Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.34022

A Secure and Lightweight Trust-Aware Data Sharing Scheme for Digital Cousin Networks in IoT

Jiaming Du

School of Telecommunications Engineering, Xidian University, Xi'an, China
23009100536@stu.xidian.edu.cn

Abstract: The Internet of Things (IoT) keeps growing fast, and that's pushed Digital Twin (DT) technology into wider use—letting physical objects and their virtual copies interact in real time. But traditional DT-based systems often run into high communication costs and poor scalability, mainly because they rely on constant data syncing and high-precision modeling. To get around these limitations, we propose a lightweight yet semantically consistent idea called Digital Cousin (DC). Based on that, we also design a secure data sharing scheme for Digital Cousin Networks (DCNs). Our scheme combines three pieces: registration with a trusted authority, proxy delegation, and trust-aware participant matching. It also uses a mutual authentication protocol built on proxy signatures to set up secure session keys. Security analysis shows the scheme can guarantee mutual authentication, anonymity, unlinkability, and traceability, and it holds up well against replay attacks and man-in-the-middle attacks. We ran experiments to test how this approach holds up, the results tell us it cuts communication overhead by up to 23.1% when compared with DT-based methods. It also brings that overhead down by more than 58% against baseline schemes. This seems to mean it works quite well in large-scale, dynamic IoT environment.

Keywords: Internet of things, digital cousin, data sharing, mutual authentication, proxy signature.

1. Introduction

Communication technologies keep getting better every year, Internet of Things (IoT) has grown really fast, letting all sorts of different devices connect on large scale. In this environment, Digital Twin (DT) technology seems to be a really useful tool. In cloud contexts, people often use it to make virtual copies of physical objects. Using artificial intelligence and machine learning, DTs can take real-time data collected from physical devices, interpret it, and send smart feedback through IoT communication infrastructures (such as device-to-cloud interface), this boosts decision-making accuracy and operational efficiency. Then, when you link multiple DTs together, you get a Digital Twin Network (DTN). This network lets each twin share data with others effectively, and it mirrors the physical IoT system in cyberspace. Instead of requiring direct device-to-device communication, this design cuts down on strain on wireless resources, it also makes the system easier to expand as needed.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.34022

However, despite these advantages, conventional DT systems still rely significantly on continual synchronization and high-fidelity modeling to maintain a tight match with their physical counterparts. Particularly in IoT situations with limited resources, the type of architecture nearly always results in significant communication overhead and restricts scalability. The concept of a Digital Cousin (DC) has lately surfaced as a lighter substitute for DT in order to circumvent these disadvantages. In contrast to traditional DTs, DCs adopt a loosely connected approach, creating virtual entities with comparable geometric and semantic characteristics that are appropriate for task-oriented surrogate interaction [1]. DCs can significantly reduce transmission costs while providing greater flexibility and adaptability across various contexts by eschewing the requirement for full-state synchronization and concentrating instead on semantic consistency. For large-scale and extremely dynamic IoT implementations, a Digital Cousin Network (DCN) is therefore more appropriate.

DCN does boost system efficiency, but it also introduces new security and reliability problems. Bad actors might create fake DCs and plug them into the network, they could also take over legitimate ones to spread false information. This can mislead people who make critical decisions, or let attackers interested on getting private data take advantage of trust-based relationships. Because DCs communicate with each other over open network channels, the data they transfer is still vulnerable to common cyberthreats like eavesdropping and tampering. Even reputable DCs may generate inconsistent or wrong data. Sometimes this happens due to environmental interference, other times because sensors fail. All these problems can seriously hurt system reliability, and may even lead to incorrect control actions in physical system. So building a secure, trustworthy data-sharing mechanism for DCN is still a major challenge we have to deal with.

A fair amount of research has looked into secure data transmission in twin-enabled IoT systems. For instance, [2] proposes a soft actor-critic based deep reinforcement learning method to jointly optimize synchronization strategies and resource allocation. The goal there is to improve DT synchronization efficiency for user devices (UDs) while lowering overall synchronization latency. Meanwhile, [3] introduces a sharding-based Practical Byzantine Fault Tolerance (PBFT) framework that uses parallel validation and a reputation enhancement mechanism to boost throughput and cut down latency during data sharing. Likewise, [4] looks into using homomorphic encryption in federated learning to protect data privacy in DT environments. More recently, some quantum-resistant twin systems have started using lattice-based cryptographic approaches, like Kyber-1024, to deal with emerging threats such as harvest-now-decrypt-later attacks [5]. However, most of these studies were developed for conventional DT architectures. They don't fully account for the unique characteristics of DCN, especially its loosely coupled semantic interaction model and the need for efficient, trust-aware participant selection. In particular, existing solutions often lack effective ways to identify reliable data providers before any data sharing actually happens—something that's critical for keeping decision accuracy in DC environments.

To address these issues, this paper proposes a secure, lightweight, and scalable data-sharing scheme that's tailored specifically for DCN. In our framework, a Trusted Authority (TA) takes on the job of registering physical entities (PE) and generating corresponding DCs along with cryptographic credentials. Using authorization tokens issued by the PE, DCs can obtain proxy delegation from the TA and then join network interactions. After that, a requester DC runs a similarity evaluation combined with reputation-based filtering to find trustworthy candidate providers. Finally, we adopt a proxy signature-based mechanism to carry out mutual authentication and session key establishment, which ensures data confidentiality and integrity during transmission. The proposed scheme strengthens security and privacy protection in DCN while also reducing communication overhead, thanks to

DOI: 10.54254/2755-2721/2026.34022

the lightweight nature of DCs. Both theoretical analysis and simulation results show that our method offers better scalability and efficiency compared to existing DT-based approaches.

The remainder of this paper proceeds in the following order. In Section 2,the requisite preliminaries and underlying cryptographic foundations are outlined. Section 3 then sets out the system model,threat model,and design objectives. A detailed account of the proposed scheme is given in Section 4. In Section 5,the security analysis and performance evaluation are presented. Section 6 concludes the paper.

2. Preliminaries

2.1. Digital cousin: definition and generation

When dealing with different IoT contexts, strict one-to-one mapping between virtual models and physical objects often limits scalability, it also hurts how well these systems can generalize. People have been using DTs a lot lately to represent PE in cyberspace, but this rigid mapping creates big problems for them. Models based in DTs may find it hard to adjust properly when real-world circumstances are dynamic, constantly shifting, and full of unexpected changes. So this study puts forward the idea of DCs to get past these constraints.

DCs are artificial virtual things that, without needing precise replication of physical traits or behaviors, retain statistical and semantic consistency with real-world items. DCs are more concerned with maintaining high-level characteristics and functional semantics than DTs, which prioritize behavioral synchronization and fine-grained geometric precision. By allowing a single physical entity to correspond to numerous DC instances, this loose modeling approach enhances cross-domain generalization, scalability, and robustness. To enable efficient DC construction, this paper adopts the Automated Creation of Digital Cousins (ACDC) framework, which supports scalable and automated generation of virtual environments. DINOv2 is the feature representation model used in this framework to gauge how comparable candidate virtual objects and real-world assets are. More precisely, candidates with smaller embedding distances are considered more semantically meaningful and are chosen as legitimate DCs when similarity is assessed in the feature embedding space [6].

DC system operates through three core processing phases, let me walk you through each step. First comes the Extraction stage: we take input RGB image and generate semantic object masks, these masks help us isolate specific elements in scene that need to be separated from the rest of the visual content. Then we move to the Matching stage. Here, every object we've extracted gets paired with a suitable DC candidate based in embedding-space similarity—this ensures each object matches the right DC type for the task. Next is the Generation stage, where we take all the selected DCs, combine them carefully and polish the tiny details to create virtual environment that's cohesive, interactive, and physically consistent. This entire pipeline means ACDC framework can efficiently build DC-based environments, it also keeps semantic coherence with the original physical scene so the virtual space doesn't feel disconnected from the real one.

2.2. Elliptic curve cryptography

This work uses Elliptic Curve Cryptography (ECC) to provide safe and lightweight data sharing in resource-constrained IoT contexts due to its robust security assurances and comparatively minimal computational cost [7].

Let q be a large prime number, and let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. The curve can be expressed as follows: $E_q(a,b) : y^2 \equiv x^3 + ax + b \pmod{q}$. To guarantee that the curve

DOI: 10.54254/2755-2721/2026.34022

stays non-singular, the parameters $a, b \in \mathbb{F}_q$ must meet the requirement $4a^3 + 27b^2 \not\equiv 0 \pmod{q}$. Let P be the generator of the group of prime order q, and let G be a cyclic group of points on E.

In the ECC setting, a private key $x \in \mathbb{Z}_q^*$ is randomly selected, and the corresponding public key is generated through point multiplication, i.e., $Q = xP$. The security of ECC relies on the computational difficulty of its underlying mathematical problems, which provides the foundation for confidentiality and authentication.

2.3. Security assumptions

The proposed scheme is built upon two widely accepted hardness assumptions in elliptic curve cryptography: the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the Elliptic Curve Computational Diffie-Hellman Problem (ECDHP). For a cyclic group G with prime order q and generator P, the ECDLP indicates that it is difficult to determine the scalar x when $Q = xP \in G$. This property guarantees that an attacker cannot obtain the private key from the corresponding public key. Similarly, the ECDHP assumes that it is computationally infeasible to calculate xyP when given $(P, xP, yP) \in G$, where x and y belong to $\mathbb{Z}_q^*$. If we run our key agreement over insecure channel, it may not stay secure without this assumption. This assumption plus a second one team up, they provide the basic cryptography for our scheme, and protect against key recovery attacks and unauthorized release of information.

3. System model and threat model

This section first introduces the system model of the proposed IoT-oriented DCN by describing the major participating entities. Afterward, the threat model considered in this study is presented.

3.1. System Model

As illustrated in Figure 1, the proposed digital cousin network architecture is composed of three main components: the TA, PE, and DC.

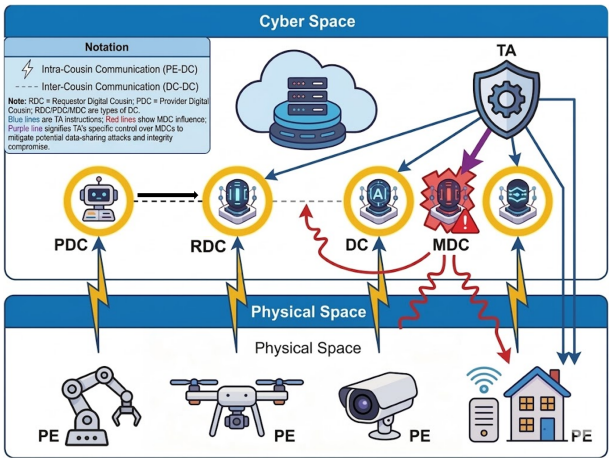

Figure 1: System model of the proposed IoT-oriented digital cousin network.

DOI: 10.54254/2755-2721/2026.34022

3.1.1. TA

When we get to the initial access stage, every PE and their corresponding DC must register with TA to grab valid public-private key pairs, without these pairs, they may not be able to take part in any network activities. TA is a trusted central entity, it takes charge at system initialization and identity registration for the whole network. Also, it issues proxy authorization credentials to legitimate DCs, these credentials let them securely act as both data requesters and data providers during network interactions.

3.1.2. PE

Each PE generates authorization tokens for its associated DCs, it has other core tasks too. These are real IoT devices set up in physical environment, not simulated ones you'd see in lab. Most have limited sensing capability, barely enough storage space, and they can't run heavy computational processes for long periods without slowing down or crashing. Then, alongside making those tokens, each PE uploads physical state information on regular basis and executes control instructions through PE–DC interaction. When a DC submits these tokens to the TA, it can obtain proxy delegation rights that let it take part at system-wide data sharing legally without any access-related issues. Also, a single PE may correspond to multiple DC instances, this helps meet different functional demand.

3.1.3. DC

A DC isn't like traditional Digital Twins that stress strict physical replication, it's a virtual representation of a PE in cyberspace where the connection relies mainly on semantic similarity and functional abstraction. Once the TA issues valid proxy credentials and proxy signature keys, a DC can operate in two distinct roles: Requester Digital Cousin (RDC) and Provider Digital Cousin (PDC). The RDC starts data requests for its linked PE, meanwhile the PDC responds to these requests and provides relevant data to back up collaborative decision-making and system optimization. But network remains vulnerable to threats from Malicious Digital Cousins (MDCs) made by attackers, as well as legitimate DCs that have been compromised. These harmful entities may launch attacks or inject false data, this can seriously hurt the reliability of the whole data-sharing process, to some extent.

3.2. Threat model

The security of the proposed scheme is analyzed under the Dolev-Yao model, in which the adversary is assumed to have full control over the communication channels. Since interactions among DCs take place over public networks, MDCs in the Physical Entity Digital Cousin Network (PEDCN) may attempt to break data confidentiality and integrity, interfere with PE operations, or infer sensitive information by analyzing communication patterns between DCs. Moreover, even authorized DCs may behave maliciously and deliberately disseminate incorrect, outdated, or low-quality data during the sharing process. To mitigate these threats, the proposed framework introduces a trusted TA that is responsible for system initialization, distribution of cryptographic parameters, and tracing of malicious identities once abnormal behaviors are detected.

4. Proposed secure data sharing and authentication scheme

DOI: 10.54254/2755-2721/2026.34022

4.1. Framework Overview

The proposed DCN framework is organized into five consecutive stages, namely system initialization, registration of PEs and DCs, proxy delegation for authorized participation, trust-driven matching with candidate filtering, and finally mutual authentication followed by session key establishment. Together, these stages form a complete process that supports secure and reliable data sharing among DCs [8]. Together, these stages form a complete process that supports secure and reliable data sharing among DCs [8].

4.1.1. System initialization and parameter setup

During system initialization stage, the TA is in charge of bootstrapping the entire DCN. It generates all the basic cryptographic parameters that the network needs to operate properly, these parameters seem to form foundation for secure communication and trustworthy data exchange among every participating entity.

4.1.2. Registration of Physical Entities and Digital Cousins

During the registration process, both PEs and their corresponding DCs deployed on dedicated servers are required to register with the TA. Through this procedure, they obtain valid public-private key pairs. This step ensures that each physical entity and its digital counterpart are bound to verifiable cryptographic identities, which are necessary for secure participation in the network.

4.1.3. Proxy delegation for authorized participation

To get valid public-private key pairs, every PE and its corresponding DC — deployed on dedicated servers — must register with TA during the registration process. This isn't just a formality. It binds each physical entity and its digital counterpart to a verifiable cryptographic identity, this is necessary for them to join network securely.

4.1.4. Trust-based matching and candidate filtering

In this phase, RDCs size up potential collaborators. They base their assessment in trust metrics and attribute similarity, which help them pick right partners, since using both to lead to more reliable choices. Candidates with low trust values or weak relevance are filtered out in advance, so that only highly trusted and suitably matched PDCs proceed to the subsequent authentication and data sharing steps.

4.1.5. Mutual authentication and session key establishment

After candidate selection, eligible RDCs and PDCs perform mutual authentication using proxy signature mechanisms. Once both parties are successfully verified, they establish pairwise session keys, which are then used to ensure the confidentiality and integrity of all data exchanged within the DCN.

4.2. Detailed design of the proposed scheme

4.2.1. System initialization and cryptographic setup

Within the DCN, the TA first initializes the network by selecting a security parameter λ and a large prime number q, where $|q| = \lambda$ bits. Subsequently, an elliptic curve $E : y^2 = x^3 + ax + b \bmod q$, where $a, b \in \mathbb{Z}_q^*$ and $4a^3 + 27b^2 \neq 0$, is defined over the finite field $\mathbb{Z}_q^*$.

DOI: 10.54254/2755-2721/2026.34022

The TA further defines an additive cyclic group G on E with generator P, selects a random system secret key $sk_{ta} \in \mathbb{Z}_q^*$, and computes the corresponding public key $pk_{ta} = sk_{ta} \cdot P$. Secure cryptographic hash functions are specified as $H_0 : \{0,1\}^* \times G^2 \to \mathbb{Z}_q^*$, $H_1 : G \times \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : \mathbb{Z}_q^* \times G \to \{0,1\}^l$, and $H_3 : G \times \{0,1\}^* \to \{0,1\}^l$. Finally, the TA publishes the public parameters $\{q, \lambda, E_q(a,b), P, pk_{ta}, H_0, H_1, H_2, H_3\}$ and securely stores sk_{ta}.

4.2.2. Registration of physical entities and digital cousins

The registration phase ensures that PEs and DCs are cryptographically identifiable and can securely join the DCN. The process consists of six steps:

1) **PE → TA: Registration request**
   1a. The PE randomly selects $sk_{pe} \in \mathbb{Z}_q^*$ and computes $pk_{pe} = sk_{pe} \cdot P$.
   1b. The PE sends (pk_{pe}, ID_{pe}) to the TA over an offline secure channel, where ID_{pe} is PE's true identity such as license number.
2) **TA → PE: Registration response**
   2a. The TA verifies ID_{pe} uniqueness and computes $R_{pe} = r_{pe} \cdot P$, $y_{pe} = r_{pe} + sk_{ta} \cdot H_0(ID_{pe}, pk_{pe}, R_{pe}) \bmod q$ with random $r_{pe} \in \mathbb{Z}_q^*$.
   2b. The TA stores (ID_{pe}, R_{pe}) and returns (R_{pe}, y_{pe}) to the PE.
3) **PE → cloud server: DC creation request**
   3a. The PE verifies $y_{pe} \cdot P = R_{pe} + H_0(ID_{pe}, pk_{pe}, R_{pe}) \cdot pk_{ta}$.
   3b. If verification passes, PE stores (y_{pe}, sk_{pe}) and (pk_{pe}, R_{pe}) and requests DC creation.
4) **PE → DC: Authorization information**
   4a. For the new DC with identity ID_{dc}, PE computes $X_{dc} = x_{dc} \cdot P$ and $\mu_{dc} = x_{dc} + sk_{pe} \cdot H_1(X_{dc}, ID_{pe}, ID_{dc}) \bmod q$ where $x_{dc} \in \mathbb{Z}_q^*$.
   4b. PE sends $(ID_{pe}, pk_{pe}, X_{dc}, \mu_{dc})$ to DC securely.
   4c. DC forwards $(X_{dc}, \mu_{dc}, ID_{pe}, ID_{dc})$ to TA for registration.
5) **TA → DC: Registration response**
   5a. TA verifies μ_{dc} as $\mu_{dc} \cdot P = X_{dc} + H_1(X_{dc}, ID_{pe}, ID_{dc}) \cdot pk_{pe}$.
   5b. TA computes $PID_{dc} = ID_{pe} \oplus H_2(sk_{ta}, X_{dc})$ and $sk_{dc} = H_1(sk_{ta}, PID_{dc}, n_{dc})$.
   5c. TA securely delivers (PID_{dc}, sk_{dc}) to DC and stores (PID_{dc}, X_{dc}).
6) **DC ↔ PE: Long-term key negotiation**
   6a. DC computes $pk_{dc} = sk_{dc} \cdot P$.
   6b. DC and PE negotiate the long-term session key k_{ij} to secure intra-cousin data transmission.

4.2.3. Delegation and proxy key generation

To participate in DCN data sharing, a DC must acquire proxy delegation from the TA:

1) **DC → TA: Delegation request**
   1a. DC computes $\rho_{dc} = H_3(X_{dc}, PID_{dc}, TS_0)$ and encrypts $C_{dc} = Enc_{pk_{ta}}(PID_{dc}, TS_0)$.
   1b. DC sends (C_{dc}, ρ_{dc}) to TA.
2) **TA → DC: Delegation response**
   2a. TA decrypts C_{dc}, checks TS_0, and validates ρ_{dc}.
   2b. TA generates proxy warrant w_{dc}, selects $r_{dc} \in \mathbb{Z}_q^*$ to compute $R_{dc} = r_{dc} \cdot P$, $h_{dc} = H_1(R_{dc}, w_{dc})$, and $\sigma_{dc} = r_{dc} + h_{dc} \cdot sk_{ta} \bmod q$.
   2c. TA sends $(\sigma_{dc}, R_{dc}, w_{dc})$ to DC and records (w_{dc}, R_{dc}) on blockchain.
3) **DC: Proxy key computation**
   3a. DC verifies w_{dc} and σ_{dc}.
   3b. DC computes proxy private key $psk_{dc} = \sigma_{dc} + h_{dc} \cdot sk_{dc} \bmod q$ and proxy public key $ppk_{dc} = psk_{dc} \cdot P$.

DOI: 10.54254/2755-2721/2026.34022

4.2.4. Trust-based matching and candidate filtering

Requester DCs evaluate potential provider DCs based on trustworthiness and attribute relevance:

1) **Trust assessment**
   1a. Direct trust $dt_{ij} = \frac{\alpha_{ij}+1}{\alpha_{ij}+\beta_{ij}+2}$ is calculated from past interactions.
   1b. Overall trust $tv_{ij} = dt_{ij} \times qa_j$ incorporates service qualification.
   1c. Candidates with $tv_{ij} < \xi$ are discarded.
2) **Attribute matching**
   2a. Compute average Jaccard similarity js_{ij} over n interactions.
   2b. Comprehensive matching $md_{ij} = tv_{ij} \times js_{ij}$; candidates with $md_{ij} \geq \zeta$ form the final provider set $\mathcal{P}$.

4.2.5. Mutual authentication and session key establishment

Eligible DCs in $\mathcal{P}$ perform mutual authentication to establish secure session keys:

1. $DC_i \rightarrow DC_j$: Authentication Request
2. $DC_j \rightarrow DC_i$: Authentication Response
3. $DC_i \rightarrow DC_j$: Authentication Acknowledgement
4. $DC_j \rightarrow DC_i$: Secure Data Sharing

5. Performance evaluation and security analysis

This section presents the evaluation of the proposed secure access control and authentication framework for the IoT-oriented DCN. The analysis mainly considers computational cost, communication overhead, and the performance of the trust-based matching mechanism. All simulations are carried out on a standard hardware environment equipped with a 12th Gen Intel(R) Core(TM) i5-12500H 2.50 GHz processor and 16.0 GB RAM. The implementation is developed in Python 3.9, where the gmpy2 and secrets libraries are used to support efficient numerical computation, randomness-based statistical analysis, and communication overhead simulation.

5.1. Security testing and analysis

Randomness testing is used to evaluate the statistical properties of sequences generated by random number generators (RNGs) or cryptographic algorithms, based on principles of probability and statistics. In this work, the NIST Special Publication 800-22 test suite [9] is adopted to assess the randomness of binary sequences of arbitrary length produced by either hardware-based or software-based pseudorandom number generators (PRNGs). The main goal of this test suite is to detect possible non-random characteristics in a given sequence. It consists of multiple independent tests, each of which produces a p-value as output. A binary sequence is regarded as passing a specific test when $0.01 < p \leq 1$. In general, larger p-values indicate stronger randomness characteristics.

The proposed scheme is evaluated and compared with the Flexible Cross-Domain Authorization Scheme [10] and the ID-based Signature Authentication Scheme [11]. In the experimental configuration, the Advanced Encryption Standard (AES) key length for both encryption and decryption is set to 128 bits, while the element size of the cyclic group is configured as 256 bits. The Message Authentication Code (MAC) length is defined as 160 bits. In addition, the hash output size, the bit length of random numbers in $\mathbb{Z}_q^*$, and the identity length are all set to 128 bits, whereas the proxy authorization value and timestamp are set to 64 bits and 32 bits, respectively. A total of 100 independent runs are

DOI: 10.54254/2755-2721/2026.34022

conducted to compute the average results. As shown in Table 1, the obtained numerical results of the proposed secure data sharing scheme are very close to those of the baseline schemes, which indicates a comparable level of security. To further verify the robustness of the proposed design, the detailed security properties achieved by the scheme are presented in the following discussion.

Table 1: Comparison of NIST SP 800-22 Statistical Test Results

Test Items	Proposed Scheme	Flexible Scheme	ID-based Scheme
Approximate Entropy	1.000000 (PASS)	1.000000 (PASS)	1.000000 (PASS)
Block Frequency	0.182457 (PASS)	0.181674 (PASS)	0.182474 (PASS)
Cumulative Sums	0.255356 (PASS)	0.260321 (PASS)	0.264121 (PASS)
Frequency	0.325311 (PASS)	0.319324 (PASS)	0.308324 (PASS)
Linear Complexity	0.159350 (PASS)	0.153439 (PASS)	0.157439 (PASS)
Longest Run	0.266934 (PASS)	0.244201 (PASS)	0.243161 (PASS)
Non-Overlapping Template	0.176521 (PASS)	0.178822 (PASS)	0.181932 (PASS)
Random Excursions	0.190181 (PASS)	0.191319 (PASS)	0.194419 (PASS)
Random Excursions Variant	0.036165 (PASS)	0.036447 (PASS)	0.037157 (PASS)
Runs	0.233485 (PASS)	0.230981 (PASS)	0.235981 (PASS)
Serial	0.487100 (PASS)	0.501102 (PASS)	0.481002 (PASS)
Universal	0.362181 (PASS)	0.359127 (PASS)	0.349097 (PASS)

Mutual authentication: In the proposed scheme, mutual identity verification between DC_i and DC_j serves as the primary security mechanism for ensuring trustworthy data sharing. By validating the correctness of signatures s_j, u_j, and the proxy warrant w_j, DC_i can confirm that DC_j is an authorized data provider within the DCN.

$$\begin{aligned}(s_i + u_i) \cdot P &= s_i \cdot P + u_i \cdot P \\ &= (\sigma_{i2} \cdot psk_i - u_i) \cdot P + u_i \cdot P \\ &= \sigma_{i2} \cdot psk_i \cdot P \\ &= (\sigma_i + h_i \cdot sk_i) \cdot \sigma_{i2} \cdot P \\ &= (r_i + h_i \cdot sk_{ta} + h_i \cdot sk_i) \cdot \sigma_{i2} \cdot P \\ &= \sigma_{i2} \cdot R_i + h_i \cdot \sigma_{i2} \cdot pk_i + h_i \cdot \sigma_{i2} \cdot pk_{ta} \\ &= \sigma_{i2} \cdot (R_i + h_i \cdot (pk_i + pk_{ta}))\end{aligned} \tag{1}$$

Key negotiation: In the proposed protocol, DC_i and DC_j establish a pairwise transient key PTK_{ij} derived from the shared secret $SK_{ij} = a_i \cdot B_j = b_j \cdot A_i = a_i \cdot b_j \cdot P$, which is utilized to secure subsequent data transmission. Based on the hardness assumptions of ECDLP and ECDHP, adversaries cannot derive SK without obtaining the private random values a_i and b_j, even when A_i, B_j, and proxy warrants are exposed over public communication channels.

Anonymity and unlinkability: In both the matching and authentication phases, the actual identity of a DC is hidden by means of proxy warrants recorded on the blockchain, which prevents adversaries from deriving the identity of the associated PE. In addition, each communication session incorporates fresh random numbers together with timestamps, which makes it computationally infeasible for an attacker to link multiple messages to the same sender within the DCN.

Traceability: Malicious digital cousins (MDC) may attempt to disrupt normal data-sharing operations in the DCN by injecting forged or misleading information. When such abnormal behavior is identified, the TA can first extract ID_{DC} from the proxy warrant w_i stored in its local database, and then recover the corresponding physical identity using $ID_{PE} = ID_{DC} \oplus H_2(sk_{TA}, X_{DC})$. In this way, the real identities of malicious participants can be exposed, and the associated (w_i, R_i) credentials can be revoked on the blockchain.

DOI: 10.54254/2755-2721/2026.34022

Protocol attack resistance: The proposed protocol introduces fresh session parameters K_i and timestamps TS in every communication round, which effectively prevents replay attacks based on previously captured messages. Moreover, cryptographic primitives such as hash functions, message authentication codes, and digital signatures are jointly employed to resist impersonation attacks and man-in-the-middle attacks. In addition, batch verification and aggregate response mechanisms are adopted to reduce computational overhead and to enhance robustness against Denial-of-Service attacks targeting DCs.

5.2. Communication overhead evaluation

To thoroughly assess the communication efficiency of the proposed scheme, a comparative evaluation is carried out focusing on the data transmission overhead during the authentication and key establishment phases. The study includes the proposed DC-based scheme, the corresponding DT-based approach, as well as representative schemes from [10] and [11]. For consistency and fairness, all schemes are evaluated under the same security parameter settings. In particular, the total size of transmitted message elements over a complete interaction is measured, and the number of data requester nodes is used as a scaling factor to observe how communication overhead evolves with network growth.

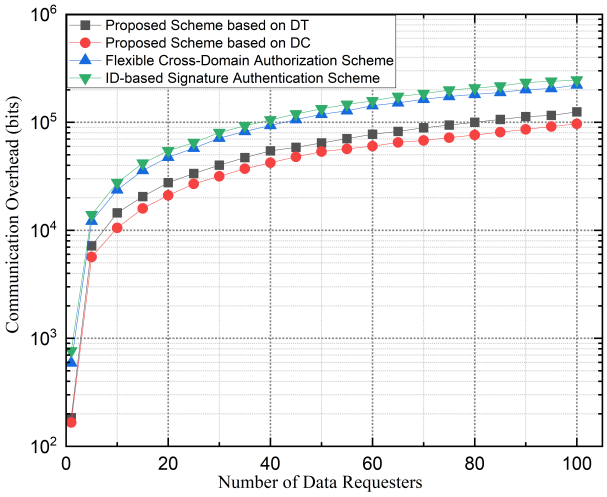

Figure 2: The comparison of communication cost overhead.

As shown in Figure 2, the proposed scheme introduces a communication overhead of approximately $97{,}521$ bits per full interaction, which is noticeably lower than that of the DT-based scheme ($126{,}825$ bits). By comparison, the Flexible Cross-Domain Authorization Scheme [10] and the ID-based Signature Authentication Scheme [11] incur much higher overheads of $232{,}233$ bits and $252{,}525$ bits, respectively. Quantitatively, the proposed scheme achieves approximately a 23.1% reduction in communication overhead compared with the DT-based approach. When compared with [10] and [11], the reductions further increase to more than 58% and 61%, respectively. Moreover, as the number of participating nodes continues to grow, the communication overhead of the proposed scheme increases at a relatively slower pace, indicating better scalability in large-scale IoT environments.

DOI: 10.54254/2755-2721/2026.34022

The improvement in communication efficiency mainly comes from two aspects: protocol-level optimization and the inherent advantages of the DC paradigm itself. Batch authentication lets a requester verify multiple responders all in one go, so you don't have to go through those tedious back-and-forth authentication steps that just waste everyone's time. Next, in matching stage, we use pre-filtering strategy. That prevents unnecessary authentication and key agreement processes, cutting redundant communication right at source. Also, we use proxy credential mechanism, which replaces repeated identity exchanges with lightweight credential representations, this cuts down on communication costs while still preserving anonymity and traceability. Then, we adopted a compact message structure to eliminate unnecessary parameter transmission. It directly lowers communication overhead in each interaction round. And an aggregation-based confirmation mechanism may further cut data transmission frequency by merging multiple nodes' feedback into one message.

Beyond these protocol-level optimizations, the bigger reason for the communication reduction lies in how the DC and DT paradigms differ, especially in modeling strategy and interaction granularity. Traditional DT systems usually rely on high-fidelity digital replicas of physical entities. They need continuous synchronization of device states, environmental information, and various contextual data to keep the physical and virtual spaces tightly consistent. This process inevitably leads to frequent data exchanges during updates, authentication, and interactions, which adds up to considerable communication overhead. In contrast, the DC paradigm takes a task-oriented abstraction approach. Virtual entities are built based on semantic and functional similarity rather than strict one-to-one replication. As a result, only task-relevant key features need to be exchanged. Full-state synchronization is avoided, which cuts down both the amount of data transmitted and how often updates happen [12].

DC-based systems limit communication to only necessary steps—things like authentication and task-level matching—since they're built around loosely coupled semantic interactions, not the nonstop two-way sync that DT systems use. DT systems need that constant back-and-forth to maintain real-time consistency between physical and virtual environments, every part of their interaction mechanism revolves around this continuous synchronization. This looser setup for DC systems can avoid heavy communication load that frequent synchronization brings. Also, the DC paradigm relaxes strict modeling accuracy requirements and uses lightweight representation strategy, so it may offer better scalability and adaptability across different scenarios. It also cuts down communication overhead tied to identity binding, keeping states synchronized, and environmental mapping.

6. Conclusions

Traditional DT-based systems have two major flaws, they're stuck with a heavy communication burden and really bad scalability in IoT field. In this paper, we came up with a secure, lightweight data-sharing structure for DCNs to fix these issues. It focuses at using the inherent characteristics of DC paradigm to tackle both flaws, so it works way better than those old DT-based setups. To make sure DCs can interact safely and reliably, we merged three components into a single unified architecture. Proxy-based authorization manages access to shared data, trust-oriented matching connects DCs that have mutual trust, and mutual authentication. What is particularly significant is the trust-based filtering system, a preliminary screening layer which eliminates unreliable participants prior to data exchange. Consequently, the data quality is enhanced and thereby improves the accuracy of the decisions. Moreover, the proxy signature-based protocol ensures the confidentiality and integrity of the information and resists the network attacks frequently appearing in the open communication environment. From both the theoretical investigation and the simulation outcomes, the scheme shows excellent security effects along with relatively low communication burden. Compared with the existing DT-based methods, it demonstrates better scalability and higher operational efficiency, making it more appropriate for the large-scale and dynamic IoT situations. In the future, we will investigate the dynamic attribute revocation, adaptive trust management and practical implementation in the heterogeneous edge-cloud environments.

DOI: 10.54254/2755-2721/2026.34022

References

[1] S. Datta, "Digital Twin, Didymos, Meets Digital Cousin, Didymium. From Paradox to Paradigm or a Paradoxical Paradigm?" Working paper, MIT DSpace, Nov. 2024.

[2] B. Li *et al.*, "Delay-Aware Digital Twin Synchronization in Mobile Edge Networks With Semantic Communications," *IEEE Trans. Veh. Technol.*, vol. 74, no. 7, p. 10974–10983, Jul. 2025.

[3] S. D. Okegbile, J. Cai, J. Chen, and C. Yi, "A reputation-enhanced shard-based byzantine fault-tolerant scheme for secure data sharing in zero trust human digital twin systems," *IEEE Internet Things J.*, vol. 11, no. 12, p. 22726–22741, Jun. 2024.

[4] J. Li *et al.*, "Federated Learning for Digital Twin Applications: A Privacy-Preserving and Low-Latency Approach," *PeerJ Comput. Sci.*, vol. 11, p. e2877, Aug. 2025.

[5] S. B, "Neuro-Symbolic Federated Learning with Quantum-Safe Cognitive Twins for Personality-Aware Human-AI Collaboration," Preprints.org, Jan. 2026.

[6] T. Dai *et al.*, "Automated Creation of Digital Cousins for Robust Policy Learning," Oct. 2024. [Online]. Available: https://arxiv.org/abs/2410.07408

[7] V. Kumar *et al.*, "Design of Secure and Efficient Framework for Vehicular Digital Twin Networks Using ECC," *IEEE Access*, vol. 12, p. 194352–194366, 2024.

[8] G. Li *et al.*, "A Secure and Lightweight Data Sharing Scheme in Vehicular Digital Twin Network," *Peer-to-Peer Networking Appl.*, vol. 18, no. 1, Dec. 2024.

[9] H.-J. Park *et al.*, "Parameter Optimization Method for Improving the Reliability of the NIST SP 800-22 Test Suite in TRNG Verification," *IEEE Access*, vol. 14, p. 31159–31173, Feb. 2026.

[10] J. Sun *et al.*, "Secure Data Sharing With Flexible Cross-Domain Authorization in Autonomous Vehicle Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, p. 7527–7540, Jul. 2023.

[11] Q. Fan *et al.*, "A Secure and Efficient Authentication and Data Sharing Scheme for Internet of Things based on Blockchain," *J. Syst. Archit.*, vol. 117, p. 102112, Aug. 2021.

[12] Z. Zeng *et al.*, "A Novel Digital Cousin based Intelligent Resource Allocation Scheme for Secure Low-Altitude UAV Spectrum Sharing in AMN," *Chin. J. Aeronaut.*, vol. 39, no. 5, p. 103983, May 2026.