Journal of Industrial Information Integration 48 (2025) 100946

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.sciencedirect.com/journal/journal-of-industrial-information-integration

Full Length Article

HCDA: A hidden cross-domain authentication protocol for embodied intelligence in smart manufacturing

Huaiyao Yang [a], Xiangwei Meng [a,*], Jiale Liang [b], Yanrong Zhang [b], Keqin Li [c]

[a] College of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China
[b] School of Computer and Information Engineering, Harbin University of Commerce, Harbin, 150000, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Authentication
Cross-domain
Embodied intelligence
Smart manufacturing

ABSTRACT

In smart manufacturing utilizing embodied intelligent robots, frequent cross-domain data transmissions introduce significant challenges on key management. While existing authentication protocols for cross-domain smart manufacturing offer certain advantages in terms of key storage security, their complex network structures and the necessity for repeated session key updates introduce risks related to master key loss, as well as elevated computation cost and communication overhead. To overcome these challenges, this paper proposes a hidden cross-domain authentication (HCDA) protocol for embodied intelligence in smart manufacturing. The domain servers in intelligent manufacturing, functioning as consensus nodes, collaboratively establish a blockchain consortium to securely record public keys and public authentication parameters. Besides, the HCDA protocol based on encryption migration method to reduce the authentication delay of embodied intelligent robots. Specifically, embodied intelligent robots perform symmetric-key encryption/decryption operations and one-way hash functions for authentication request, while domain servers execute the Elliptic Curve Cryptography (ECC) algorithm to generate session key. The security of HCDA protocol is proved by informal analysis. Finally, the simulation results for computation cost and communication overhead demonstrate that the HCDA protocol exhibits significant performance advantages compared with the related protocols.

1. Introduction

The smart manufacturing integrates physical entities into digital networks by leveraging sensors, communication technologies, and network protocols [1]. This integration enables comprehensive environmental perception, seamless inter-device communication, information sharing, and coordinated decision-making [2–4]. However, as the manufacturing industry increasingly demands efficiency, quality assurance, personal safety, and intelligence-driven decision-making capabilities persist, traditional smart manufacturing systems exhibit inherent limitations, particularly in real-time performance, reliability, and adaptability to complex environments. In response to this global trend towards Industry 5.0 development initiatives arises the embodied intelligence empowering smart manufacturing, which serves as an extension of smart manufacturing specifically tailored for industrial applications [5]. This paradigm not only addresses critical technical challenges in production safety and efficiency but also fosters a novel human-machine collaborative co-creation production model [6].

As embodied intelligence continues to advance in empowering smart manufacturing, it increasingly necessitates cross-domain collaborative frameworks to orchestrate comprehensive production processes. Although advanced communication networks enable seamless interoperability among embodied intelligent robots across domains, establishing cross-domain collaboration within smart manufacturing still faces persistent challenges, including security vulnerabilities, efficiency bottlenecks, and system compatibility constraints [7,8]. To mitigate security threats such as embodied intelligent robots spoofing and hijacking while ensuring data integrity alongside reliable communications, smart manufacturing systems must implement robust authentication protocols that leverage encryption techniques along with digital signature technologies. This measure is crucial for verifying the legitimacy of embedded intelligent robots and domain servers, while simultaneously establishing secure and dynamic cross-domain communication channels [9,10]. The security of cross-domain authentication protocols emerges as a critical challenge for intelligent manufacturing, primarily due to the inherent disparities between cross-domain frameworks and

* Corresponding author.
E-mail address: xiangwei_meng@126.com (X. Meng).

https://doi.org/10.1016/j.jii.2025.100946

Available online 4 September 2025
2452-414X/© 2025 Elsevier Inc. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

authentication protocols, compounded by the fragility of public communication channels to various attacker threats. For example, heterogeneous communication protocols and fragmented security standards among different domains create interoperability barriers. Furthermore, authentication parameters transmitted over public channels are susceptible to brute-force attacks and cryptanalysis by attackers [11,12].

Traditional authentication protocols primarily depend on the well-established Public Key Infrastructure (PKI) mechanism, wherein a Certificate Authority (CA) issues digital certificates that bind entities' identity information to their public keys. PKI is susceptible to Distributed Denial-of-Service (DDoS) attacks, while its centralized architecture introduces a critical single point of failure risk [13,14]. Additionally, as the number of system entities increases, the operational costs of certificate storage and management may exhibit a nonlinear growth trend. In recent years, Identity-Based Cryptography (IBC) and multi-factor authentication protocols have emerged within smart manufacturing. These protocols are typically implemented within closed domains where trusted administrators distribute cryptographic keys to devices within those domains. Nevertheless, the absence of uniformly enforced cross-domain authentication and key agreement mechanisms severely impedes the establishment of secure communication relationships among cross-domain entities [15,16].

The advent of blockchain technology presents a novel solution to the aforementioned problems. As a decentralized and tamper-resistant distributed ledger, it eliminates single points of failure, facilitates data security synchronization across peer nodes, and demonstrates robust scalability [17–19]. By capitalizing on these inherent features, numerous security frameworks integrating blockchain technology with smart manufacturing have been developed to enable cross-domain trust establishment and secure data transmission. Specifically, the administrator distributes authorization certificates on the blockchain and facilitate rapid authentication and key agreement through predefined protocols. However, most existing blockchain-based cross-domain authentication protocols continue to face new challenges. Firstly, the public authentication parameters stored in blockchain ledgers may expose user identities or device behavior patterns through vulnerability analysis. For instance, the attackers can infer sensitive information by correlating embodied intelligent robot identity with transmitted data during the authentication process. Furthermore, within their respective blockchain networks, different domains utilize distinct authentication protocols, consensus mechanisms, and smart contracts, thereby engendering additional compatibility costs and security risks during cross-domain channel establishment.

To address the above-mentioned challenges, this study introduces a Hidden Cross-Domain Authentication (HCDA) protocol tailored for embodied intelligence in smart manufacturing. By leveraging blockchain technology to enable the distributed management of authentication parameters-such as public keys-a secure cross-domain authentication protocol is established for large-scale embodied intelligent robots. The main contributions of this work are as follows:

1) We introduce a blockchain-based embodied intelligence network from the perspective of cross-domain. Considering the incompatibility of authentication protocols in different domains, we use domain servers as consensus nodes to store public keys and public authentication parameters.
2) In order to address the communication trust issue among embodied intelligent robots from different domains, we propose a hidden cross-domain authentication protocol in which the risk of data leakage during authentication process is reduced by reducing the authentication parameters recorded in the blockchain ledger.
3) We propose a key agreement mechanism to establish a session key among embodied intelligent robots and the domain servers. The established session key enables embodied intelligent robots to securely exchange data across domains for collaborative operations. Furthermore, when an embodied intelligent robot leaves the domain, the session key is revoked to ensure forward and backward secrecy of the encrypted data.
4) Finally, we conduct a comprehensive security analysis to demonstrate that HCDA ensures identity authenticity and the security of authentication parameters. Additionally, we perform simulation-based experiments to evaluate its advantages in computation cost and communication overhead compared to related protocols.

Section 2 reviews the related works. Section 3 introduces the network model and threat model. Section 4 presents a detailed process of the proposed HCDA protocol. Section 5 analyzes the security features of the proposed protocol. Section 6 provides a performance comparison of the proposed protocol and related protocol. Finally, Section 7 summarizes the paper.

2. Related work

Currently, a multitude of authentication protocols have been developed to secure smart manufacturing environments. These protocols can be systematically classified into two distinct categories based on their authentication scope and trust domain configurations: intra-domain authentication and cross-domain authentication. Intra-domain authentication protocol focuses on verifying device identities within a single trusted network or administrative domain, typically employing methods like symmetric/asymmetric cryptography, challenge-response mechanisms, or biometric verification. Cross-domain authentication protocol addresses identity verification across heterogeneous networks or organizational boundaries, often leveraging decentralized technologies (e.g., blockchain-based authentication frameworks) and standardized trust frameworks to ensure interoperability and security.

2.1. Intra-domain authentication protocol

In 2020, Vinoth et al. [20] proposed a lightweight secure multi-factor authentication key agreement protocol for smart manufacturing based on secret sharing techniques and the CRT. However, their protocol is vulnerable to sensor node capture attack, DDoS attack, replay attack, and desynchronization attack. Moreover, the direct connection between users and sensors generates excessive power consumption, rendering it unsuitable for smart manufacturing applications and low-capacity devices. In 2021, Rangwani et al. [21] proposed a robust privacy-preserving authentication protocol for smart manufacturing. The protocol leverages Elliptic Curve Cryptography (ECC) and one-way hash functions to achieve three-factor strong identity verification by integrating user identity, password, and biometric features.

In 2022, Hajian et al. [22] proposed a secure anonymity protocol for Device-to-Device (D2D) mutual authentication and key agreement for the Internet of things. The protocol achieves device anonymity and tracking resistance through temporary identity and hash chaining techniques, and solves the security challenge of D2D communication where the authentication server is not involved in the authentication and key agreement process. Rafique et al. [23] proposed an efficient certificateless multi-factor authentication and key agreement protocol for smart manufacturing. The protocol achieves efficient authentication and key agreement for resource-constrained devices through the use of symmetric encryption, XOR operations, and hash functions. Tarveer et al. [24] proposed a resource-efficient authentication protocol that leverages lightweight cryptography primitives and hash functions, integrating fuzzy extractors to achieve user biometric binding while supporting mutual authentication and secure session key establishment.

In 2023, Xu et al. [25] proposed an ECC-based three-factor anonymous authentication and key agreement protocol for smart manufacturing environments. This protocol employs pseudonym mechanisms and fuzzy biometric extraction techniques, encapsulating registered users' anonymous identities and authentication parameters

within control nodes to enable dynamic user enrollment while ensuring identity protection. Tanveer et al. [26] proposed a lightweight cryptography-based authentication protocol. The framework establishes session keys through local authentication and gateway-assisted mutual authentication mechanisms, which combined with Chebyshev chaotic mapping and efficient encryption with ASCON, significantly reduces the computation and communication overheads while ensuring security. Deebak et al. [27] proposed a blockchain-based trust-aware seamless authentication protocol for large-scale smart manufacturing. This protocol addresses privacy protection and single point of failure issues through distributed ledger technology and lightweight cryptographic techniques, while optimizing device identity management efficiency by integrating dynamic data traffic patterns with smart contracts.

In 2024, Dhar et al. [28] proposed an innovative scheme integrating blockchain and quantum cryptography. This approach employs Quantum Key Distribution (QKD) technology to generate secure cryptographic keys, utilizes blockchain to store data hash values for ensuring integrity, and incorporates Zero-Knowledge Proofs (ZKP) along with lightweight encryption algorithms to enhance data privacy protection and anonymity.

2.2. Cross-domain authentication protocol

In 2020, Shen et al. [30] proposed a blockchain-assisted secure device authentication mechanism for cross-domain smart manufacturing, in which the identity management mechanism is used to achieve anonymity in device authentication. In 2021, Wang et al. [31] proposed a handover authentication and key agreement for cross-domain intelligent telehealth system. During the authentication process, edge nodes assist in reducing users' computation cost. Singh et al. [32] proposed a cross-domain secure data sharing framework for smart manufacturing applications based on blockchain technology. This framework employs a hybrid authentication mechanism for secure authentication and key agreement, and ensures data integrity and authenticity through the combined use of smart contracts and multi-layer signatures.

In 2022, Zhang et al. [33] proposed a blockchain-based cross-domain authentication protocol for devices of smart manufacturing. The protocol introduces a hardware-assisted multi-factor key derivation method via Physical Unclonable Function (PUF) and a dynamic accumulative approach on-chain for cross-domain trust establishment, significantly reducing on-chain storage overhead while ensuring security. Tong et al. [34] proposed a consortium blockchain-based Comprehensive Cross-domain Authentication Protocol (CCAP) for IoT systems. This scheme enables seamless interoperability between heterogeneous authentication domains through pseudonym mechanisms and threshold cryptography, achieving device privacy protection with lawful traceability while reducing administrative overhead via decentralized architecture. Cui et al. [35] proposed an blockchain-based anonymous cross-domain authentication protocol for smart manufacturing. Leveraging blockchain's decentralized and immutable properties, this scheme employs dynamic accumulator techniques to compress authentication protocols and integrates smart contracts for rapid verification, thereby eliminating the delay issues caused by frequent on-chain operations in traditional blockchain-based solutions.

In 2023, Khashan et al. [36] proposed an efficient hybrid centralized and blockchain-based authentication architecture for heterogeneous internet of things systems. This architecture provides centralized authentication for associated internet of things devices through deployed edge servers and establishes a blockchain network composed of these edge servers to ensure efficient decentralized authentication and verification among devices across diverse and heterogeneous internet of things systems.

Most of these authentication protocols focus on the cross-domain authentication scenario of a single device, which is difficult to adapt to the blockchain-based embodied intelligence in smart manufacturing [37]. In addition, they ignore the hidden problem of authentication parameters being recorded in the blockchain ledger, and certificates stored in domain servers are also difficult to avoid leakage [29].

3. System model

3.1. Network model

Leveraging highly advanced communication networks, the interconnection of embodied intelligence expands from single-domain manufacturing applications to interconnected multi-domain ecosystems [38,39]. As illustrated in Fig. 1, a simplified cross-domain smart manufacturing scenario depicts two independently managed manufacturing domains, which may be operated by different business partners. Embodied intelligent robots are distributed across production lines in each domain, capable of perceiving environmental parameters and product manufacturing states in real time, and dynamically adjusting production behaviors based on data analysis to optimize efficiency and ensure quality. In this network model, the entities within domain *A* (including domain server and embodied intelligent robots) collaboratively engage with entities in domain *B* to execute distinct yet interrelated manufacturing processes for the same product. The embodied intelligent robots require high-frequency cross-domain communication, necessitating a robust cross-domain authentication protocol to balance the requirements of energy consumption, identity verification, and communication security.

Consortium blockchain (*CB*): It is deployed across all domain servers. As a decentralized and tamper-resistant distributed ledger, *CB* encapsulates public keys, authentication parameters, and associated operations into traceable transactions that are synchronized among consensus nodes. All nodes can collaboratively detect malicious activities through cross-verification of *CB* transactions. Additionally, smart contracts enable cross-domain parties to jointly compute digital signatures and directly store authorization results on *CB*, thereby eliminating reliance on third-party trusted entity and significantly enhancing the security and efficiency of cross-domain collaboration mechanisms.

Domain server (*DS*): As a fully trusted management node within the domain, *DS* is equipped with a high-performance processor and a massive storage array, delivering data aggregation and real-time analytics. The *DS* securely stores domain-specific management rules, device identity information, and authentication parameters, enabling automated identity management for intra-domain devices. For cross-domain operations, *DS* functions as a consensus node in the consortium blockchain, establishing trusted interoperability with other domains. It not only provides cross-domain information verification for intra-domain embodied intelligent robot but also actively participates in smart con-

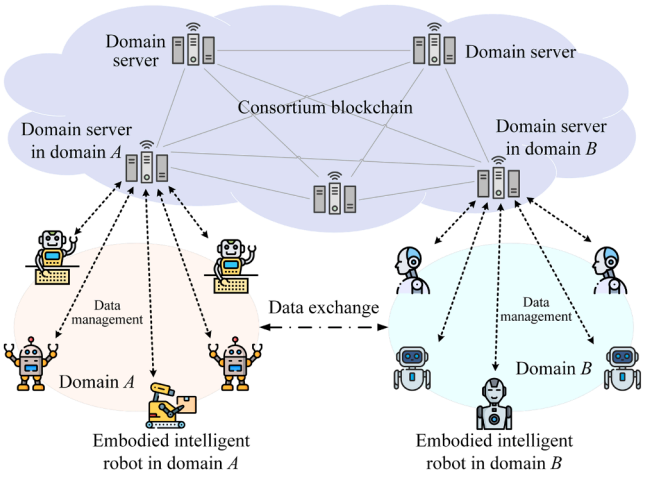

Fig. 1. Cross-domain network model of embodied intelligence in smart manufacturing.

tract execution and distributed ledger updates on the blockchain.

Embodied intelligent robot (*D*): It is deployed within a specific industrial domain and performs functions such as production task execution and real-time data collection. The data collected by *D* is transmitted to *DS*. In this architecture, *D* possesses the computational capacity to participate in cryptographic operations (e.g., elliptic curve encryption) and holds a unique legal identity. However, its trust authority is confined to its own domain. Consequently, cross-domain access requires proxy authentication by *DS*.

3.2. Threat model

Attackers operating under the Dolev-Yao (DY) model can exploit fully controllable public channels to intercept sensitive data, impersonate legitimate entities (e.g., *DS* and *D*), and launch attacks such as eavesdropping, replay attacks, and more. These actions may result in authentication failures or key leakage. Additionally, attackers can disrupt legitimate authentication processes by coordinating multiple *DS* or disguising themselves as trusted *D* to penetrate across domain, thereby facilitating collusion attacks. The key generation center (*KGC*), responsible for generating master key and publishing system parameters, may cause trust authority failures if it malfunctions.

4. Proposed HCDA protocol

In this section, we detail the HCDA protocol proposed for embodied intelligence in smart manufacturing. Table 1 provides a guide to the symbols used, and Fig. 2 illustrates the main steps of the HCDA protocol. Firstly, *KGC* distributes public parameters to all system entities. Then, *DS* registers its identity and public-private key pair on the consortium blockchain. After *DS* configuration, each *D* registers its identity with its designated *DS*. Due to disparities in cross-domain trust mechanisms, *D* can only perform cross-domain authentication via *DS* proxies. Once the session key is generated, *D* from different domains can leverage symmetric encryption to enable efficient and secure communication.

4.1. System initialization phase

KGC selects a non-singular elliptic curve $E_p : y^2 = x^3 + ax + b(modp)$ over the finite field F_p, where a, $b \in F_p$, $p > 3$, and $4a^3 + 27b^2(modp) \neq 0$. It then defined a cyclic group G of prime order q on E_p, where $P \in G$ is the generator point of the elliptic curve. Next, *KGC* selects two secure one-way hash functions $H_1 : \{0,1\}^* \rightarrow Z_q^*$ and $H_2 : G \rightarrow Z_q^*$. Finally, *KGC* publishes the system parameters: $params = <G, q, P, H_1, H_2>$.

Table 1
The symbols and definitions.

Symbols	Definitions
DA	Embodied intelligent robot deployed in domain *A*
DB	Embodied intelligent robot deployed in domain *B*
DSA	Domain server deployed in domain *A*
DSB	Domain server deployed in domain *B*
CB	Consortium blockchain
KGC	Key generation center
id	Unique identity of entity
sk	Private key of entity
pk	Public key of entity
pid	Pseudonym of entity
tk	Temporary key of entity
tid	Hash signature
k	Session key
$E_{tk}(.)$ /$D_{tk}(.)$	Symmetric encryption/decryption using temporary key *tk*
$H(.)$	Secure one-way hash function
$\oplus$	Bitwise XOR operation

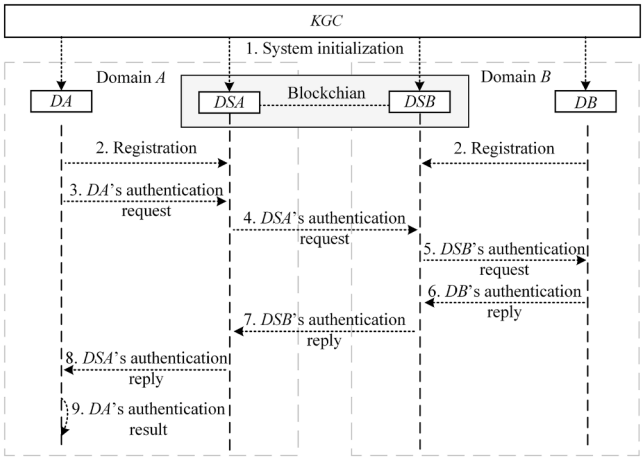

Fig. 2. HCDA protocol process overview.

4.2. Registration phase

At this phase, *DS* has completed registration on the blockchain, while each *D* registers with its corresponding *DS*. The details are as follows:

1) ***DS*** registration

Let *DSA* and *DSB* denote the servers deployed in domains *A* and *B*, respectively. *KGC* selects the unique identities (id_{DSA}, id_{DSB}) and the random numbers (x_{DSA}, x_{DSB}) to generate the private keys $sk_{DSA} = H_1(id_{DSA}, x_{DSA})$, $sk_{DSB} = H_1(id_{DSB}, x_{DSB})$, and computes the corresponding public keys $pk_{DSA} = sk_{DSA} \cdot P$, $pk_{DSB} = sk_{DSB} \cdot P$. Then, *KGC* sends the unique identities (id_{DSA}, id_{DSB}) and their respective private keys (sk_{DSA}, sk_{DSB}) to *DSA* and *DSB* via a secure channel. Finally, *KGC* uploads the public keys (pk_{DSA}, pk_{DSB}) on the blockchain ledger, where the recorded data is publicly accessible. The registration process of *DSA* is illustrated in Fig. 3.

2) ***D*** registration

Let *DA* and *DB* denote the embodied intelligent robots deployed in domains *A* and *B*, respectively. *DA* selects its unique identities id_{DA} and submits a registration request to its domain server *DSA* via a secure channel. Upon receiving the request, *DSA* checks whether the hash value $H_1(id_{DA})$ already exists in its authentication database. If the hash exists, the request is rejected; otherwise, *DSA* generates a pseudonym $pid_{DA} = H_1(sk_{DSA}) \oplus id_{DA}$ and transmits pid_{DA} back to *DA* via a secure channel. Subsequently, D_i computes $H_1(sk_{DSA}) = pid_{DA} \oplus id_{DA}$ and stores $H_1(sk_{DSA})$ alongside id_{DA} in secure memory. The process of *DA* is registered with *DSA* as shown in Fig. 4. The registration process for *DB* is identical to that of *DA*.

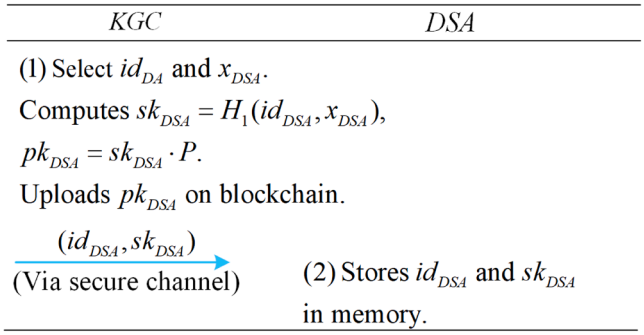

Fig. 3. Domain server registration process of HCDA protocol.

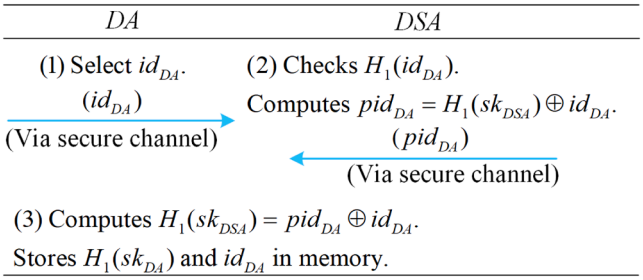

Fig. 4. Embodied intelligent robot registration process of HCDA protocol.

4.3. Authentication and key agreement phase

If *DA* in domain *A* intends to initiate a cross-domain session with *DB* in domain *B*, authentication and key establishment must be performed through their respective domain servers *DSA* and *DSB*. The detailed authentication process between *DA* and *DB* is shown in Fig. 5, with the critical steps as follows:

1) ***DA***'s authentication request

DA generates a fresh timestamp t_{DA} and selects a random number r_{DA}. To ensure confidentiality and integrity of r_{DA}, *DA* computes a temporary key $tk_{DA} = H_1(id_{DA} \oplus r_{DA})$ and a parameter $X_{DA} = pid_{DA} \oplus r_{DA}$. Next, *DA* encrypts the tuple (id_{DA}, r_{DA}) using the temporary key tk_{DA} to generate $Y_{DA} = E_{tk_{DA}}(id_{DA}, r_{DA})$. Simultaneously, *DA* generates a hash signature $tid_{DA} = H_1(id_{DA}, r_{DA}, t_{DA})$. Finally, *DA* sends the authentication request with message tuple $(X_{DA}, Y_{DA}, tid_{DA}, t_{DA})$ to its *DSA*.

2) ***DSA***'s authentication request

Upon receiving the request at time t^*_{DA}, *DSA* first checks the timestamp freshness $|t^*_{DA} - t_{DA}| \le \Delta T$. If the verification fails, *DSA* rejects the request with a timeout error. Next, *DSA* computes a temporary key $tk_{DA} = H_1(X_{DA} \oplus H_1(sk_{DSA}))$ and decrypts the ciphertext $(id_{DA}, r_{DA}) = D_{tk_{DA}}(Y_{DA})$. Using the parameters $(id_{DA}, r_{DA}, t_{DA})$, *DSA* verifies the integrity by a hash signature $tid^*_{DA} = H_1(id_{DA}, r_{DA}, t_{DA})$, and checks $tid^*_{DA} \stackrel{?}{=} tid_{DA}$. If the signature does not match, the request is rejected. After that, *DSA* selects a random number r_{DSA}, generates a fresh timestamp t_{DSA}, and computes the parameters $N_{DSA} = H_1(r_{DSA} \cdot t_{DSA}) \cdot sk_{DSA} \cdot pk_{DSB}$, $X_{DSA} = N_{DSA} + t_{DSA} \cdot sk_{DSA} \cdot pk_{DSB}$. Afterwards, *DSA* computes a hash signature $tid_{DSA} = H_1(r_{DSA}, r_{DA}, t_{DSA})$ and a temporary key $tk_{DSA} = H_2(N_{DSA})$ before encrypting secret parameters $Y_{DSA} = E_{tk_{DSA}}(r_{DSA}, r_{DA})$. At last, *DSA* forwards $(X_{DSA}, Y_{DSA}, tid_{DSA}, t_{DSA})$ to *DSB*.

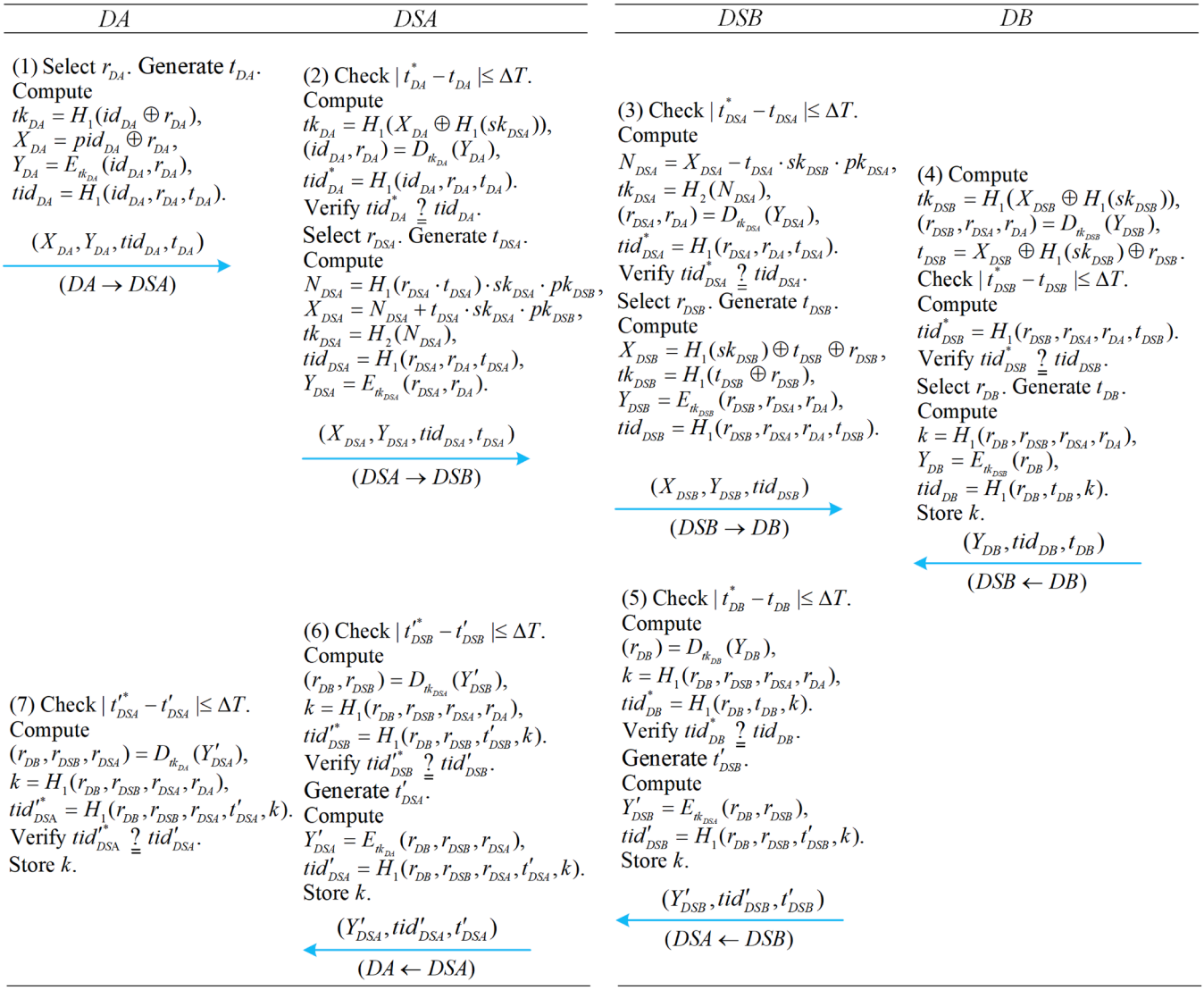

Fig. 5. Authentication and key agreement phase of HCDA protocol.

3) ***DSB***'s authentication request

Upon receiving the message from *DSA*, *DSB* first checks the timestamp t_{DSA} of the message by the condition $|t^*_{DSA} - t_{DSA}| \leq \Delta T$, where the timestamp t^*_{DSA} is the local timestamp at *DSB* when the message is received. Then, *DSB* computes a parameter $N_{DSA} = X_{DSA} - t_{DSA} \cdot sk_{DSB} \cdot pk_{DSA}$, a temporary key $tk_{DSA} = H_2(N_{DSA})$, and decrypts $(r_{DSA}, r_{DA}) = D_{tk_{DSA}}(Y_{DSA})$. Then, *DSB* performs verification on the hash signature $tid^*_{DSA} \stackrel{?}{=} tid_{DSA}$, where $tid^*_{DSA} = H_1(r_{DSA}, r_{DA}, t_{DSA})$. If the integrity verification succeeds, *DSB* also selects a random number r_{DSB}, generates a fresh timestamp t_{DSB}, computes the parameters $X_{DSB} = H_1(sk_{DSB}) \oplus t_{DSB} \oplus r_{DSB}$, $tk_{DSB} = H_1(t_{DSB} \oplus r_{DSB})$, and uses a temporary key tk_{DSB} to encrypt $Y_{DSB} = E_{tk_{DSB}}(r_{DSB}, r_{DSA}, r_{DA})$. Afterwards, *DSB* computes a hash signature $tid_{DSB} = H_1(r_{DSB}, r_{DSA}, r_{DA}, t_{DSB})$ and transmits the message tuple (X_{DSB}, Y_{DSB}, tid_{DSB}) to *DB*.

4) ***DB***'s authentication reply

Upon receiving the message from *DSB*, *DB* first computes a temporary key $tk_{DSB} = H_1(X_{DSB} \oplus H_1(sk_{DSB}))$ and decrypts the ciphertext Y_{DSB} to recover the tuple $(r_{DSB}, r_{DSA}, r_{DA})$ using the temporary key tk_{DSB}. It then computes a timestamp $t_{DSB} = X_{DSB} \oplus H_1(sk_{DSB}) \oplus r_{DSB}$ and verifies its validity. Meanwhile, *DB* computes a hash signature $tid^*_{DSB} = H_1(r_{DSB}, r_{DSA}, r_{DA}, t_{DSB})$ and verifies whether it matches the received signature $tid^*_{DSB} \stackrel{?}{=} tid_{DSB}$. If the verification succeeds, *DB* selects a random number r_{DB}, generates a fresh timestamp t_{DB}, computes a session key $k = H_1(r_{DB}, r_{DSB}, r_{DSA}, r_{DA})$, and stores the session key k in the secure memory for subsequent secure communication. Subsequently, *DB* encrypts $Y_{DB} = E_{tk_{DSB}}(r_{DB})$ and computes a hash signature $tid_{DB} = H_1(r_{DB}, t_{DB}, k)$. Finally, *DB* transmits the message tuple (Y_{DB}, tid_{DB}, t_{DB}) back to *DSB*.

5) ***DSB***'s authentication reply

The *DSB* receives the message (Y_{DB}, tid_{DB}, t_{DB}) from *DB*, and checks the freshness of timestamp t_{DB}. If the verification holds, *DSB* decrypts $(r_{DB}) = D_{tk_{DB}}(Y_{DB})$, computes and stores the session key $k = H_1(r_{DB}, r_{DSB}, r_{DSA}, r_{DA})$. Then, *DB* computes a hash signature $tid^*_{DB} = H_1(r_{DB}, t_{DB}, k)$ and verifies $tid^*_{DB} \stackrel{?}{=} tid_{DB}$. If it does not hold, *DSB* immediately stops the process. Otherwise, *DSB* generates a fresh timestamp t'_{DSB}, encrypts $Y'_{DSB} = E_{tk_{DSA}}(r_{DB}, r_{DSB})$ and computes a hash signature $tid'_{DSB} = H_1(r_{DB}, r_{DSB}, t'_{DSB}, k)$. Finally, *DSB* sends the message (Y'_{DSB}, tid'_{DSB}, t'_{DSB}) back to *DSA*.

6) ***DSA***'s authentication reply

Upon receiving the message (Y'_{DSB}, tid'_{DSB}, t'_{DSB}) from *DSB*, *DSA* first checks the freshness of timestamp t'_{DSB}. If the condition satisfies, *DSA* decrypts $(r_{DB}, r_{DSB}) = D_{tk_{DSA}}(Y'_{DSB})$, computes and stores the session key $k = H_1(r_{DB}, r_{DSB}, r_{DSA}, r_{DA})$. It then computes a hash signature $tid'^*_{DSB} = H_1(r_{DB}, r_{DSB}, t'_{DSB}, k)$ and verifies $tid'^*_{DSB} \stackrel{?}{=} tid'_{DSB}$. If the verification holds, *DSA* generates a fresh timestamp t'_{DSA}, computes a parameter $Y'_{DSA} = E_{tk_{DA}}(r_{DB}, r_{DSB}, r_{DSA})$ and a hash signature $tid'_{DSA} = H_1(r_{DB}, r_{DSB}, r_{DSA}, t'_{DSA}, k)$. Finally, *DSA* transmits the message (Y'_{DSA}, tid'_{DSA}, t'_{DSA}) back to *DA*.

7) ***DA***'s authentication result

DA also checks the freshness of timestamp t'_{DSA} after receiving the message tuple (Y'_{DSA}, tid'_{DSA}, t'_{DSA}) from *DSA*. Then, *DA* decrypts $(r_{DB}, r_{DSB}, r_{DSA}) = D_{tk_{DA}}(Y'_{DSA})$, computes and stores $k = H_1(r_{DB}, r_{DSB}, r_{DSA}, r_{DA})$. At last, *DA* computes $tid'^*_{DSA} = H_1(r_{DB}, r_{DSB}, r_{DSA}, t'_{DSA}, k)$ and verifies $tid'^*_{DSA} \stackrel{?}{=} tid'_{DSA}$. If the condition satisfies, the session key k is considered valid.

4.4. Revocation phase

The revocation phase is deployed in HCDA protocol for *DS* to manage (using Algorithm 1) the embodied intelligent robot identity and prevent the loss of session key. The *DS*'s revocation process is presented as follows:

1) ***DA***'s revocation request

DA first validates generates a fresh timestamp t_{DA} and computes $hash.signature = H_1(tk_{DA}, id_{DA}, t_{DA})$ via invoking *DA*'s revocation request function. Then, the signature message will be sent to *DSA*.

2) ***DSA***'s revocation request

When *DSA* obtains the parameters, it triggers the *DSA*'s revocation request function to verify the timestamp t_{DA} and *hash.signature*. If not, *DA* revocation fails; otherwise, the session key k is released by *DSA*. Then, *DSA* computes $request.signature = Sign(sk_{DSA}, id_{DSA}, t_{DSA})$ and $hash.signature = H_1(request.signature, t_{DSA})$. Finally, the signature message will be sent to *DSA*.

3) ***DSB***'s revocation reply

DSB uses the *DSB*'s revocation reply function to check the validity of *DSA*'s revocation request. If the equation $Verify(request.signature) = 1$ and $Verify(hash.signature) = 1$, the session key k is released by *DSA*.

5. Security analysis

In this section, we analyze the security of the HCDA protocol, including mutual authentication, entity anonymity, conditional traceability, man-in-the-middle attack, replay attack, eavesdropping attack, key guessing attack, stolen-verifier attack, pseudonym disclosure attack, perfect forward secrecy, impersonation attack on embodied intelligent robot, impersonation attack on domain server, and desynchronization attack.

1) Mutual authentication

In the HCDA protocol, the entities participating in the authentication include *DA*, *DB*, *DSA*, and *DSB*. From the perspective of the protocol process, in each authentication step, the receiving party determines the validity of the received message by checking the timestamp of the received message, decrypting the message content, performing relevant hash calculations, and conducting comparisons. This series of operations ensures that each participating party can confirm the authenticity of the identities of other parties with whom it interacts, thereby achieving mutual authentication among the participating parties.

2) Entity anonymity

Take the *DA* as an example. Its real identity id_{DA} is not directly exposed during the authentication process. When *DA* initiates an authentication request to *DSA*, *DA* computes $X_{DA} = pid_{DA} \oplus = H_1(sk_{DSA}) \oplus id_{DA} \oplus r_{DA}$ and $Y_{DA} = E_{tk_{DA}}(id_{DA}, r_{DA})$, where id_{DA} has been encrypted and hidden. From the perspective of the security of encryption and hash functions, it is difficult to solve second preimage resistance. The attacker $\mathscr{A}$ cannot easily obtain the key information required for decryption, and thus it is difficult to calculate id_{DA}. Hence, we conclude that the HCDA protocol provides entity anonymity.

3) Conditional traceability

The participating entities (*DA*,*DB*, *DSA*, *DSB*) select random numbers r_{DA}, r_{DSA}, r_{DSB}, and r_{DB} generate dynamic timestamps t_{DA}, t_{DSA}, t_{DSB}, and

Algorithm 1
DA revocation process.

Input: DA revocation request.
Input: Session key released.
Begin
Function DA's revocation request **( )**
1: Generates a fresh timestamp t_{DA};
2: Computes $tk_{DA} = H_1(id_{DA} \oplus t_{DA})$,
3: $hash.signature = H_1(tk_{DA}, id_{DA}, t_{DA})$;
Return ( );
% DA *sends revocation request to* DSA.
Function DSA's revocation request **( )**
7: **If** $|t^*_{DA} - t_{DA}| \leq \Delta T$ **then**
8: **If** $Verify(hash.signature) = 1$ **then**;
4: The verification succeed, session key released;
4: Generates a fresh timestamp t_{DSA},
5: $request.signature = Sign(sk_{DSA}, id_{DSA}, t_{DSA})$,
6: $hash.signature = H_1(request.signature, t_{DSA})$;
Return ( );
% DSA *sends revocation request to* DSB.
Function DSB's revocation reply **( )**
7: **If** $|t^*_{DSA} - t_{DSA}| \leq \Delta T$ **then**
8: **If** $Verify(request.signature) = 1$ **then**;
9: **If** $Verify(hash.signature) = 1$ **then**;
10: The verification succeed, session key released;
Return ( );
End

t_{DB} to computes temporary ciphertext messages. Therefore, $\mathscr{A}$ cannot trace the behaviors of the participating parties. Moreover, due to the dynamic changes of the timestamps, as well as various encryption protections for identity information, the parameters of each session are independent of each other. As a result, $\mathscr{A}$cannot link different sessions at all, and the conditional traceability of our proposed scheme is guaranteed.

4) Man-in-the-middle attack

In this attack scenario, assume that $\mathscr{A}$ can intercept transmission messages during the authentication and key agreement phases, and attempts to modify the messages to deceive *DA*, *DB*, *DSA*, and *DSB*. To achieve this goal, $\mathscr{A}$ must obtain the secret parameters *r*, *id* and *sk* in order to generate legitimate request messages. However, such secret parameters are not publicly available. If $\mathscr{A}$ attempts to pose as a legitimate entity to launch an attack, due to the lack of the correct private key and the key information required for relevant calculations, it cannot accurately generate message content that complies with the verification rules. The protocol has a corresponding verification mechanism in place, which will verify the received messages. Once a message does not conform to the verification rules, it will be detected. Therefore, the HCDA protocol can effectively defend against man-in-the-middle attack.

5) Replay attack

Similar to the man-in-the-middle attack, in a replay attack, $\mathscr{A}$ can monitor the communication among *DA*, *DB*, *DSA*, and *DSB*. For example, $\mathscr{A}$ can intercept the message $(X_{DA}, Y_{DA}, tid_{DA}, t_{DA})$. The message involves the use of a timestamp t_{DA} with a relatively short usage cycle and a hash signature $tid_{DA} = H_1(id_{DA}, r_{DA}, t_{DA})$. Therefore, without knowing the timestamp t_{DA}, $\mathscr{A}$cannot computes a valid tid_{DA}. Thus, the HCDA protocol can resist replay attack.

6) Eavesdropping attack

In the authentication phase, $\mathscr{A}$ may eavesdrop on the messages. Take the message $(X_{DA}, Y_{DA}, tid_{DA}, t_{DA})$ sent by *DA* to *DSA* as an example. Even if $\mathscr{A}$ eavesdrops on this message, it cannot obtain the identity id_{DA}. For instance, although $\mathscr{A}$ may be able to compute $Y_{DA} = E_{tk_{DA}}(id_{DA}, r_{DA})$, it cannot derive the values of id_{DA} and r_{DA} from Y_{DA}. The identity id_{DA} is also protected by the timestamp t_{DA} and the random number r_{DA}, so it is secure. In conclusion, the eavesdropping attack on the public channels is ineffective.

7) Key guessing attack

In key agreement process, $\mathscr{A}$ can adopt power analysis attacks to obtain the parameters $X_{DA} = pid_{DA} \oplus r_{DA}$ and $Y_{DA} = E_{tk_{DA}}(id_{DA}, r_{DA})$. However, without knowing id_{DA} and r_{DA}, it is infeasible for $\mathscr{A}$ to compute the session key *k*.

8) Stolen-verifier attack

In the registration phase of this protocol, the private key sk_{DSA} of *DSA* is stored in secure memory. The public key pk_{DSA} is uploaded on the blockchain. By leveraging the immutability property of the blockchain, it prevents $\mathscr{A}$ from stealing the private key sk_{DSA} and impersonating *DSA*. When *DA* registers, it submits $H_1(id_{DA})$. According to the collision resistance of the hash function $H(.)$, even if $\mathscr{A}$ intercepts this hashed value, it remains computationally infeasible to derive the original identity id_{DA}. After *DSA* verifies the received identity hash value, it hides the real identity by generating $pid_{DA} = H_1(sk_{DSA}) \oplus id_{DA}$. Consequently, even if $\mathscr{A}$ obtains pid_{DA}, it remains infeasible to decrypt and recover either the private key sk_{DSA} or the real identity id_{DA}.

9) Pseudonym disclosure attack

In the HCDA protocol, consider the pseudonym $pid_{DA} = H_1(sk_{DSA}) \oplus id_{DA}$ as a case study. Even if pid_{DA} is compromised, $\mathscr{A}$ remains unable to derive the domain server's private key $sk_{DSA} = H_1(id_{DSA}, x_{DSA})$ due to the computational intractability of inverting the collision-resistant hash function $H(.)$. This cryptographic property guarantees that the HCDA protocol effectively thwarts pseudonym disclosure attack, thereby establishing robust safeguards against identity inference resulting from pseudonym leakage.

10) Perfect forward secrecy

In the HCDA protocol, even if *id* is compromised, $\mathscr{A}$ remain unable to compromise the security of the previously generated session key $k = H_1(r_{DB}, r_{DSB}, r_{DSA}, r_{DA})$. Specifically, due to the computational hardness of the Computational Diffie-Hellman (CDH) problem, it is infeasible for $\mathscr{A}$ to compute the session key k without knowing r_{DB}, r_{DSB} , r_{DSA}, and r_{DA}. Therefore, the proposed HCDA protocol satisfies forward secrecy.

11) Impersonation attack on embodied intelligent robot

Suppose $\mathscr{A}$ intercepts the valid message tuple (X_{DA}, Y_{DA}, tid_{DA}, t_{DA}) transmitted by *DA*. The purpose of $\mathscr{A}$ is to use this message to leverage this intercepted message to construct a legitimate request message that deceives either *DSA* or *DSB* into authentication. However, $\mathscr{A}$ has not obtained r_{DA}, id_{DA}, and sk_{DSA} embedded within the message. Therefore, $\mathscr{A}$ cannot generate a valid request message within polynomial time. In other words, the proposed scheme can resist the impersonation attacks on embodied intelligent robot.

12) Impersonation attack on domain server

If $\mathscr{A}$ attempts to execute an impersonation attack by disguising as the domain server (taking *DSA* as an example), $\mathscr{A}$ must generate a valid message pair $X_{DSA} = N_{DSA} + t_{DSA} \cdot sk_{DSA} \cdot pk_{DSB}$ and $Y_{DSA} = E_{tk_{DSA}}(r_{DSA}, r_{DA})$, where the parameter N_{DSA} contains both sk_{DSA} and r_{DSA} to prevent $\mathscr{A}$ from replacing *DSA*. *DSB* uses $tid_{DSA} = H_1(r_{DSA}, r_{DA}, t_{DSA})$ to determine whether the message has been altered. Since $\mathscr{A}$ does not know r_{DSA} and r_{DA}, it cannot construct a valid hash signature tid_{DSA}. Therefore, $\mathscr{A}$ cannot carry out an impersonation attack on domain server.

13) Desynchronization attack

In the HCDA protocol, a multi-layered defense mechanism is implemented to effectively counter desynchronization attack. Timestamp verification is utilized. For example, by checking $|t^* - t| \leq \Delta T$, messages with timestamps beyond the allowable range are rejected to prevent old messages from causing confusion. Moreover, with the generation of random numbers r_{DB}, r_{DSB}, r_{DSA}, and r_{DA}, due to their uncertainty, it is very difficult for $\mathscr{A}$ to use old random numbers to interfere. In addition, when the domain server receives a message, it will compare it to determine whether the received message has been tampered with. If the comparison is inconsistent, it will be aborted.

6. Performance evaluation

In this section, we comprehensively evaluate the performance of the HCDA protocol by analyzing its computational cost and communication overhead. In addition, we conduct a comparative analysis between the HCDA protocol and with some of related works [30,31,37] to assess its relative advantages in efficiency overhead.

6.1. Comparison of computational cost

To facilitate a quantitative comparison of computational cost between the HCDA protocol and related works [30,31,37], we evaluate performance through the execution time of core cryptographic operations. To simultaneously simulate the computational overhead of both robot and the server, we employ an embedded simulation environment. All cryptographic primitives were implemented on Raspberry Pi 3 Model *B*+ device featuring a Cortex-A53 (ARMv8) 64-bit System-on-Chip (SoC) operating at 1.4 GHz with 1GB SDRAM. The experimental execution time measurements follow the methodology established in [40], as detailed in Table 2. Notably, the runtime of the lightweight XOR operation is excluded from our analysis due to its significantly shorter execution duration relative to other cryptographic primitives.

For cross-domain authentication protocol in smart manufacturing, entities that execute authentication protocols include embodied intelligent robots and domain servers in local and remote domains. Table 3 counts the numbers of time-consuming cryptographic operations used by each entity during cross-domain authentication in each of the four protocols. The total execution time to complete an authentication process in HCDA protocol is $0.05 + 8.332 + 4.219 + 0.05 = 12.651$ ms, the Shen et al.'s [30] protocol has a total execution time of $18.476 + 68.159 + 68.153 + 18.47 = 173.258$ ms. Similarly, the Wang et al.'s [31] protocol incurs a total execution time of $32.952 + 41.166 + 24.726 = 98.844$ ms during an authentication process, the Roy et al.'s [37] protocol requires $12.351 + 12.339 + 8.244 = 32.934$ ms for total execution time.

Table 2
The average execution time of encryption operations.

Symbol	Description	Time (ms)
T_{em}	Execution time for an ECC point multiplication	4.107
T_{ea}	Execution time for an ECC point addition	0.018
T_h	Execution time for a hash operation	0.006
T_{bp}	Execution time of a bilinear pairing	12.52
T_{sed}	Execution time of a symmetric encryption/decryption	0.013
T_{ex}	Execution time of a modular exponentiation	6.143

Fig. 6 shows a computational cost comparison of the four protocols. It can be observed that our proposed scheme requires the least computational time. Specifically, bilinear pairing operations, exponential computations, and ECC point multiplications dominate the overall execution time. However, in HCDA protocol, *DA* does not need to perform these operations, and the number of operations on the server side is significantly fewer than in related works. The HCDA protocol also holds advantages in energy efficiency, making it more suitable for embodied intelligent robots with strict resource and computational constraints.

Fig. 7 gives specific computational cost comparison of *DSA*. It is evident that as the number of *DA* increases, the computational cost of HCDA protocol grows at a significantly slower rate, thereby demonstrating strong scalability in group network environments. Notably, when the number of *DA* is 10, *DSA* achieves a computational cost of merely 83 ms, further validating its efficiency under high- device-density conditions.

6.2. Comparison of communication overhead

In this subsection, we analyze the communication overhead of the HCDA protocol. Based on experimental parameters, we assume the size of a point in group G is $|G| = 40$ bytes, while the output size of the hash function, the size of identity, and the size of random number are all $|h| = |id| = |r| = 20$ bytes. Additionally, the block size of symmetric encryption/decryption and the size of timestamp are set to $|ed| = 16$ bytes and $|t| = 4$ bytes, respectively.

DA in HCDA protocol sends $M_1 = \{X_{DA}, Y_{DA}, tid_{DA}, t_{DA}\}$ to *DSA*, the

Table 3
The average execution time of encryption operations.

Protocol	*DA*	*DSA*	*DSB*	*DB*	Total
Shen et al. [30]	$3T_{em} + 1T_{ex} + 2T_h$	$6T_{em} + 2T_{ea} + 3T_{ex} + 2T_{bp} + 2T_h$	$6T_{em} + 2T_{ea} + 3T_{ex} + 2T_{bp} + T_h$	$3T_{em} + T_{ex} + T_h$	$18T_{em} + 4T_{ea} + 8_{ex} + 4T_{bp} + 6T_h$
Wang et al. [31]	$8T_{em} + 3T_{ea} + 7T_h$	$10T_{em} + 3T_{ea} + 7T_h$	$6T_{em} + 3T_{ea} + 5T_h$	–	$24T_{em} + 9T_{ea} + 19T_h$
Roy et al. [37]	$3T_{em} + 5T_h$	$3T_{em} + 3T_h$	$2T_{em} + 5T_h$	–	$8T_{em} + 13T_h$
HCDA	$4T_h + 2T_{sed}$	$2T_{em} + T_{ea} + 8T_h + 4T_{sed}$	$T_{em} + T_{ea} + 7T_h + 4T_{sed}$	$4T_h + 2T_{sed}$	$3T_{em} + 2T_{ea} + 23T_h + 12T_{sed}$

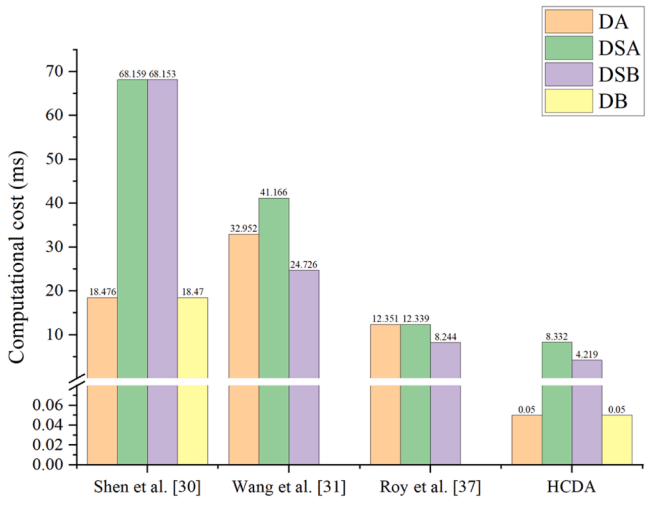

Fig. 6. Computational cost comparison.

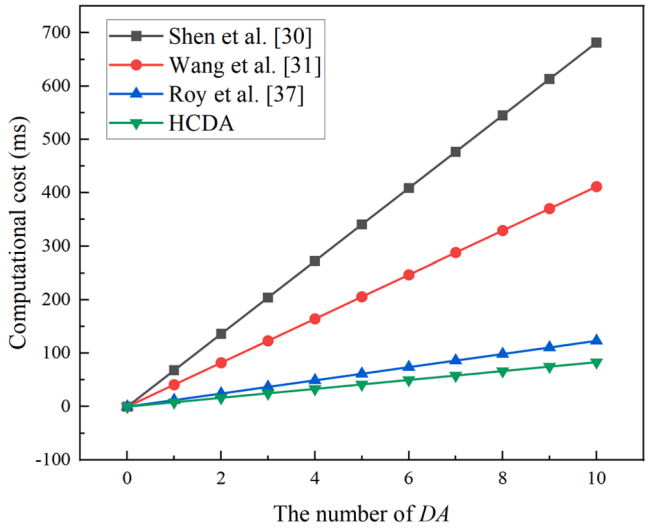

Fig. 7. Computational cost comparison of *DSA* with different number of *DA*.

required communication overhead is $|Z_q^*| + |h| + |ed| + |t| = 20 + 20 + 16 + 4 = 60$ bytes; *DSA* sends $M_2 = \{X_{DSA}, Y_{DSA}, tid_{DSA}, t_{DSA}\}$ to *DSB* and $M_6 = \{Y'_{DSA}, tid'_{DSA}, t'_{DSA}\}$ to *DA*, all messages incur the communication overhead of $|G| + 2|h| + 2|ed| + 2|t| = 40 + 40 + 32 + 8 = 120$ bytes; *DSB* sends $M_3 = \{X_{DSB}, Y_{DSB}, tid_{DSB}\}$ to *DB* and $M_5 = \{Y'_{DSB}, tid'_{DSB}, t'_{DSB}\}$ to *DSA*, the communication overhead is $|r| + 2|h| + 2|ed| + |t| = 20 + 40 + 32 + 4 = 96$ bytes. *DB* sends $M_5 = \{Y_{DB}, tid_{DB}, t_{DB}\}$ to *DSB*, where the size of this data is $|h| + |ed| + |t| = 20 + 16 + 4 = 40$ bytes.

In Shen et al.'s [30] protocol, *DA* sends $M_1 = \left\{\left(h, S, ID_{e_i^A}\right), \left(N_{e_i^A}, ID_{e_i^A}\right), \left(N_{e_i^A}, ID_{e_i^A}, h, S\right)\right\}$ to *DSA*, the communication overhead is $2|G| + 3|id| + 2|h| + 2|r| = 40 + 60 + 40 + 40 = 220$ bytes; *DSA* sends $M_2 = \left\{N, \left(h, S, N_{e_i^A}, ID_{e_i^A}\right)\right\}$ of size $|G| + |id| + |h| + 2|r| = 40 + 20 + 20 + 40 = 120$ bytes to *DSB*; $M_3 = \{(Request), (Verification\ Result)\}$ is transmitted by *DSB*, and the data size is $2|r| = 40$ bytes; *DB* sends $M_1 = \left\{\left(N'_{e_i^A}, ID'_{e_i^A}, h', S'\right), (Response)\right\}$ to *DSB*, *DB*'s communication burden is $|G| + |id| + |h| + 2|r| = 40 + 20 + 20 + 40 = 120$ bytes.

In Wang et al.'s [31] protocol, *DA* sends $M_1 = \{A, W_U, \sigma, T_U\}$ and $M_3 = \{M\}$ to *DSA*, which occupy totally $|G| + 3|r| + |t| = 40 + 60 + 4 = 104$ bytes communication overhead; *DSB* sends $M_2 = \{B, w_1, T_1\}$ to *DA* and $M_2 = \{A', B, W_1, l, T_2\}$ to *DSB*, then $3|G| + 2|r| + |h| + 2|t| = 120 + 40 + 20 + 8 = 188$ bytes communication overhead is needed; *DSB* sends $M_2 = \{B', w_2, T_3\}$ to *DA*, the communication overhead is $|G| + |h| + |t| = 40 + 20 + 4 = 64$ bytes.

In Roy et al.'s [37] protocol, *DA* sends $M_1 = \{ID_H, XID_M^*, A\}$ and $M_5 = \{K\}$ to *DSA*, *DA*'s communication overhead is $|G| + |r| + |id| + |h| = 40 + 20 + 20 + 20 = 100$ bytes; *DSA* sends $M_2 = \{ID_S, XID_M^*, K_B, A, B\}$ to *DSB* and $M_4 = \{V_2, K, B\}$ to *DA*, and the corresponding data size is $3|G| + |r| + |id| + 3|h| = 120 + 20 + 20 + 60 = 220$ bytes; *DSB*sends $M_3 = \{V_1, V_2\}$ to *DSA*, the communication overhead of $2|h| = 40$ bytes.

Fig. 8 shows the results of the communication overhead comparison of the four protocols. Compared with Shen et al.'s [30] protocol, Wang et al.'s [31] protocol, and Roy et al.'s [37] protocol, the total communication overhead of the HCDA protocol is reduced by 36.8 %, 11.2 %, and 12.2 % respectively. Moreover, HCDA protocol shows 72.7 %, 42.3 %, and 40 % lower *DA*'s communication overhead than Shen et al.'s [30] protocol, Wang et al.'s [31] protocol, and Roy et al.'s [37] protocol, respectively. The results demonstrate that the HCDA protocol achieves the highest communication efficiency among all compared related works [30,31,37]. Further, the HCDA protocol incurs the lowest communication cost on the device side, making it particularly suitable for resource-constrained *DA*.

Fig. 9 compares the aggregated communication overhead of protocols across 1–10 session key agreement cycles. It is worth mentioning that, on average, the communication overhead of HCDA protocol is better than that of related works [30,31,37].

7. Conclusion and future work

In this paper, we propose a blockchain-based embodied intelligence network model for smart manufacturing, in which the domain server as a consensus node to achieve consistent management of authentication parameters. Then, a hidden cross-domain authentication protocol based on authentication migration method is proposed for the network model, which can ensure the cross-domain communication security and reduce the computational cost of the authentication phase of embodied intelligent robot. We use widely accepted informal security analysis to demonstrate that the proposed HCDA protocol can resist a variety of known well-known attacks. In addition, in terms of computational cost and communication overhead, the HCDA protocol expends less resource consumption as compared to related works, which is verified by simulation experiments.

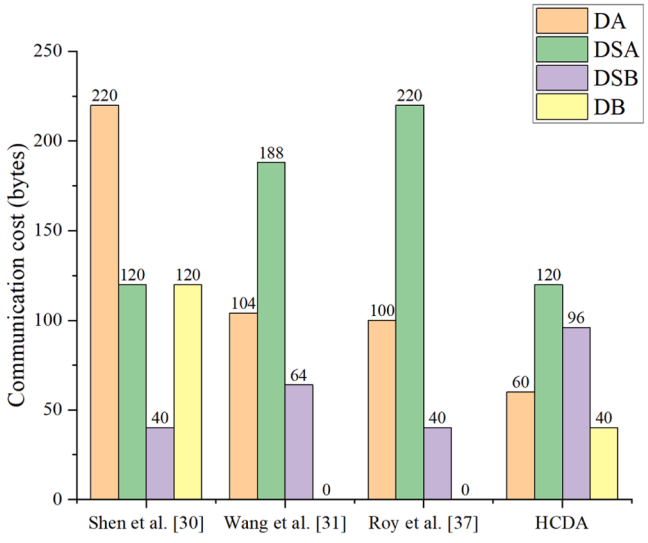

Fig. 8. Communication overhead comparison among different entities.

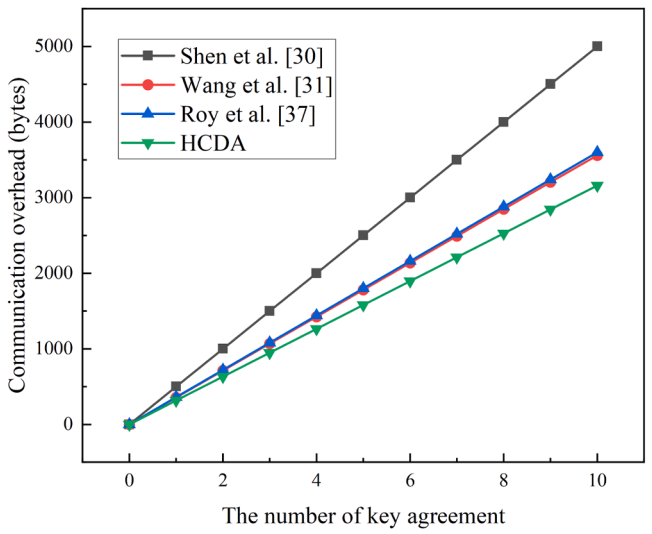

Fig. 9. Total communication overhead comparison among different protocols.

Although the proposed protocol realizes anonymous cross-domain authentication for multiple embodied intelligent robots, ensuring data trustworthiness and communication security between smart manufacturing entities and other network entities remains a challenge. In the future, we aim to extend the current protocol by exploring the adoption of distributed Ring Learning with Errors (Ring-LWE) to enable a multi-group cross-domain secure communication model.

CRediT authorship contribution statement

Huaiyao Yang: Writing – original draft, Conceptualization. **Xiangwei Meng:** Writing – review & editing. **Jiale Liang:** Formal analysis, Conceptualization. **Yanrong Zhang:** Formal analysis. **Keqin Li:** Writing – original draft, Supervision.

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.

Acknowledgment

This work is supported in part by the Major Science and Technology Tackling Project of Hunan Province under Grant 2025QK2008, and in part by the Heilongjiang Provincial Support Program for Basic Research of Outstanding Young Teachers under Grant YQJH2024095.

Data availability

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

References

[1] K. Gai, Q. Xiao, M. Qiu, G. Zhang, J. Chen, Y. Wei, Y. Zhang, Digital twin-enabled AI enhancement in smart critical infrastructures for 5G, ACM Trans. Sens. Netw. 18 (3) (2022) 1–20.
[2] J. Li, M. Qiu, J.W. Niu, L.T. Yang, Y. Zhu, Z. Ming, Thermal-aware task scheduling in 3D chip multiprocessor with real-time constrained workloads, ACM Trans. Embed. Comput. Syst. 12 (2) (2013) 1–22.
[3] N. Tekin, A. Aris, A. Acar, S. Uluagac, V.C. Gungor, A review of on-device machine learning for IoT: an energy perspective, Ad Hoc Netw 153 (2024) 103348.
[4] J.B. Minani, F. Sabir, N. Moha, Y.G. Guéhéneuc, A systematic review of IoT systems testing: objectives, approaches, tools, and challenges, IEEE Trans. Softw. Eng. 50 (4) (2024) 785–815.
[5] V.R. Kebande, A.I. Awad, Industrial internet of things ecosystems security and digital forensics: achievements, open challenges, and future directions, ACM Comput. Surv. 56 (5) (2024) 1–37.
[6] A. De Benedictis, F. Flammini, N. Mazzocca, A. Somma, F. Vitale, Digital twins for anomaly detection in the industrial internet of things: conceptual architecture and proof-of-concept, IEEE Trans. Ind. Inform. 19 (12) (2023) 11553–11563.
[7] Y. Liu, W. Liang, K. Xie, S. Xie, K. Li, W. Meng, Lightpay: a lightweight and secure off-chain multi-path payment scheme based on adapter signatures, IEEE Trans. Serv. Comput. 17 (4) (2023) 1622–1635.
[8] C. Ni, S.C. Li, Machine learning enabled industrial IoT security: challenges, trends and solutions, J. Ind. Inf. Integr. 38 (2024) 100549.
[9] W. Liang, Y. Li, K. Xie, D. Zhang, K.C. Li, A. Souri, K. Li, Spatial-temporal aware inductive graph neural network for C-ITS data recovery, IEEE Trans. Intell. Transp. Syst. 24 (8) (2022) 8431–8442.
[10] J. Liu, J. Lou, L. Xiong, J. Liu, X. Meng, Cross-silo federated learning with record-level personalized differential privacy, in: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, 2024, pp. 303–317.
[11] M. Seifelnasr, R. AlTawy, A. Youssef, A conditional privacy-preserving protocol for cross-domain communications in VANET, IEEE Trans. Intell. Transp. Syst. 26 (4) (2025) 5251–5263.
[12] Z. Shao, M. Wang, Y. Chen, C. Xue, M. Qiu, L.T. Yang, E.H.M. Sha, Real-time dynamic voltage loop scheduling for multi-core embedded systems, IEEE Trans. Circuits Syst. II-Express Briefs 54 (5) (2007) 445–449.
[13] X. Meng, B. Liu, X. Meng, Y. Liang, H. Deng, A lightweight group authentication protocol for blockchain-based vehicular edge computing networks, IEEE Trans. Intell. Transp. Syst. 25 (8) (2024) 8556–8567.
[14] J. Höglund, M. Furuhed, S. Raza, Lightweight certificate revocation for low-power IoT with end-to-end security, J. Inf. Secur. Appl. 73 (2023) 103424.
[15] S. Alagarsamy, V. Nagarajan, M.M.Y. Devi, OMIBC: optimal modified identity-based cryptography for signcryption and private key extraction using fuzzy model, Wirel. Netw. 30 (4) (2024) 2159–2172.
[16] S. Zhang, Y. Pan, Q. Liu, Z. Yan, K.K.R. Choo, G. Wang, Backdoor attacks and defenses targeting multi-domain AI models: a comprehensive review, ACM Comput. Surv. 57 (4) (2024) 1–35.
[17] A. Pasdar, Y.C. Lee, Z. Dong, Connect API with blockchain: a survey on blockchain oracle implementation, ACM Comput. Surv. 55 (10) (2023) 1–39.
[18] J. Tan, J. Shi, L. Wu, B. Chen, H. Tang, C. Zhang, W. Zhang, J. Wan, S. Wang, Embodied intelligence empowering customized manufacturing: architecture, opportunities, and challenges, IEEE Access 13 (2025) 2169–3536.
[19] S. Kumar, M.K. Barua, Exploring the hyperledger blockchain technology disruption and barriers of blockchain adoption in petroleum supply chain, Resour. Policy 81 (2023) 103366.
[20] 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.
[21] D. Rangwani, D. Sadhukhan, S. Ray, M.K. Khan, M. Dasgupta, A robust provable-secure privacy-preserving authentication protocol for Industrial Internet of Things, Peer Peer Netw. Appl. 14 (3) (2021) 1548–1571.
[22] R. Hajian, A. Haghighat, S.H. Erfani, A secure anonymous D2D mutual authentication and key agreement protocol for IoT, Internet Things 18 (2022) 100493.
[23] F. Rafique, M.S. Obaidat, K. Mahmood, M.F. Ayub, J. Ferzund, S.A. Chaudhry, An efficient and provably secure certificateless protocol for Industrial Internet of Things, IEEE Trans. Ind. Inform. 18 (11) (2022) 8039–8046.
[24] M. Tanveer, A. Alkhayyat, A.U. Khan, N. Kumar, A.G. Alharbi, REAP-IIoT: resource-efficient authentication protocol for the Industrial Internet of Things, IEEE Internet Things J 9 (23) (2022) 24453–24465.
[25] H. Xu, C. Hsu, L. Harn, J. Cui, Z. Zhao, Z. Zhang, Three-factor anonymous authentication and key agreement based on fuzzy biological extraction for Industrial Internet of Things, IEEE Trans. Serv. Comput. 16 (4) (2023) 3000–3013.
[26] M. Tanveer, A. Badshah, H. Alasmary, S.A. Chaudhry, CMAF-IIoT: Chaotic map-Based Authentication Framework For Industrial Internet of Things, 23, Internet of Things, 2023 100902.
[27] B.D. Deebak, F.H. Memon, K. Dev, S.A. Khowaja, W. Wang, N.M.F. Qureshi, TAB-SAPP: a trust-aware blockchain-based seamless authentication for massive IoT-enabled industrial applications, IEEE Trans. Ind. Inform. 19 (1) (2022) 243–250.
[28] S. Dhar, A. Khare, A.D. Dwivedi, R. Singh, Securing IoT devices: a novel approach using blockchain and quantum cryptography, Internet Things 25 (2024) 101019.
[29] W. Liang, Y. Li, J. Xu, Z. Qin, D. Zhang, K.C. Li, Qos prediction and adversarial attack protection for distributed services under dlaas, IEEE Trans. Comput. 73 (3) (2023) 669–682.
[30] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, M. Guizani, Blockchain-assisted secure device authentication for cross-domain industrial IoT, IEEE J. Sel. Areas Commun. 38 (5) (2020) 942–954.
[31] W. Wang, H. Huang, L. Xue, Q. Li, R. Malekian, Y. Zhang, Blockchain-assisted handover authentication for intelligent telehealth in multi-server edge computing environment, J. Syst. Architect. 115 (2021) 102024.
[32] P. Singh, M. Masud, M.S. Hossain, A. Kaur, Cross-domain secure data sharing using blockchain for industrial IoT, J. Parallel Distrib. Comput. 156 (2021) 176–184.
[33] Y. Zhang, B. Li, J. Wu, B. Liu, R. Chen, J. Chang, Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT, IEEE Internet Things J. 9 (22) (2022) 22501–22515.
[34] Y. Zhang, B. Li, J. Wu, B. Liu, R. Chen, J. Chang, CCAP: a complete cross-domain authentication based on blockchain for Internet of Things, IEEE Trans. Inf. Forensic Secur. 17 (2022) 3789–3800.

[35] J. Cui, N. Liu, Q. Zhang, D. He, C. Gu, H. Zhong, Efficient and anonymous cross-domain authentication for IIoT based on blockchain, IEEE Trans. Netw. Sci. Eng. 10 (2) (2022) 899–910.
[36] O.A. Khashan, N.M. Khafajah, Efficient hybrid centralized and blockchain-based authentication architecture for heterogeneous IoT systems, J. King Saud Univ.-Comput. Inf. Sci. 35 (2) (2023) 726–739.
[37] P.K. Roy, A. Bhattacharya, Secure and authentic anonymous roaming service, Wirel. Pers. Commun. 125 (1) (2022) 819–839.
[38] Y. Li, W. Liang, K. Xie, D. Zhang, K. Li, N.N. Xiong, EventMon: real-time event-based streaming network monitoring data recovery, IEEE Trans. Dependable Secur. Comput. (2024).
[39] W. Liang, S. Xie, K.C. Li, X. Li, X. Kui, A.Y. Zomaya, MC-DSC: a dynamic secure resource configuration scheme based on medical consortium blockchain, IEEE Trans. Inf. Forensic Secur. 19 (2024) 3525–3538.
[40] S.A. Chaudhry, J. Nebhan, K. Yahya, F. Al-Turjman, A privacy enhanced authentication scheme for securing smart grid infrastructure, IEEE Trans. Ind. Inform. 18 (7) (2021) 5000–5006.