequate servers, which are used to generate attribute keys for devices for access requests. Similarly, AS is equipped with a TPM either.

(**4**) **DU** (**Data User**) : DUs are IIoT end devices with limited storage and computation resources. Considering the future IIoT, those devices are equipped with SE. For clarity, we note two devices involved in one authentication as DU and DUP, which belong to two trust domains.

4.2. System procedure

The basic procedures of our BP-AKAA scheme are illustrated in Fig. 3, which consist of the following four phases:

1. **System Initialization**

The AS runs the $\mathsf{Setup}(1^{\kappa}) \rightarrow (\mathsf{MPK}, \mathsf{ASK})$ algorithm to initialize the system, which takes a security parameter κ to generate the master public key MPK and the attribute secret key ASK. Then it publishes MPK in the blockchain while keeping MSK secret.

2. **Registration**

2.1 RS registration. This algorithm is conducted between an RS and the AS, where the AS acts as a KGC.

Firstly, a RS with a TPM performs the $\mathsf{IdKeyGen}_{\mathsf{u}}(\mathsf{MPK}) \rightarrow (\mathsf{IPK}', \mathsf{ISK})$ algorithm. Then it constructs a zero-knowledge proof Ψ_{PoK} of its secret id key ISK and sends it to the AS for registration. Then, the AS verifies this proof. If passed, the AS runs the $\mathsf{IdKeyGen}(\mathsf{MPK}, \mathsf{IPK}') \rightarrow \mathsf{IPK}$. Next, the AS sends IPK back to the RS. After all the RSs are registered, they initialize a blockchain system by writing their public keys in the genesis block/ underlying code.

2.2 DU registration. This algorithm is conducted between a DU, its RS (BC), and the AS (responsible for attribute key generation).

Firstly, the DU with a SE performs the $\mathsf{IdKeyGen}_{\mathsf{u}}(\mathsf{MPK}) \rightarrow (\mathsf{IPK}', \mathsf{ISK})$ algorithm. Then it constructs a zero-knowledge proof Ψ_{PoK} of its secret id key ISK and sends it to the RS for registration.

Secondly, the RS verifies this proof. If passed, the RS runs the $\mathsf{IdKeyGen}(\mathsf{MPK}, \mathsf{IPK}') \rightarrow \mathsf{IPK}$. Next, it stores IPK in the blockchain ledger (after consensus) and sends $(\mathsf{IPK}, \mathsf{IPK}')$ to the AS. (The RS sends the signed message to the AS for verification.)

Third, the AS runs the $\mathsf{AKeyGen}(\mathsf{MPK}, \mathsf{ASK}, \mathsf{IPK}, \mathsf{IPK}') \rightarrow \mathsf{AK}$ to generate the attribute key based on a set of attributes (the authorization of this user attribute set is accomplished offline, which is beyond the scope of this paper) and sends it back to the DU.

Finally, the DU runs the $\mathsf{UKeyGen}_{\mathsf{u}}(\mathsf{MPK}, \mathsf{IPK}, \mathsf{AK}) \rightarrow$ $(\mathsf{AuK}, \mathsf{AUK})$ algorithm to gain its valid authentication key and attribute key.

3. **Authentication**

The authentication process involved the DU and the DUP with the assistance of the BC.

First, the DU constructs a NIZKP signature Ψ_{PoE} on a self-selected random number using its secret id key and sends this proof to the BC.

Then, the BC verifies this proof. If that passes, it sends an acknowledgment Ack to the DU and the DUP.

4. **Access Control and Key Agreement**

This phase involves the DU and the DUP.

After receiving a valid acknowledgment from the BC, the DU performs the $\mathsf{AxTokGen}(\mathsf{MPK}, \mathsf{AUK}, \mathsf{Ack}) \rightarrow \mathsf{ATok}$ algorithm to get an access token. Then it sends the token to the DUP.

The DUP runs the $\mathsf{Vrf}(\mathsf{MPK}, \mathsf{ATok}, \mathsf{IPK}, \mathsf{Ack}) \rightarrow 0/1$ algorithm. If the result equals 1, it performs the key agreement.

The DUP runs the $\mathsf{KeyAgm}(\mathsf{MPK}, \mathsf{ATok}) \rightarrow (\mathsf{SeK}', \mathsf{SeKey})$ algorithm. Moreover, it sends a parameter SeK' and the corresponding NIZKP to the DU. Finally, the DU verifies the proof and performs the $\mathsf{KeyAgm}'(\mathsf{MPK}, \mathsf{SeK}') \rightarrow \mathsf{SeKey}$ algorithm to gain the same session key.

4.3. Threat model

The widely adopted "Dolev–Yao" (DY) threat model [43] indicates that the communicating entities, including DUs (IIoT devices), RSs, and the AS, are not fully trustworthy, and data sharing is realized over insecure public channels. More specifically, in our scheme, based on capability, we divide adversaries into the following three types:

(1) Outsider attackers. Generally speaking, an outsider attacker has the following capabilities, eavesdropping on all communication links, recording and replaying messages, and decomposing and reassembling messages.

(2) Corrupted users (normal insider attackers). An attacker can break a DU and use the DU's private key to deceive the servers or

decrypt messages. But an attacker cannot forge valid messages with the user's private id key stored in the SE.

(3) Corrupted servers (strong insider attackers). In addition to the above two situations, an attacker can further manipulate the AS or the RSs or access their database to gain information.

5. Concrete construction

This section demonstrates details for constructing the BP-AKAA scheme, which contains the following four phases.

Phase 1. **System Initialization**.

(1) The AS takes a security parameter $\kappa \in \mathrm{N}$ to generate two multiplicative cyclic groups $\mathbb{G}, \mathbb{G}_T$ of prime order p. Let g be a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Let $\mathbb{D} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$ be a bilinear group. Then, the AS selects two one-way hash functions: $\mathtt{H}_1 : \{0,1\}^* \to \mathbb{G}$ and $\mathtt{H}_2 : \{0,1\}^* \to \mathbb{Z}_p^*$ for constructing NIZKP. Moreover, a key derivation function (KDF) [44] is selected for generating symmetric encryption keys.

(2) The attribute server AS selects $\mathrm{RSK} = \alpha \leftarrow \mathbb{Z}_p^*$ as its private key, then it computes and publishes the publish key $\mathrm{RPK} = g^{\alpha}$ in the blockchain and the whole system.

(3) Let $\mathrm{U_A}$ be the attribute universe with N attributes. The AS selects $t_i \leftarrow \mathbb{Z}_p^*$ for each attribute $a_i \in \mathrm{U_A}$ to compute attribute public keys $T_i = \{g^{t_i}\}$.

Finally, the AS publishes the master public key MPK in the blockchain and keeps the attribute secret key ASK private (kept in its TPM):

$$\mathrm{MPK} = (\mathbb{D}, \mathtt{H}_1, \mathtt{H}_2, \{T_i\}_{i \in [1,N]}, \mathrm{KDF}, \mathrm{RPK}),$$

$$\mathrm{ASK} = (\{t_i\}_{i \in [1,N]}, \mathrm{RSK}).$$

Note that there are two ways to construct a blockchain. The first one is initializing a blockchain independently. In this way, the master public key MPK will be sent to the blockchain through a transaction and be stored in the first block publicly. The second way is constructing a blockchain with those registration servers of all domains. Then, the master public key MPK will be written in the genesis block or the underlying code.

Phase 2. **Registration**

I : RS Registration.

An RS chooses $\mathrm{ISK} = \beta'$ as its private id key (using its TPM to generate and store) and computes its incomplete public id key $\mathrm{IPK}' = g^{\beta'}$. Then, it constructs a NIZKP of its private id key and then sends it to the AS. Next, the AS verify the NIZKP. If that passes, the AS calculates the complete public id key of the RS as $\mathrm{IPK} = (g^{\beta'})^{\gamma'}$ with a newly selected random number γ'. Then the AS sends the public key back to the RS. The interactions between an RS and the AS are depicted in Fig. 4. Note that the computational details of the NIZKP of knowledge are as described in Section 3.2 Definition 1.

II : DU registration.

As demonstrated in Fig. 5, the user registration phase contains the following four steps.

(1) To avoid the key escrow issue of a central RS, we adopt the joint key generation method in certificateless schemes. Firstly, the DU (e.g. an IIoT device) with an embedded device number GID generates its private id key using its SE as $\mathrm{ISK} = \beta = \mathtt{H}_2(GID \| psw)$, where $psw \leftarrow \{0,1\}^*$ is a self-defined password. Then, it computes its incomplete public id key $\mathrm{IPK}' = g^{\beta}$, and constructs a NIZKP of knowledge of its private id key as follows:

It selects $\mu \leftarrow \mathbb{Z}_p^*$ to compute

$$u = g^{\mu}, \quad c = \mathtt{H}_2(g, \mathrm{IPK}', u),$$

$$\xi = \mu - c\beta \ (mod p).$$

Finally, it sends $\Psi_{\mathrm{PoK}} = \langle \mathrm{IPK}', u, \xi \rangle$ to the RS.

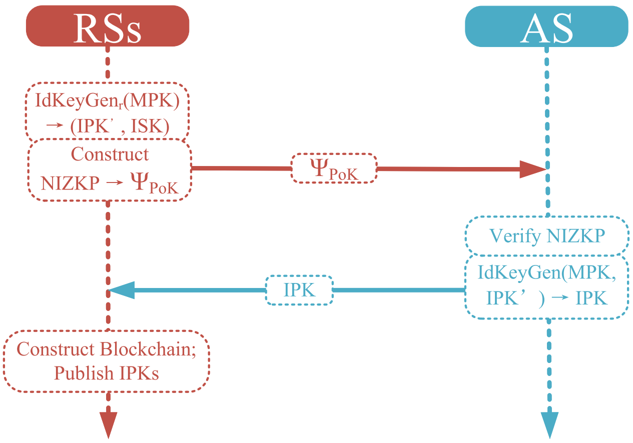

RSs
AS
IdKeyGen$_r$(MPK) → (IPK' , ISK)
Construct NIZKP → Ψ_{PoK}
Ψ_{PoK}
Verify NIZKP
IdKeyGen(MPK, IPK') → IPK
IPK
Construct Blockchain; Publish IPKs

Fig. 4. RS registration.

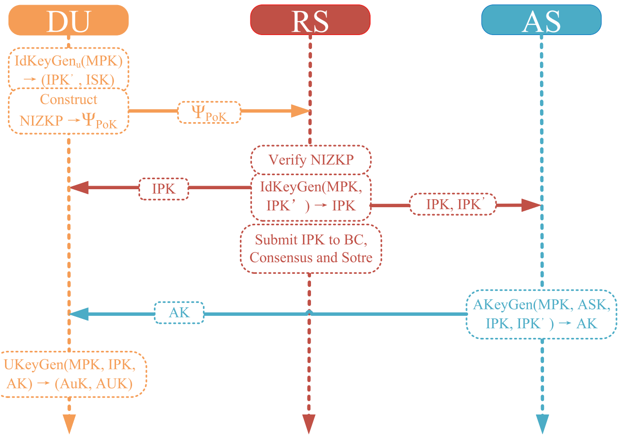

DU
RS
AS
IdKeyGen$_u$(MPK) → (IPK' , ISK)
Construct NIZKP → Ψ_{PoK}
Ψ_{PoK}
Verify NIZKP
IdKeyGen(MPK, IPK') → IPK
IPK
IPK, IPK'
Submit IPK to BC, Consensus and Sotre
AK
AKeyGen(MPK, ASK, IPK, IPK') → AK
UKeyGen(MPK, IPK, AK) → (AuK, AUK)

Fig. 5. DU registration.

(2) The RS verifies the NIZKP Ψ_{PoK}. It computes $c = \mathtt{H}_2(g, \mathrm{IPK}', u)$ first. If

$$u = g^{\xi} \cdot (\mathrm{IPK}')^{c},$$

proceed. Otherwise, the RS rejects this registration request.

The RS selects $\gamma \leftarrow \mathbb{Z}_p^*$ to compute the DU's public id key as $\mathrm{IPK} = (g^{\beta})^{\gamma}$. Then it sends IPK to the BC to store (as the RSs are part of the BC, storing a key is submitting a **signed** transaction with its secret id key to the BC, and the key will be written in the ledger after a consensus process, which essentially is a signature verification process. In this way, the validity of the public id key is guaranteed). Next, the RS sends $(\mathrm{IPK}', \mathrm{IPK})$ to the AS to request the attribute key for the DU.

(3) After receiving the key request from the RS, the AS first check the validity (The AS verifies the signature of the RS). If yes, it gets the DU's attribute set $\mathrm{A_U} = \{a_i\}_{i \in [n]}$ $(n < \mathrm{N})$ that is authorized offline. Then, it selects $\delta \leftarrow \mathbb{Z}_p^*$ to compute

$$\mathrm{AK}_1 = (\mathrm{IPK})^{\alpha} \cdot (g^{\sum_n t_i})^{\delta}, \quad \mathrm{AK}_2 = (\mathrm{IPK}')^{\delta}.$$

Finally, the AS sends the attribute key $\mathrm{AK} = (\mathrm{AK}_1, \mathrm{AK}_2)$ along with IPK back to the DU.

(4) The DU computes its authentication key $\mathrm{AuK} = IPK^{1/\beta}$ and its valid attribute user key $\mathrm{AUK}_1 = \mathrm{AK}_1, \mathrm{AUK}_2 = (\mathrm{AK}_2)^{1/\beta}$. Finally, the DU stores (ISK, IPK, AuK, AUK) in its SE and sends AuK in the BC.

It is worth mentioning that the registration phase in our scheme requires no secret channel between the DU and the RS or the DU and the AS.

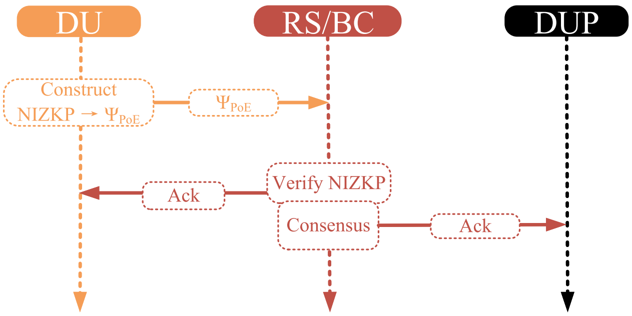


Fig. 6. Authentication.

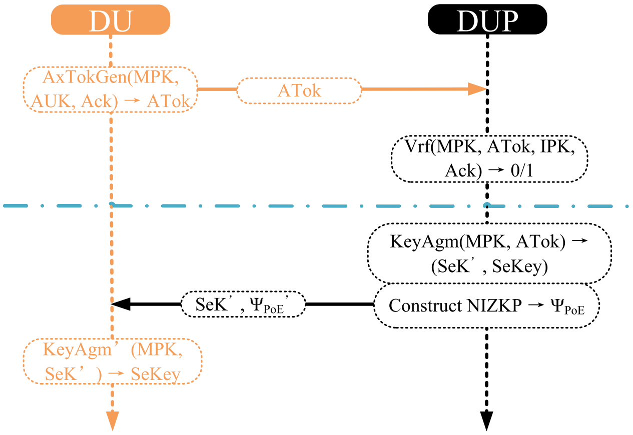


Fig. 7. Access and Key Agreement.

Phase 3. Authentication

Fig. 6 illuminates the authentication phase of our BP-AKAA scheme. It can be seen that the authentication between devices DU and DUP requires the assistance of the blockchain (not direct authentication or relying on a central server). Specifically, it costs one interaction between the DU and the BC, and the DUP can accept the DU after receiving an acknowledgment from the BC. As the smart contract used for authentication is performed automatically, the DUP can approve the result without extra checking.

Assuming the aim of the authentication is to build a secure communication channel between device DU (sender) and device DUP (receiver). Let IPK_r and IPK_s be the public id key of the receiver DUP and the sender DU, respectively.

(1) Let $msg \leftarrow \{0,1\}^*$ be the authentication request message and TS be a timestamp. The DU selects random numbers $r \leftarrow \mathbb{Z}_p^*$ and $\text{R} \leftarrow \mathbb{G}$, and constructs a NIZKP of equality as follows:

The DU computes $h = \mathtt{H}_1(\text{TS}, msg, \text{R})$ to calculate $\text{Z} = h^{\beta}$ with it identity secret key β. Then, it calculates

$$p_1 = h^r, \quad p_2 = \text{AuK}_\text{s}{}^r,$$

$$c = \mathtt{H}_2(\text{AuK}_\text{s}, h, \text{IPK}_\text{s}, \text{Z}, p_1, p_2), \quad y = r + c\beta\ (mod p).$$

Finally, it sends $\Psi_{\text{PoE}} = \langle \text{IPK}_\text{s}, \text{Z}, p_1, p_2, y\rangle$ along with $(\text{TS}, msg, \text{R}, \text{AuK}_\text{s})$ to the BC.

(2) The BC (smart contract) verifies this proof by firstly checking the validity of the received timestamp $\overline{TS} - TS < \Delta T$, where $\overline{TS}$ is the time when the proof is received, and ΔT is a pre-setted maximum transmission delay.

If passes, it computes $h = \mathtt{H}_1(\text{TS}, msg, \text{R})$ and $c = \mathtt{H}_2(\text{AuK}_\text{s}, h, \text{IPK}_\text{s}, \text{Z}, p_1, p_2)$ with BC-stored keys of the sender $\text{AuK}_\text{s}, \text{IPK}_\text{s}$. Next, it checks if both the following two equations hold.

$$h^y = p_1 \cdot \text{Z}^c, \quad \text{AuK}_\text{s}{}^y = p_2 \cdot \text{IPK}_\text{s}^c.$$

If yes, it accepts this authentication request and sends an acknowledgment $\text{Ack} \in \mathbb{G}$ to the DU while sending $(\text{Ack}, \text{TS}, \text{IPK}_\text{s})$ to the DUP. This step is accomplished by smart contract automatically.

Phase 4. Access and Key Agreement

The access and key agreement between the DU and the DUP are depicted in Fig. 7.

(1) After receiving the valid Ack, the DU selects three numbers $\theta, s, y_A \leftarrow \mathbb{Z}_p^*$ to generate an attribute access token.

$$\sigma_1 = \text{AUK}_1 \cdot (\prod_n (T_i))^{\theta} \cdot \text{Ack}^s,$$

$$\sigma_2 = \text{AUK}_2 \cdot g^{\theta}, \quad \sigma_3 = g^s, \quad \sigma_4 = g^{y_A}.$$

Next, it sends $\text{ATok} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, A_U)$ to the DUP.

(2) The DUP estimates if the DU can access its data by checking the following equation:

$$\frac{e(g, \sigma_1)}{e(\prod_n (T(i)), \sigma_2) \cdot e(\text{Ack}, \sigma_3)} = e(g^{\alpha}, \text{IPK}_s).$$

If yes, it proceeds.

Correctness:

$$\begin{aligned} \frac{e(g, \sigma_1)}{e(\prod_n (T(i)), \sigma_2) \cdot e(\text{Ack}, \sigma_3)} &= e(g^{\alpha}, \text{IPK}_\text{s}) \\ \frac{e(g, \text{AUK}_1 \cdot (\prod_n (T_i))^{\theta} \cdot \text{Ack}^s)}{e(\prod_n (T(i)), \text{AUK}_2 \cdot g^{\theta}) \cdot e(\text{Ack}, g^s)} &= e(g^{\alpha}, (g^{\beta})^{\gamma}) \\ \frac{e(g, (g^{\beta\cdot\gamma\cdot\alpha}) \cdot (\prod_n (T_i))^{\delta} \cdot (\prod_n (T_i))^{\theta} \cdot \text{Ack}^s)}{e(\prod_n (T(i)), g^{\delta} \cdot g^{\theta}) \cdot e(\text{Ack}, g^s)} &= e(g^{\alpha}, (g^{\beta})^{\gamma}) \\ e(g, (g^{\beta\cdot\gamma\cdot\alpha})) &= e(g^{\alpha}, (g^{\beta})^{\gamma}) \end{aligned}$$

(3) The DUP performs the key agreement section. It selects $y_B \leftarrow \mathbb{Z}_p^*$ to compute the seed of the session key as $\text{Seed} = e(\sigma_4, g^{\alpha})^{y_B} \| \text{TS} \| TS'$. Then, it uses the KDF to generate the session key as $\text{SeKey} = \text{KDF}(\text{Seed}, l)$. Next, it sends $\text{SeK}' = g^{y_B}$ along with a NIZKP Ψ_{PoE} of y_B (requires DUP's private id key and a timestamp TS′) to the DU.

(4) The DU first verifies the NIZKP of SeK′ with the DUP's public id key. If this passes, it uses the parameter g^{y_B}, the parameter y_A and the timestamp TS′ to compute the session key $\text{SeKey} = \text{KDF}(e(g^{y_B}, g^{\alpha})^{y_A} \| \text{TS} \| \text{TS}', l)$. Finally, these two data users (IIoT devices) can implement secure communication with symmetric encryption.

6. Security analysis

According to the threat model defined in Section 4.3, there are three levels of adversaries. In this section, we prove that our BP-AKAA scheme can resist not only general attacks, such as the impersonation attack, the replay attack, and the man-in-the-middle attack but also more advanced attacks, such as the RS corruption attack and the user-server attack. Note that the security against chosen-message attacks of a ZKP-based signature scheme is proved in [7]. Due to similarity, we omit details of the CMA security of our proposed authentication scheme.

(1) Resistant to impersonation attack: Impersonation attack means that an adversary $\mathcal{A}$ tries to masquerade as a registered DU. In our scheme, $\mathcal{A}$ knowing the identity and the attribute set of the registered user cannot generate a valid NIZKP proof Ψ_{PoE} as long as the identity secret key of the registered user is confidential. Specifically, $\mathcal{A}$ can eavesdrop $\text{IPK}' = g^{\beta}$, $\text{IPK} = g^{\beta\cdot\gamma}$, AK and even $\text{AuK} = g^{\gamma}$, but the identity secret key β and valid AUK are kept secret from $\mathcal{A}$. As a result, our BP-AKAA scheme can eliminate the user impersonation attack.

(2) Resistant to replay attack: Essentially, the NIZKP proof Ψ_{PoE} is a signature with the private signing key β. Benefiting from the usage of timestamps in our scheme, an adversary $\mathcal{A}$ who intercepts an authentication message from the public channel and resends it after a while cannot pass the verification. In a word, our BP-AKAA scheme can resist replay attacks.

(3) Resistant to man-in-the-middle attack: Assuming an adversary $\mathcal{A}$ obtains a valid message $\Psi_{\text{PoE}} = \langle \text{IPK}, \text{Z}, p_1, p_2, y\rangle$ along with

$(\mathrm{TS}, msg, \mathrm{R}, \mathrm{AuK})$. $\mathcal{A}$ tris to implement the man-in-the-middle attack by generating its own timestamp TS'. $\mathcal{A}$ can compute $h' = \mathtt{H}_1(\mathrm{TS}', msg, \mathrm{R})$, but it cannot calculate the corresponding Z as $\mathrm{Z}' = (h')^{\beta}$ that can pass the verification without knowing the identity secret key β of the original DU. Hence, our BP-AKAA scheme has the ability to resist the man-in-the-middle attack.

It is worth mentioning that our BP-AKAA scheme also satisfies non-repudiation, which is a critical characteristic of a signature scheme. Because the identity secret key of a DU is seen only for the DU, and our scheme is proven to resist the abovementioned three attacks, no adversary or any participant can forge a valid message to pass the authentication verification. Thus, given a valid message, a corresponding user cannot later deny the transmission of this message.

(4) Resistant to corrupted RS (AS): It is straightforward that our authentication scheme can resist dishonest registration servers as the registration phase of our scheme is key-escrow, including the RS registration and the DU registration. In other words, even the RS (AS) of a domain responsible for registering a valid user cannot impersonate the user because it did not know the paired identity secret key. Furthermore, although all public keys of valid users are stored in the distributed ledger in our scheme, no RS or blockchain node can realize this attack successfully owing to the same reason.

Another security advantage that can be gained by no key escrow, is resistance to server collaboration attacks. This means that, with the collaboration between the AS and an RS, they cannot forge valid keys for unregistered users, or forge valid NIZKP with invalid keys for registered users (frame).

(5) Resistant to database insertion attack: Recall that in our scheme, the id key generation and the attribute key generation for a DU are conducted by two servers. Assuming $\mathcal{A}$ can insert the database of RSs (basically equal to the public ledger) to get a valid public key of a user, yet it cannot generate a valid attribute key for the user as it did not know the RSK of the AS (which is stored in the security component, TPM). It is notable that the adversary cannot deceive the AS that he is the RS, because there is a verification process between the AS and the RS. Furthermore, assuming $\mathcal{A}$ can insert the database of the AS, it still cannot generate a valid attribute key for the user due to the same reason.

(6) Resistant to users collaboration attack: This attack mainly targets attribute key abuse, which indicates that several legal users may collaborate and combine their attribute keys to gain unauthorized access. Recall part of a valid attribute key is formed as $\mathrm{AK}_1 = (g)^{\beta\gamma\alpha} \cdot (g^{\sum_n t_i})^{\delta}$. The belonged DU only controls the parameter β, thus the combined attribute key cannot pass the access verification. In a word, DUs in our scheme cannot collaborate to gain extra access power.

(7) Resistant to user-server collaboration attack: A DU may conspire with its RS to violate the privacy of another DU (within or beyond its domain) by revealing its own private id key. However, the private id key of each DU in our scheme is chosen by the user itself, which indicates strong independence. Thus, even the RSs or the blockchain nodes that know all public id keys and the private id key of the malicious user cannot disclose the privacy of other users. This analysis shows that our BP-AKAA scheme can resist the user-server collaboration attack.

7. Performance analysis

7.1. Functionality comparison

We compare the functionality comparison of our system with related schemes in Table 2, including identity-based authentication, attribute-based authentication, and (NI)ZKP-based authentication.

Concerning id-based authentication (and key agreement) schemes [9,10], and [11,21,22] (designed for D2D IIoT applications), all of them did not consider some critical functions, such as cross-domain, key escrow resistance, identity privacy protection, and access control. Scheme [14] and scheme [19] adopted attribute-based access control and attribute-based encryption separately, yet both two schemes did not support key agreement and blockchain-assisted cross-domain mutual authentication. Similarly, another two (NI)ZKP-based privacy-protected schemes [25,28] are not blockchain-assisted either.

Although schemes [35,36] combined the blockchain and ZKP to accomplish authentication, they did not realize functions such as key agreement, key escrow resistance, and secret channel free. Scheme [34] incorporated NIZKP and blockchain to accomplish authentication between devices through transactions without considering access control and key agreement, thus cannot be used for secure communication. Nevertheless, our proposed BP-AKAA scheme accomplishes not only some essential characteristics for authentication, including mutuality, privacy-preserving, secret channel free, and key agreement, but also some additional benefits for secure communications, such as blockchain-assisted cross-domain, key escrow free, and attribute-based threshold access control. Thus, it can be concluded that our BP-AKAA scheme has excellent functional superiority compared with other authentication schemes.

7.2. Performance comparison

We compare the performance of our system with existing ones in three aspects, communication overhead, storage overhead, and computation overhead (which contains theoretical analysis and simulation results).

7.2.1. Communication overhead

The communication overheads between different participants are depicted in Table 3. The initialization phase generates the master public key that contains N attribute public keys, which need to be sent to the BC. Besides, all communication costs between the RSs and the DUs, or between other participants are constant. Therefore, it is concluded that our BP-AKAA scheme is applicable and practical for IIoT scenarios with tons of end devices.

In our scheme, there are two types of NIZKP messages, Ψ_{PoK} and Ψ_{PoE}. As defined in Section 3.2, the size of the first type is one element of the group $\mathbb{G}$ and one element of the group $\mathbb{Z}_p^*$, and the size of the second type is four elements of the group $\mathbb{G}$ and one element of the group $\mathbb{Z}_p^*$.

7.2.2. Storage overhead

The storage overhead of our scheme is concluded in Table 4. In our BA-AKAA scheme, the public keys of the RSs and DUs are publicly stored in the BC and the authentication section does not involve the AS, thus the storage overhead of the AS is only $(1+\mathrm{N})|\mathbb{Z}_p^*|$ (no space is costed for storing public keys) and that of the BC is related to the number of attributes, users and RSs $((\mathrm{N} + \mathrm{N}_{\mathrm{u}} + \mathrm{N}_{\mathrm{s}})|\mathbb{G}|)$. For the same reason, the storage overhead of each RS is only one element in group $\mathbb{Z}_p^*$ (its secret key). Furthermore, the storage overhead of a DU in our scheme is constant (irrelative to the number of attributes N_a contained in its attribute set), which is suitable for resource-limited IIoT devices.

The storage advantage of our BP-AKAA scheme is highlighted by comparison with three other schemes, [14,19,34]. For attribute-based AKA scheme [14] (single registration server, no cross-domain, no key agreement, with access control) and [19] (multi-attribute-authority, no key agreement, with attribute-based encryption and access control), the storage overheads of all participants are related to the number of users or attributes, which means the DU (IIoT device) requires more storage space to govern its keys. Scheme [34] (cross-domain, with blockchain assistance, no access control) realized blockchain-assisted authentication without supporting access control, thus the storage overhead of it is solely related to the number of users.

Table 2
Comparison of functionality.

Scheme	Theory	BC (distributed)	CroDom	No Key-Esc	Pri-P	Mu-Authen	AccCont	Key-Agree	No SecChan
[9]	IB+PW	×	×	✓	✓	✓	×	×	×
[10]	IB-group	×	×	×	Anonymity	✓	×	✓	×
[11]	IB+CL	×	✓	✓	✓	✓	×	×	×
[21]	IB	×	×	×	ID-Conf	✓	×	✓	×
[22]	IB	×	×	×	×	✓	×	✓	×
[14]	AB+NIZKP	×	×	✓	✓	×	✓	×	✓
[19]	AB+ZKP	×	×	✓	×	×	✓	×	×
[25]	NIZKP	×	×	×	✓	✓	×	✓	✓
[28]	ZKP+PW	×	×	×	✓	×	×	×	✓
[34]	NIZKP	✓	✓	✓	✓	×	×	×	✓
[35]	ZKP	✓	×	×	✓	×	×	×	×
[36]	ZKP	✓	×	×	✓	×	×	×	×
Ours	AB+NIZKP	✓	✓	✓	✓	✓	AB-Threshold	✓	✓

Abbreviations: ✓: supports functionality. ×: do not support functionality. CroDom: cross-domain. Key-Esc: key escrow. Pri-P: privacy-preserving. Mu-Authen: mutual authentication. AccCont: access control. Key-Agree: key agreement. SecChan: secret channel.

Table 3
Communication overhead.

Phases	Communication Overhead
Initialization (AS → BC)	$\vert\mathbb{D}+2\vert\mathrm{H}\vert+(\mathrm{N}+1)\vert\mathbb{G}\vert+\vert\mathrm{KDF}\vert$
RS Rg (RS→AS; AS→RS; RS→BC)	$1\vert\Psi\vert$; $1\vert\mathbb{G}\vert$; $\vert\mathrm{Txt}\vert$
DU Rg (DU→RS; RS→DU; RS→BC)	$1\vert\Psi\vert$; $1\vert\mathbb{G}\vert$; $\vert\mathrm{Txt}\vert$
DU Rg (RS → AS; AS → DU)	$2\vert\mathbb{G}\vert$; $2\vert\mathbb{G}\vert$
Au (DU→RS; RS→DU(P); RS→BC)	$1\vert\Psi\vert$; $1\vert\mathbb{G}\vert$; $\vert\mathrm{Txt}\vert$
AcKA (DU→DUP; DUP→DU)	$4\vert\mathbb{G}\vert+\vert\mathrm{UAtS}\vert$; $1\vert\Psi\vert+1\vert\mathbb{G}\vert$

7.2.3. Computational overhead

A: Theoretical Analysis

Based on phases and participants, Table 5 depicts theoretical analysis and comparison of computation overheads. For simplicity and without loss of generality, cost-lighted operations, including scalar multiplication, modular multiplication, KDF, and modular addition and subtraction are omitted (focused on exponentiation and pairing operations).

Compared with the two AKAA schemes [14,19], the computational overhead of user registration of our scheme's DU is lightweight (no multiple exponentiations and pairings). Moreover, unlike schemes [14, 19], the operations for access control of our scheme are constant and irrelevant to the number of attributes. Besides, as the attributes of scheme [19] are controlled by multiple authorities, the DU needs to calculate multiple ZKP messages for authenticating with related authorities, which quite increases the communication and computation overheads. On the other hand, focusing on the authentication phase of the blockchain-assisted scheme [34] (the fourth approach) and ours, our scheme has computational superiority in both the DU and the BC part in spite of that scheme [34] only achieves authentication without access control and key agreement. In a word, the theoretical analysis demonstrates that our scheme is more efficient and practical for resource-constrained IIoT devices on the bases of satisfying reliable authentication and access control.

B: Simulation Comparison

All simulation experiments are performed in a VMware virtual machine, including the blockchain part. The detailed settings of the experiment environment are summarized in Table 6. We compare the computation performance of our scheme with three related approaches [14,19,34], where [14,19] are two ZKP-based authentication schemes with attribute-based access control, and [34] is a blockchain-assisted ZKP-based authentication scheme.

The computation overhead of the setup phase, the user registration phase of the user, and the key generation server are indicated in Fig. 8(a), (b), and (c). It can be gain straightforward that the setup overheads of the three schemes grow linearly with the number of attributes. Regarding the user registration phase, i.e. the authentication between the user and the verifier (key generation server), Fig. 8(b) shows that the overhead of users in our scheme is a constant (including Phase 2.1.1 and 2.1.4 in Section 5) while that of the other two schemes are linearly related with the number of attributes in a user attribute set. Similarly, the overhead of the server (including the RS and the AS) in our scheme is also a constant. Yet the result of scheme [14] is relative to the number of attributes, and that of scheme [19] is relative to the number of attributes and the overhead of the polynomial generation.

Fig. 9 compares the authentication and verification algorithms of the four schemes, where the prover represents the DU, and the verifier represents the blockchain node in our scheme. Scheme [19] utilized PoK instead of PoE thus its time consumptions are less than the other three schemes that utilize PoE. Moreover, the results of scheme [34] are higher than the others as this scheme realizes additional features such as unlinkability, sender trackability, and malleability attack resistance. As we can see from Fig. 9 that our scheme's overhead is on the medium and is slightly high than that of scheme [19] because our scheme assigned the verifier with the associated identity public key.

Furthermore, in addition to simulating ZKP algorithms with the Go language, we also test the transaction per second (TPS) of ZKP verification algorithms of those four schemes with the Hyperledger Fabric platform. The results are depicted in Fig. 10, where the TPS decreases with the number of endorsers (blockchain nodes that participate in the consensus process). It can get that the ZKP used for user registration in our scheme (PoK defined in Section 3.2) is roughly equal to that in scheme [19], and the ZKP used for authentication (PoE defined in Section 3.2) essentially resembles that in schemes [14,34].

To better comparison, we divided the access part of the three schemes from the authentication phase and set the threshold of the scheme [14] as t $= n = N_a$, as illuminated in Fig. 11, where the sub-fig (a) and the sub-fig (b) contains the overhead of users (DU) and the verifier (DUP in our scheme) respectively. Fig. 11 highlights an excellent advantage of our scheme, which is that both overheads of the DU and the DUP are independent of the number of attributes required for access (more suitable for resource-constrained IoT devices), while the overheads of the other two schemes are burdensome.

We built a more realistic test environment to further test the feasibility and practicality of our proposed scheme, as shown in Fig. 12. The computer on the left is connected to the server, the two red Raspberry Pi devices in the lower right corner are controlled through the computer screen in the middle, and the wireless local area network of the whole environment is set up through the router on the right. The resource configuration of the server is the same as the one used in the above simulation experiments, while the CPU, memory, and disk capacity of the Raspberry Pi 4B are Broadcom BCM2711 64-bits 1.5 GHz, 8G, and 32G.

Table 4
Storage overhead.

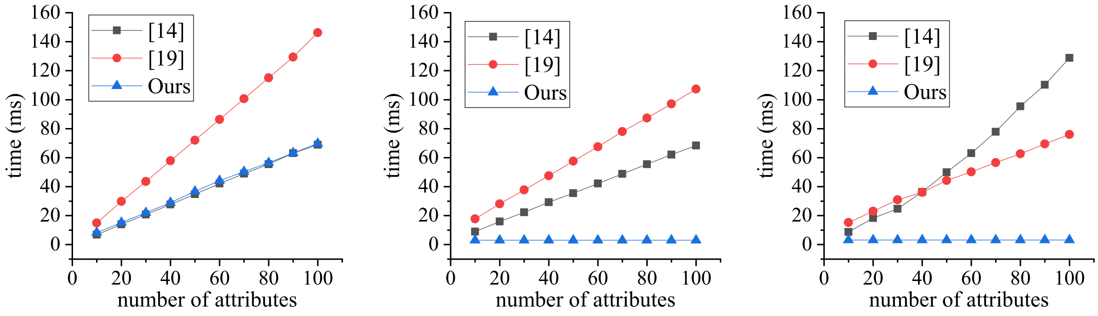
Key Type	[14]	[19]	[34]	Ours
AttKey	$RS : N + \lvert\mathbb{G}\rvert + N_u\lvert\mathbb{G}_T\rvert$	$AA : N_{AA}(\lvert\mathbb{G}\rvert + 2\lvert\mathbb{Z}_p^*\rvert + \lvert\mathbb{G}_T\rvert)$	–	$AS : (1 + N)\lvert\mathbb{Z}_p^*\rvert$
SysKey	$RA : N\lvert\mathbb{Z}_p^*\rvert + N\lvert\mathbb{G}\rvert$	–	–	$RS : 1\lvert\mathbb{Z}_p^*\rvert$
UPK	–	–	$BC : N_u\lvert\mathbb{G}\rvert$	$BC : (N + N_u + N_s)\lvert\mathbb{G}\rvert$
USK	$DU : 1\lvert\mathbb{Z}_p^*\rvert + N_a\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_T\rvert + \lvert UAtS\rvert$	$DU : 1\lvert\mathbb{Z}_p^*\rvert + 1\lvert\mathbb{G}\rvert + N_a\lvert\mathbb{G}\rvert + \lvert UAtS\rvert$	$DU : 1\lvert\mathbb{Z}_p^*\rvert$	$DU : 1\lvert\mathbb{Z}_p^*\rvert + 4\lvert\mathbb{G}\rvert + \lvert UAtS\rvert$

Abbreviations: RS: cloud server (= AS). RA: registration authority (= RS).

Table 5
Computation overhead.

Phases	[14]	[19]	[34]	Ours
Initialization	$RS : NE$	$AA : 2N_{AA}E + N_{AA}E_T$	–	$AS : (N + 1)E$
RS Regist	–	–	–	$RS : 1E + 1H + [1E + 1H]$; $AS : [2E + 1H] + 1E$
DU Regist	$DU : 1E_T + 1P + [1E + 1H]\lvert N_aE$; $RA : [2E + 1H] + Poly + N_aE + 1E_T$	$DU : 1H + [2E + 1H] + 1H + 1E$ $\mid 2H + (N_a')(2P)$ $AA : [2E + 1H] + 3H + 2P + (N_a')E$	$DU : 1E$	$DU : 1H + 1E + [1E + 1H] \mid 2E$; $RS : [2E + 1H] + 1E + BC_c$; $AS : 3E$
Authen	$DU : [3E + 2H]$; $RS : [4E + 2H]$	$DU : [1E + 1H]$; $ES : [2E + 1H]$	$DU : [7E + 3P + 1H]$; $BC : [3E + 1P + 1H] + BC_c$	$DU : [3E + 2H]$; $BC : [4E + 2H] + BC_c$
Acc	$DU : 0$; $RS : tE_{T+tP}$	$DU : N_aE$ $ES : (2N_a + 2)P + 3H$	–	$DU : 5E$; $DUP : 4P$
KA	–	–	–	$DUP : 1P + 1E_T + 1E + [2H + 3E]$; $DU : [2H + 4E] + 1P + 1E_T$

Abbreviations: Poly: operations for constructing a polynomial. t: the threshold value of access. AA: attribute authority. N_{AA}: number of attributes governed by an AA. ES:edge server. N_a': equals to $N_{AA} \cap N_a$.



(a)

(b)

(c)

Fig. 8. Setup and user registration.

Table 6
Simulation setting.

Name	Setting
Operation System	Ubuntu 20.04.4 LTS 64-bit
Memory	15.6 GiB
Processor	11th Gen Intel® Core™ i7-11700K @ 3.60 GHz
Disk Capacity	53.7 GB
Language	Go 1.11.4
Cryptography Library	PBC 0.5.14
Blockchain Platform	Hyperledger Fabric 1.4.2 with Solo Consensus
Docker	Docker 20.10.12
TPS Test Tool	Tape 0.1.2

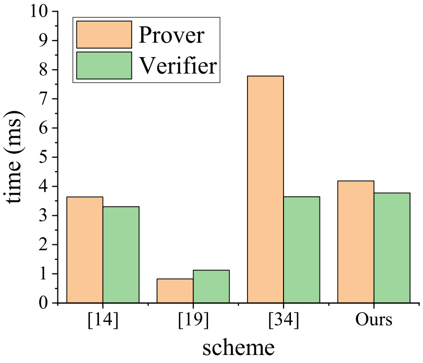


Fig. 9. Authentication.

Table 7
Delay time.

Phase	Dealy (ms)
Registration	13.606
Authentication	38.744
Access and Key Agreement	55.537

In this environment, we tested the delay of three main phases of our scheme: first is Raspberry Pi 1 generating Ψ_{PoK} to register with the server to get the required keys, i.e. as in Fig. 5; second is Raspberry Pi 1 generating Ψ_{PoE} and send it to the server for authentication, i.e. as in Fig. 6; finally is Raspberry Pi 1 generating access token AToK and send it to Raspberry Pi 2 for access and key agreement, i.e. as in Fig. 7. The delay times of the above three phases are summarized in Table 7.

In conclusion, our scheme achieves user registration and authentication by two classical ZKP protocols and access control by attribute-based encryption. The simulation results show that our scheme performs better in setup, user registration, and access control. Besides, the blockchain throughput result indicates that the proposed blockchain-enforced authentication scheme is feasible and efficient.

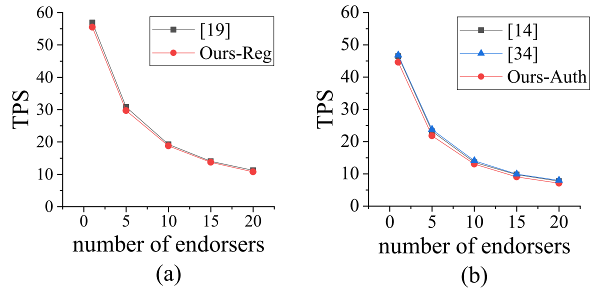


Fig. 10. Throughputs of BC authentication.

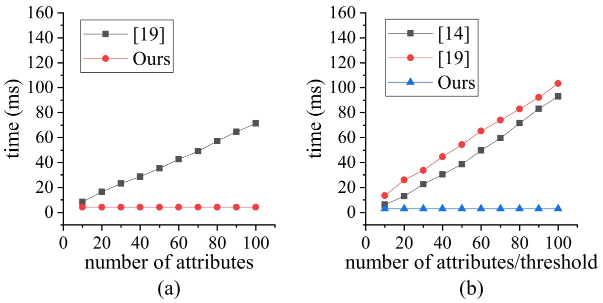


Fig. 11. Access control.

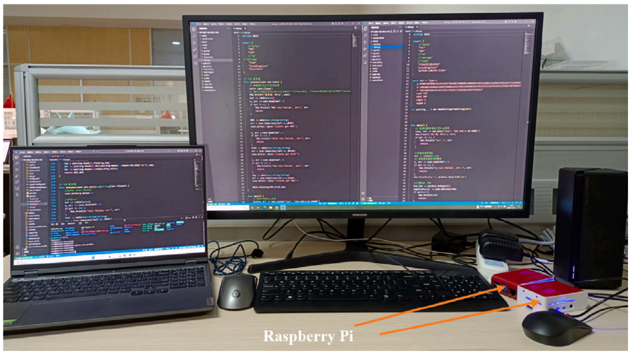


Fig. 12. Test environment.

8. Conclusion

This paper proposed a blockchain-enforced private-preserving authentication and key agreement and access control (BP-AKAA) scheme for IIoT D2D communication. Specifically, blockchain technology and non-interactive zero-knowledge proof are utilized to ensure cross-domain and privacy protection, respectively. Besides, we designed a comprehensive and secure communication framework for industrial IoT applications to realize authentication, key agreement, and attribute-based access control. Theoretical and simulation analysis indicates that the BP-AKAA scheme has predominant performances in security and computational overheads and is suitable for resource-limited IIoT devices.

As this paper solely focused on D2D communications, we plan to exploit more potentials of blockchain in other communication types in IIoT, such as cross-domain S2D.

CRediT authorship contribution statement

Suhui Liu: Conceptualization, Scheme design, Formal security analysis and performance analysis, Writing – original drat. **Liquan Chen:** Supervision, Writing – review & editing. **Hongtao Yu:** Simulation, Performance analysis. **Shang Gao:** Scheme design, Table making, Formula review. **Huiyu Fang:** Writing – review, Figure drawing.

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

No data was used for the research described in the article.

Acknowledgments

The authors thank the editors and the anonymous reviewers for their valuable and enriching comments and suggestions. This work is supported by the National Key R&D Program of China (2020YFE0200600) and the National Natural Science Foundation of China (No. U22B2026).

References

[1] Hazra A, Adhikari M, Amgoth T, Srirama SN. A comprehensive survey on interoperability for IIoT: Taxonomy, standards, and future directions. ACM Comput Surv 2021;55(1):1–35.

[2] Ling Z, Hao ZJ. Intrusion detection using normalized mutual information feature selection and parallel quantum genetic algorithm. Int J Semant Web Inform Syst (IJSWIS) 2022;18(1):1–24.

[3] Ling Z, Hao ZJ. An intrusion detection system based on normalized mutual information antibodies feature selection and adaptive quantum artificial immune system. Int J Semant Web Inform Syst (IJSWIS) 2022;18(1):1–25.

[4] Figueroa-Lorenzo S, Añorga J, Arrizabalaga S. A survey of IIoT protocols: A measure of vulnerability risk analysis based on CVSS. ACM Comput Surv 2020;53(2):1–53.

[5] Jiang T, Zhang J, Tang P, Tian L, Zheng Y, Dou J, Asplund H, Raschkowski L, D'Errico R, Jämsä T. 3GPP standardized 5G channel model for IIoT scenarios: A survey. IEEE Internet Things J 2021;8(11):8799–815.

[6] Chander S, Vijaya P, Dhyani P. A parallel fractional lion algorithm for data clustering based on MapReduce cluster framework. Int J Semant Web Inform Syst (IJSWIS) 2022;18(1):1–25.

[7] Bellare M, Goldwasser S. New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In: Conference on the theory and application of cryptology. Springer; 1989, p. 194–211.

[8] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Bus Rev 2008;21260.

[9] Zhou L, Li X, Yeh K-H, Su C, Chiu W. Lightweight IoT-based authentication scheme in cloud computing circumstance. Future Gener Comput Syst 2019;91:244–51.

[10] Lopes APG, Gondim PR. Group authentication protocol based on aggregated signatures for D2D communication. Comput Netw 2020;178:107192.

[11] Xiong H, Wu Y, Jin C, Kumari S. Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT. IEEE Internet Things J 2020;7(12):11713–24.

[12] Kumar A, Abhishek K, Liu X, Haldorai A. An efficient privacy-preserving id centric authentication in iot based cloud servers for sustainable smart cities. Wirel Pers Commun 2021;117(4):3229–53.

[13] Liu Y, Wang H, Li T, Li P, Ling J. Attribute-based handshake protocol for mobile healthcare social networks. Future Gener Comput Syst 2018;86:873–80.

[14] Ibrahim MH, Kumari S, Das AK, Odelu V. Attribute-based authentication on the cloud for thin clients. J Supercomput 2018;74(11):5813–45.

[15] Zhang Q, Gan Y, Liu L, Wang X, Luo X, Li Y. An authenticated asymmetric group key agreement based on attribute encryption. J Netw Comput Appl 2018;123:1–10.

[16] Lin H-Y, Ting P-Y, Wu H-R. An attribute-based mutual authentication scheme with time-bounded keys. In: Proceedings of the 3rd International conference on telecommunications and communication engineering. 2019, p. 75–9.

[17] Sun J, Su Y, Qin J, Hu J, Ma J. Outsourced decentralized multi-authority attribute based signature and its application in IoT. IEEE Trans Cloud Comput 2019;9(3):1195–209.

[18] Sucasas V, Mantas G, Papaioannou M, Rodriguez J. Attribute-based pseudonymity for privacy-preserving authentication in cloud services. IEEE Trans Cloud Comput 2021.

[19] Zhang Z, Zhou S. A decentralized strongly secure attribute-based encryption and authentication scheme for distributed internet of mobile things. Comput Netw 2021;201:108553.

[20] Karati A, Islam SH, Karuppiah M. Provably secure and lightweight certificateless signature scheme for IIoT environments. IEEE Trans Ind Inf 2018;14(8):3701–11.

[21] Esfahani A, Mantas G, Matischek R, Saghezchi FB, Rodriguez J, Bicaku A, Maksuti S, Tauber MG, Schmittner C, Bastos J. A lightweight authentication mechanism for M2M communications in industrial IoT environment. IEEE Internet Things J 2017;6(1):288–96.

[22] Gupta DS, Islam SH, Obaidat MS, Vijayakumar P, Kumar N, Park Y. A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments. IEEE Syst J 2020;15(2):1732–41.

[23] Abdi Nasib Far H, Bayat M, Kumar Das A, Fotouhi M, Pournaghi SM, Doostari M-A. LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. Wirel Netw 2021;27(2):1389–412.

[24] Xu D, Yu K, Ritcey JA. Cross-layer device authentication with quantum encryption for 5G enabled IIoT in industry 4.0. IEEE Trans Ind Inf 2021.

[25] Martín-Fernández F, Caballero-Gil P, Caballero-Gil C. Authentication based on non-interactive zero-knowledge proofs for the internet of things. Sensors 2016;16(1):75.

[26] Walshe M, Epiphaniou G, Al-Khateeb H, Hammoudeh M, Katos V, Dehghantanha A. Non-interactive zero knowledge proofs for the authentication of iot devices in reduced connectivity environments. Ad Hoc Netw 2019;95:101988.

[27] Rasheed AA, Mahapatra RN, Hamza-Lup FG. Adaptive group-based zero knowledge proof-authentication protocol in vehicular ad hoc networks. IEEE Trans Intell Transp Syst 2019;21(2):867–81.

[28] Soewito B, Marcellinus Y. IoT security system with modified Zero knowledge proof algorithm for authentication. Egyptian Inform. J. 2021;22(3):269–76.

[29] Gaba GS, Hedabou M, Kumar P, Braeken A, Liyanage M, Alazab M. Zero knowledge proofs based authenticated key agreement protocol for sustainable healthcare. Sustainable Cities Soc 2022;80:103766.

[30] Cui Z, Fei X, Zhang S, Cai X, Cao Y, Zhang W, Chen J. A hybrid blockchain-based identity authentication scheme for multi-WSN. IEEE Trans Serv Comput 2020;13(2):241–51.

[31] Wang L, Tian Y, Zhang D. Toward cross-domain dynamic accumulator authentication based on blockchain in internet of things. IEEE Trans Ind Inf 2021;18(4):2858–67.

[32] Yang Y, He D, Vijayakumar P, Gupta BB, Xie Q. An efficient identity-based aggregate signcryption scheme with blockchain for IoT-enabled maritime transportation system. IEEE Trans Green Commun Netw 2022.

[33] Leng J, Ye S, Zhou M, Zhao JL, Liu Q, Guo W, Cao W, Fu L. Blockchain-secured smart manufacturing in industry 4.0: A survey. IEEE Trans Syst Man Cybern Syst 2020;51(1):237–52.

[34] Andola N, Yadav VK, Venkatesan S, Verma S, et al. SpyChain: A lightweight blockchain for authentication and anonymous authorization in IoD. Wirel Pers Commun 2021;119(1):343–62.

[35] Gabay D, Akkaya K, Cebe M. Privacy-preserving authentication scheme for connected electric vehicles using blockchain and zero knowledge proofs. IEEE Trans Veh Technol 2020;69(6):5760–72.

[36] Feng T, Yang P, Liu C, Fang J, Ma R. Blockchain data privacy protection and sharing scheme based on Zero-Knowledge proof. Wirel Commun Mob Comput 2022;2022.

[37] Kumar R, Kumar P, Tripathi R, Gupta GP, Islam AN, Shorfuzzaman M. Permissioned blockchain and deep-learning for secure and efficient data sharing in industrial healthcare systems. IEEE Trans Ind Inf 2022.

[38] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems. SIAM J Comput 1989;18(1):186–208.

[39] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the theory and application of cryptographic techniques. Springer; 1986, p. 186–94.

[40] Goldreich O, Micali S, Wigderson A. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J ACM 1991;38(3):690–728.

[41] Dabbagh M, Choo K-KR, Beheshti A, Tahir M, Safa NS. A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities. Comput Secur 2021;100:102078.

[42] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: International workshop on public key cryptography. Springer; 2011, p. 53–70.

[43] Dolev D, Yao A. On the security of public key protocols. IEEE Trans Inform Theory 1983;29(2):198–208.

[44] Krawczyk H. Cryptographic extraction and key derivation: The HKDF scheme. In: Annual cryptology conference. Springer; 2010, p. 631–48.

Suhui Liu received her B.S. degree and M.S. degree in f Science, Qufu Normal University, Shandong, China in 2018 and 2021 respectively. She is currently pursuing a Ph.D. in the School of Cyber Science and Engineering, Southeast University, Jiangsu, China. Her main research interests include cloud-assisted IoT data security, functional cryptography, and blockchain technology.

Liquan Chen received the Ph.D. degree from Southeast University, China in 2005. He worked as a postdoc in Southeast University from 2005 to 2007, and an associate professor at Southeast University from 2008 to 2018. He worked as visiting scholar in National University of Singapore, Singapore from 2011 to 2012, and became a member of IEEE since 2010. He now is a professor in School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include information security, cryptography and network security protocol, etc.

Hongtao Yu received his B.E. degree in Information Security, Guangdong University of Technology, China in 2021. He is currently pursuing a M.E. degree in the School of Cyber Science and Engineering, Southeast University, Jiangsu, China. His main research interests include information security, blockchain and authentication.

Dr. Shang Gao is currently a research assistant professor in the Department of Computing in the Hong Kong Polytechnic University. He received his B.S. degree from Hangzhou Dianzi University, China, in 2010, M.E. degree from Southeast University, China, in 2014, and Ph.D. degree from the Hong Kong Polytechnic University, Hong Kong, in 2019. After graduation, he worked in Microsoft China for one year. His research interests include information security, network security, software-defined networks, blockchain security, and applied cryptography. His work has been published in several top-tier conferences and journals, including CCS, INFOCOM, TDSC, ToN, etc.

Huiyu Fang received the B.S. degree in information engineering and M.S. degree in software engineering from Southeast University, Nanjing, China, in 2014 and 2018, respectively. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include privacy preserving, information security, and IoT security.