Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CPBS: A Collaborative Proxy Blind Signature Scheme with Threshold Authorization for Privacy-Preserving Medical IoT

Jinnan Li
Naval University of Engineering
Qing Ye
Naval University of Engineering
Yi Yang
yangyi.ip@whu.edu.cn

Naval University of Engineering
Qian Zhou
Naval University of Engineering
Zhimin Yuan
Naval University of Engineering
Shiwei Huo
Information Support Force Engineering University

Research Article

Keywords: Medical Internet of Things, Proxy Blind Signature, Threshold Authorization, Secret Sharing, Blockchain Auditing

Posted Date: June 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10018166/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CPBS: A Collaborative Proxy Blind Signature Scheme with Threshold Authorization for Privacy-Preserving Medical IoT

Jinnan Li[1], Qing Ye[1], Yi Yang[1*], Qian Zhou[1], Zhimin Yuan[1], Shiwei Huo[2]

[1*]Department of Information Security, Naval University of Engineering, Wuhan, 430030, Hubei, China.
[2]Information Support Force Engineering University, Wuhan, 430030, Hubei, China.

*Corresponding author(s). E-mail(s): yangyi.ip@whu.edu.cn;
Contributing authors: M25183905@nue.edu.cn; dis_yeqing@sohu.com;
D24381005@nue.edu.cn; 1209051035@nue.edu.cn; shiweihuo@163.com;

Abstract

Secure data collection in Peer-to-Peer Medical Internet of Things (MIoT) networks faces a fundamental tension between resource-limited edge devices and high-strength security requirements. Existing solutions rely on a single centralized proxy to authorize signatures, creating a single point of trust and lacking the multi-party oversight essential for medical compliance. To address this, we propose CPBS, a Collaborative Proxy Blind Signature scheme that replaces centralized authorization with decentralized, peer-to-peer collaboration. Specifically, patients, physicians, and ethics committees operate as independent peer nodes that jointly authorize a ward monitoring gateway via a (t, n)-threshold mechanism, distributing trust across multiple parties without any centralized authority. Core cryptographic operations are offloaded to edge gateways, while resource-constrained medical devices perform only hashing and lightweight encryption. The blind signature ensures that gateways gain no knowledge of raw data, and on-chain record storage provides end-to-end auditability. Formal analysis verifies correctness, blindness, and existential unforgeability under the random oracle model. Performance results show that CPBS reduces terminal computational and communication overhead by up to 52.3% and 76.8%, respectively, compared with existing proxy signature schemes, making it particularly suitable for decentralized, multi-source health monitoring in P2P MIoT environments.

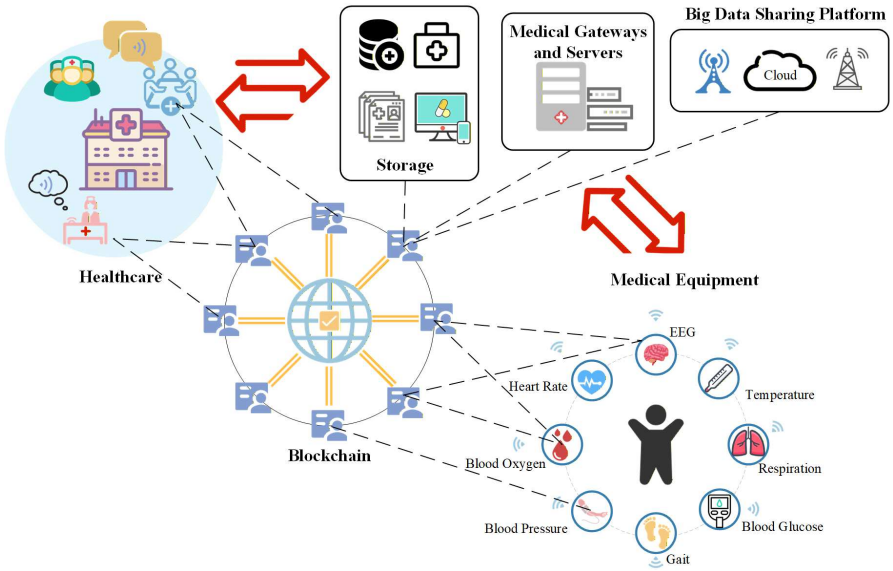


Fig. 1: Healthcare Internet of Things model

Keywords: Medical Internet of Things, Proxy Blind Signature, Threshold Authorization, Secret Sharing, Blockchain Auditing

1 Introduction

With the deep integration and rapid deployment of fifth-generation mobile communications (5G), the Internet of Things (IoT), and artificial intelligence (AI) technologies, the field of smart healthcare is undergoing profound transformation [1]. The widespread adoption of MIoT and wearable medical devices has enabled continuous, real-time monitoring and collection of patients' physiological signs, generating massive amounts of highly sensitive physiological data [2, 3]. This marks a shift in healthcare data management, moving away from data silos toward a timely, efficient, and multiparty collaborative paradigm. Under this model, a large number of medical sensing devices serve as data sources, transmitting collected physiological data through proxy nodes such as edge gateways, and ultimately gathering at hospital big data platforms [4, 5]. The core significance of this model lies in shifting data collection, preprocessing, and storage from reliance on specific components within a single institution to a distributed, data-driven, multi-party collaborative effort. This transformation not only expands the temporal and spatial boundaries of health monitoring, supporting telemedicine and personalized health management, but also serves as the core data driver enabling the development of precision medicine.

1.1 Motivation

It is important to highlight that after transmitting encrypted or raw data to proxy nodes and the cloud, sensing devices typically delete local data promptly to save their limited storage space and computational resources. This results in data subjects, including patients, losing direct physical control over their data. Once proxy nodes or cloud service providers maliciously alter or delete data, or data is destroyed due to system failures, it will cause immeasurable harm to patient privacy, medical diagnosis, and even life safety [6–8]. In fact, a surge in security incidents involving the leakage and tampering of health data in recent years has highlighted the urgent need to ensure data integrity in this field [9]. Therefore, for the MIoT, it is crucial to design an authentication and auditing scheme capable of effectively verifying data integrity throughout the entire process. From a technical perspective, a typical method for ensuring data integrity involves sensing devices generating digital signatures or authentication tags for data files during the data collection or transmission phase. In the subsequent verification phase, data users such as hospital big data platforms verify these signatures to determine whether the data has remained intact and authentic since the signature was generated. In fact, many healthcare data security solutions adopt this approach for source authentication [10–12].As illustrated in Fig.1, our proposed healthcare MIoT model is designed to enable secure multi-party data collection by allowing patients, physicians, and ethics committees to jointly authorize the ward gateway to perform signature generation without exposing their private keys or granting any single party direct access to raw physiological data.

1.2 Challenge

Although the importance of healthcare data security is self-evident, existing digital signature schemes deployed in MIoT environments still face three core challenges.

1. **Excessive computational burden on edge devices:** MIoT sensing devices are typically miniaturized and battery-powered, operating under strict constraints on computational power, storage capacity, and energy consumption. This makes it difficult to directly deploy traditional digital signature schemes that involve complex cryptographic operations [13].
2. **High risk of device key compromise:** Frequent use of built-in private keys for signing operations significantly increases the risk of these keys being stolen through side-channel attacks and other methods in resource-constrained environments; once a key is compromised, an attacker can forge arbitrary medical data, posing a serious threat to patient safety [14].
3. **Excessive power of a single agent and lack of collaborative authorization:** Entrusting a single agent node to sign data offloads computation from the terminal but introduces a single point of authority. This concentration of power may lead to unauthorized data manipulation. Furthermore, it conflicts with the multi-party authorization mandates of medical scenarios, which often require joint consent from patients, physicians, and ethics committees.

Existing research has revealed a technical gap between terminal computing efficiency, key security, and multi-party collaborative supervision. Notably, while lightweight solutions based on proxy signatures [15] can shift core computations to edge nodes, they fail to prevent proxy nodes from tampering with original data, thereby unable to guarantee data integrity from the source and losing their trusted foundation. Solutions based on periodic key updates [16], while capable of reducing the scope of key leakage, require devices to perform additional key generation computations. Furthermore, the re-registration and widespread distribution of public keys impose significant coordination and management overheads on networks that are already resource-constrained. Solutions based on multi-party collaborative signatures [17, 18], while achieving decentralized checks and balances on signing authority, are mostly designed for general-purpose computing environments. In resource-constrained MIoT scenarios, they face challenges including high computational overhead and numerous communication rounds, making direct deployment difficult. At the same time, existing improved solutions often address individual issues in isolation, failing to systematically resolve these comprehensive problems.

1.3 Contribution

For secure data collection in MIoT scenarios, there is currently a lack of comprehensive solutions that combine lightweight terminal operations, threshold-based collaborative authorization, and end-to-end privacy protection. To address this issue, this paper proposes CPBS, a collaborative proxy blind signature scheme for secure MIoT data collection. The main contributions are summarized as follows:

- First, we design a (t, n)-threshold collaborative proxy blind signature protocol. Multiple original signers jointly generate authorization shares via secret sharing; the proxy signer combines at least t shares to sign. Blind interactions ensure the proxy signer gains no knowledge of the plaintext, achieving decentralized authority and data privacy.
- Second, based on this protocol, we develop a CPBS scheme for MIoT. Patients, physicians, and ethics committees co-authorize a gateway as proxy signer; medical devices perform only hashing and lightweight encryption. The scheme supports dynamic revocation, threshold adjustment, and node onboarding without changing the public verification key. Authorization records are stored on a medical consortium blockchain for end-to-end auditing, and zero-knowledge proofs defend against malicious authorization and collusion.
- Formal security analysis shows that CPBS satisfies correctness, blindness, and existential unforgeability under the random oracle model. Experiments indicate that compared with traditional direct signing and proxy signing, CPBS reduces terminal computation by up to 52.3% and communication by up to 76.8%.

1.4 Organization

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 introduces the preliminary concepts. Section 4 presents the system and threat

model. Section 5 provides a detailed description of the proposed cooperative proxy blind signature protocol and its correctness proof. Section 6 elaborates on the CPBS scheme. Section 7 presents the security analysis. Section 8 reports the performance evaluation results, and Section 9 concludes the paper.

2 RELATED WORKS

This section reviews and analyzes existing research, with a focus on the application of proxy signatures and blind signature techniques in privacy protection, as well as collaborative signature schemes related to threshold cryptography.

2.1 Proxy Signature

To alleviate the signing burden on resource-constrained devices, Mambo et al. [19] introduced the concept of proxy signature in 1996, enabling original signers to delegate signing authority. This concept has since been extended to IoT and medical scenarios. By combining with other cryptographic primitives, Lin et al. [20] proposed the first proxy blind signature scheme in 2000, while Lin et al. [21] constructed a linear homomorphic proxy signature scheme in 2018. More recently, Wang and Wu [22] enhanced the security model and introduced a blockchain-based variant, and Zhang et al. [23] designed a lightweight proxy-oriented integrity checking mechanism for cloud-assisted transportation systems. Despite these advances, existing proxy signature schemes suffer from a critical problem: an authorized proxy signer can unilaterally tamper with original data and generate valid signatures for the altered version. In MIoT scenarios, this weakness poses severe risks. A compromised or malicious proxy could forge patient data, leading to misdiagnosis or medical accidents. Xue et al. [24] further highlighted the issue of centralized authority abuse in proxy signature models, exposing the inherent security flaws of single-proxy authorization.

In contrast to prior work, we introduce a collaborative authorization scheme involving multiple original signers. This approach distributes signing authority across multiple parties, preventing any single node from independently completing authorization, thereby mitigating the risk of proxy power abuse.

2.2 Privacy-Preserving Blind Signature Technique

To protect message content from the signer, Chaum [25] introduced blind signatures in 1982, enabling signature generation without exposing the plaintext. This property is naturally suited to preventing proxy or auditor access to sensitive physiological data in medical scenarios. Subsequent works have extended this idea. Bhavin et al. [26] proposed a quantum blind signature scheme targeting MIoT security, offering advantages in resource and transmission overhead. Gao et al. [27] developed a vehicular-network blind signature that reduces information leakage via message blinding. Yang et al. [28] designed a blind signature-based authorization mechanism for cloud-assisted private set intersection, ensuring semi-trusted verifiers learn nothing about the data. Samal et al. [29] presented a lightweight ECC-based blind signature authentication protocol

for resource-constrained devices, yet its computational burden remains considerable for continuous monitoring.

Despite these advances, existing schemes exhibit clear limitations. Some focus exclusively on privacy protection without addressing the delegation of signing authority required in medical contexts. Others struggle to balance privacy, security, and computational-communication efficiency.

To address these gaps, our work integrates blind signatures with proxy signatures. By designing a lightweight blinding protocol, we enable the proxy signer to generate valid signatures without accessing raw medical data, while imposing only lightweight operations on sensing devices. This achieves a unified balance between data privacy protection and terminal efficiency.

2.3 Multi-party Collaborative Signature Technique

To prevent signature forgery from key leakage and enable distributed oversight, threshold cryptography and multi-party signing have been developed. Lindell [30] proposed the first two-party Elliptic Curve Digital Signature Algorithm (ECDSA) signing protocol in 2017 using zero-knowledge proofs. He et al. [31] subsequently proposed a two-party distributed signing protocol for the IEEE P1363 standard, enabling joint signature generation without full key reconstruction. Mu et al. [32] designed a secure two-party SM9-DSA signing protocol by exploiting Paillier's additive homomorphism and zero-knowledge proofs. Liu et al. [33] proposed an asymmetric secure multi-party signing protocol for IEEE P1363, provably secure against static malicious adversaries. For smart home systems, Liu et al. [34] introduced a two-party collaborative SM9-based signing scheme with distributed key generation. In the MIoT domain, Xie et al. [35] presented LiteCrypt, a lightweight threshold signature protocol tailored for resource-constrained devices, significantly reducing computation time.

Despite these advances, existing collaborative signing schemes are designed for stable two-party or fixed-group environments. They lack an integrated mechanism combining lightweight terminal operations, threshold-based collaborative authorization, end-to-end privacy, and tamper-proof auditing, all essential for complex MIoT scenarios. To address these gaps, we propose CPBS scheme tailored for secure data collection in MIoT environments.

3 PRELIMINARIES

In this section, we first give a list of notations as shown in Table 1 and then give some preliminaries used in this paper.

3.1 Bilinear pairing

Let G be an additive cyclic group of prime order p, and let G_T be a multiplicative cyclic group of prime order p. Let g be a generator of G. A bilinear pairing is a map $e : G \times G \to G_T$ that satisfies the following three properties:

Table 1: Notations

Symbol	Meaning
q	Prime.
G	Additive cyclic groups (order q).
G_T	Multiplicative cyclic group (order q).
H_1	Hash: $\{0,1\}^* \times \{0,1\}^* \times G \to \mathbb{Z}_q^*$.
H_2	Hash: $G_T \times G_T \to \mathbb{Z}_q^*$.
P_1, P_2	Generator of G.
F_1	Hash: $\{0,1\}^{l_2} \to \{0,1\}^{l_1}$.
F_2	Hash: $\{0,1\}^{l_1} \to \{0,1\}^{l_2}$.
e	Bilinear pairing: $G \times G \to G_T$.
l_1, l_2	Positive integers: $l_1 + l_2 = \|q\|$.
$S_P, S_D, S_{AP}, S_G, S_C, S_{BDP}$	Private keys.
$P_P, P_D, P_{AP}, P_G, P_C, P_{BDP}$	Public keys.
M_w	Authorization policy.
Q_w	Authorization ID.
R	Proxy commitment.
$\|G\|$	Bit length of G elements.
$\|q\|$	Bit length of $\mathbb{Z}_q$ elements.
$\|M_w\|$	Bit length of M_w.
$\|proof\|$	ZK proof bit length.
$\|data\|$	Single data block bit length.
n	Total data blocks.
t	Authorization threshold.
r	Challenge blocks.
T_{mul}^G	Scalar mul. time on G.
T_{pair}	Bilinear pairing time.
T_{exp}	Exponentiation time.
T_H	Hash time.
T_{inv}	Modular inverse time.
T_{sym}	Symmetric enc/dec time.
T_{pke}	Public key enc/dec time.
T_{zkp}	ZK proof gen. time.
T_{zkv}	ZK proof ver. time.
T_{muz}	Mul. time on $\mathbb{Z}_q$.
T_{enc}	Paillier enc. time.

1. **Bilinearity**: For all $a, b \in \mathbb{Z}_p$ and all $u, v \in G$,

$$e(u^a, v^b) = e(u, v)^{ab}.$$

2. **Non-degeneracy**: $e(g, g) \neq \mathbf{1}_{G_T}$, where $\mathbf{1}_{G_T}$ denotes the identity element of group G_T.
3. **Computability**: There exists an efficient algorithm to compute $e(u, v)$ for all $u, v \in G$.

A map e satisfying all three properties is called a bilinear pairing.

3.2 Threshold signature

Threshold digital signature is an extension of standard signature schemes, where the private key is split into n shares. Each participant P_i holds a corresponding private key share. A valid signature can only be generated if at least t participants participate in the signing process, which is denoted as a (t, n)-threshold signature scheme.

Threshold signature schemes are built on a cyclic group G of prime order q generated by an element g. They employ two hash functions: H maps arbitrary strings to elements in $\mathbb{Z}_q$, and H' maps elements in G to elements in $\mathbb{Z}_q$. The workflow of a threshold signature scheme is briefly described as follows:

1. **Key Generation**: Each participant selects a private key $x_i \in \mathbb{Z}_q^*$, computes the corresponding public key share $T_i = g^{x_i}$, and broadcasts T_i. The aggregated public key is finally generated as $T = \prod_{i=1}^{n} T_i$.
2. **Signing**: Let $P_1, P_2, \ldots, P_t$ be the t participants involved in signing. Each participant samples a random number $k_i \in \mathbb{Z}_q^*$ and broadcasts it. Then all participants compute $k = \sum_{i=1}^{t} k_i$ and $K = g^k$. Each participant further computes $r = H'(K)$, $m = H(M)$, and $s_i = k_i m + k \cdot x_i \cdot r$, where M is the message to be signed, and broadcasts s_i. All participants aggregate the partial signatures as $s = \sum_{i=1}^{t} s_i = k(m + xr)$, and the final signature is $\Sigma = (r, s)$.
3. **Verification**: Any verifier parses the signature (r, s) and computes $R' = g^{ms^{-1} \bmod q} T^{rs^{-1} \bmod q} \in G$. The verifier checks whether $H'(R') = r$ holds. If the verification passes, return 1; otherwise, return $\perp$.

3.3 Two-party collaborative signature

Two-party collaborative signature consists of four algorithms: 1) Initialization; 2) Key Distribution; 3) Collaborative Signing; 4) Verification. The detailed description is as follows:

1. **Initialization** $(1^\lambda) \to (pk, msk)$: This algorithm generates a public key pk and a master private key msk based on the security parameter.
2. **Key Distribution** $(msk) \to (psk_1, psk_2)$: This algorithm outputs the corresponding partial private keys psk_1 and psk_2 to Alice and Bob respectively.
3. **Collaborative Signing** $(M, psk_1, psk_2) \to (\sigma)$: In this algorithm, Alice initiates a signing request and interacts with Bob to generate a signature σ for message M based on their respective partial private keys.
4. **Verification** $(M, \sigma, pk) \to \{1/0\}$: This algorithm is used to verify the validity of the received (M, σ). It outputs 1 if the verification is correct, and 0 otherwise.

4 PROBLEM STATEMENT

This section introduces the system model, describing the involved entities, their functional roles, and the overall collaboration process. It then describes the threat model, clarifying the security assumptions of our scheme and the limits of an attacker's capabilities.

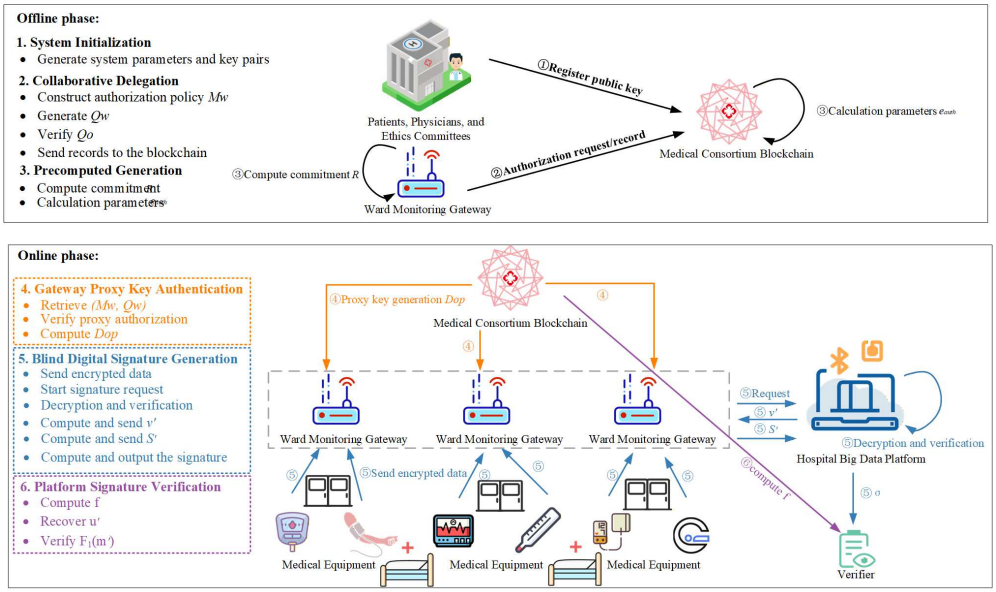


Fig. 2: The system architecture of scheme

4.1 System Model

We propose a collaborative proxy blind signature model for secure MIoT data collection, consisting of five core entities: 1) Original Signers Set; 2) MIoT Sensing Devices; 3) Ward Monitoring Gateway; 4) Hospital Big Data Platform; 5) Medical Consortium Blockchain. The system architecture of the scheme is shown in Fig. 2. Their roles are described as follows:

1. Original Signers Set: Patients, attending physicians, and hospital ethics committees jointly authorize the ward monitoring gateway via a threshold mechanism. They hold private key shares and store authorization records on the medical consortium blockchain.
2. MIoT Sensing Devices: Wearable sensors and other medical devices collect raw physiological data under strict resource constraints. They perform only hashing and lightweight encryption before sending the blinded data to the ward monitoring gateway.
3. Ward Monitoring Gateway: A resource-rich edge node that aggregates at least t authorization shares to generate a proxy signing key, and computes the partial signature on the blinded data without learning the plaintext.
4. Hospital Big Data Platform: Serving as the final data user and verifier, the platform decrypts the data, verifies its consistency, and interacts with the gateway to generate full signatures. It can validate the signature validity at any time.

5. Medical Consortium Blockchain: By leveraging the inherent immutability and transparency of distributed ledgers, the blockchain stores public keys, authorization records, and timestamps. This enables traceability and public verification without reliance on trusted third parties.

The core goal is to enable collaborative authorization by medical entities for ward monitoring gateway signing, ensuring data privacy, low terminal burden, and signature verifiability, unforgeability and traceability.

The interaction follows an offline preprocessing and online lightweight interaction architecture. MIoT sensing devices communicate only with the ward monitoring gateway. The ward monitoring gateway interacts with the hospital big data platform and serves as the sole intermediary for transacting with the medical consortium blockchain. The original signers set only participates in the authorization phase.

Formally, the CPBS scheme Π takes security parameter k, threshold t, and participant number n as input, and outputs a valid proxy blind signature σ for message m. It runs in two phases:

1. **Offline Phase**: Sequentially completes System Setup, Participant Key Registration, Collaborative Delegation, and the offline part of Gateway Pre-computation, without requiring real-time online participation of sensing devices.
2. **Online Phase**: Sequentially completes Gateway Proxy Key Retrieval, Blind Data Signing, and Hospital Platform Signature Verification. Sensing devices only perform lightweight data processing operations in this phase.

Π must ensure: 1) The ward monitoring gateway cannot access raw data plaintext; 2) Collusion of fewer than t original signers cannot generate valid authorization; 3) Any legitimate verifier can independently verify signature validity via public information on the medical consortium blockchain.

4.2 Threat Model

We consider two types of adversaries with different capabilities, corresponding to the two main security properties we aim to prove.

Blindness.

For blindness, we assume the proxy signer is *semi-honest*. It follows the protocol specification exactly, but may try to infer additional information from the messages it receives. This is the standard model for proving blindness in blind signature schemes, as it captures the fact that a honest-but-curious signer cannot link a signature to a specific signing session.

Unforgeability and Authorization.

For unforgeability, we consider a stronger, *malicious* adversary. Such an adversary can arbitrarily deviate from the protocol, forge signatures, corrupt up to $t-1$ original signers, or collude with the proxy signer. This is necessary to prove that no coalition

of fewer than t parties can produce a valid signature. Our proofs for unforgeability are given in the malicious model, relying on the q-SDH assumption.

External adversaries.

Adversaries that do not hold any private key are also considered malicious; they can attempt to forge signatures or break blindness by modifying messages. Their capabilities are covered by the unforgeability and blindness games, where the adversary acts as a man-in-the-middle.

Blockchain Security Assumptions.

Our scheme relies on the underlying medical consortium blockchain for decentralized trust. We assume it provides the following properties:

- **Immutability:** Once a transaction is confirmed and a sufficient number of blocks have been appended, it cannot be altered or removed by any adversary. This is guaranteed by the consensus protocol under the assumption that at most f out of $3f + 1$ nodes are faulty.
- **Traceability:** Every valid authorization record can be retrieved by its policy identifier M_w, and the blockchain provides a verifiable proof of inclusion.

These properties are taken as fundamental security assumptions of the system infrastructure and are not proven within our cryptographic framework.

We stress that the semi-honest model for blindness is standard and sufficient, because blind signatures are typically defined with an honest-but-curious signer; if the signer is fully malicious, zero-knowledge proofs or additional mechanisms would be required to force correct behavior. Our future work includes extending the blindness proof to the malicious setting.

5 THRESHOLD COOPERATIVE PROXY BLIND SIGNATURE PROTOCOL

In this section, we present the construction method for the cooperative proxy blind signature protocol based on (t, n) thresholds that we have designed, and analyze its correctness.

5.1 Overview

The (t, n)-threshold collaborative proxy blind signature protocol addresses the core tension between multi-party authorization, data privacy, and lightweight signing in distributed environments. Three entity types are involved: original signers holding private keys, a proxy signer authorized to sign, and users requesting signatures. The protocol enables the proxy signer to produce signatures on behalf of multiple original signers without learning the message content, while satisfying threshold authorization, blind-signature privacy, and unforgeability.

The execution proceeds as follows. First, system parameters are generated and each party establishes its key pair. Next, in the collaborative authorization phase, original signers jointly issue authorization shares via threshold secret sharing; the proxy signer aggregates at least t shares into an authorization token and verifies its validity. During blind signing, the user blinds the message, the proxy signer returns a partial signature, and the user unblinds it to obtain the final signature. Finally, any entity holding the public parameters can verify the signature. By integrating threshold authorization with blind signatures, the protocol achieves multi-party collaborative control and lightweight online signing while preserving data privacy.

5.2 Construction

The detailed structure of our proposed (t, n)-threshold collaborative proxy blind signature protocol is as follows:

- **System Initialization**: All participants complete system parameter and key generation in this phase. First, input the security parameter k, generate a bilinear group system (G, G_T) of prime order q and a bilinear map $e : G \times G \to G_T$, and select generators $P_1, P_2 \in G$. Then define cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : G_T \to \{0,1\}^{l_1+l_2}$, and message encoding functions $F_1 : \{0,1\}^{l_2} \to \{0,1\}^{l_1}$, $F_2 : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$. Output the system public parameters $Params = (G, G_T, P_1, P_2, e, q, H_1, H_2, F_1, F_2, l_1, l_2, t, n)$. Each participant independently generates a key pair: original signer O_i selects private key $sk_{o_i} \xleftarrow{R} \mathbb{Z}_q^*$ and computes public key $pk_{o_i} = sk_{o_i} \cdot P_2$; proxy signer P selects private key $sk_p \xleftarrow{R} \mathbb{Z}_q^*$ and computes public key $pk_p = sk_p \cdot P_2$; signature requester U selects private key $sk_u \xleftarrow{R} \mathbb{Z}_q^*$ and computes public key $pk_u = sk_u \cdot P_2$. All public keys are made public, and private keys are kept secret by each participant.
- **Joint Authorization**:To authorize the proxy signer P for a specific authorization policy M_w, the n original signers first let $w = H_1(M_w, ID_D, Q_D, ID_P, Q_P)$. They then execute a distributed key generation (DKG) protocol to generate a secret $\alpha = (s_o + w)^{-1}$ and distribute shares α_i via Shamir's (t, n)-threshold secret sharing, where s_o is the aggregated private key of the original signers. After the DKG, each signer O_i computes its authorization share $Q_{w_i} = \alpha_i P_1$ and sends it to P. P collects at least t valid shares, computes Lagrange coefficients λ_i, and aggregates $Q_w = \sum_{i \in S} \lambda_i Q_{w_i}$. It also computes $Q_o = \sum_{i \in S} \lambda_i Q_{o_i}$ (where $Q_{o_i} = s_{o_i} P_2$). Finally, P verifies $e(Q_w, Q_o + wP_2) = e(P_1, P_2)$. If the equality holds, P stores (M_w, Q_w, Q_o, w) and records the authorization on the blockchain.
- **Proxy Blind Signature Generation**: The signature requester and the proxy signer interact to generate a blind signature in this phase. The proxy signer P first selects a random number $r \xleftarrow{R} \mathbb{Z}_q^*$, computes the commitment value $R = r \cdot P_1$, and generates the proxy signing key $D_{op} = sk_p \cdot Q_w$ using its own private key and the authorization point. The signature requester U selects random blinding factors $\alpha, \beta \xleftarrow{R} \mathbb{Z}_q^*$, precomputes the authorization verification parameter $e_{auth} = e(P_1, Q_o + w \cdot P_2)$. Then encode the original message m as

$u = F_1(m) \parallel [F_2(F_1(m)) \oplus m]$ and perform blinding processing: $x = e(\alpha \cdot R, pk_p)$, $y = (e_{auth} \cdot e(P_1, pk_p))^{\beta}$, $h = H_2(x \cdot y)$, $v = h \oplus u$, $v' = \alpha^{-1}(v + \beta) \bmod q$. U sends the blinded message v' to P. P computes the partial signature $S' = (r + v') \cdot D_{op}$ and returns it to U. U performs unblinding operation to obtain the final signature $S = \alpha \cdot S' + \beta \cdot P_1$, and outputs the complete signature $\sigma = (S, v, M_w)$.

- **Signature Verification**: Any verifier can independently verify the validity of the signature in this final phase. After receiving the signature and message, the verifier V first computes the verification parameter $f = e(S, Q_o + w \cdot P_2) \cdot e(P_1, pk_p)^{-v}$. Then recover the encoded message $h = H_2(f)$, $u' = v \oplus h$. Split u' into two parts $u' = A \parallel B$, and decode to recover the original message $m' = F_2(A) \oplus B$. Finally, verify: if $F_1(m') = A$, accept the signature; otherwise, reject it.

5.3 Correctness Proof

We prove the correctness of the proposed (t, n)-threshold collaborative proxy blind signature protocol. A signature σ passes verification if and only if all participants honestly follow the protocol steps.

Case 1: Correctness of Joint Authorization. By the property of Shamir secret sharing, for any set S of size t, $\sum_{i \in S} \lambda_i \alpha_i = \alpha$. Thus $Q_w = \sum_{i \in S} \lambda_i (\alpha_i P_1) = \alpha P_1$. Also $Q_o = \sum \lambda_i (s_{o_i} P_2) = s_o P_2$. Since the DKG ensures $\alpha = (s_o + w)^{-1}$, we have $e(Q_w, Q_o + wP_2) = e(\alpha P_1, (s_o + w)P_2) = e(P_1, P_2)^{\alpha(s_o + w)} = e(P_1, P_2)$. Hence the verification holds.

Case 2: Correctness of Proxy Blind Signature. Let $S' = (r + v')D_{op}$ with $D_{op} = sk_p Q_w$. After unblinding, $S = \alpha S' + \beta P_1 = \alpha sk_p Q_w (r + v') + \beta P_1$. Substitute $v' = \alpha^{-1}(v + \beta)$ to obtain $S = \alpha sk_p Q_w r + sk_p Q_w (v + \beta) + \beta P_1$. Compute the verification value $f = e(S, Q_o + wP_2) \cdot e(P_1, Q_p)^{-v}$. Insert S and use bilinearity: $e(S, Q_o + wP_2) = e(\alpha sk_p Q_w r,\ Q_o + wP_2) \cdot e(sk_p Q_w (v + \beta),\ Q_o + wP_2) \cdot e(\beta P_1,\ Q_o + wP_2)$. Now apply the authorization correctness $e(Q_w, Q_o + wP_2) = e(P_1, P_2)$ and the fact $Q_p = sk_p P_2$. Then $e(\alpha sk_p Q_w r,\ Q_o + wP_2) = e(\alpha r P_1,\ sk_p P_2) = e(\alpha R, Q_p)$ where $R = rP_1$; $e(sk_p Q_w (v + \beta),\ Q_o + wP_2) = e((v + \beta)P_1,\ sk_p P_2) = e(P_1, Q_p)^{v+\beta}$; $e(\beta P_1,\ Q_o + wP_2) = e_{\text{auth}}^{\beta}$ where $e_{\text{auth}} = e(P_1, Q_o + wP_2)$. Hence $e(S, Q_o + wP_2) = e(\alpha R, Q_p) \cdot e(P_1, Q_p)^{v+\beta} \cdot e_{\text{auth}}^{\beta}$. Substituting back into f yields $f = e(\alpha R, Q_p) \cdot e(P_1, Q_p)^{v+\beta} \cdot e_{\text{auth}}^{\beta} \cdot e(P_1, Q_p)^{-v} = e(\alpha R, Q_p) \cdot (e_{\text{auth}} \cdot e(P_1, Q_p))^{\beta} = x \cdot y$, where $x = e(\alpha R, Q_p)$ and $y = (e_{\text{auth}} \cdot e(P_1, Q_p))^{\beta}$. Thus the verification equation holds.

Case 3: Correctness of Signature Verification. For an honestly generated signature, $f = x \cdot y$, thus $h = H_2(f) = H_2(x \cdot y)$. The verifier computes $u' = v \oplus h = v \oplus H_2(x \cdot y)$. By definition $v = h \oplus u$, we get $u' = (h \oplus u) \oplus h = u$. Split $u' = A \parallel B$ and recover $m' = F_2(A) \oplus B$. Since $u = F_1(m) \parallel (F_2(F_1(m)) \oplus m)$, we have $A = F_1(m)$ and $B = F_2(F_1(m)) \oplus m$. Substitute to get $m' = F_2(F_1(m)) \oplus (F_2(F_1(m)) \oplus m) = m$. Hence $F_1(m') = F_1(m) = A$, and the signature is accepted.

6 CPBS SCHEME FOR MIoT

In this section, we will describe the specific architecture of the Collaborative Proxy Blind Signature for Medical Internet of Things scheme and prove its correctness.

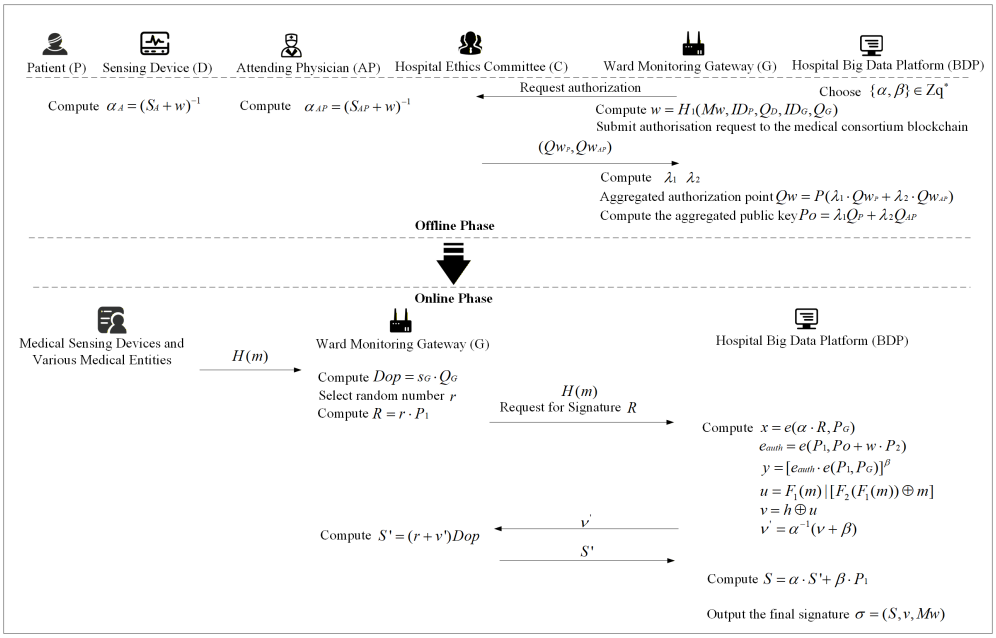


Fig. 3: The process of the CPBS scheme.

6.1 Overview

To balance data security and terminal performance in MIoT, we propose a collaborative proxy blind signature scheme that shifts heavy cryptography offline to enable lightweight online data upload. During the offline phase, participants generate system parameters, register public keys on a medical consortium blockchain, and the ward gateway requests authorization; patients, physicians, and the ethics committee jointly issue (t, n)-threshold Boneh–Boyen shares; the gateway aggregates them, records authorization on-chain, and precomputes commitments and blinding parameters. In the online phase, sensing devices hash and encrypt physiological data, sending the result to the gateway; the gateway verifies authorization, derives a proxy signing key, and interacts with the hospital big data platform for blind signing; the platform decrypts, verifies, unblinds, and outputs the final signature, which anyone can verify via on-chain information. CPBS retains proxy blind signature security properties: data privacy, signature unforgeability, and authorization traceability. By offloading pairings and authorization offline and delegating signing to edge nodes, it sharply reduces terminal computation and communication latency, achieving an optimal privacy–performance balance in MIoT.

6.2 Construction

The CPBS scheme operates in offline and online phases, as shown in Fig. 3. The offline preprocessing phase includes four steps: System Setup, Participant Key Registration, Collaborative Delegation, and Gateway Pre-computation. The online phase includes

three steps: Gateway Proxy Key Retrieval, Blind Data Signing, and Hospital Platform Signature Verification.

1. **System Initialization**: Input security parameter k, output system public parameters $Params = (G, P, e, q, H, F, k, l, t, n)$.
2. **Participant Key Registration**: All participants generate key pairs. Hospital big data platform selects $s_{BDP} \xleftarrow{R} \mathbb{Z}_q^*$ as private key and sets $P_{BDP} = s_{BDP} \cdot P_2$ as public key. Patient P, attending physician AP, hospital ethics committee C, ward monitoring gateway G and medical device D generate their respective key pairs (s_{o_i}, P_{o_i}), (s_G, P_G) and (s_D, P_D). All public keys are registered on the medical consortium blockchain.
3. **Collaborative Delegation:** The gateway G constructs the authorization policy M_w and computes $w = H_1(M_w, ID_D, Q_D, ID_G, Q_G)$. It sends w to all original signers. Upon receiving w, the original signers run a DKG to generate a secret $\alpha = (s_o + w)^{-1}$ and obtain shares α_i. Each signer then sends $Q_{w_i} = \alpha_i P_1$ to G. G collects at least t shares, interpolates $Q_w = \sum \lambda_i Q_{w_i}$, computes $Q_o = \sum \lambda_i Q_{o_i}$, and verifies $e(Q_w, Q_o + wP_2) = e(P_1, P_2)$. If valid, G stores (M_w, Q_w, Q_o, w) and records the authorization on the blockchain.
4. **Gateway Pre-computation**: This step includes Commitment and Blinding processes. Gateway G selects random number $r \xleftarrow{R} \mathbb{Z}_q^*$ and computes commitment $R = rP_1$. Hospital platform BDP selects blinding factors $\alpha, \beta \xleftarrow{R} \mathbb{Z}_q^*$ and precomputes authorization verification parameter $e_{auth} = e(P_1, P_o + wP_2)$.
5. **Gateway Proxy Key Authentication**: Gateway P retrieves (M_w, Q_w) from the blockchain and re-verifies authorization validity. If valid, compute proxy signing key $D_{op} = s_G \cdot Q_w$.
6. **Blind Digital Signature Generation**: This step includes Blinding, BlindSign and Unblinding processes. Medical device D collects data, computes hash value, encrypts data with P_{BDP} and sends to G. G forwards the data and initiates signature request to BDP. BDP decrypts and verifies data, then computes $u = F_1(m) \parallel [F_2(F_1(m)) \oplus m]$, $x = e(\alpha \cdot R, P_G)$, $y = (e_{auth} \cdot e(P_1, P_G))^{\beta}$, $h = H_2(x \cdot y)$, $v = h \oplus u$, $v' = \alpha^{-1}(v + \beta) \bmod q$ and sends v' to G. G computes partial signature $S' = (r + v') \cdot D_{op}$ and returns to BDP. BDP unblinds to get final signature $S = \alpha \cdot S' + \beta \cdot P_1$ and outputs complete signature $\sigma = (S, v, M_w)$.
7. **Signature Verification**: After receiving $\sigma = (S, v, M_w)$ and medical data, the verifier computes verification parameter $f = e(S, P_o + w \cdot P_2) \cdot e(P_1, P_G)^{-v}$, recovers $u' = v \oplus H_2(f)$. Split $u' = A \parallel B$ and recover message $m' = F_2(A) \oplus B$. Accept the signature if $F_1(m') = A$; otherwise reject.

7 SECURITY ANALYSIS

In this section, we first analyze the security of the proposed threshold-based cooperative proxy blind signature protocol, and then present a formal security proof for the CPBS scheme.

Theorem 1: In the random oracle model, if the q-SDH assumption holds in bilinear groups (G_1, G_2), the proposed protocol satisfies: Unforgeability: against a malicious

adversary who may corrupt up to $t-1$ original signers and arbitrarily deviate from the protocol . Blindness:against a semi-honest proxy signer.

We prove these properties via the following lemmas.

*Lemma 1:*No PPT adversary that corrupts up to $t-1$ original signers and may arbitrarily deviate from the protocol can forge a valid signature, assuming the q-SDH problem is hard.

Proof: For a corrupted proxy P and up to $t-1$ malicious original signers S' ($|S'| < t$), simulator Sim constructs an indistinguishable view:

1. Parameter Simulation: Sim generates public parameters $Params = (G, P, e, q, H, F, k, l, t, n)$. It selects proxy private key $x_p \in \mathbb{Z}_q^*$, sets $Q_p = x_p P_2$, and implicitly sets aggregated original public key $Q_o = sP_2$. It selects $q-1$ distinct $w_i \in \mathbb{Z}_q$, computes $\frac{1}{s+w_i} P_1$ from the q-SDH instance sP_1, and stores them in L_w.
2. Authorization Query: For an honest signer O_i ($i \notin S'$), Sim computes $w = H_1(M_w, ID_p, Q_p)$. If $w \in L_w$, it returns a randomized share; otherwise, it aborts.
3. Signature Query: For message m and policy M_w, Sim selects random $\alpha, \beta, r, v \in \mathbb{Z}_q^*$, computes $R = rP_1$, $D_{op} = x_p Q_w$, $v' = \alpha^{-1}(v+\beta) \bmod q$, $S' = (r+v')D_{op}$, and outputs $\sigma = (\alpha S' + \beta P_1, v, M_w)$. It maintains H_2 consistency.

Indistinguishability is proven via hybrid models:

- **Hybrid 1**: Authorization shares are replaced with precomputed values. By threshold Boneh-Boyen security, $Q_w = (s_o + w)^{-1} P_1$ has identical distribution to the real protocol.
- **Hybrid 2**: Blinding factors α, β are replaced with random values. Blinded challenge v' is uniformly distributed over $\mathbb{Z}_q$, identical to the real protocol.
- **Hybrid 3**: H_2 responses are replaced with ideal random values. This is indistinguishable as H_2 is a random oracle.

If the adversary forges σ^* for (m^*, M_w^*) with $w^* \notin L_w$, Sim uses the forking lemma to obtain two signatures (S_1, v_1) and (S_2, v_2) ($v_1 \neq v_2$). From the verification equation: $e(S_1, Q_o + w^* P_2) e(P_1, Q_p)^{-v_1} = e(S_2, Q_o + w^* P_2) e(P_1, Q_p)^{-v_2}$ Rearranging gives $e(S_1 - S_2, (s+w^*)P_2) = e(x_p(v_1 - v_2)P_1, P_2)$. By non-degeneracy: $\frac{1}{s+w^*} P_1 = \frac{1}{x_p(v_1 - v_2)} (S_1 - S_2)$ This solves the q-SDH problem, a contradiction. Thus, the adversary's success probability is negligible.

*Lemma 2:*No external PPT adversary who does not hold any private key can forge a valid signature, assuming the q-SDH problem is hard in the bilinear group.

*Proof:*We define the EUF-CMA game as follows. A challenger $\mathcal{C}$ runs the setup, generates all key pairs, and gives the public parameters and all public keys to the adversary $\mathcal{A}$. $\mathcal{A}$ may adaptively query hash queries to H_1, H_2, authorization queries on $(M_w, ID_d, Q_d, ID_p, Q_p)$, and signature queries on (m, M_w) . At the end, $\mathcal{A}$ outputs a forgery (m^*, σ^*) such that σ^* is valid and m^* was never queried.

We construct a simulator $\mathcal{S}$ that uses $\mathcal{A}$ to break the q-SDH assumption. Given a q-SDH instance $(P, sP, s^2P, \ldots, s^qP) \in G^{q+1}$, $\mathcal{S}$ sets $P_2 = P$, chooses a random

generator $P_1 \in G$, picks $x_p \in \mathbb{Z}_q^*$ and sets $Q_p = x_p P_2$. It sets the aggregated original public key $Q_o = sP_2$. It also selects $q-1$ distinct values $w_1, \ldots, w_{q-1} \in \mathbb{Z}_q$ and pre-computes $\frac{1}{s+w_i} P_1$ using the given $s^j P_1$, storing the pairs $(w_i, \frac{1}{s+w_i} P_1)$ in a list L_w.

$\mathcal{S}$ answers $\mathcal{A}$'s queries as follows. Hash queries are answered by random oracles with consistency. For an authorization query, compute $w = H_1(\cdots)$. If $w \in \{w_1, \ldots, w_{q-1}\}$, return the pre-computed $Q_w = \frac{1}{s+w} P_1$; otherwise abort. For a signature query, pick random $\alpha, \beta, r, \nu \in \mathbb{Z}_q^*$, compute $R = rP_1$, $v' = \alpha^{-1}(\nu + \beta)$, $D_{op} = x_p Q_w$, $S' = (r+v')D_{op}$, $S = \alpha S' + \beta P_1$, then program H_2 so that $H_2(e(S, Q_o + wP_2)e(P_1, Q_p)^{-\nu}) = \nu_{\text{bits}} \oplus u\text{S}$. Return $\sigma = (S, \nu, M_w)$.

Assume $\mathcal{A}$ outputs a forgery (m^*, σ^*) with w^* not in $\{w_1, \ldots, w_{q-1}\}$ (otherwise E_1 occurs). By the forking lemma, replaying $\mathcal{A}$ with different H_2 outputs yields two valid signatures (S_1, ν_1, M_w^*) and (S_2, ν_2, M_w^*) with $\nu_1 \neq \nu_2$. Both satisfy the verification equation $e(S_1, Q_o + w^* P_2)e(P_1, Q_p)^{-\nu_1} = e(S_2, Q_o + w^* P_2)e(P_1, Q_p)^{-\nu_2}$. Rearranging gives $e(S_1 - S_2, (s + w^*)P_2) = e(x_p(\nu_1 - \nu_2)P_1, P_2)$. By non-degeneracy, $\frac{1}{s+w^*} P_1 = \frac{S_1 - S_2}{x_p(\nu_1 - \nu_2)}$. Thus $\mathcal{S}$ outputs $(w^*, \frac{1}{s+w^*} P_1)$ as a solution to the q-SDH problem. The probability analysis follows that of Lemma 1, showing that the forgery advantage is negligible.

Lemma 3: A semi-honest proxy signer who follows the protocol cannot distinguish the signing views of two distinct messages.

*Proof:*We define the following blindness game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

1. $\mathcal{C}$ runs the setup and gives the public parameters to $\mathcal{A}$.
2. $\mathcal{A}$ outputs two equal-length messages m_0, m_1.
3. $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and executes the signing protocol with $\mathcal{A}$ twice: once for message m_b and once for m_{1-b}. $\mathcal{A}$ receives the two transcript views $\text{View}_0, \text{View}_1$.
4. $\mathcal{A}$ outputs a guess b'. $\mathcal{A}$ wins if $b' = b$.

We show that $\mathcal{A}$'s advantage is negligible.
Hybrid 0. $\mathcal{A}$ sees (R, v_0', S_0') for m_b and (R, v_1', S_1') for m_{1-b}, where

$$v_i' = \alpha_i^{-1}(\nu_i + \beta_i), \quad S_i' = (r_i + v_i')D_{op},$$

and α_i, β_i, r_i are chosen uniformly at random for each session.
Hybrid 1. Replace each v_i' with a uniformly random element $v_i' \leftarrow \mathbb{Z}_q$. In the real protocol, because α_i, β_i are independent and uniform, v_i' is uniformly distributed over $\mathbb{Z}_q$. Thus the statistical distance between Hybrid 0 and Hybrid 1 is at most $2^{-|q|}$, which is negligible.
Hybrid 2. After fixing v_i' as random, replace each S_i' with a uniformly random element from the subgroup generated by D_{op}. In the real protocol, $S_i' = (r_i + v_i')D_{op}$; since r_i is uniform and independent of v_i', S_i' is also uniformly distributed over $\langle D_{op} \rangle$. Hence Hybrid 2 is perfectly indistinguishable from Hybrid 1.

In Hybrid 2, the views View_0 and View_1 are both independent of the messages m_0, m_1 . Therefore, $\mathcal{A}$'s probability of correctly guessing b is exactly $1/2$.

Since each successive hybrid differs only by a negligible statistical distance, the advantage of $\mathcal{A}$ in the real game is also negligible. Consequently, the scheme satisfies blindness.

8 PERFORMANCE ANALYSIS

To evaluate the computational efficiency of the proposed scheme, we conduct a series of comparative experiments. The performance of our scheme is compared with those of Tang et al. [36], Chen et al. [37], and Wang et al. [38] in terms of computation, communication, and storage overhead.We select these three schemes for comparison based on the following rationales. Tang et al. [36] is a threshold proxy signature scheme applied to IoT-blockchain scenarios, allowing us to compare the overhead of threshold-based authorization. Chen et al. [37] is a threshold signature-based authentication scheme for edge computing, which helps compare the computational and communication costs of threshold mechanisms in a similar multi-party setting. Wang et al. [38] is a threshold adaptor signature scheme designed for federated learning, serving as a representative of recent threshold cryptography works. None of these schemes simultaneously provide blind signatures for data privacy, decentralized multi-party authorization, and on-chain auditability. Therefore, by comparing CPBS with these three baselines, we can clearly demonstrate that our scheme achieves a superior balance between functional richness and practical performance for MIoT applications.

8.1 Environment and Benchmark

The experiments are conducted on a host machine running Windows 11 with an AMD Ryzen 7 5800H processor (3.20 GHz) and 16 GB RAM. A virtual machine running Ubuntu 24.04.4 LTS is allocated 4 GB RAM, 40 GB disk space, and 4 virtual CPU cores.

Bilinear pairings and group operations are implemented using the PBC library, with multi-precision arithmetic supported by the GMP library. Hash functions are invoked via the `element_from_hash` interface provided by PBC. Symmetric encryption, required for lightweight terminal operations in CPBS, employs the AES-256-GCM authenticated encryption mode from the OpenSSL library.

The experimental parameters are configured as follows: security parameter $\lambda = 128$ bits, Type-A elliptic curve (PBC), with $|G| = 512$ bits, $|q| = 160$ bits. The threshold is set to $t = 10$, with a total number of original signers $n_s = 20$. The raw input data are sourced from the MIT-BIH Arrhythmia Database.Per the CPBS protocol, each ECG segment first undergoes terminal-side hashing and symmetric encryption, after which the resulting authenticated payload is logically partitioned into 160-bit blocks.This uniform block abstraction enables a controlled comparison of core cryptographic overheads across all schemes, independent of variable-length waveform inputs.The data file is divided into $n = 100, 200, 300, \ldots, 1000$ such blocks.

8.2 Computational Overhead

The computational overhead is divided into the following phases. 1) Authorization Phase: Original signers generate shares, each performing 1 inversion, 1 scalar multiplication in G, and optional zero-knowledge proof. The gateway verifies t proofs, computes $O(t^2)$ Lagrange coefficients, and aggregates t shares. Total cost: $C_{del}^{ours} = t(T_{inv} + T_{mul}^{G} + T_{zkp}) + tT_{zkv} + O(t^2)T_{muz} + tT_{mul}^{G}$. 2) Signature Generation Phase: Sensing devices perform 1 hash, 1 symmetric, and 1 public-key encryption. The gateway performs 2 scalar multiplications. The hospital platform performs 2 pairings, 1 exponentiation, 1 hash, 1 inversion, and 2 scalar multiplications. 3) Verification Phase: The verifier performs 1 pairing, 1 exponentiation, and 1 hash.

A comparison is summarized in Table 2. Our authorization overhead matches that of [36] but exceeds that of [37], as the latter relies on centralized key distribution. This one-time cost is amortized over many data blocks. The scheme in [38] incurs quadratic overhead due to Paillier encryption.In the signature phase, our scheme outperforms [36]–[38] by separating authorization and signing, requiring only constant-time operations. The schemes in [36] and [37] need per-signature multi-party interaction, while [38] employs heavy Paillier and MtA protocols.In verification, our overhead is lower than that of [37] and [38], and slightly lower than that of [36]. Moreover, [36] lacks blind signature privacy; [37] requires multiple pairings without blockchain attestation; and [38] suffers from threshold-dependent overhead.Overall, our scheme achieves lower overall computational overhead while providing blind signature privacy, threshold authorization, and blockchain auditability that the others lack.

Table 2: Computational Overhead Comparison

Overhead	Our Scheme	Tang et al.[36]	Chen et al.[37]	Wang et al.[38]
Signature Generation	$2T_{pair} + T_{exp} + 2T_H + T_{inv} + T_{sym} + T_{pke} + 4T_{mul}^{G}$	$t(3T_{mul}^{G_1} + T_{inv} + (t-1)T_{muz})$	$t(T_{mul}^{G} + 2T_{exp} + T_H) + (t-1)T_{mul}^{G}$	$O(t) \cdot (T_{enc} + T_{mul}^{G} + T_{muz})$
Verification	$T_{pair} + T_{exp} + T_H$	$2T_{mul}^{G} + T_H$	$nT_{pair} + T_{exp} + T_H$	$2T_{inv} + 2T_{exp} + T_H$

8.3 Communication Overhead

According to the protocol flow, communication overhead includes three phases. 1) Authorization phase: Each original signer sends the authorization share $Q_{w_i} \in G$ and optional zero-knowledge proof to the ward monitoring gateway, incurring a cost of $t(|G| + |\text{proof}|)$. The aggregated record uploaded to the blockchain costs $|M_w| + 2|G| + |q|$. 2) Data uploading phase: MIoT sensing devices send encrypted data blocks and temporary public keys to the gateway, costing $|\text{data}| + |G|$. 3) Signature interaction

phase: The gateway sends commitment $R \in G$ to the hospital big data platform, receives blinded value $v' \in \mathbb{Z}_q$, and returns partial signature $S' \in G$, with a total cost of $2|G| + |q|$.The total communication overhead of our scheme is $C_{comm}^{ours} = t(|G| + |proof|) + 5|G| + 2|q| + |data| + |M_w|$.

Table 3 summarizes the communication overhead comparison. Our scheme incurs a higher one-time cost in the authorization phase due to multi-party transmission and blockchain storage, which ensures decentralized and auditable authorization. In the online phase, our communication cost is lower than [36]–[38]. Specifically, [36] requires per-signature interaction with t participants and provides no blind signature privacy; [37] lacks blind signing, agent delegation, and blockchain attestation; and [38] involves multi-round interactions with $O(t^2)$ complexity, which is inefficient for large-scale IoT deployments. In summary, our scheme combines blind signatures, threshold authorization, proxy signing, and blockchain auditability, while achieving high online communication efficiency with linear complexity, striking a practical balance between functionality and performance for MIoT applications.

Table 3: Communication Overhead Comparison (bits)

Item	Our Scheme	Tang et al.[36]	Chen et al.[37]	Wang et al.[38]
Authorization	$t(\|G\| + \|proof\|) + \|M_w\| + 2\|G\| + \|q\|$	$2\|G\| + \|q\|$	$(2n_s + t + 6)\|G_1\|$	$n_s\|E\| + n_s\|G\| + n_s(n_s - 1)\|q\| + n_s t\|G\|$
Online Authentication	$3\|G\| + \|q\| + \|data\|$	$t(\|G\| + \|q\|) + 2\|G\| + \|q\| + \|data\|$	$(t + 1)\|q\| + \|data\| + (t + 2)\|G\|$	$O(t^2)\|C\| + O(t)\|G\|$
Blockchain Interaction	Included in Authorization	$\|M_w\| + 2\|G\| + \|q\|$	0	/
Total	$t(\|G\| + \|proof\|) + 5\|G\| + 2\|q\| + \|data\| + \|M_w\|$	$(t + 6)\|G_\| + (t + 4)\|q\| + \|M_w\| + \|data\|$	$(2n_s + 2t + 8)\|G\| + (t + 1)\|q\| + \|data\|$	/

8.4 Storage Overhead

In our scheme, storage overhead is distributed across five entities: 1) MIoT sensing devices: Only store private key $s_d \in \mathbb{Z}_q$, with overhead $|q|$ bits. 2) Original signers (n_s total): Each stores private key $s_{o_i} \in \mathbb{Z}_q$ with overhead $|q|$ bits, for total overhead $n_s \cdot |q|$. 3) Ward monitoring gateway: Stores private key $s_p \in \mathbb{Z}_q$, aggregated authorization point $Q_w \in G$ and authorization policy M_w, with overhead $|q| + |G| + |M_w|$ bits. 4) Hospital big data platform: Stores proxy blind signatures for data blocks. Each signature (S, v) occupies $|G| + |q|$ bits. For n data blocks, total signature overhead is $n \cdot (|G| + |q|)$. 5) Medical consortium blockchain: Permanently stores authorization records including M_w, Q_w, Q_o and identifier w, with overhead $|M_w| + 2|G| + |q|$ bits.

The total storage overhead after full deployment is $S_{ours} = (n_s + n + 3)|q| + (n + 3)|G| + 2|M_w|$. As shown in the comparison, our scheme offloads major storage burden to the medical consortium blockchain and hospital big data platform, while sensing devices and original signers only maintain minimal secret keys. Schemes [36] and [37] require head and signature nodes to store secret values linear to threshold t, and [37] lacks blockchain audit capability. Scheme [38] brings heavy storage overhead due to Paillier keys held by each participant.

Our platform storage scales linearly with the number of data blocks n, since independent signatures are generated for each block to support fine-grained individual verification. In contrast, [36]–[38] adopt aggregated signatures with fixed storage independent of block count, but cannot verify single data entry separately. This design makes our scheme better suited for MIoT scenarios requiring independent auditing of individual vital data records.

Table 4: Storage Overhead Comparison (bits)

Entity	Our Scheme	Tang et al.[36]	Chen et al.[37]	Wang et al.[38]
IoT Device	$\|q\|$	/	/	/
Original Signer	$n_s\|q\|$	$(t+2)\|q\|$	$(t+2)\|q\|$	/
Proxy Signer	$\|q\| + \|G\| + \|M_w\|$	$\|q\| + \|G\|$	$\|q\| + \|G\|$	/
User/Verifier	$n(\|G\| + \|q\|)$	$3\|G\|$	$3\|G\|$	$\|G\|$
Blockchain	$\|M_w\| + 2\|G\| + \|q\|$	$\|M_w\| + 2\|G\| + \|q\|$	/	/
Participant	/	/	/	$\|q\| + 2048 + \|G\| +$ others

8.5 Experimental Analysis

Fig. 4 and Fig. 5 compare the computational time of the signing and verification phases between our scheme and schemes [36]–[38] under increasing numbers of data blocks.

Authorization Phase Overhead Our scheme performs one-time threshold collaborative authorization before data collection, taking only 3.94 ms and independent of the number of data blocks. Schemes [36] and [37] also include one-time authorization, whose overhead grows linearly with threshold t. Scheme [38] requires heavy Paillier encryption and multi-round interaction in key generation, leading to much higher one-time cost. This one-time overhead can be amortized over massive subsequent data blocks, thus having negligible impact on overall performance.

Signature Generation Overhead Our scheme achieves lower total signing time than [36]–[38] with 100 to 1000 data blocks. For 1000 blocks, our total time is about 6.667 s, while [36] is 13.195 s, [37] is 13.966 s, and [38] is 11.924 s. The optimal signing efficiency benefits from the optimized proxy blind signature design: terminal devices run lightweight operations, and gateway and user-side computation are well optimized. By contrast, [36]–[38] require multi-party aggregation, Paillier encryption and multi-round MtA protocols, resulting in faster growth and higher signing latency. While

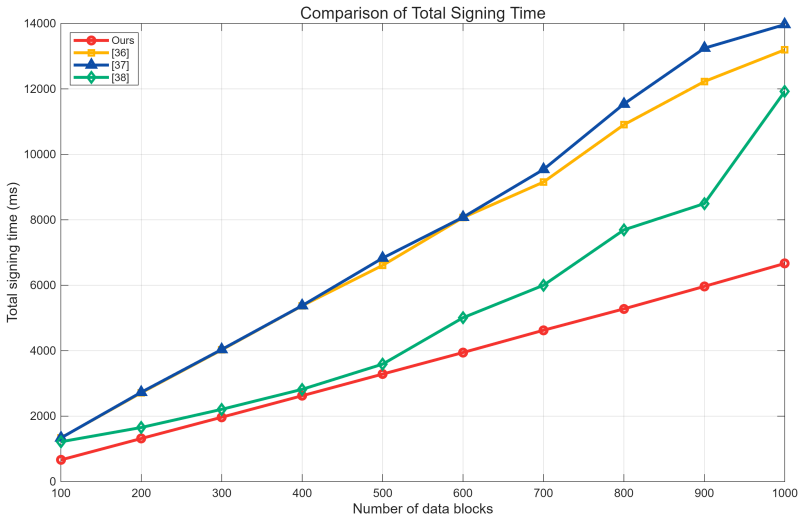


Fig. 4: Comparison of Computational Overhead During Signature Phase on ECG data blocks.

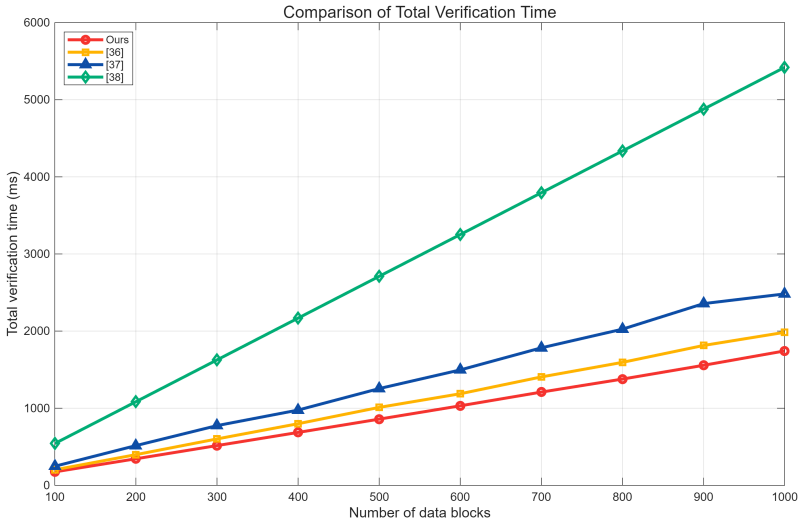


Fig. 5: Comparison of Computational Overhead During Verification Phase on ECG data blocks.

supporting blind signature privacy, threshold authorization and blockchain auditability, our scheme achieves the best signing efficiency, and is well suited for real-time MIoT data collection.

Verification Phase Overhead Our verification time outperforms [37] and [38], and is slightly better than [36]. Our verification only requires one bilinear pairing and one exponentiation. In comparison, [37] requires multiple pairings, and [38] involves

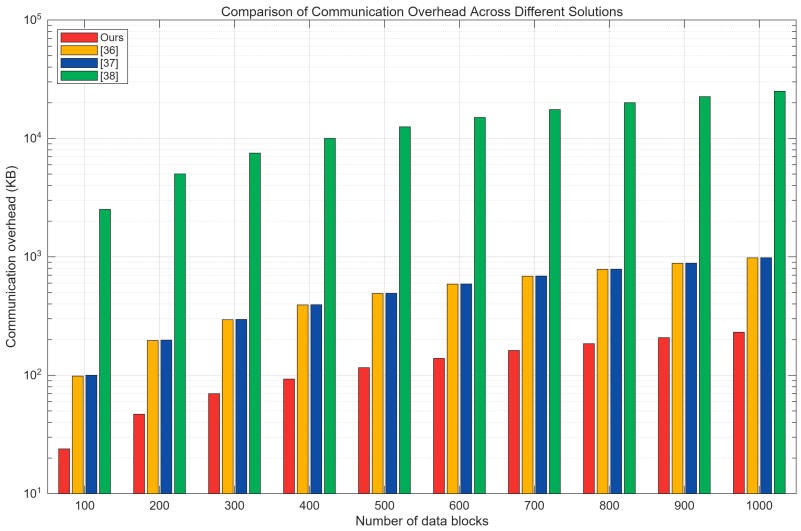


Fig. 6: Comparison of Total Communication Overhead on ECG data blocks.

multiple modular inversions and exponentiations. Our scheme maintains high verification efficiency while realizing data privacy preservation via blind signatures, suitable for frequent signature verification scenarios in MIoT.

Communication Overhead As shown in Fig. 6, with data blocks increasing from 200 to 1000, our total communication overhead grows from 46.89 KB to 230.89 KB, while [36] increases from 202.41 KB to 1010.41 KB, [37] from 204.06 KB to 1012.06 KB, and [38] from 5015 KB to 25015 KB. Our per-block online cost is only 0.23 KB, significantly lower than the 1.01 KB and 25 KB of the other schemes. This advantage stems from the separation of authorization and signing, allowing the proxy signer to complete online signatures independently without multi-party interaction. In contrast, [36] and [37] require per-signature interaction with t nodes, and [38] adopts Paillier ciphertexts with multi-round MtA protocols, leading to quadratic growth and high absolute overhead. Although our one-time authorization cost is slightly higher than [36], it occurs only once and can be amortized. Overall, our scheme achieves the lowest online communication overhead with full functional support, making it suited for large-scale data collection.

Storage Overhead Total storage overhead depends on data size stored by each entity. As shown in Fig. 7, when data blocks increase from 200 to 1000, our total storage overhead rises from 17.69 KB to 82.92 KB, while [36] remains stable at 2.33 KB, [37] at 2.06 KB, and [38] at 10.66 KB.

Our storage overhead increases linearly with data blocks, because independent signatures are generated for each block to support fine-grained individual verification and audit tracing. In contrast, [36]–[38] adopt aggregated signatures with fixed storage independent of block count, but cannot verify single data entry separately. This extra overhead is reasonable: the cloud platform has abundant storage capacity, and independent traceability and verification for vital medical data are critical. Meanwhile,

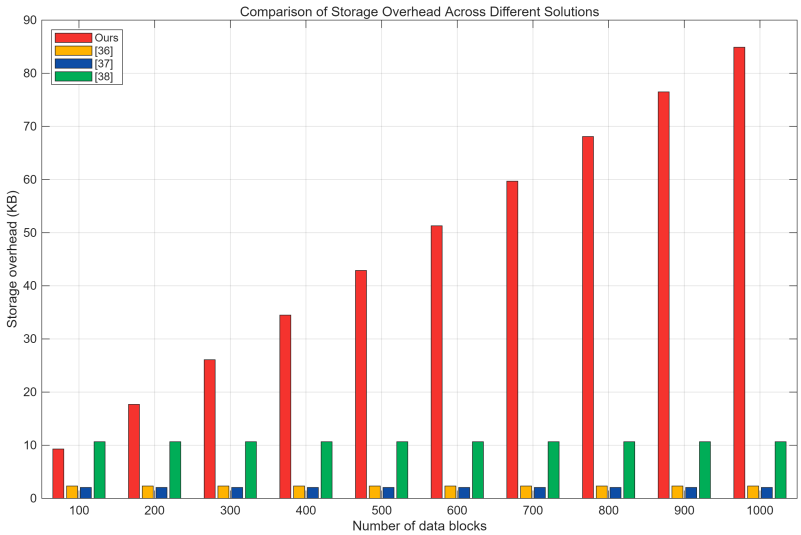


Fig. 7: Comparison of Total Storage Overhead on ECG data blocks.

IoT devices and the gateway in our scheme only store minimal secret keys, imposing lighter local storage burden than the signature nodes of [36], [37] and the Paillier key storage of participants in [38]. Overall, our scheme balances functional advantages including blind signature, threshold authorization, blockchain auditability and fine-grained verification, with reasonable cloud-side storage distribution, matching modern IoT architecture requirements.

Blockchain Module Overhead We evaluate the overhead introduced by the blockchain module under the same environment using Hyperledger Fabric v2.5 with Raft consensus. Table 5 and Fig. 8 show that the per-verification query latency is 18.6–23.8 ms for 4–8 peers, which is comparable to cryptographic verification, while the one-time commit latency of 132.4–189.0 ms is amortizable. Table 6 confirms that even with 100,000 records, ledger size per peer remains only 174 MB.

Table 5: Commit latency and query latency for authorization record

Peers	Commit latency (ms)	Query latency (ms)
4	132.4	18.6
8	189.0	23.8
16	331.7	35.2
32	521.3	58.4

In summary, the blockchain module introduces a one-time write latency of 132–189 ms and a per-verification query latency of 18.6–23.8 ms, with storage growing to only 174 MB for 100,000 records. These results confirm that using a consortium

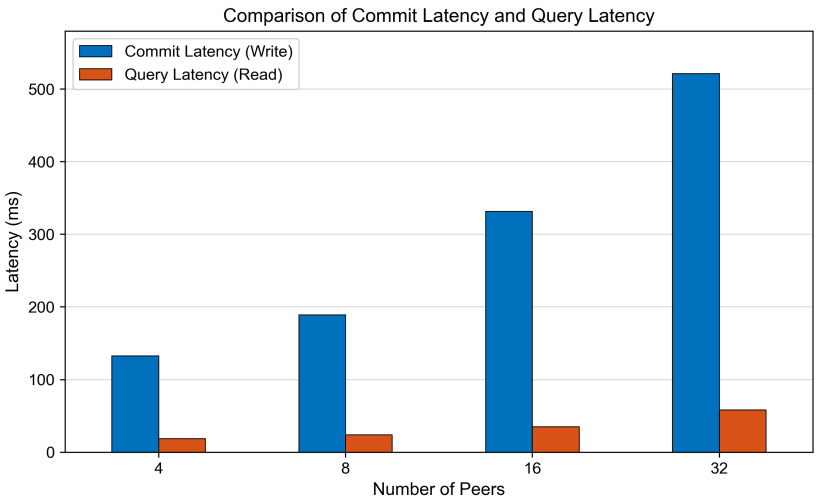


Fig. 8: Comparison of commit latency and query latency

Table 6: Ledger size per peer vs. cumulative records (8 peers)

Cumulative records	Ledger size (MB)
1,000	3.92
5,000	10.79
10,000	19.39
50,000	88.14
100,000	174.07

blockchain for authorization record storage is practical and does not compromise the overall efficiency of the CPBS scheme.

9 CONCLUSION

This paper proposed CPBS, a collaborative proxy blind signature scheme for secure MIoT data collection. It introduces a (t, n)-threshold collaborative authorization to distribute signing rights, stores tamper-proof authorization records on a medical consortium blockchain, and uses blind signatures to protect raw data privacy. By separating authorization from signing and offloading heavy cryptography to an offline phase, terminals perform only lightweight hashing and encryption. Security analysis proves correctness, blindness, and existential unforgeability in the random oracle model. Experiments show reductions in computation and communication overhead by up to 52.3% and 76.8%, respectively, supporting long-term multi-source health monitoring. Future work will extend security guarantees against fully malicious adversaries

and explore more efficient on-chain authorization mechanisms for large-scale MIoT deployments.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China.

Author contribution. Jinnan Li proposed the research concept, designed the research methodology and core framework, and drafted the initial manuscript.Qian Zhou and Shiwei Huo revised the manuscript. Qing Ye, Yi Yang and Zhimin Yuan provided research guidance and technical support. All authors reviewed, revised, and approved the final draft.

Funding. This work was supported in part by the National Natural Science Foundation of China under Grant 62402513 and Grant 62572355.

References

[1] Ullah S, Li J, Chen J, Ali I, Khan S, Ahad A, Ullah F, Leung VCM. (2024) A Survey on Emerging Trends and Applications of 5G and 6G to Healthcare Environments. ACM Comput Surv 57(4):1–36. https://doi.org/10.1145/3703154

[2] Younesi A, Oustad E, Ansari M, Fahringer T, Buyya R. (2026) Health-Care 5.0: An industry 5.0 perspective for next-generation medical systems with synergistic integration of IoT, AI, and 6G. Internet of Things 35:101815. https://doi.org/10.1016/j.iot.2025.101815

[3] Taherdoost H. (2024) Wearable Healthcare and Continuous Vital Sign Monitoring with IoT Integration. Comput Mater Continua 81(1):79–104. https://doi.org/10.32604/cmc.2024.054378

[4] Alshuhail A, Alshahrani A, Mahgoub H, Ghaleb M, Darem AA, Aljehane NO, Alotaibi M, Alzahrani F. (2025) Machine edge-aware IoT framework for real-time health monitoring: Sensor fusion and AI-driven emergency response in decentralized networks. Alex Eng J 129:1349–1361. https://doi.org/10.1016/j.aej.2025.08.030

[5] Alsharabi N, Alayba A, Alshammari G, Alsaffar M, Jadi A. (2025) An end-to-end four tier remote healthcare monitoring framework using edge-cloud computing and redactable blockchain. Comput Biol Med 189:109987. https://doi.org/10.1016/j.compbiomed.2025.109987

[6] Alsadhan A, Alhogail A, Alsalamah HA. (2025) Toward Efficient Health Data Identification and Classification in IoMT-Based Systems. Sensors 25(19):5966. https://doi.org/10.3390/s25195966

[7] Akinbi A, Raj PP. (2025) A systematic security analysis of Medical Internet of Things (MIoT) ecosystems in threat modeling scenarios. Front Internet Things 4. https://doi.org/10.3389/friot.2025.1712430

[8] Deb S, Lupu E, Drakakis EM, Bharath AA, Leung ZK, Ma GR, Chattopadhyay A. (2025) Securing the Internet of Medical Things (IoMT): Real-World Attack Taxonomy and Practical Security Measures. arXiv preprint arXiv:2507.19609. https://doi.org/10.48550/arXiv.2507.19609

[9] Yankson B, Barati M, Bondzie R, Madani R. (2025) The Rise of Hacking in Integrated EHR Systems: A Trend Analysis of U.S. Healthcare Data Breaches. J Cybersecur Priv 5(3):70. https://doi.org/10.3390/jcp5030070

[10] Ghazal TM, Hasan MK, Khairy SOF, Mokhtar UA, Islam S, Saeed RA, Ahmed N, Abdullah M. (2025) Lightweight signcryption scheme for Securing wearable sensor observed health data sharing in Internet of Medical Things paradigm. Sci Rep 15:40989.

[11] Liu J, Yang J, Huang X, Qu Y, Xiang Y. (2025) UDVRS: Achieving User-Designation and Privacy-Preserving for Verifiable Data Sharing in Healthcare System. IEEE Trans Dependable Secur Comput. https://doi.org/10.1109/TDSC.2025.3599205

[12] Fan Q, Xie Y, Zhang C, Liu X, Zhu L. (2024) An Authentic and Privacy-Preserving Scheme Towards E-Health Data Transmission Service. IEEE Trans Serv Comput 17(5):1969–1982. https://doi.org/10.1109/TSC.2024.3451145

[13] Maqsood TB, Hameed K, Junaid M, Tahir N. (2025) Enhancing IoT Healthcare Security with Post-Quantum Cryptography: A Performance and Case Study Analysis. 2025 6th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 1–8. https://doi.org/10.1109/ICACS64902.2025.10937815

[14] Perez-Tirador P, Jevtic R, Cabezaolias C, Romero T, Otero A, Caffarena G. (2025) The effect of ECG data variability on side-channel attack success rate in wearable devices. Integration 103:102385. https://doi.org/10.1016/j.vlsi.2025.102385

[15] Hu X. (2023) Security Analysis and Improvement of Proxy Signature Scheme with Certificateless and Blind Properties. 2023 2nd International Conference on Big Data, Information and Computer Network (BDICN), Xishuangbanna, China, 372–375. https://doi.org/10.1109/BDICN58493.2023.00082

[16] Bojjagani S, Brabin D, Kumar K, Sharma NK, Batta U. (2024) Secure privacy-enhanced fast authentication and key management for IoMT-enabled smart healthcare systems. Computing 106:2427–2458. https://doi.org/10.1007/s00607-024-01291-0

[17] Han G, Ma Y, Zhang Z, Liu J. (2026) Blockchain-Assisted Efficient Certificateless Aggregate Signcryption Scheme for IoMT. IEEE Internet Things J 13(5):8452–8464. https://doi.org/10.1109/JIOT.2025.3641397

[18] Xu G et al. (2025) Anonymity-Enhanced Sequential Multi-Signer Ring Signature for Secure Medical Data Sharing in IoMT. IEEE Trans Inf Forensics Secur 20:5647–5662. https://doi.org/10.1109/TIFS.2025.3574959

[19] Mambo M, Usuda K, Okamoto E. (1996) Proxy Signatures: Delegation of the Power to Sign Messages. IEICE Trans Fundam E79-A(9):1338–1354.

[20] Lin W, Jan JK. (2000) Security personal learning tools using a proxy blind signature scheme. J Chin Lang Comput.

[21] Lin Q, Li J, Huang Z, Chen W, Shen J. (2018) A Short Linearly Homomorphic Proxy Signature Scheme. IEEE Access 6:12966–12972. https://doi.org/10.1109/ACCESS.2018.2809684

[22] Wang C, Wu B. (2023) A Linear Homomorphic Proxy Signature Scheme Based on Blockchain for Internet of Things. CMES - Comput Model Eng Sci 136(2):1857–1878. https://doi.org/10.32604/cmes.2023.026153

[23] Zhang H, Zhang X, Zhang Y, Wang X, Liu Q. (2024) Blockchain-Based Proxy-Oriented Data Integrity Checking Mechanism in Cloud-Assisted Intelligent Transportation Systems. IEEE Trans Intell Transp Syst 25(11):17366–17381. https://doi.org/10.1109/TITS.2024.3431539

[24] Xue Q, Lu Z, Zhang T. (2024) Attribute-based proxy signature scheme with dynamic strong forward security. Int J Sens Netw 44(4):214–225. https://doi.org/10.1504/ijsnet.2024.138518

[25] Chaum D. (1983) Blind Signatures for Untraceable Payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds) Advances in Cryptology. Springer, Boston, MA. https://doi.org/10.1007/978-1-4757-0602-4_18

[26] Bhavin M, Tanwar S, Sharma N, Tyagi S, Kumar N. (2021) Blockchain and quantum blind signature-based hybrid scheme for healthcare 5.0 applications. J Inf Secur Appl 56:102673. https://doi.org/10.1016/j.jisa.2020.102673

[27] Gao M, Li J, Di X, Li X, Zhang M. (2024) A blind signature scheme for IoV based on 2D-SCML image encryption and lattice cipher. Expert Syst Appl 246:123215. https://doi.org/10.1016/j.eswa.2024.123215

[28] Yang Y, Yan X, Lian B, Wang X, Cui J, Zhao X, Wang L, Wang F, Chen K. (2026) A blind signature-based authorization scheme for enhancing the privacy of Cloud-Assisted private set intersection. Comput Netw 274:111844. https://doi.org/10.1016/j.comnet.2025.111844

[29] Samal K, Sunanda SK, Patnaik S. (2025) A lightweight privacy preservation authentication protocol for IoMT using ECC based blind signature. Int J Comput Sci Med. https://doi.org/10.1177/18479790251318538

[30] Lindell Y. (2021) Fast Secure Two-Party ECDSA Signing. J Cryptol 34:44. https://doi.org/10.1007/s00145-021-09409-9

[31] He D, Zhang Y, Wang D, Choo K-KR. (2020) Secure and Efficient Two-Party Signing Protocol for the Identity-Based Signature Scheme in the IEEE P1363 Standard for Public Key Cryptography. IEEE Trans Dependable Secure Comput 17(5):1124–1132. https://doi.org/10.1109/TDSC.2018.2857775

[32] Mu Y, Xu H, Li P et al. (2020) Secure two-party SM9 signing. Sci China Inf Sci 63:189101. https://doi.org/10.1007/s11432-018-9589-x

[33] Liu Y, Feng Q, Peng C, Luo M, He D. (2022) Asymmetric Secure Multi-party Signing Protocol for the Identity-Based Signature Scheme in the IEEE P1363 Standard for Public Key Cryptography. In: Chen, J., He, D., Lu, R. (eds) Emerging Information Security and Applications. EISA 2022. Communications in Computer and Information Science, vol 1641. Springer, Cham. https://doi.org/10.1007/978-3-031-23098-1_1

[34] Liu SG, Liu R, Rao SY. (2022) Secure and efficient two-party collaborative SM9 signature scheme suitable for smart home. J King Saud Univ - Comput Inf Sci 34(7):4022–4030. https://doi.org/10.1016/j.jksuci.2022.05.008

[35] Q. Xie et al., "LiteCrypt: Enhancing IoMT Security with Optimized HE and Lightweight Dual-Authorization," in *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, Belgrade, Serbia, 2024, pp. 166–175, doi: 10.1109/ICPADS63350.2024.00031.

[36] Tang F, Xu T, Peng J, Gan N. (2024) TP-PBFT: A Scalable PBFT Based on Threshold Proxy Signature for IoT-Blockchain Applications. IEEE Internet Things J 11(9):15434–15449. https://doi.org/10.1109/JIOT.2023.3347232

[37] Chen L, Guo C, Gong B, et al. (2024) A secure cross-domain authentication scheme based on threshold signature for MEC. J Cloud Comp 13:70. https://doi.org/10.1186/s13677-024-00631-x

[38] Wang Z et al. (2025) TASS: Threshold Adaptor Signature Scheme for Holographic IoT Consumer Devices in Blockchain Enabled Federated Learning. IEEE Trans Consum Electron 71(3):8662–8673. https://doi.org/10.1109/TCE.2025.3573156