electronics

Article

A Lightweight Identity-Based Network Coding Scheme for Internet of Medical Things

Kuan Wang [1,*], Mingxuan Song [1], Genqing Bian [1,*], Bilin Shao [1,*] and Kaiqi Huang [2,*]

[1] College of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; songmingxuan@xauat.edu.cn
[2] School of Cyber Engineering, Xidian University, Xi'an 710126, China
* Correspondence: wangkuan114514@163.com (K.W.); bgq_00@163.com (G.B.); sblin0462@sina.com (B.S.); 21151213631@stu.xidian.edu.cn (K.H.)

Abstract: Network coding is a potent technique extensively utilized in decentralized Internet of Things (IoT) systems, including the Internet of Medical Things (IoMT). Nevertheless, the inherent packet-mixing characteristics of network coding expose data transmission to pollution attacks, potentially compromising the integrity of original files. The homomorphic signature scheme serves as a robust cryptographic tool that can bolster network coding's resilience against such attacks. However, current schemes are computationally intensive for signature verification, making them impractical for IoMT environments. In this study, we propose a lightweight identity-based network coding scheme (IBNS) that minimizes computational overhead during the signing and verification processes. This scheme has been demonstrated to be secure against adaptive chosen-message attacks and is well-suited for IoMT applications. Furthermore, we assess the performance of our IBNS through both theoretical and experimental analyses. Simulation outcomes confirm that our scheme outperforms previous ones in terms of practicality and efficiency.

Keywords: network coding; pollution attacks; identity-based cryptography; homomorphic signature

check for updates

Citation: Wang, K.; Song, M.; Bian, G.; Shao, B.; Huang, K. A Lightweight Identity-Based Network Coding Scheme for Internet of Medical Things. *Electronics* **2024**, *13*, 1316. https://doi.org/10.3390/electronics13071316

Academic Editor: Adão Silva

Received: 4 February 2024
Revised: 23 March 2024
Accepted: 29 March 2024
Published: 31 March 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Medical Things (IoMT), as current popular technology, connects doctors, patients, and wearable devices, thus achieving more convenient and accurate disease management for patients [1–3]. Since the patient's medical detection device can collect data and transmit it in real time to a medical cloud server via the network, the medical staff can access patients' data without going offline.

Network coding introduces a novel mechanism where intermediate nodes combine incoming packets before forwarding, enhancing throughput, robustness, and reducing power consumption [4]. Its application within the Internet of Medical Things (IoMT) is particularly significant. It should be noted that the IoMT and network coding technology is also gaining popularity. For example, Peng et al. proposed an efficient network coding-based fault-tolerant mechanism in WBAN for smart healthcare monitoring systems [5]. In addition, Takabayash proposed a network coding scheme [6] in the IoMT environment based on the IEEE 802.15.6 ultra-wideband physical layer. Finally, Ar-Reyouchi et al. proposed a health data coding scheme [7] in the wireless medical sensor network environment.

At present, the network coding scheme in the IoMT environment needs to solve three problems: well-documented pollution attack [8,9], adaptive chosen-message attacks, and how to reduce the communication cost of network coding as much as possible [10].

Firstly, the well-documented pollution attack poses a critical security threat to network coding due to the potential for packet mixing, which can rapidly propagate pollution throughout the network [11]. As a result, even with sufficient packets, the destination node may fail to recover the original data, squandering network resources. Existing countermeasures against pollution attacks generally fall into two categories: information-theoretic

methods [12–14] and cryptographic methods [15–17]. Information-theoretic methods introduce redundancy to recover from malicious faults but merely tolerate pollution attacks passively at the destination. In contrast, cryptographic methods empower intermediate nodes to detect and filter polluted packets during transmission, effectively mitigating pollution attacks at their source. Cryptographic methods can be further classified into symmetric key [18–21] and public key approaches [22,23], with the latter gaining prominence due to their solution to the key distribution problem, making them particularly suitable for network coding scenarios involving multiple source and destination nodes. This paper focuses on homomorphic signatures within public key methods, characterized by the ability of intermediate nodes to detect and filter polluted packets by combining all received signatures into a unified signature that must correspond to the combined data packet, allowing intermediate and destination nodes to identify potential errors.

Moreover, how to prevent adaptive election-message attacks is also a problem that needs to be solved. Under adaptive chosen-message attacks, attackers can select the messages they want to attack based on previously received information [24,25]. In an adaptive selective message attack, the attacker can use the signer as an oracle throughout the attack [26,27]. They can select messages based on the signer's public key and previous messages or signatures obtained during the attack. At present, there are two ways to prevent it. One is to use encryption or coding schemes that limit the plasticity of ciphertext and the proof of system security [28]. The second is to use digital signatures [29].

Finally, how to effectively reduce the communication overhead of coding schemes is also a problem that needs to be solved. Currently, the Public Key Infrastructure (PKI) is a prevalent mechanism for certificate management, enabling the hierarchical transmission of trusted messages among users. However, PKI deployment can be prohibitively expensive, and certificate management is onerous, particularly within the context of the Internet of Medical Things (IoMT). To circumvent the use of certificates, Shamir introduced the concept of identity-based public key cryptography (IBC) [30], which employs a user's identifying information—such as an ID card, email address, or telephone number—as their public key. A trusted authority, such as a Key Generation Center (KGC), then derives the private key. This innovative approach significantly streamlines certificate management, prompting scholars to develop numerous identity-based network coding schemes [31–33]. Nonetheless, existing schemes are challenged by their high computational requirements for signature verification, a process that intermediate nodes with limited computational resources may struggle to perform without introducing network latency. Consequently, few existing schemes are suitable for the IoMT. To address this, we have designed a lightweight identity-based linearly homomorphic signature scheme tailored for network coding.

Our Contributions: In this study, we introduce a novel, lightweight identity-based network coding scheme (IBNS). The uniqueness of our scheme stems from the development of innovative signing and verification algorithms, which can prevent well-documented pollution attacks. In addition, we provide evidence that our IBNS is resilient to adaptive chosen-message attacks under the k-CAA assumption. Both theoretical and experimental analyses indicate that our scheme surpasses related schemes regarding computational efficiency during the signature generation and signature verification phases. Our contributions are threefold:

(1) IBNS adopts the homomorphic signature method in the public key method. This method makes a one-to-one correspondence between the signature combination and the packet. Through this correspondence, both the intermediate node and the destination node can identify potential errors. This is why IBNS can prevent well-documented pollution attacks.
(2) In the related discussion of security analysis, we prove the security of the IBNS signature and authentication algorithm. Our proof process is based on the k-CAA assumption. Compared with the security proof of other schemes, based on the k-CAA hypothesis, it can be shown that this scheme can fully demonstrate that IBNS can effectively prevent adaptive chosen-message attacks.

(3) To reduce the overhead in the IoMT environment, we design a lightweight identity-based linear homomorphic signature scheme for network coding. The experimental results show that our scheme performed well.

Organization: This paper is structured as follows. Section 2 introduces the related works of IBNS. The system and security models of IBNS are described in Section 3. Section 4 provides details on the components of IBNS. An analysis of the correctness and security of IBNS is provided in Section 5. In Section 6, a performance analysis of IBNS is conducted. Moreover, In Section 7, we discuss the meaning of IBNS in IoMT. Finally, Section 8 concludes the paper and provides a summary of the findings on IBNS.

2. Related Works

2.1. Internet of Medical Things (IoMT)

The Internet of Things (IoT) concept was first introduced by Ashton et al. in 2009 [34]. As life expectancy increases and the population ages, healthcare services encounter significant challenges.

The development of smart sensors, smart devices, and advanced lightweight communication protocols created the possibility of interconnecting medical things to monitor biomedical signals and diagnose the diseases of patients without human intervention, termed the Internet of Medical Things (IoMT) [35]. IoMT has a variety of application forms; the two most mainstream forms are (1) implantable medical devices (IMDs) and (2) Internet of Wearable Devices (IoWDs) [36,37]. IoMT has been fully integrated into our lives. In addition, with the application of deep learning and blockchain technology [38], IoMT will become more intelligent and versatile. Moreover, relevant studies show that the output value of IoMT is expected to be USD 13.5 billion by 2025 [39]. Therefore, IoMT has a great future.

However, IoMT still faces several security challenges [40,41]. For example, during health data collection, malicious users can attack related hardware and software devices [42]. In data storage, storage services may be subject to DDoS attacks [43]. Most importantly, security risks in transmitting medical data must be addressed urgently [44]. For example, transmitting data between sensors can be easily eavesdropped. In addition, data transmission is prone to pollution attacks and adaptive chosen-message attacks [45]. Finally, the relevant coding scheme needs to be improved, and the digital signature function needs to be added to the coding scheme [46]. Because there are many security risks in the data transmission process, we propose IBNS to solve the security problems in the data transmission process.

2.2. Homomorphic Signatures

In order to better prevent pollution attacks and adaptive election-message attacks, homomorphic signatures are adopted in our scheme. The notion of homomorphic signatures was initially introduced by Johnson et al. [47]. Following this, Lin et al. proposed an identity-based homomorphic signature scheme suitable for network coding based on the Computational Diffie–Hellman (CDH) assumption [32]. In contrast, the related key attack on this scheme was considered by Chang et al. [33]. Additionally, a network coding scheme with privacy-preserving features was introduced by Liu et al. [48], and a certificates network coding scheme for IoT applications was proposed by Wu et al. [49].

In addition, homomorphic signatures have made significant progress in recent years. First of all, authorized function homomorphic signature (AFHS) is a new primitive introduced to address the problem of data owners lacking control of an agency's behavior in homomorphic signature [50–52]. In AFHS, the agency has to obtain a confidence key from data owners to evaluate a function on the data and obtain a signature. This restricts the agency's ability to sign. Second, fully homomorphic signatures from standard lattices provide a new identity-based homomorphic trapdoor function (IBHTDF), which is claw-free and collision-resistant [53–55]. Moreover, a homomorphic algorithm for the new IBHTDF was found where the noise level of the multiplication gate is the same as that of the addition

gate. This reduces the noise level of IBHTDF when evaluating a circuit of depth. Moreover, finally, key-homomorphic signatures focus on key-homomorphic signature schemes and show that this concept turns out to be an exciting and versatile tool [56,57].

3. System Model and Security Model

This topic describes the IBNS System Model and Security Model. First of all, in order to better conduct the subsequent discussion, we introduce the relevant basic knowledge of IBNS in Preliminaries. Secondly, we introduce the System Model of the Identity-based Network Coding. Finally, we introduce the Security Model of the IBNS.

3.1. Preliminaries

Before introducing the preliminaries, we introduce some fundamental concepts used in this study. Specifically, 1^{λ} represents the security parameter, and PPT is an abbreviation for "Probabilistic Polynomial Time". For an algorithm $\mathcal{A}$, the notation $\alpha \xleftarrow{r} Q$ signifies that α is randomly selected from a set Q. Subsequently, $y \leftarrow \mathcal{A}(x_1, \cdots, x_m)$ indicates that it takes $x_1, \cdots, x_m$ as inputs and produces y as output. The symbols and their corresponding definitions are detailed in Table A1 (Appendix A).

3.1.1. Bilinear Map

We introduce the concepts related to bilinear map. The properties of bilinear map are bilinearity, non-degeneracy, and computability. We adopt $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ as three multiplicative cyclic groups. And the generators of $\mathbb{G}_1, \mathbb{G}_2$ are g_1 and g_2. The order of the three multiplicative cyclic groups is p. The properties of the bilinar map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ are as follows.

(1) Bilinearity. $\forall a \in \mathbb{G}_1, \forall b \in \mathbb{G}_2$, and $\forall x, y \in \mathbb{Z}_p$, $e(a^x, b^y) = e(a, b)^{xy}$.
(2) Non-degeneracy. For g_1 and g_2, $e(g_1, g_2) \neq 1$.
(3) Computability. $\forall P, Q \in \mathbb{G}_1$. There is a polynomial-time algorithm to compute $e(P, Q)$ efficiently.

3.1.2. Complexity Assumption

Definition 1 (k-CAA problem)**.** *Given a cyclic group G_1 with a prime order q, and there are $k+1$ elements $x, a_1, \cdots, a_k \in Z_q^*$, then suppose two elements $g, g^x \in G_1$ and k pairs $\left(a_1, g^{(x+a_1)^{-1}}\right), \cdots, \left(a_k, g^{(x+a_k)^{-1}}\right)$. For some $a^* \notin \{a_1, \cdots, a_k\}$, it outputs a new pair $\left(a^*, g^{(x+a^*)^{-1}}\right)$.*

Definition 2 (k-CAA assumption)**.** *If for any PPT algorithm $\mathcal{A}$, its advantage to solve the k-CAA problem is negligible, which can be denoted as:*

$$Adv_{\mathcal{A}} = Pr\left[\mathcal{A}\left(g, g^x, g^{\frac{1}{(x+a_1)}}, \cdots, g^{\frac{1}{1/(x+a_k)}}\right) = g^{\frac{1}{(x+a^*)}}\right],$$

then we say that the k-CAA assumption holds on G_1.

3.2. System Model of the Identity-Based Network Coding

In linearly identity-based network coding, three types of nodes need to be considered, and these are source nodes (i.e., signers), intermediate nodes (i.e., repeaters and routers), and destination nodes (i.e., receivers), respectively. In Figure 1, we give the schematic diagram, including two source nodes.

- Source nodes want to send some data files to destination nodes. For the single source node (S), which will transmit a data file V, it first splits the file into m blocks $\bar{\mathbf{v}}_1, \cdots, \bar{\mathbf{v}}_m$, and each block can be viewed as the n-dimensional vector $\bar{\mathbf{v}}_i = (v_{i,1}, \cdots, v_{i,n})$. Next, S_i augments each vector as $\mathbf{v}_i = (\bar{\mathbf{v}}_i, \mathbf{e}_i)$, where $\mathbf{e}_i$ denotes the m-dimensional unit vector and its i-th element equals 1. These augmented vectors $\mathbf{v}_1, \cdots, \mathbf{v}_m$ are also

called packets. Now, S needs to generate a tag t_i as the signature for each packet $\mathbf{v}_i$, then it sends all packet–tag pairs $\{(\mathbf{v}_i, t_i)\}_{i=1}^{m}$ into the network.

- Intermediate nodes consist of abundant repeaters and routers in the network; they process received pairs as follows. After receiving lots of pairs $(\mathbf{v}_1, t_1), \cdots, (\mathbf{v}_l, t_l)$ for $1 \leq l \leq m$, each intermediate node (N) first checks the validity of each pair, and it chooses l coefficients $\{c_i\}_{i=1}^{l} \in Z_q$ randomly. Then, N computes a combined packet and a combined tag as

$$\hat{\mathbf{v}} = \sum_{i=1}^{l} c_i \mathbf{v}_i, \quad \hat{t} = \prod_{i=1}^{l} t_i^{c_i}. \tag{1}$$

  The combined pair $(\hat{\mathbf{v}}, \hat{t})$ will continue to transmit in the network.
- Destination nodes are receivers who want to obtain the whole data file during transmission. Upon collecting m linearly independent packets $(\hat{\mathbf{v}}_1, \cdots, \hat{\mathbf{v}}_m)$, the destination node ($R$) checks the validity of all pairs $(\hat{\mathbf{v}}_1, \hat{t}_1), \cdots, (\hat{\mathbf{v}}_m, \hat{t}_m)$. If all of them are unpolluted, R can leverage the method of solving linear equations to recover the original vectors (blocks) $\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \cdots, \bar{\mathbf{v}}_m$, that is, R obtains the data file V.

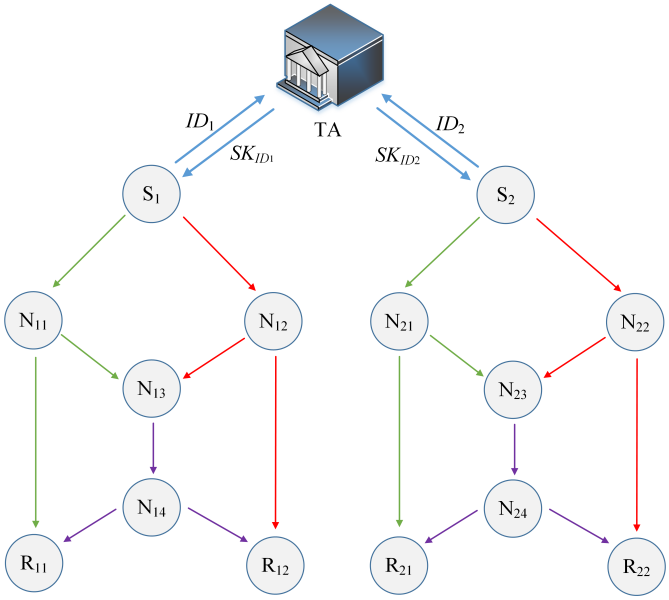


Figure 1. The system model of the identity-based network coding.

Definition 3 (IBNS). *An identity-based network coding scheme IBNS consists of six PPT algorithms and specific details are depicted as follows:*

- *Setup* $(1^\lambda) \rightarrow (msk, params)$*: TA executes this algorithm. Upon receiving a security parameter, denoted as* 1^λ*, the TA generates a master secret key, represented as msk, and system public parameters, symbolized as params.*
- *Extract-Private-Key* $(params, msk, ID) \rightarrow SK_{ID}$*: Extract-Private-Key is executed by TA, which is responsible for generating the private key* SK_{ID}*, for the source node. Specifically, when provided with an identity, ID, associated with a user, the TA generates and delivers the corresponding private key,* SK_{ID}*, to that user.*
- *TagGen* $(params, ID, SK_{ID}, vid) \rightarrow \tau$*: In this algorithm, the source node, given the public parameters params, the identity ID, the private key* SK_{ID}*, and the identifier vid of a subspace* V*, generates a file tag* τ*.*
- *Sign* $(params, ID, SK_{ID}, \tau, \boldsymbol{v}) \rightarrow t$*: The signer executes this algorithm to generate the tag t as the signature. Particularly, this algorithm takes the public parameters params, the identity ID, the private key* SK_{ID}*, the file tag* τ *and a packet* $\boldsymbol{v} \in V$ *as input, and outputs a signature t. Note that the signer should sign all packets* $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m \in V$*, then any packets and tags can continue to combine in the network.*

- *Combine* $\left(params, ID, vid, \tau, \{c_i, \boldsymbol{v}_i, t_i\}_{i=1}^{l}\right) \rightarrow (\hat{\boldsymbol{v}}, \hat{t})$*: This algorithm is executed by intermediate nodes. It accepts as input the system parameters params, the user's identity ID, the identifier vid along with the tag* τ *of subspace V, and packet-tag pairs* $\{c_i, \boldsymbol{v}_i, t_i\}_{i=1}^{l}$*, and subsequently generates a combined packet–tag pair* $(\hat{\boldsymbol{v}}, \hat{t})$*.*
- *Verify* $(params, ID, vid, \tau, \hat{\boldsymbol{v}}, \hat{t}) \rightarrow 0/1$*: This algorithm can be executed by either the intermediate node or the destination node. It accepts as input the system parameters params, the user's identity ID, the identifier vid with the tag* τ *of subspace V and a combined packet* $\hat{\boldsymbol{v}}$ *with signature tag* $\hat{t}$*, and subsequently produces an output of 1 (accept) or 0 (reject).*

Correctness. The correctness of secure IBNS requires that, for each output $(params, msk)$ generated from **Setup**, the following holds:

(1) For all user's identity ID, all file identifier vid and all vectors $\mathbf{v} \in V$, if $\tau \leftarrow$**TagGen**$(params, ID, SK_{ID}, vid)$ and $t \leftarrow$**Sign**$(params, ID, SK_{ID}, \tau, \mathbf{v})$, then

$$\textbf{Verify}(params, ID, vid, \tau, \mathbf{v}, t) \rightarrow 1.$$

(2) For all user's identity ID, any file tag τ and all tuples $\left\{(c_i, \mathbf{v}_i, t_i)_{i=1}^{m}\right\}$, if

$$\hat{t} \leftarrow \textbf{Combine}\left(params, ID, vid, \tau, \{c_i, \mathbf{v}_i, t_i\}_{i=1}^{l}\right),$$

and it equals **Sign**$(params, ID, SK_{ID}, \tau, \sum_{i=1}^{m} c_i \mathbf{v}_i)$, then

$$\textbf{Verify}\left(params, ID, vid, \tau, \sum_{i=1}^{m} c_i \mathbf{v}_i, \hat{t}\right) \rightarrow 1.$$

3.3. Security Model of the IBNS

Consider a probabilistic polynomial-time (PPT) adversary capable of executing an adaptive chosen-message attack on identity-based network coding schemes, with the ultimate goal of producing a valid forged signature. Notably, this adversary can request the private key corresponding to any user's identity and has the ability to sign any packets for any given identities. If the adversary fails to generate a valid forgery with a significant probability, the identity-based network signature (IBNS) is deemed secure. The security model of the IBNS can be characterized by a game in which the adversary interacts with a challenger.

Game-IBNS: In this game, the adversary $\mathcal{A}$ interacts with the challenger $\mathcal{C}$ as follows:

- **Initialization**: Input the security parameter 1^{λ}, the challenger $\mathcal{C}$ runs the algorithm

$$(msk, params) \leftarrow \textbf{Setup}\left(1^{\lambda}\right),$$

then, it keeps *msk* secret and returns *params* to $\mathcal{A}$.
- **Queries**: $\mathcal{A}$ is able to make a series of queries as follows adaptively.
  - *Extract-Private-Key-Queries*: The private key associated with the user's ID is queried by $\mathcal{A}$. The **Extract-Private-Key** operation is executed by $\mathcal{C}$, which results in the output of SK_{ID}. This is then returned to $\mathcal{A}$.
  - *TagGen-Queries*: The identity ID, the private key SK_{ID}, and a chosen file identifier vid are submitted by $\mathcal{A}$. Following this, the algorithm **TagGen** is executed by $\mathcal{C}$, which results in the generation of the file tag τ. This tag is then returned to $\mathcal{A}$.
  - *Signing-Queries*: The identity ID, the private key SK_{ID}, the file tag τ, and a vector $\mathbf{v} \in V$ are provided by $\mathcal{A}$. Subsequently, the algorithm **Sign** is executed by $\mathcal{C}$, which generates the signature tags $t_1, \cdots, t_m$ that correspond to packets $\mathbf{v}_1, \cdots, \mathbf{v}_m$. The packet–tag pairs $\{(\mathbf{v}_i, t_i)\}_{i=1}^{m}$ are then returned to $\mathcal{A}$.

- **Output**: Finally, the adversary $\mathcal{A}$ produces an identity ID^*, a file identifier vid^* from V, a file tag τ^* from V, a packet $\mathbf{v}^* \in V$, and a tag t^*. The game is won by $\mathcal{A}$ if the following conditions are met.
  (1) **Verify**$(params, ID^*, vid^*, \tau^*, \mathbf{v}^*, t^*) \rightarrow 1$.
  (2) The identity ID^* does not appear in *Extract-Private-Key-Queries*.
  (3) For all τ_k queried in *TagGen-Queries*, the file tag $\tau^* \neq \tau_k$ and $\mathbf{v}^* \neq 0$.
  (4) The file tag $\tau^* = \tau_k$ for some τ_k queried in *TagGen-Queries*, but $\mathbf{v} \notin V_k$. Here, V_k is the subspace spanned by all packets $\{\mathbf{v}_i\}_{i=1}^{m}$ with the same queried file tag τ_k, where $1 \leq k \leq m$.

We denote $\varepsilon_{\mathcal{A}} = Pr[\mathcal{A}wins]$ as the advantage of the adversary $\mathcal{A}$ wins the game **Game-IBNS**.

Definition 4 (Security against $\mathcal{A}$)**.** *For any PPT adversaries $\mathcal{A}$, if their advantages $\varepsilon_{\mathcal{A}}$ are negligible, then we say that the scheme IBNS is secure under chosen-identity and chosen-subspace attacks. In short, IBNS is a secure identity-based network coding scheme.*

4. A Concrete Scheme in the IoMT

In this section, according to the bilinear map and the IBNS, we propose a concrete identity-based network coding scheme applicable to the IoMT environment, then carry out the correctness analysis and security analysis.

4.1. An Overview

In the IoMT environment, the IBNS consists of three entities, namely the trusted authority, the data owner, and the medical cloud server. They can be defined as follows.

- TA (trusted authority): This entity functions as the center for key generation. Its responsibilities include the creation of the private key in accordance with the identity of the DO and the dissemination of the system's public parameters.
- DO (data owner): Consider a patient equipped with medical detection devices as the DO. They have the ability to send encrypted data files to medical cloud servers by gathering and encrypting health data. Notably in IBNS, the identity of the patient could be represented by their name or medical number. To alleviate the computational load on patients, a security mediator can assist in processing data files and creating packet–tag pairs for transmission.
- MCS (medical cloud server): The MCS consists of several cloud servers that store data files from patients. Their goal is to gather sufficient uncontaminated packets to restore the original data file. Once the data file is acquired, healthcare professionals like doctors and nurses have the ability to download these data files from the MCS, allowing them to stay informed about the patients' conditions.

Figure 2 illustrates the process of communication between the above three entities. Note that intermediate nodes are undertaking the verify-and-forward tasks in the network.

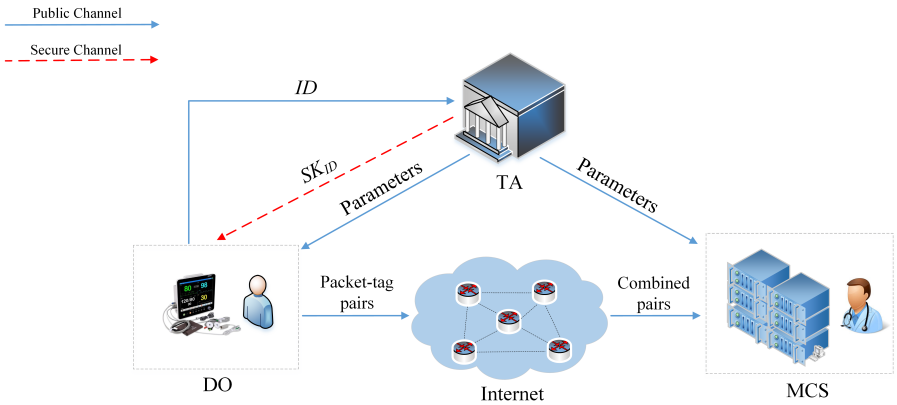


Figure 2. The system model of our proposed IBNS in an IoMT environment.

4.2. Our Proposed Scheme

In this subsection, we give a concrete construction of our proposed IBNS, which can be described as follows.

- **Setup**$(1^\lambda) \rightarrow (msk, params)$: This algorithm is executed by TA. When given a security parameter 1^λ, it produces system parameters params and TA's master key msk. Initially, it selects two cyclic groups G_1, G_2 of identical prime order q and a bilinear map e: $G_1 \times G_1 \rightarrow G_2$. Subsequently, TA opts for a generator $g \in G_1$, assigns $msk = s$, and then calculates $P = g^s$, which serves as a master public key. Define three hash functions as

$$H_1 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q^*, \tag{2}$$

$$H_2 : \{0,1\}^* \times \{0,1\}^* \times G_1 \rightarrow Z_q^*, \tag{3}$$

$$H_3 : \{0,1\}^* \times Z_n \times G_1 \rightarrow Z_q^*. \tag{4}$$

  Then, TA keeps msk secret, sets $w = e(g, g)$ and publishes the system parameters

$$params = (q, G_1, G_2, g, e, P, H_1, H_2, H_3, w). \tag{5}$$

- **Extract-Private-Key**$(params, msk, ID) \rightarrow SK_{ID}$: The Teaching Assistant (TA) also executes this algorithm. Specifically, it accepts the system parameters $params$, the master key msk, and a unique identifier ID for the Data Owner (DO). Following this, the TA makes a random selection of $x_{ID} \xleftarrow{r} Z_q^*$ and proceeds with the computation

$$R_{ID} = g^{x_{ID}}, \quad z_{ID} = x_{ID} + sH_1(ID||P||R_{ID}). \tag{6}$$

  Then, TA returns the private key $SK_{ID} = (x_{ID}, z_{ID})$ and R_{ID} to the DO through a secure channel.
- **TagGen**$(params, ID, SK_{ID}, vid) \rightarrow \tau$: The DO executes this algorithm. Concretely, it inputs the system parameters $params$, the identity ID, the private key SK_{ID} and a subspace identifier vid, then it verifies

$$g^{z_{ID}} \stackrel{?}{=} R_{ID} \cdot P^{H_1(ID||P||R_{ID})}. \tag{7}$$

  Hereafter, the DO randomly chooses $x \xleftarrow{r} Z_q^*$, sets $E = w^x$ and computes

$$h = H_2(ID||vid||E), \quad \tau = \frac{R_{ID}}{g^{z_{ID}+hx_{ID}}}. \tag{8}$$

  Finally, the DO outputs a file (subspace) tag τ.
- **Sign**$(params, ID, SK_{ID}, \tau, \mathbf{v}) \rightarrow t$: The DO is responsible for executing this algorithm. Initially, it partitions the file F with tag τ into m vectors, denoted as $\bar{\mathbf{v}}_1, \cdots, \bar{\mathbf{v}}_m$. Each vector can be interpreted as an n-dimensional vector $\bar{\mathbf{v}}_i = (v_{i,1}, \cdots, v_{i,n}) \in Z_q^n$. Following this, the DO extends each $\bar{\mathbf{v}}_i$ to

$$\mathbf{v}_i = (\bar{\mathbf{v}}_i, v_{i,n+1}, v_{i,n+2}, \cdots, v_{i,n+m}) \in Z_q^{n+m}, \tag{9}$$

  where $1 \leq i \leq m$. Subsequently, it defines $V = span\{\mathbf{v}_1, \cdots, \mathbf{v}_m\}$ as the subspace spanned by the m packets. It then generates signature tags $t_i = (t_{i,1}, t_{i,2})$ for each packet $\mathbf{v}_i \in V$ as

$$t_{i,1} = w^{\frac{x \cdot x_{ID}}{x + x_{ID}} \cdot \left(\sum_{j=1}^{n} v_{i,j} H_3(ID||j||\tau)\right)}, \tag{10}$$

$$t_{i,2} = w^{\frac{x^2}{x + x_{ID}} \cdot \left(\sum_{j=n+1}^{n+m} v_{i,j} H_3(ID||j||\tau)\right)}. \tag{11}$$

Finally, the DO sends the subspace tag τ, the two public parameters R_{ID} and E, the packets $\mathbf{v}_1, \cdots, \mathbf{v}_m$ and their corresponding tags $t_1, \cdots, t_m$ to all neighboring nodes. When designing the signature, we first divide the file F into m vectors. The function of the partition vector is to facilitate the node in checking the data. In the implementation and development process, we used Python 3.8 and numpy for vector partitioning. It should be noted that the vector's size depends on the file size and does not have a fixed value. We use bilinear mapping to generate signature tags. We use pypbc in Ubuntu 20.04 LTS 64-bit operating system to achieve bilinear mapping. The operation of bilinear mapping can be easily realized by using pypbc. In the process of implementation, we first initialize the bilinear map, generating g and w. Then, we generate signature tags t_i through bilinear mapping using the way of large integer operation.

- **Combine**$\left(params, ID, vid, \tau, \{c_i, \mathbf{v}_i, t_i\}_{i=1}^{l}\right) \rightarrow (\hat{\mathbf{v}}, \hat{t})$: This algorithm is performed by intermediate nodes. It first checks the validity of received packet–tag pairs $\{(\mathbf{v}_i, t_i)\}_{i=1}^{l}$. If the check passes, the intermediate node randomly selects l combination coefficients $c_1, \cdots, c_l \xleftarrow{r} Z_q$ and computes the combined packet–tag pairs

$$\hat{\mathbf{v}} = \sum_{i=1}^{l} c_i \mathbf{v}_i \in V, \tag{12}$$

$$\hat{t}_1 = \prod_{i=1}^{l} t_{i,1}^{c_i}, \quad \hat{t}_2 = \prod_{i=1}^{l} t_{i,2}^{c_i}. \tag{13}$$

Then, the intermediate node forwards $(\hat{\mathbf{v}}, \hat{t}) = (\hat{\mathbf{v}}, (\hat{t}_1, \hat{t}_2))$ to adjacent nodes in the network.

- **Verify**$(params, ID, vid, \tau, \hat{\mathbf{v}}, \hat{t}) \rightarrow 0/1$: This algorithm is executed by either intermediate nodes or MCSs, and two situations need to be taken into account.

$$e\left(\tau, R_{ID}^{h} \cdot P^{H_1(ID||P||R_{ID})}\right) = w, \tag{14}$$

$$t_{i,1}^{\frac{1}{\sum_{j=1}^{n} v_{i,j} H_3(ID||j||\tau)}} \cdot t_{i,2}^{\frac{1}{\sum_{j=n+1}^{n+m} v_{i,j} H_3(ID||j||\tau)}} = E. \tag{15}$$

Upon receiving the combined packet–tag pairs, the intermediate node initially separates $(\hat{\mathbf{v}}, \hat{t})$ into $(\hat{\mathbf{v}}, (\hat{t}_1, \hat{t}_2))$, then it checks if Equations (14) and (15) are valid. If they are, it outputs 1, and the intermediate node merges all clean packets into the combined one and computes its signature tag. At the same time, it discards contaminated packets and forwards the new mixed packet–tag pairs to neighboring nodes. Once the MCS obtains the combined packet–tag pairs, it first inspects the subspace V's tag τ, then it checks if Equations (14) and (15) are valid. As soon as it gathers sufficient clean data packets, it can restore the original file. Concurrently, it discards the contaminated data packets.

And in the design and implementation of the signature verification process. We still use Pypbc in the Ubuntu 20.04 LTS 64-bit operating system. The characteristics of digital signatures determine that the verification results that need to be satisfied in the design process of the verification algorithm are $e(g, g)$ and E. In implementing the verification algorithm, we still use the large integer operation. Large integer operation and bilinear mapping are the main techniques used to implement the algorithm.

In addition, our validation algorithm only needs to use signature tags t_i. In addition, F is not used directly in the signature process, but *tau* is signed. An essential feature of homomorphic signatures is that they are signed without using encrypted data directly. Our scheme using bilinear mapping also ensures that our homomorphic signature can satisfy the multiplicative homomorphism.

5. Correctness Analysis and Security Analysis

5.1. Correctness Analysis

In the correctness proof, we first explain the correctness of the verification algorithm. To verify the correctness of the algorithm, we only need to prove $e\left(\tau, R_{ID}^{h} \cdot P^{H_1(ID\|P\|R_{ID})}\right) = w$ and $t_{i,1}^{\frac{1}{\sum_{j=1}^{n} v_{i,j} H_3(ID\|j\|\tau)}} \cdot t_{i,2}^{\frac{1}{\sum_{j=n+1}^{n+m} v_{i,j} H_3(ID\|j\|\tau)}} = E$ is established. Since the signature algorithm needs to ensure that the calculation process of signature and verification is reciprocal, according to this feature, we need to explain the correctness of the signature algorithm $(\hat{t}_1, \hat{t}_2) \Leftrightarrow$ **Sign** $left(params, ID, SK_{ID},\ tau,\ hat\ textbf{fn}\ right)$.

The security of our proposed IBNS is mainly analyzed from the following aspects:

(1) Given the file tag τ of the subspace V, then

$$\begin{aligned} e\left(\tau, R_{ID}^{h} \cdot P^{H_1(ID||P||R_{ID})}\right) &= e\left(g^{x_{ID}-(z_{ID}+hx_{ID})}, g^{hx_{ID}+sH_1(ID||P||R_{ID})}\right) \\ &= e\left(g^{-hx_{ID}-sH_1(ID||P||R_{ID})}, g^{hx_{ID}+sH_1(ID||P||R_{ID})}\right) \\ &= e(g,g) = w. \end{aligned} \tag{16}$$

(2) If the packet signature is output from the algorithm **Sign**, then

$$t_{i,1}^{\frac{1}{\sum_{j=1}^{n} v_{i,j} H_3(ID||j||\tau)}} \cdot t_{i,2}^{\frac{1}{\sum_{j=n+1}^{n+m} v_{i,j} H_3(ID||j||\tau)}} = e(g,g)^{\frac{x^2+x\cdot x_{ID}}{x+x_{ID}}} = e(g,g)^{x} = E. \tag{17}$$

(3) If the packet signature is output from the algorithm **Combine**, then $(\hat{t}_1, \hat{t}_2) \Leftrightarrow$ **Sign** $(params, ID, SK_{ID}, \tau, \hat{\mathbf{v}})$, that is

$$\begin{aligned} \hat{t_1} &= \prod_{s=1}^{m} t_{i,1}^{c_s} = \prod_{s=1}^{m} e(g,g)^{\frac{x\cdot x_{ID}}{x+x_{ID}} \cdot \left(c_s \sum_{j=1}^{n} v_{s,j} H_3(ID||j||\tau)\right)} \\ &= e(g,g)^{\frac{x\cdot x_{ID}}{x+x_{ID}} \cdot \sum_{s=1}^{m} \left(c_s \sum_{j=1}^{n} v_{s,j} H_3(ID||j||\tau)\right)} = e(g,g)^{\frac{x\cdot x_{ID}}{x+x_{ID}} \cdot \left(\sum_{j=1}^{n} \hat{\mathbf{v}}_j H_3(ID||j||\tau)\right)}. \end{aligned} \tag{18}$$

$$\begin{aligned} \hat{t_2} &= \prod_{s=1}^{m} t_{i,2}^{c_s} = \prod_{s=1}^{m} e(g,g)^{\frac{x^2}{x+x_{ID}} \cdot \left(c_s \sum_{j=n+1}^{n+m} v_{s,j} H_3(ID||j||\tau)\right)} \\ &= e(g,g)^{\frac{x^2}{x+x_{ID}} \cdot \sum_{s=1}^{m} \left(c_s \sum_{j=n+1}^{n+m} v_{s,j} H_3(ID||j||\tau)\right)} = e(g,g)^{\frac{x^2}{x+x_{ID}} \cdot \left(\sum_{j=n+1}^{n+m} \hat{\mathbf{v}}_j H_3(ID||j||\tau)\right)}. \end{aligned} \tag{19}$$

5.2. Security Analysis

During the security proof process, we performed a security analysis based on the "Game-IBNS" mentioned in the "Security Model" section. According to the setting of "Game-IBNS", the security proof needs to be expressed in terms of "Queries" and "Output". Moreover, in the process of security proof, we introduce the mimic $\mathcal{B}$; the mimic will reference the behavior of the adversary $\mathcal{A}$. During the process of "Queries", we assume the result that the mimic $\mathcal{B}$ might give. In "Output", we give all the possibilities for $\mathcal{A}$ to win in "Game-IBNS". Finally, we pass on whether $\mathcal{A}$ can win in "Game-IBNS".

Now, we provide proof that our proposed IBNS is secure in the random oracle model.

Theorem 1. *Suppose $\mathcal{A}$ is the adversary who can break our proposed IBNS; his advantage is at most $\varepsilon_{\mathcal{A}}$ and running time is at most η. If the k-CAA problem is $(\varepsilon_{\mathcal{A}}, \eta)$-hard, then the IBNS is $(\eta, q_{ke}, q_{H_1}, q_{\eta}, \varepsilon_{\mathcal{A}})$-secure. In other words, for any $\varepsilon_{\mathcal{A}}$ and η satisfying*

$$\varepsilon'_{\mathcal{A}} \geq \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ke}} \left(1 - \frac{1}{q_{\eta}+1}\right)^{q_{\eta}} \frac{1}{(q_{\eta}+1)q_{H_1}} \varepsilon_{\mathcal{A}}, \tag{20}$$

where q_{H_1}, q_{ke}, q_η indicate the number of queries respecting to H_1-Queries, Extract-Key-Queries and TagGen-Queries.

Proof. Consider $params = (q, e, G_1, G_2, g)$ as the public parameters of the system. If $\mathcal{A}$ is a PPT adversary, we designate another PPT adversary $\mathcal{B}$ as a simulator.

Let $k+1 = q_{H_2}$, and given the k-CAA problem generator g, $W = g^a$ and k pairs $\left(b_1, g^{(a+b_1)^{-1}}\right), \cdots, \left(b_k, g^{(a+b_k)^{-1}}\right)$ as input, the simulator $\mathcal{B}$ produces a new pair $\left(b^*, g^{(a+b^*)^{-1}}\right)$, where b^* is not in $b_1, b_2, \cdots, b_k$. Then, $\mathcal{B}$ selects the master public key $P = W$ and provides $params = (q, e, G_1, G_2, g, W, w, H_3)$ to $\mathcal{A}$, where $w = e(g, g)$ and $H_3 : \{0,1\}^* \times Z_n \times G_1 \to Z_q^*$ is a hash function. In addition, the hash functions H_1 and H_2 are treated as random oracles. $\mathcal{A}$ can make the following queries adaptively.

- **H_1-Queries**. Given the i-th identity ID_i as the identity of $\mathcal{A}$, and it guesses the challenge identity in η-th H_1 queries, where $\eta \in \{1, \cdots, q_{H_1}\}$, $\mathcal{B}$ maintains a list L_{H_1} including the tuple (ID_i, R_{ID_i}, b_i) and initializes L_{H_1} empty. For a H_1 query on (ID_i, R_{ID_i}), $\mathcal{B}$ checks if (ID_i, R_{ID_i}) exists in L_{H_1}. If it is, $\mathcal{B}$ returns b_i to $\mathcal{A}$. Otherwise, $\mathcal{B}$ randomly chooses a value $b_i \xleftarrow{r} Z_q^*$, returns it to $\mathcal{A}$ and adds (ID_i, R_{ID_i}, b_i) to L_{H_1}.
- **Extract-Private-Key-Queries**. $\mathcal{B}$ maintains a list L_1 including the tuple (ID_i, x_{ID_i}) and initializes L_1 empty. For an identity ID_i query from $\mathcal{A}$, $\mathcal{B}$ first checks if (ID_i, x_{ID_i}) exists in L_1. If it is, $\mathcal{B}$ returns x_{ID_i} to $\mathcal{A}$. Otherwise, $\mathcal{B}$ randomly chooses $x_{ID_i} \xleftarrow{r} Z_q^*$, returns it to $\mathcal{A}$, and adds (ID_i, x_{ID_i}) to L_1.
- **H_2-Queries**. $\mathcal{B}$ holds a list L_{H_2} consisting of $(ID_i, vid_i, E_i, \beta_i, c_i)$ and initializes L_{H_2} empty. For a query on the tuple (ID_i, vid_i, E_i), $\mathcal{B}$ checks if the tuple exists in L_{H_2}. If it is, $\mathcal{B}$ returns β_i to $\mathcal{A}$. Otherwise, $\mathcal{B}$ answers as follows.
  - If $i \neq \eta$, $\mathcal{B}$ randomly selects $\beta_i \xleftarrow{r} Z_q^*$ and sends β_i to $\mathcal{A}$. Finally, it adds $(ID_i, vid_i, E_i, \beta_i, \perp)$ into L_{H_2}.
  - If $i = \eta$, $\mathcal{B}$ randomly selects $c_i \xleftarrow{r} \{0,1\}$ and sets ξ equal to the probability when $c_i = 1$. Similarly, $1 - \xi$ equals the probability when $c_i = 0$.
    (1) If $c_i = 1$, $\mathcal{B}$ randomly chooses a $b_i' \xleftarrow{r} Z_q^*$, where $b_i' \notin \{b_1, b_2, \cdots, b_k\}$. Then, it continues to compute $\beta_i = b_i b_i' / x_{ID_i}$ and returns $H_2(ID_i || vid || E_i) = \beta_i$ to $\mathcal{A}$. Finally, it adds the tuple $(ID_i, vid_i, c_i, \beta_i, E_i)$ into L_{H_2}.
    (2) If $c_i = 0$, $\mathcal{B}$ randomly picks $b_i' \in \{b_1, b_2, \cdots, b_k\}$ which has not been chosen. Then, it computes $\beta_i = b_i b_i' / x_{ID_i}$ and assigns $H_2(ID_i || vid || E_i) = \beta_i$. Lastly, it returns b_i' to $\mathcal{A}$ and adds the tuple $(ID_i, vid_i, c_i, \beta_i, E_i)$ to L_{H_2}.
- **TagGen-Queries**. $\mathcal{A}$ holds a list L_3 including a tuple $(ID_i, vid_i, V, x_i, \tau_i)$ and initializes L_3 empty. $\mathcal{A}$ can adaptively select an identity ID_i, an identifier vid_i of the subspace V and a user's private key x_{ID_i}. If there exists the tuple (ID_i, vid_i, V) in L_3, then $\mathcal{B}$ returns τ_i to $\mathcal{A}$. Else, $\mathcal{B}$ answers as follows.
  - If $t = \eta$, it implies that $\mathcal{B}$ does not possess the private key. Later, $\mathcal{B}$ iterates the list L_{H_2}.
    (1) If $c_\eta = 1$, then $\mathcal{B}$ aborts the simulation and outputs $\perp$.
    (2) If $c_\eta = 0$, then $\mathcal{B}$ randomly selects $x_i \xleftarrow{r} Z_q^*$ and calculates $E_i = w^{x_i}$. Thereafter, it verifies if the tuple (ID_i, vid_i, E_i) is present in L_{H_2}. If it is, it reselects $x_i \xleftarrow{r} Z_q^*$ at random, imitates $\mathcal{A}$ to perform a H_2 query of the tuple (ID_i, vid_i, E_i) and extracts the β_i from L_{H_2}. Ultimately, $\mathcal{B}$ computes the subspace V's tag

$$\tau_i = \left(g^{\frac{1}{(a+b_i')}} \right)^{\frac{1}{b_i}}. \tag{21}$$

The correctness of τ_i can be confirmed from

$$e\left(\tau_i, R_{ID_i}^{\beta_i} \cdot P^{b_i}\right) = e\left(g^{\frac{1}{(a+b_i')b_i}}, g^{\beta_i x_{ID_i} + ab_i}\right) = w. \tag{22}$$

- **Signing-Queries**. $\mathcal{A}$ adaptively selects an identity ID_i, an idenfifier of the subspace V, an subspace V's output tag τ of *TagGen-Queries* and the private key x_{ID_i}.
  - If $i \neq \eta$, $\mathcal{B}$ extracts x_i and τ_i from L_3 and performs as follows.
    (1) Given the packets $\mathbf{v}_1, \cdots, \mathbf{v}_m \in V$, $\mathcal{B}$ computes their signature tags $t_{i,j} = \left(t_{i,j1}, t_{i,j2}\right)$, where
    $$t_{i,j1} = e(g,g)^{\frac{x_i \cdot x_{ID_i}}{x_i + x_{ID_i}} \cdot \left(\sum_{k=1}^{n} v_{j,k} H_3(ID_i||k||\tau_i)\right)}, \tag{23}$$
    $$t_{i,j2} = e(g,g)^{\frac{x_i^2}{x_i + x_{ID_i}} \cdot \left(\sum_{k=n+1}^{n+m} v_{j,k} H_3(ID_i||k||\tau_i)\right)}. \tag{24}$$
    (2) $\mathcal{B}$ returns $t_{i,1}, \cdots, t_{i,m}$ to $\mathcal{A}$.
    The correctness of t_i can be verified from
    $$t_{i,j1}^{\frac{1}{\sum_{k=1}^{n} v_{j,k} H_3(ID_i||k||\tau_i)}} \cdot t_{i,j2}^{\frac{1}{\sum_{j=n+1}^{n+m} v_{j,k} H_3(ID_i||k||\tau_i)}} = E_i. \tag{25}$$
  - If $i = \eta$, it means $\mathcal{B}$ does not know the private key. $\mathcal{B}$ extracts x_i and τ_i from L_3 and answers as follows.
    (1) Given the packets $\mathbf{v}_1, \cdots, \mathbf{v}_m \in V$, $\mathcal{B}$ computes their signature tags $t_{i,j} = \left(t_{i,j1}, t_{i,j2}\right)$, where
    $$t_{i,j1} = e(g,g)^{\frac{x_i \cdot x_{ID_i}}{x_i + x_{ID_i}} \cdot \left(\sum_{k=1}^{n} v_{j,k} H_3(ID_i||k||\tau_i)\right)}, \tag{26}$$
    $$t_{i,j2} = e(g,g)^{\frac{x_i^2}{x_i + x_{ID_i}} \cdot \left(\sum_{k=n+1}^{n+m} v_{j,k} H_3(ID_i||k||\tau_i)\right)}. \tag{27}$$
    (2) $\mathcal{B}$ sends $t_{i,1}, \cdots, t_{i,m}$ back to $\mathcal{A}$.
    The correctness of t_i can be verified from
    $$t_{i,j1}^{\frac{1}{\sum_{k=1}^{n} v_{j,k} H_3(ID_i||k||\tau_i)}} \cdot t_{i,j2}^{\frac{1}{\sum_{j=n+1}^{n+m} v_{j,k} H_3(ID_i||k||\tau_i)}} = E_i. \tag{28}$$
- **Output**. At last, $\mathcal{A}$ outputs $(ID^*, vid^*, \tau^*, \mathbf{v}^*, t^*)$, and it wins the game if only if the following conditions are established.
  (1) The identity ID^* has never been queried in *Extract-Private-Key-Queries*.
  (2) The adversary never queried $(ID^*, x_{ID^*}, vid^*, V^*)$ in *TagGen-Queries*.
  (3) The adversary has queried $(ID^*, x_{ID^*}, vid^*, V^*)$ in *TagGen-Queries*, but $\tau \neq \tau^*$.
  (4) **Verify**$(params, ID^*, vid^*, \tau^*, \mathbf{v}^*, t^*) \rightarrow 1$.

$$\tau^* = g^{\frac{1}{z_{ID^*} + \beta^* x_{ID^*} - x_{ID}^*}} = g^{\frac{1}{(a+b^*)b^*}} \Rightarrow \tau^{*b^*} = g^{\frac{1}{(a+b^*)}}. \tag{29}$$

  - If $ID^* \neq ID_\eta$, then $\mathcal{B}$ cancels the simulation and outputs $\perp$.
  - If $ID^* = ID_\eta$, then $\mathcal{B}$ iterates L_{H_2}. If $c^* = 0$, it cancels the simulation and outputs $\perp$. Otherwise, if $c^* = 1$ and $b^* \notin \{b_1, b_2, \cdots, b_k\}$, then it outputs $\left(b^*, \tau^{*b^*}\right)$ as the solution of the k-CAA problem.
- **Probability-Analysis**. If the adversary $\mathcal{B}$ obtains the solution of the k-CAA problem, then the following events should be satisfied: (1) Ω_1: $\mathcal{B}$ has never output $\perp$.

(2) Ω_2: The adversary wins the game. (3) Ω_3: $ID^* = ID_\eta$ and $c^* = 1$. In this way, the probability of $\mathcal{B}$ succeeding can be indicated as

$$\begin{aligned} \varepsilon_{\mathcal{A}} &= Pr[\Omega_1 \wedge \Omega_2 \wedge \Omega_3] \\ &= Pr[\Omega_1] \cdot Pr[\Omega_2|\Omega_1] \cdot Pr[\Omega_3|\Omega_1 \wedge \Omega_2]. \end{aligned} \tag{30}$$

- If the event Ω_1 occurs, then we consider the following situations.
  (1) The adversary $\mathcal{B}$ never outputs $\perp$ in *Extract-Private-Key-Queries*, and the probability of terminating the simulation equals $\left(1 - 1/q_{H_1}\right)^{q_{ke}}$.
  (2) The adversary $\mathcal{B}$ never outputs $\perp$ in *TagGen-Queries*, and the probability of terminating the simulation is $\left(1 - \left(1/q_{H_1}\right) \cdot \xi\right)^{q_\eta}$.

  Hence, we have

$$Pr[\Omega_1] = \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ke}} \left(1 - \frac{1}{q_{H_1}}\xi\right)^{q_\eta} \geq \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ke}} (1 - \xi)^{q_\eta}. \tag{31}$$

- If Ω_2 occurs, then $Pr[\Omega_2|\Omega_1] = \varepsilon_{\mathcal{A}}$.
- If Ω_3 occurs, then $Pr[\Omega_3|\Omega_1 \wedge \Omega_2] = \frac{\xi}{q_{H_1}}$.

To sum up, we have

$$\varepsilon_{\mathcal{A}'} \geq \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ke}} (1 - \xi)^{q_\eta} \frac{\xi}{q_{H_1}} \varepsilon_{\mathcal{A}}. \tag{32}$$

Specifically, when $\xi = \frac{1}{q_\eta + 1}$, the function $(1 - \xi)^{q_\eta}\xi$ obtains the maximum value. Consequently, we can obtain that

$$\varepsilon'_{\mathcal{A}} \geq \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ke}} \left(1 - \frac{1}{q_\eta + 1}\right)^{q_\eta} \frac{1}{(q_\eta + 1)q_{H_1}} \varepsilon_{\mathcal{A}}. \tag{33}$$

The proof of Theorem 1 ends here, and our proposed scheme IBNS is secure. □

The above proof shows that the probability of the opponent $\mathcal{A}$ winning IBNS-Gaming in PPT is negligible. The security analysis results show that our scheme is secure under the k-CAA assumption. On the one hand, the IBNS introduces intermediate nodes, whose role is to examine medical data. The IBNS can prevent pollution attacks through inspection by intermediate nodes. On the other hand, through A security analysis based on the k-CAA assumption, we conclude that the probability of the opponent $\mathcal{A}$ winning IBNS-Gaming in PPT is negligible. k-CAA assumes that the security analysis can prevent adaptive chosen-message attacks to a certain extent. Therefore, our scheme has strong security.

6. Performance Analysis

We divide the performance analysis into two parts: "Theoretical Analysis" and "Experimental Analysis". In "Theoretical Analysis", we present an analysis of the time overhead required for relevant operations. "Theoretical Analysis" of "time overhead" includes the execution cost of a hash operation, bilinear pairing, exponential operation, and multiplicative operation. In addition, we compared IBNS with [31,32] schemes. In order to verify our correlation analysis for "Theoretical Analysis", we conducted simulation tests for the relevant steps in "Experimental Analysis". The simulation test experiments include the Extract-Private-Key, TagGen, and Signature phases. The relevant results of the experiment are presented in the form of time overhead. We also compare the results of our experiments with the relevant schemes and prove that IBNS has better efficiency than other schemes.

6.1. Theoretical Analysis

In IBNS design requirements, in addition to security, the most important purpose is to meet the current IoMT running performance requirements, and one of the most important indicators to measure computing performance is the running time of related operations. Therefore, we analyze the time taken for each operation. We believe that the computing time of IBNS is mainly distributed in hash operation, bilinear pairing, exponential operation, and multiplicative operation.

We use T_h, T_p, T_e, and T_m to denote the execution cost of a hash operation, a bilinear pairing, an exponential operation, and a multiplicative operation, respectively. In addition, we omit the execution cost of hash operation to Z_q, additive operation on Z_q, and inverse operation on Z_q^*, because their cost is negligible compared with other operations. For convenience, we assume the file F will be split into m vectors, and each of them will be augmented as a $(m+n)$-dimensional vector. For convenience, we define $N = m + n$.

In our IBNS, the computational cost of the signature generation phase can be computed by the algorithm **TagGen** and the algorithm **Sign**. For a source node, the computational cost for signing equals $(2m+1) \cdot T_e$. For a destination node or intermediate node, we need to evaluate the algorithms **Combine** and **Verify**, which constitute the signature verification phase. According to our calculations, the computational cost of the signature verification phase is $2l \cdot T_m + (4l+2) \cdot T_e + T_p$. In the same way, we estimate the computational cost of schemes in [31,32].

The evaluation results are listed in Table 1. Obviously, our proposed IBNS is comparable to the other schemes in terms of computational cost. Hence, our IBNS can improve the efficiency of data transmission.

Table 1. Comparison of computation cost.

Scheme	Signature Generation Phase	Signature Verification Phase
[31]	NT_e	$(N+2l)T_e + T_h + 2(2l-1)T_m + 2T_p$
[32]	$m(N+3)T_e + (N+1)T_h + m(N+1)T_m$	$(N+l+2)T_e + (N+1)T_h + (N+l)T_m + 2T_p$
Our CLNS	$(2m+1)T_e$	$(4l+2)T_e + 2T_h + 2lT_m + T_p$

6.2. Experimental Analysis

In this subsection, we conduct experiments on the processes of the above signature generation phase and signature verification phase. Based on the "Charm" [58,59] framework, all experiments are implemented on a Lenovo laptop with Intel Core i5-8300H CPU @2.30 GHz, 8 GB RAM, Ubuntu 20.04 LTS 64-bit and Python 3.8. (The equipment manufacturer is Lenovo and the procurement place is Xi'an, China.)

6.2.1. Extract-Private-Key Phase

The Extract-Private-Key Phase is the basis for subsequent signature and Tag Generation steps. We believe that generating Extract-Private-Key is needed to meet the requirements of efficiency. If it takes a long time for the user to generate Extract-Private-Key, the uptime overhead of the system will increase significantly. Because of this, we need to test the generation time of Extract-Private-Key here. To test the performance of generating Extract-Private-Keys, we tested the time required to generate extract-private-keys of different lengths. The experimental results are shown in Figure 3. The experimental results show that the time required for generating a key of 128 bits, 192 bits, 256 bits, and 320 bits 5.1954 ms, 5.4758 ms, 5.7013 ms, and 5.9954 ms, respectively. Therefore, the generation time of Extract-Private-Key is acceptable.

In addition, we compare the performance differences between different schemes in the Extract-Private-Key Phase. We select the private key generation schemes of Li's [53] and Qiu's [56] schemes and compare the time cost. Our default key length is 256 bits. Relevant experimental results are shown in Figure 4. From the figure, we can see that the time cost of Li's scheme is 7.593 milliseconds, and the time cost of Qiu's scheme is

6.429 milliseconds. The key generation time for IBNS is 5.701 milliseconds. Therefore, IBNS has a significant performance advantage in key generation.

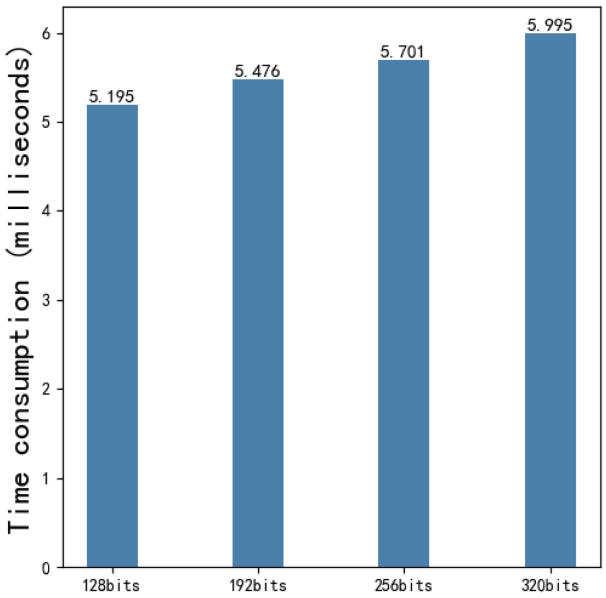


Figure 3. Time consumption of the Extract-Private-Key phase.

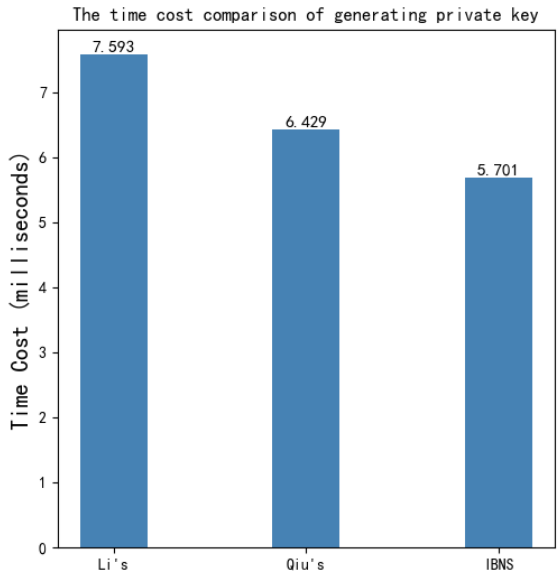


Figure 4. The time cost comparison of Private Key Generation.

6.2.2. TagGen Phase

The TagGen Phase is also the basis for subsequent signature and verification algorithms. For the same reason as testing the Extract-Private-Key Phase, we test the TagGen Phase to ensure that the steps before signing do not take too much time, thus keeping the system efficient. To test the performance of generating a tag, we tested the average tag generation time. We conducted 50 groups of experiments; each group of experiments needs to record 100 instances of tag generation time, and they take the average value as the tag generation time of each group of experiments. Figure 5 shows that the average time consumed in 50 groups of experiments is 10.6664 ms. It can be seen that the tag generation time of our scheme does not significantly increase the running time cost of the system.

In addition, we compare the performance differences between different schemes in the TagGen Phase. We choose the label generation scheme of Tan's [60] and Yang's [61]

and compare the time cost. Relevant experimental results are shown in Figure 6. From the figure, we can see that the label generation time of Tan's scheme is 12.5749 milliseconds, and the label generation time of Yang's scheme is 13.6136 milliseconds. The IBNS label generation time is 10.6664 ms. From the above experimental results, it can be seen that IBNS can show good performance in the TagGen Phase.

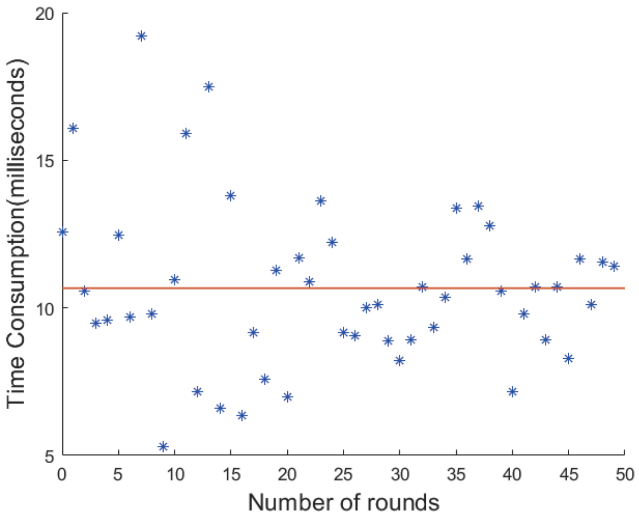


Figure 5. The average time of the TagGen phase. (the points indicate all the experimental results, and the lines reflect the average of all the experimental results.)

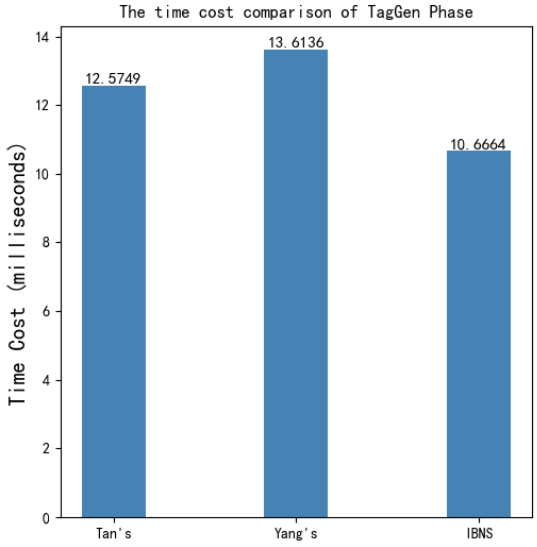


Figure 6. The time cost comparison of TagGen Phase.

6.2.3. Combine Phase

Combine requires the combination of many packet–tag pairs, which is an indispensable operation before transmission. As one of the important steps of network coding, the efficiency of Combine directly determines whether the encoding scheme is efficient in the transmission process. That is why we need to test the time it takes to combine. To test the performance of Combination, we tested the change in time after the number of packet–tag pairs was increased during combination. The experimental results are shown in Figure 7. Figure 7 shows that even in the case of the largest 50 packet–tag pairs, the time cost of the Combination is only 17.3947 ms. Therefore, Combination has no significant impact on the overall efficiency of IBNS.

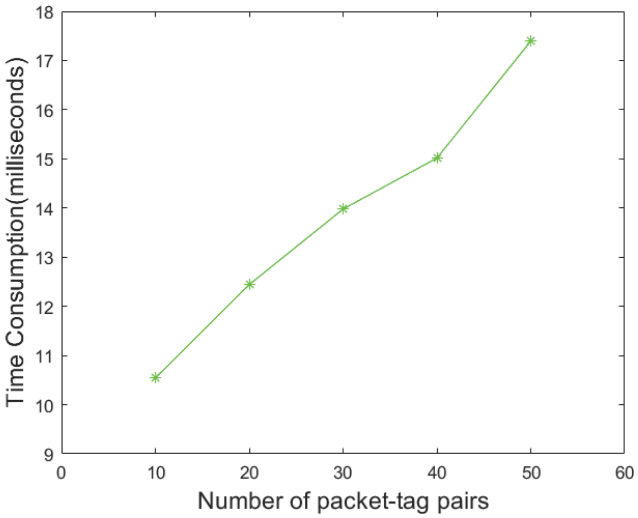


Figure 7. The average time of the Combine phase.

In addition, we compare the performance differences between different schemes in the combined phase. We choose the combined scheme of Jin's [62] and Li's [63] and compare the time cost. Let us assume that the number of packet–tag pairs is 50. Relevant experimental results are shown in Figure 8. From the figure, we can see that the combined time of Jin's scheme is 19.7198 milliseconds, and the combined time of Li's scheme is 20.3885 milliseconds. The IBNS combined time is 17.3947 ms. From the above experimental results, IBNS performed well in the combine phase.

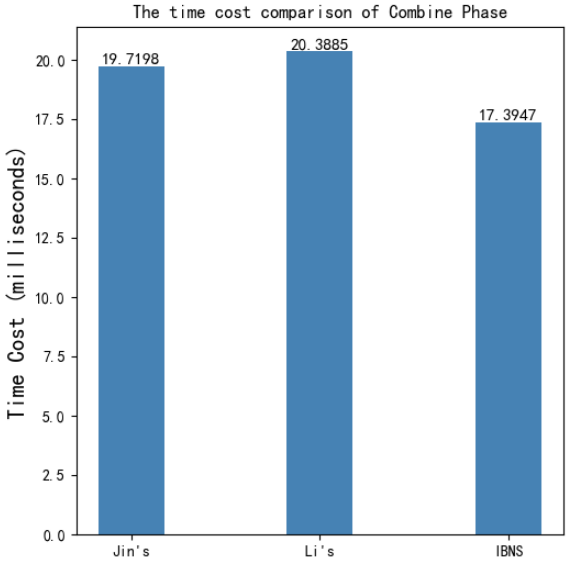


Figure 8. The time cost comparison of Combine Phase.

6.2.4. Signature Phase

As the core step of IBNS, the efficiency of the IBNS solution must be guaranteed. In order to test the efficiency of the IBNS signature algorithm, we need to compare it with similar schemes. We selected Jiang et al.'s [31] scheme and Lin et al.'s [32] scheme. To make the experiment more manageable, we equate the number of vectors to the dimension of each vector, i.e., $m = n$. We also opt for the 512-bit SS elliptic curve from the PBC library [59]. Simultaneously, we adjust the parameter m or (n) to simulate the signature generation phase and the signature verification phase. Specifically, we partition a 1 MB data file into blocks ranging from 10 to 70 in increments of 10. We then conduct 50 experiments on the same packet and compute the average time consumption for both phases. As illustrated

in Figure 9, our proposed IBNS shows a slight efficiency advantage over the other two schemes in the signature generation phase. Regarding the signature verification phase, our IBNS significantly enhances its efficiency, as depicted in Figure 10. In comparison, our IBNS is more practical and suitable for IoMT applications.

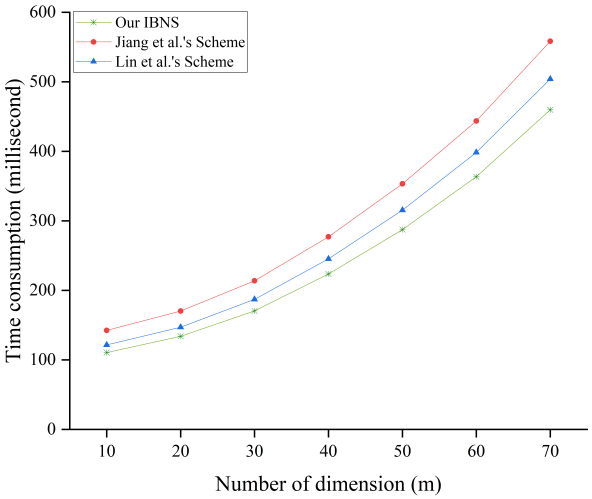


Figure 9. Time consumption of the signature generation phase. (The two schemes are [31] and [32].)

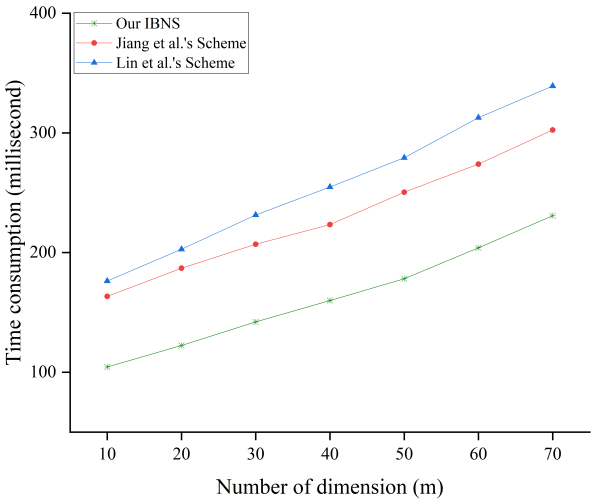


Figure 10. Time consumption of the signature verification phase. (The two schemes are [31] and [32].)

According to the results of Figure 9 and 10, the IBNS signature algorithm performs well. As the number of dimensions increases, the generation and verification time of the signature algorithm increases. However, IBNS has the lowest time overhead of the three scenarios, with a signature generation of only 462 milliseconds, even at 70 dimensions. Moreover, even at 70 dimensions, the signature verification is only 228 milliseconds. Therefore, the IBNS signature scheme has good performance.

Through theoretical analysis and experimental analysis, it can be seen that IBNS has good performance. In the performance analysis, we analyze the computation operation time of similar schemes, and through the analysis, we can see that IBNS has good performance in the signature algorithm. In order to prove the results of the theoretical analysis and the time required to detect other operating steps, we conducted a simulation experiment. In the simulation experiments, the results of various experiments show that IBNS has good performance. The conclusion above shows that IBNS can complete signature and coding in a short time under an IoMT environment. It should be noted that as IoMT continues to evolve, more medical data need to be encoded and transmitted quickly. Therefore, taking care of both security and performance is also a contribution of IBNS.

7. Discussion

In this section, we discuss the benifits of the IBNS, its meanings, and its limits.

7.1. Benefits

IBNS's strength lies in its ability to balance the need for security and efficiency.

Firstly, we design a lightweight identity-based network coding scheme. The highlight of IBNS is the signature algorithm and the verification algorithm. In the signature algorithm, we use homomorphic signatures. Moreover, intermediate nodes are set in the IBNS. Intermediate nodes can detect and filter polluted packets by combining all received signatures into a unified signature corresponding to the combined data packet. In the security analysis, we prove the scheme's security based on the k-CAA hypothesis, which shows that our scheme can prevent pollution attacks and adaptive chosen-message attacks. Because of these, IBNS can effectively prevent attacks.

Secondly, we give proof based on the k-CAA problem in the security analysis. In the proof process, we give the conditions the opponent $\mathcal{A}$ needs to meet during the IBNS game. At the same time, we also show the behavior of the mimic $\mathcal{B}$ in this process and the conditions that need to be met to solve the k-CAA problem. In Probability Analysis, we analyze whether $\mathcal{B}$ can solve the k-CAA problem in PPT. The relevant results indicate that IBNS is secure against the adaptive chosen-message attacks under the k-CAA assumption.

Finally, we analyzed the time overhead in the performance analysis section. Firstly, the time cost of the IBNS signature algorithm is analyzed theoretically, and the time cost of IBNS is compared with [31,32]. After that, we conducted simulation experiments on the Extract-Private-Key Phase, TagGen Phase, and Combine Phase of IBNS, and the experimental results showed that our scheme was efficient. In order to ensure that the signature algorithm is still effective in the multi-dimensional case, we tested the time overhead of the signature algorithm in the case of increased dimension, and the experimental results show that the signature algorithm of IBNS has higher efficiency.

7.2. The Meaning of IBNS

IBNS can be applied to IoMT. The main application scenario of IBNS is to meet the security and efficient transmission of IoMT. On the one hand, the security performance of IBNS can ensure the security of data transmission. In the IoMT environment, the transmission process is vulnerable to attacks [64,65], and IBNS can effectively prevent pollution attacks and adaptive chosen-message attacks. On the other hand, IBNS has good performance, and the relevant experimental data show the computational overhead of IBNS. In IoMT, data need to be transferred quickly [66,67]. IBNS has high signature and coding efficiency, which can effectively reduce communication time and ensure efficient data transmission.

IBNS also has challenges in practical application; the biggest problem is ensuring the fast transmission of data. To solve the problem of fast data transmission, we choose to segment medical data packets into several matrices. The IBNS combines the packets after delivery. This method can quickly complete the data transmission. In addition, ensuring the security of the key is also a challenge that needs to be considered. After investigation, handing the keys to the data owner can effectively avoid the security problems caused by centralized storage compared with the traditional centralized storage of keys.

7.3. Limits

Although the IBNS solves the security problem of data transmission in the IoMT environment, some points still need improvement. In data storage, the IBNS uses MCS, which are subject to potential attacks such as DDoS attacks, and dishonest MCS may also lead to data leakage. In addition, IBNS also needs to explore the area of protection against other attacks further. For example, we need to explore how IBNS can prevent MIMT, impersonation, and other attacks in future studies.

8. Conclusions

In IBNS, we propose a homomorphic signature and authentication algorithm for IoMT. Furthermore, based on the signature and verification algorithm, intermediate nodes are introduced to detect polluted packets by combining all received signatures into a unified signature that must correspond to the combined data packet. Secondly, in the process of security proof, we conduct security proof based on the k-CAA hypothesis. The security proof showed that the opponent could not solve IBNS-Gaming in PPT. Because of this, our scheme also protects against adaptive chosen-message attacks. Finally, we analyze the performance of IBNS, starting from theoretical analysis and experimental results. Both results indicate that IBNS performs well under the premise of ensuring security. IBNS is suitable for scenarios where efficient and secure transmission of medical data is required in an IoMT environment.

In the future work, we will continue to improve IBNS. On the one hand, we need to improve the data storage scheme of IBNS. We see blockchain technology as a technology that can be applied to data storage. On the other hand, we will continue improving the security mechanism of IBNS to cope with more security challenges.

Author Contributions: Conceptualization, G.B.; Methodology, K.W. and M.S.; Software, K.W. and M.S.; Validation, M.S. and B.S.; Formal analysis, B.S.; Resources, M.S. and G.B.; Data curation, K.W. and M.S.; Writing—original draft, K.W. and M.S.; Writing—review & editing, K.H.; Supervision, G.B.; Project administration, G.B.; Funding acquisition, G.B.All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Shaanxi Provincial Natural Science Basic Research Program Project (no. 2023-YBGY-021).

Data Availability Statement: The datasets of this article are available on request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Symbols and Definitions

Table A1. Symbols and definitions.

Symbols	Definitions
IoMT	Internet of Medical Things
TA	Trusted Authority
DO	Data Owner
MCS	Medical Cloud Server
IBHS	Identity-based Homomorphic Signature
PPT	Probabilistic Polynomial Time
k-CAA	Collusion Attack Algorithm with k-traitor
v_{id}	the identifier of a subspace
$\mathbf{v}_i$	the i-th packet
$v_{i,j}$	the j-th component of the i-th packet
t_i	the i-th signature of $\mathbf{v}_i$
1^λ	the security parameter
q	the sizeable prime number
G_1, G_2	cyclic groups
g	the generator of G_1
Z_q	the finite field $[0, 1, \cdots, q-1]$
e	the bilinear map from $G_1 \times G_1$ to G_2
w	$e(g, g)$ in G_2
Z_q^*	the finite field $[1, 2, \cdots, q-1]$
$\mathcal{A}, \mathcal{C}$	the adversary and the challenger
msk	the master secret key
$params$	the system public parameters
ID	the user's identity
SK_{ID}	ID's private key
$\perp$	the termination symbol

References

1. He, P.; Meng, S.; Cui, Y.; Wu, D.; Wang, R. Compression and Encryption of Heterogeneous Signals for Internet of Medical Things. *IEEE J. Biomed. Health Inform.* **2023**, 1–12. [CrossRef]
2. Wei, K.; Zhang, L.; Guo, Y.; Jiang, X. Health Monitoring Based on Internet of Medical Things: Architecture, Enabling Technologies, and Applications. *IEEE Access* **2020**, *8*, 27468–27478. [CrossRef]
3. Chaudhary, S.; Johari, R.; Bhatia, R.; Gupta, K.; Bhatnagar, A. CRAIoT: Concept, Review and Application(s) of IoT. In Proceedings of the 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 18–19 April 2019; pp. 1–4.
4. Ahlswede, R.; Cai, N.; Li, S.Y.; Yeung, R.W. Network information flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216. [CrossRef]
5. Peng, Y.; Wang, X.; Guo, L.; Wang, Y.; Deng, Q. An Efficient Network Coding-Based Fault-Tolerant Mechanism in WBAN for Smart Healthcare Monitoring Systems. *Appl. Sci.* **2017**, *7*, 817. [CrossRef]
6. Takabayashi, K.; Tanaka, H.; Sakakibara, K. Toward Dependable Internet of Medical Things: IEEE 802.15.6 Ultra-Wideband Physical Layer Utilizing Superorthogonal Convolutional Code. *Sensors* **2022**, *22*, 2172. [CrossRef]
7. Ar-Reyouchi, E.M.; Ghoumid, K.; Ar-Reyouchi, D.; Rattal, S.; Yahiaoui, R.; Elmazria, O. Protocol Wireless Medical Sensor Networks in IoT for the Efficiency of Healthcare. *IEEE Internet Things J.* **2022**, *9*, 10693–10704. [CrossRef]
8. Bahache, M.; Tahari, A.E.K.; Herrera-Tapia, J.; Lagraa, N.; Calafate, C.T.; Kerrache, C.A. Towards an Accurate Faults Detection Approach in Internet of Medical Things Using Advanced Machine Learning Techniques. *Sensors* **2022**, *22*, 5893. [CrossRef]
9. Mahmoud, M.S.; Hammood, D.A.; Alkhayyat, A. Integrating device to device network with internet of health things: Towards minimum power allocation. *Bull. Electr. Eng. Inform.* **2020**, *9*, 2334–2341. [CrossRef]
10. Alizadehsani, R.; Roshanzamir, M.; Izadi, N.H.; Gravina, R.; Kabir, H.D.; Nahavandi, D.; Alinejad-Rokny, H.; Khosravi, A.; Acharya, U.R.; Nahavandi, S.; et al. Swarm intelligence in internet of medical things: A review. *Sensors* **2023**, *23*, 1466. [CrossRef]
11. Vasudevan, V.A.; Tselios, C.; Politis, I. On security against pollution attacks in network coding enabled 5g networks. *IEEE Access* **2020**, *8*, 38416–38437. [CrossRef]
12. Zhou, X.; Zheng, X.; Shu, T.; Liang, W.; Wang, K.I.K.; Qi, L.; Shimizu, S.; Jin, Q. Information Theoretic Learning-Enhanced Dual-Generative Adversarial Networks with Causal Representation for Robust OOD Generalization. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–14. [CrossRef]
13. Aubret, A.; Matignon, L.; Hassas, S. An Information-Theoretic Perspective on Intrinsic Motivation in Reinforcement Learning: A Survey. *Entropy* **2023**, *25*, 327. [CrossRef]
14. Mielniczuk, J. Information Theoretic Methods for Variable Selection—A Review. *Entropy* **2022**, *24*, 1079. [CrossRef] [PubMed]
15. Ning, Z.; Shi, W.; Xiao, L.; Liang, W.; Weng, T.H. A novel approach for anti-pollution attacks in network coding. *Connect. Sci.* **2021**, *33*, 447–462. [CrossRef]
16. Chang, J.; Shao, B.; Ji, Y.; Bian, G. Comment on "A Tag Encoding Scheme Against Pollution Attack to Linear Network Coding". *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 2618–2619. [CrossRef]
17. Kolumban-Antal, G.; Lasak, V.; Bogdan, R.; Groza, B. A Secure and Portable Multi-Sensor Module for Distributed Air Pollution Monitoring. *Sensors* **2020**, *20*, 403. [CrossRef] [PubMed]
18. Cheng, C.; Lee, J.; Jiang, T.; Takagi, T. Security analysis and improvements on two homomorphic authentication schemes for network coding. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 993–1002. [CrossRef]
19. Esfahani, A.; Mantas, G.; Rodriguez, J. An efficient null space-based homomorphic MAC scheme against tag pollution attacks in RLNC. *IEEE Commun. Lett.* **2016**, *20*, 918–921. [CrossRef]
20. Wu, Y.; Chang, J.; Xue, R.; Zhang, R. Homomorphic MAC from Algebraic One-Way Functions for Network Coding with Small Key Size. *Comput. J.* **2017**, *60*, 1785–1800. [CrossRef]
21. Chang, J.; Ji, Y.; Xu, M.; Xue, R. General transformations from single-generation to multi-generation for homomorphic message authentication schemes in network coding. *Future Gener. Comput. Syst.* **2019**, *91*, 416–425. [CrossRef]
22. Cheng, Z.; Chi, K.; Tian, X.; Li, Y. Secure network coding based on homomorpuic signature against pollution attacks. In Proceedings of the 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Hangzhou, China, 30 October–1 November 2012; Volume 3, pp. 1092–1096.
23. Li, T.; Chen, W.; Tang, Y.; Yan, H. A homomorphic network coding signature scheme for multiple sources and its application in IoT. *Secur. Commun. Netw.* **2018**, *2018*, 9641273. [CrossRef]
24. Yehia, M.; AlTawy, R.; Aaron Gulliver, T. Hash-based signatures revisited: A dynamic fors with adaptive chosen message security. In Proceedings of the International Conference on Cryptology in Africa, Cairo, Egypt, 20–22 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 239–257.
25. Yu, H.; Ren, R. Certificateless elliptic curve aggregate signcryption scheme. *IEEE Syst. J.* **2021**, *16*, 2347–2354. [CrossRef]
26. Lukas, N.; Diaa, A.; Fenaux, L.; Kerschbaum, F. Leveraging optimization for adaptive attacks on image watermarks. *arXiv* **2023**, arXiv:2309.16952.
27. Kar, J.; Naik, K.; Abdelkader, T. A secure and lightweight protocol for message authentication in wireless sensor networks. *IEEE Syst. J.* **2020**, *15*, 3808–3819. [CrossRef]
28. Li, Y.; Huang, Y.; Tang, M.; Ren, S.; Xu, H. Adaptive Chosen Plaintext Side-Channel Attacks for Higher-Order Masking Schemes. In *Wireless Algorithms, Systems, and Applications: Proceedings of the 16th International Conference, WASA 2021, Nanjing, China, 25–27 June 2021*; Proceedings, Part II 16; Springer: Berlin/Heidelberg, Germany, 2021; pp. 173–185.

29. Shao, H.; Piao, C. A Provably Secure Lightweight Authentication Based on Elliptic Curve Signcryption for Vehicle-to-Vehicle Communication in VANETs. *IEEE Trans. Ind. Inform.* **2023**, *20*, 3738–3747. [CrossRef]
30. Shamir, A. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 47–53.
31. Jiang, Y.; Zhu, H.; Shi, M.; Shen, X.S.; Lin, C. An efficient dynamic-identity based signature scheme for secure network coding. *Comput. Netw.* **2010**, *54*, 28–40. [CrossRef]
32. Lin, Q.; Yan, H.; Huang, Z.; Chen, W.; Shen, J.; Tang, Y. An ID-based linearly homomorphic signature scheme and its application in blockchain. *IEEE Access* **2018**, *6*, 20632–20640. [CrossRef]
33. Chang, J.; Ma, H.; Zhang, A.; Xu, M.; Xue, R. RKA security of identity-based homomorphic signature scheme. *IEEE Access* **2019**, *7*, 50858–50868. [CrossRef]
34. Ashton, K. That 'internet of things' thing. *RFID J.* **2009**, *22*, 97–114.
35. Ghubaish, A.; Salman, T.; Zolanvari, M.; Unal, D.; Al-Ali, A.; Jain, R. Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet Things J.* **2020**, *8*, 8707–8718. [CrossRef]
36. Papaioannou, M.; Karageorgou, M.; Mantas, G.; Sucasas, V.; Essop, I.; Rodriguez, J.; Lymberopoulos, D. A survey on security threats and countermeasures in internet of medical things (IoMT). *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4049. [CrossRef]
37. Vishnu, S.; Ramson, S.J.; Jegan, R. Internet of medical things (IoMT)—An overview. In Proceedings of the 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 5–6 March 2020; pp. 101–104.
38. Al-Turjman, F.; Nawaz, M.H.; Ulusar, U.D. Intelligence in the Internet of Medical Things era: A systematic review of current and future trends. *Comput. Commun.* **2020**, *150*, 644–660. [CrossRef]
39. Yaacoub, J.P.A.; Noura, M.; Noura, H.N.; Salman, O.; Yaacoub, E.; Couturier, R.; Chehab, A. Securing internet of medical things systems: Limitations, issues and recommendations. *Future Gener. Comput. Syst.* **2020**, *105*, 581–606. [CrossRef]
40. Thamilarasu, G.; Odesile, A.; Hoang, A. An intrusion detection system for internet of medical things. *IEEE Access* **2020**, *8*, 181560–181576. [CrossRef]
41. Manickam, P.; Mariappan, S.A.; Murugesan, S.M.; Hansda, S.; Kaushik, A.; Shinde, R.; Thipperudraswamy, S. Artificial intelligence (AI) and internet of medical things (IoMT) assisted biomedical systems for intelligent healthcare. *Biosensors* **2022**, *12*, 562. [CrossRef] [PubMed]
42. Dwivedi, R.; Mehrotra, D.; Chandra, S. Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review. *J. Oral Biol. Craniofacial Res.* **2022**, *12*, 302–318. [CrossRef] [PubMed]
43. Saba, T.; Haseeb, K.; Ahmed, I.; Rehman, A. Secure and energy-efficient framework using Internet of Medical Things for e-healthcare. *J. Infect. Public Health* **2020**, *13*, 1567–1575. [CrossRef] [PubMed]
44. Ashfaq, Z.; Rafay, A.; Mumtaz, R.; Zaidi, S.M.H.; Saleem, H.; Zaidi, S.A.R.; Mumtaz, S.; Haque, A. A review of enabling technologies for Internet of Medical Things (IoMT) Ecosystem. *Ain Shams Eng. J.* **2022**, *13*, 101660. [CrossRef]
45. Jain, S.; Nehra, M.; Kumar, R.; Dilbaghi, N.; Hu, T.; Kumar, S.; Kaushik, A.; Li, C.Z. Internet of medical things (IoMT)-integrated biosensors for point-of-care testing of infectious diseases. *Biosens. Bioelectron.* **2021**, *179*, 113074. [CrossRef]
46. Rasool, R.U.; Ahmad, H.F.; Rafique, W.; Qayyum, A.; Qadir, J. Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML. *J. Netw. Comput. Appl.* **2022**, *201*, 103332. [CrossRef]
47. Johnson, R.; Molnar, D.; Song, D.; Wagner, D. Homomorphic signature schemes. In Proceedings of the Cryptographers' Track at the RSA Conference, San Jose, CA, USA, 18–22 February 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 244–262.
48. Liu, X.; Huang, J.; Wu, Y.; Zong, G. A privacy-preserving signature scheme for network coding. *IEEE Access* **2019**, *7*, 109739–109750. [CrossRef]
49. Wu, B.; Wang, C.; Yao, H. A certificateless linearly homomorphic signature scheme for network coding and its application in the IoT. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 852–872. [CrossRef]
50. Catalano, D.; Fiore, D.; Tucker, I. Additive-homomorphic functional commitments and applications to homomorphic signatures. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 5–9 December 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 159–188.
51. Chang, J.; Ji, Y.; Shao, B.; Xu, M.; Xue, R. Certificateless homomorphic signature scheme for network coding. *IEEE/ACM Trans. Netw.* **2020**, *28*, 2615–2628. [CrossRef]
52. Hébant, C.; Phan, D.H.; Pointcheval, D. Linearly-homomorphic signatures and scalable mix-nets. In Proceedings of the Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, 4–7 May 2020; Proceedings, Part II 23; Springer: Berlin/Heidelberg, Germany, 2020; pp. 597–627.
53. Li, R.; Wang, F.; Zhang, R.; Chen, K. Ntru-based fully homomorphic signature. *Secur. Commun. Netw.* **2022**, *2022*, 9942717. [CrossRef]
54. Wang, C.; Wu, B.; Yao, H. Leveled adaptively strong-unforgeable identity-based fully homomorphic signatures. *IEEE Access* **2020**, *8*, 119431–119447. [CrossRef]
55. Gu, Y.; Shen, L.; Zhang, F.; Xiong, J. Provably secure linearly homomorphic aggregate signature scheme for electronic healthcare system. *Mathematics* **2022**, *10*, 2588. [CrossRef]
56. Qiu, R.; Ai, M.; Zheng, F.; Liang, L.; Li, Y. Privacy-Preserving of Power Consumption Big Data Based on Improved Group Signature and Homomorphic Encryption. In Proceedings of the 2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 20–22 November 2020; pp. 25–29.

57. Trivedi, H.S.; Patel, S.J. Homomorphic cryptosystem-based secure data processing model for edge-assisted IoT healthcare systems. *Internet Things* **2023**, *22*, 100693. [CrossRef]
58. Akinyele, J.A.; Garman, C.; Miers, I.; Pagano, M.W.; Rushanan, M.; Green, M.; Rubin, A.D. Charm: A framework for rapidly prototyping cryptosystems. *J. Cryptogr. Eng.* **2013**, *3*, 111–128. [CrossRef]
59. Lynn, B. On the Implementation of Pairing-Based Cryptosystems. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2007.
60. Tan, P.; Qin, Z.; Wan, W.; Zhang, S.; Zhang, J.; Xia, J. An Improved Certificateless Partial Blind Signature Scheme Based on Homomorphic Encryption. In Proceedings of the International Conference on Artificial Intelligence and Security, Qinghai, China, 15–20 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 207–221.
61. Yang, X.; Zhou, H.; Ren, N.; Tian, T. Homomorphic proxy re-signcryption scheme and its application in edge computing-enhanced IoT. In Proceedings of the 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 27–29 December 2021; pp. 644–649.
62. Jin, X.; Wang, F.; Zhang, R.; Lian, B.; Chen, K. Leveled fully homomorphic signcryption from lattices. *IEEE Access* **2023**, *11*, 35232–35242. [CrossRef]
63. Li, L.; Wang, X.; Chen, T. AFHPS: An Authorized Function Homomorphic Proxy Signature Scheme with Sampling Batch Verification. In Proceedings of the International Conference on Frontiers in Cyber Security, Chengdu, China, 21–23 August 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 295–308.
64. Gaytan, J.T. A literature survey of security and privacy issues in internet of medical things. *Int. J. Comput. Inf. Manuf.* **2022**, *2*. [CrossRef]
65. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A.; Yelamarthi, K. Prospect of internet of medical things: A review on security requirements and solutions. *Sensors* **2022**, *22*, 5517. [CrossRef] [PubMed]
66. Alsaeed, N.; Nadeem, F. Authentication in the Internet of Medical Things: Taxonomy, review, and open issues. *Appl. Sci.* **2022**, *12*, 7487. [CrossRef]
67. Bajaj, A.; Bhatnagar, M.; Chauhan, A. Recent trends in internet of medical things: A review. In *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 645–656.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.