electronics

MDPI

Article

Lightweight Transmission Behavior Audit Scheme for NDN Industrial Internet Identity Resolution and Transmission Based on Blockchain

Yunhua He [1,*], Yuliang Ma [1], Qing Hu [2], Zhihao Zhou [1], Ke Xiao [1] and Chao Wang [1]

[1] Information Science and Technology, North China University of Technology, Beijing 100144, China; qdumyl@yeah.net (Y.M.)

[2] China Weapon Industry Information Center, Beijing 100144, China

* Correspondence: heyunhua@ncut.edu.cn

Abstract: The Named Data Network (NDN) enables efficient content dissemination through interest-based retrieval, name-based routing, and content caching. In the industrial Internet architecture based on NDN, device identity distribution, identification, resolution, and routing rely on identification resolution technology. However, this approach presents challenges such as cache poisoning, interest packet flood attacks, and black hole attacks. Existing security schemes primarily focused on routing forwarding and verification fail to address critical concerns, including routing environment credibility and data leakage, while exhibiting poor time and space efficiency. To address these challenges, this paper proposes a lightweight behavior auditing scheme using blockchain technology. The scheme utilizes an improved Bloom filter to compress behavioral information like interest and data packets during the identification transmission process. The compressed data are subsequently uploaded to a blockchain for auditing, achieving efficient space and time utilization while maintaining feasibility.

Keywords: IIoT; NDN; identity resolution; blockchain; audit; bloom filter

Citation: He, Y.; Ma, Y.; Hu, Q.; Zhou, Z.; Xiao, K.; Wang, C. Lightweight Transmission Behavior Audit Scheme for NDN Industrial Internet Identity Resolution and Transmission Based on Blockchain. *Electronics* **2023**, *12*, 2538. https://doi.org/10.3390/electronics12112538

Academic Editor: Keke Gai

Received: 31 March 2023
Revised: 24 May 2023
Accepted: 31 May 2023
Published: 4 June 2023

1. Introduction

The Industrial Internet is a crucial aspect of digital and intelligent manufacturing as it links industrial assets, such as machines and control systems to achieve optimal industrial operations. Its importance lies in its ability to coordinate various information in the industrial production process, leading to more efficient production and new services [1]. With the rapid development of the Internet of Things, 5G networks, and industrial technology, new applications such as smart cities, virtual reality, and industrial intelligent production continue to emerge. The number of wearable devices, industrial machines, and sensors transitioning to production is also increasing. According to the 2018 Cisco VNI report, by 2022, the number of machine device connections will reach 14.6 billion, accounting for 51% of the world's connected devices [2].

However, the particularity of industrial production requires industrial networks to perceive environmental information through intelligent means, support the access of a large number of heterogeneous devices, support massive multi-source, multi-modal data high-speed transmission, and have stronger security. This poses significant challenges to the traditional internet in terms of architecture, security, and performance. In recent years, industrial security incidents have occurred frequently, such as the 2010 malware (Stuxnet) attack on Iranian nuclear power plants, the 2014 physical infrastructure attack on a German steel plant, ransomware attacks on aluminum companies in 2019, and on Honda's production line in 2020. As an emerging fusion field, the Industrial Internet faces new and severe security issues. Therefore, security must be the fundamental guarantee for its large-scale and healthy development.

Today, many researchers are focusing on the information-centric network (ICN). This network is message-centric and provides several benefits, such as content naming and transparent network content caching. These features improve network performance, reduce traffic, and lower latency. NDN (Named Data Networking) [3] is one of the most extensively studied ICNs, and it is compatible with various scenarios, including the Internet of Things and 5G. Moreover, it meets the requirements for industrial Internet identification resolution and production. Therefore, NDN is a promising next-generation network architecture. Nodes in NDN maintain three data structures: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). Content data packets are forwarded step by step to the requester based on the PIT entries. Each Interest packet's forwarding interface is stored in the corresponding FIB entry. Once the desired content is received, the corresponding PIT entry is removed. FIB routes the request data packet to its destination. Unlike an IP router's FIB, an NDN router's FIB contains information name prefixes instead of IP address prefixes. It also supports simultaneous forwarding of requests to multiple interfaces for a given name prefix. CS functions similarly to an IP router's cache but retains content across communications, improving content download speed and reducing network bandwidth usage. The policy module in each NDN router determines the forwarding direction for each Interest packet, playing a crucial role in the routing decision-making process.

The identification resolution process in the NDN network involves identification encoding, transmission, and decoding. The security of logo coding and decoding is ensured by corresponding algorithms. However, the identification transmission process in the network environment has resulted in new security issues. The most harmful security attacks on routing include black hole attacks, cache pollution, and content poisoning attacks [4]. The unique content caching mechanism in NDN allows routes to cache recently requested content, reducing request delays but also making them susceptible to attacks [5]. Although many scholars have developed defense schemes such as encrypting communication information, local traffic detection, and attack identification on routers, and bypassing attacks based on reputation tables, there are still unanswered questions [6].

(1) Routing multi-party node trust issues in identification resolution transmission process. The identification resolution transmission process in the NDN network is vulnerable to trust issues due to semi-honest or malicious routes that may tamper with cached data and calculation results. This unreliability creates a need for local verification and reputation management by routes, resulting in data security issues.
(2) Verification efficiency problem in identification resolution transmission. During the process of transmitting identification resolution data packets, there is a risk of tampering at each hop along the route. To ensure the integrity of the data packet, it may be verified by the route. However, using authentication based on digital signatures in NDN can lead to significant overheads. Similarly, performing traffic analysis locally on the router can also result in additional overheads.

Blockchain offers immutability, traceability, and transparency, making it an appealing solution for addressing security and scalability concerns. In trustless group environments, blockchain serves as an ideal data storage solution. It ensures data consistency through consensus mechanisms and mitigates the risk of single-point failures with its distributed storage approach. These characteristics make blockchain well-suited for industrial Internet scenarios. To address the routing trust and verification efficiency problems, we propose a lightweight blockchain [7] auditing scheme. This approach uses the blockchain as a trusted third party to upload routing information and audit routing behavior via smart contracts. To improve space and verification efficiency, we employ an improved Bloom filter [8] to compress large data before uploading it to the chain. This not only enhances the blockchain's space utilization rate and auditing efficiency but also preserves the privacy of real data, as the compressed and original data are different.

The main contributions of this paper are:

(1) A Blockchain Behavior Audit Scheme: We propose a lightweight identification resolution behavior auditing scheme based on blockchain to address the threats associated with NDN industrial Internet identification resolution transmission. Our scheme uses blockchain as a trusted third party to build a real storage platform that stores routing

audit information. The blockchain nodes jointly complete the audit work through a contract, solving the single-point trust problem of traditional solutions.

(2) A Data Compression Scheme based on Bloom Filter: To compress the data on the real storage chain, we use an improved Bloom filter, which is an efficient data structure. First, the routing behavior is recorded in a table, and then compressed using the corresponding Bloom filter. We use a counting Bloom filter to enable repeated modification of the filter and limit the output data size by limiting the number of bits. This approach significantly improves efficiency while retaining important features.

In conclusion, our proposed blockchain behavior audit and data compression schemes offer promising solutions to the challenges associated with industrial Internet identification resolution. By leveraging blockchain and Bloom filter technology, we can provide a secure, efficient, and reliable environment for NDN industrial Internet identification resolution transmission.

We organize the rest of the paper as follows: In Section 2, we present an overview of existing research on IoT and Industrial Internet security, highlighting their shortcomings, such as privacy breaches, unreliable routing, and mismatched scenarios. In Section 3, we establish the scenario and threat models, outlining the design objectives. Section 4 delves into the lightweight audit scheme proposed in this paper. Section 5 evaluates the scheme's contract security, effectiveness, and performance. Section 6 outlines future research directions to address the remaining challenges in this scheme.

2. Related Work

The related work section provides an overview of the security research related to the transmission process of the Named Data Networking (NDN) Industrial Internet of Things (IIoT). However, the application of NDN in the Internet of Things needs to solve the security problems of NDN itself. For example, typical security attacks include denial of service, resource exhaustion, cache pollution, unauthorized access, etc. Some scholars have published relevant research and proposed solutions [5]. However, the security solutions applied to NDN cannot be directly applied to NDN in IoT because the connections between IoT devices are intermittent, heterogeneous, and dynamic [1]. The current research on related security schemes is as follows: At present, the main research focuses on three aspects: encrypted transmission, routing forwarding and verification, and blockchain technology.

(1) Encrypted transmission: To protect security during message transmission, it is common practice to encrypt information, use tokens for access control, or use certificates to authenticate messages. Previous architectures have relied on asymmetric authentication mechanisms used throughout the NDN stack, however, Enguehard et al. [9] quantifies the time and energy overhead of such schemes on constrained devices, and finally came to the conclusion that its cost is too high. Compagno et al. [10] proposed a solution based on symmetric cryptography, which solved the initial authentication and key distribution problems of IoT. However, it does not consider the needs of the routing protocols. Mick et al. [11] proposed a lightweight authentication and hierarchical routing framework for device authentication security. Through nodes and basic, the shared key between facilities and equipment integrates routing and secure login into a framework, and authentication and routing can be performed at the same time, thereby reducing the consumption in the continuous process. Although the authentication is completed, data security is not considered. Kar et al. [12] use the hop count of the message to generate a public key to encrypt the information, and then the message receiver receives the message with the help of the decryption function sent by the sender, using the number of hops is decrypted as a private key to protect the security of the message. At the same time, a cooperative Stackelberg game model is used to determine the best defense strategy for the defenders and attackers. While this method is more efficient than other key schemes, a malicious node in the route can easily obtain the hop count of other layers through its forwarding table, leading to data leakage. To further enhance data security, Tariq et al. [13]

proposed an NDN authentication scheme using an elliptic curve algorithm and bilinear mapping to compress public keys, sign and verify data, making it suitable for resource-constrained IoT devices with space-saving benefits. However, the scheme requires key pairs to be distributed in advance, leading to management difficulties and vulnerability to man-in-the-middle attacks during the distribution process. To further improve data security, Qu et al. [14] proposed an effective and lightweight countermeasure scheme, which consists of a token-based routing monitoring strategy (TRM), hierarchical consensus-based trust management (HCT), and popularity-based probability composed of cache and cache replacement strategy (PPC). It uses tokens to control the sending of packets, establishes a trusted environment through hierarchical consensus, and uses cache strategies to improve verification strategies. However, the disadvantage is that hierarchical trust is evaluated by core routing, and the core router is not considered an existing security issue.

(2) Routing forwarding and verification: Several scholars have developed routing forwarding and routing verification schemes based on routing. In a routing forwarding strategy, DiBenedetto et al. [15] propose selecting the next hop based on its forwarding success rate as a defense against attackers. However, discarding interest packets can hinder other routes on the path from receiving the requested data and diminish their rating as potential next-hop routes. Consequently, this approach negatively impacts the selection of subsequent routes and significantly increases the likelihood of detours. Yang et al. [16] proposed a minimization bypass scheme SmartDetour by using the reputation mechanism, and established a new probabilistic forwarding table in each route, and bypassed when the packet forwarding failed. The forwarding candidate reputation is updated, and at the same time, with the help of the reputation-based probabilistic forwarding strategy, the interface with the highest probability in the reputation selection table is used for unicast attempts. Although this method can reduce the detour distance very well, its unicast trial and error will waste more time and record a large amount of repeated information, resulting in a waste of storage space. To avoid content poisoning attacks, signature verification can be performed on each hop route, but the verification will cause a lot of overhead. Kim et al. [17] regards content detection as the main method, proposes an efficient content verification scheme, processes limited content cache segments, and adopts the LRU algorithm to reduce cache and verify repeated popular content to improve the efficiency of data verification. However, this scheme still cannot guarantee whether the router has actually verified the data. Although it has certain robustness, it is not safe when the number of attacked nodes increases. While the above solutions may detect malicious behaviors and resist attacks, they fall short in establishing a trusted environment between devices and implementing necessary authority control.

(3) Blockchain technology: As a data storage and information encryption technology, blockchain provides new ideas and methods for transmission security. Lei et al. [18] implemented blockchain-based cache poisoning protection and privacy-aware access control in the NDN-based vehicle network, achieving key management, cache poisoning detection, and access control. However, this solution has limited efficiency and is only applicable to the vehicle network. In the UAV network, Alsamhi et al. [19] proposed a combination of Federated Learning (FL) and blockchain technology. They utilized blockchain to store model data and verify human-machine behavior, ensuring high-level security and data privacy. Additionally, they investigated a blockchain-based method for transmitting sensitive information and achieving collaborative consensus in a trustless environment [20]. In the domain of the medical internet, Myrzashova et al. [21] introduced a novel conceptual framework for FL based on blockchain in digital medical environments. Their approach ensured the accuracy of the overall FL result through blockchain validation. They also incentivized the donation of local data during training tasks, effectively addressing the challenges associated with confidential medical data leakage and security. This framework facili-

tated collaboration among multiple parties in training without the need to share or centralize datasets. However, the scenarios addressed by the aforementioned solutions do not align with those encountered in the Industrial Internet. This disparity encompasses both the quantity and nature of the data involved. Hence, it remains necessary to develop a blockchain security solution specifically tailored to the identification resolution system of the Industrial Internet. This solution aims to address the issues of untrustworthy routing and privacy breaches.

3. Model Building and Design Requirements

3.1. Industrial Internet Analysis Process Model Construction

This paper investigates the industrial Internet identification resolution system's transmission scenario under the NDN network [22]. The system includes core routing (CT), border routing (ET), and access devices (AT), using a top-down multi-layer naming scheme. The layers include the root prefix layer, task type layer, service layer, topological location, and intra-network functional layer.

To improve retrieval speed and reduce routing cache, a hash function maps name prefixes to unique NameCode. The system also uses Fibonacci encoding [23] for fixed-length encoding and establishes an NCP storage correspondence in border routing. Cross-domain requests are handled by implicitly converting the NameCode into the original name prefix, improving efficiency.

(1) Root prefix layer: Used to define the core domain or network prefix, which is related to the device's location in the network.
(2) Task type layer: Defines the IoT data namespace. Based on the required tasks it is mainly divided into two types: data collection tasks, such as the real-time collection of sensor information in the production process, and equipment status information; and instruction tasks, such as sending alarms, custom monitoring cycles, and other equipment action attributes.
(3) Service layer: Defines specific service content, such as obtaining temperature, humidity, monitoring and retrieval, and device status retrieval.
(4) Topological location and intra-network functional layer: Identifies the intra-network functions used, allowing for multi-source data retrieval.

The NDN network's longest matching principle for routing and forwarding can cause slow retrieval speeds and increased routing cache occupation with excessively long prefixes. To address this issue, a hash function is used to map name prefixes to unique NameCode, while Fibonacci encoding performs fixed-length encoding on location information. This establishes an NCP (name-code-prefix) table storage correspondence in border routing, allowing for efficient data access within the same routing range, as shown in Figure 1.

In the registration process, a new device sends a request packet to ET to register its name prefix, typically its ID name. ET converts the name prefix into a corresponding NameCode, records the relationship in the NCP table, and returns the NameCode value to the device. When sending a request, the device uses the NameCode, and any future packets of interest used in local communication also use the NameCode value instead of the long name prefix. Intermediate nodes can use the NameCode value to forward packets.

Parsing is mainly divided into two cases: intra-domain and cross-domain. In intra-domain parsing, devices in the same domain only use NameCode for identity resolution. In cross-domain parsing, when a device outside the domain is requested, the NameCode is implicitly converted to the original prefix in the route. When the router receives the packet, it will take out the NameCode and search the NCP table to obtain the name prefix corresponding to the NameCode and complete the implicit conversion of the NameCode.

By implementing a hash function and Fibonacci encoding, along with the use of NameCode for efficient data access and retrieval, the NDN network can significantly improve efficiency and reduce data length for cross-domain requests.

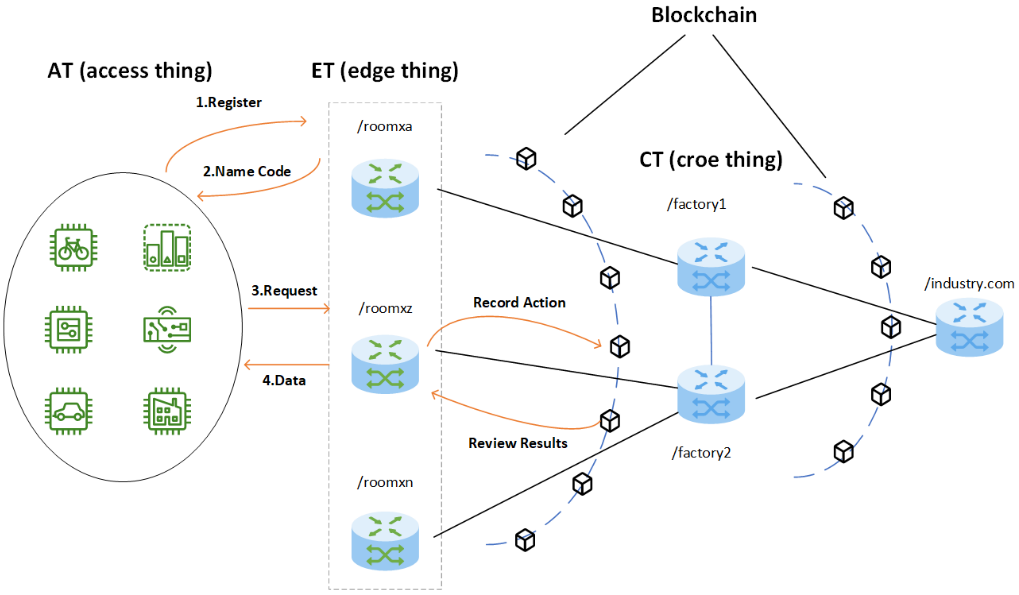

Figure 1. Transmission scenario model diagram.

3.2. Threat Model

The NDN network industrial Internet identification resolution transmission scenario is vulnerable to a variety of security threats, with core routes, border routes, access devices, and route cache data being the main assets at risk. The specific threats faced by each asset are:

(1) Request flooding attack [24]: In this attack, an attacker floods the network with a large number of useless interest packets in a short amount of time, overwhelming the network and causing the cache table to be replaced constantly. This ultimately results in the route being unable to provide forwarding services to normal users.

(2) Logo content poisoning [25]: An attacker replaces the original content with fake data, resulting in the consumer being unable to receive the legitimate data.

(3) Black hole attack [26]: Malicious routers discard interest packets or returned information during the identification resolution process, which can target a specific range or device by filtering a specific name prefix. This can isolate the device, preventing it from receiving normal resolution services.

(4) Data leakage: The industrial Internet transmits information in plain text without encryption, making it vulnerable to data leakage. Third-party behavior audits can also lead to production data leakage.

To mitigate these threats, it is recommended to implement measures such as access control, encryption, and behavior monitoring. By doing so, the security and reliability of the NDN network can be improved.

3.3. Design Goals

The application of blockchain technology in supervision has numerous advantages. Firstly, the use of distributed technology in blockchain enables the storage and synchronization of data between multiple parties, making supervision easier to access and allowing regulators to review data comprehensively and in real time. Secondly, blockchain guarantees the consistency of data stored by multiple parties through the consensus mechanism and ensures traceability, tamper-resistance, and non-repudiation of transactions through cryptography. Regulatory agencies can participate in the data consensus as nodes and can directly obtain credible arbitration evidence related to transaction data responsibilities. The Power Ledger platform, powered by blockchain technology, has gained significant traction in Australia for energy transactions and microgrid management. Modum is a solution that utilizes blockchain and IoT technology to provide temperature monitoring and supply chain data verification. Modum's solution is already widely adopted by pharmaceutical and logistics companies in Europe.

In this paper, we aim to address various types of serious threats in the scenario by combining blockchain technology to achieve the following goals:

(1) Trusted security solution: A security solution is necessary to solve various attacks encountered in the scene, and the trustworthiness of the solution ensures that the entire process is credible, private, and safe. The blockchain, being a distributed network security technology, is suitable for NDN IoT security scenarios. By building a blockchain system that uses blockchain as the carrier of the security solution, each device can register its identity and assets in the blockchain architecture, and each behavior can be encrypted and protected in the blockchain by the trusted device record. The corresponding table of the hash of the device ID, prefix name, and its NameCode is stored on the blockchain, and the compressed behavior is also recorded (Section 4.1). By verifying the hash of the device ID, the identity of the host can be determined.

(2) Lightweight behavior audit: Uploading device behavior records to the blockchain and using blockchain contracts to conduct behavior audits with the help of the non-repudiation, openness, and transparency of the blockchain can enable audit results to be traced and verified. In the process of monitoring equipment behavior, the use of Bloom filter technology and improved compression methods can compress behavior records to achieve high compression rates and improve verification efficiency.

4. Lightweight Audit Scheme Based on Blockchain

In this section, we propose a lightweight blockchain behavior audit solution for the problems and design purposes of the above scenarios, as shown in Figure 2. All devices will be registered on the blockchain, and the corresponding relationship between NameCode and Hash (ID) will be saved on the blockchain to obtain the corresponding blockchain access rights. For routing behavior, we analyzed the information necessary for behavior auditing in the NDN industrial Internet scenario in Section 4.2, and stored it on the blockchain in the form of a behavior record table after compression. In addition, the compression method is the focus of attention. We use the improved CBF, whose length is a multiple of the machine word length, to compress the behavior data, stipulate that the size of the Count should not exceed 4 bits, and deduce the best hash when the number of data is determined. The number of Hash functions improves space efficiency and time efficiency while ensuring that hash collisions are sufficiently low. The figure shows the process of obtaining information from the routing data structure and then generating a behavior record table through HashSet compression and uploading it to the blockchain. For behavioral auditing, we use contracts for on-chain auditing. After authorization authentication, the auditing contract can be invoked, and the auditing speed can be accelerated by comparing the data by byte. At the same time, the compressed data does not have data characteristics, which effectively prevents data leakage during the auditing process.

The figure consists of two parts: the upper part represents the logical layer, while the lower part represents the physical layer. The physical layer includes Consumer, Provider, NDN Network, and blockchain. In this setup, the Consumer initiates an Interest packet that is transmitted to the NDN network. If the network does not have the requested data cached, the Interest packet is forwarded to the Provider, who responds with the corresponding data packet to the NDN network. Apart from forwarding and caching various packet types, the NDN network is also responsible for compressing and uploading behavioral data.

The first half of the figure illustrates the relevant data structures and processes. The data structures comprise three types of NDN data tables and behavior record tables. The CS table contains the request prefix and corresponding data, the PIT table contains the request prefix and forwarding port with forwarding, and the FIB contains the currently known prefix and forwarding port number. The processes involve data compression using an improved CBF (Counting Bloom Filter), data upload to the blockchain, and contract invocation with entities.

Overall, the figure provides a visual representation of the system architecture, highlighting the interaction between different components and the flow of data and operations.

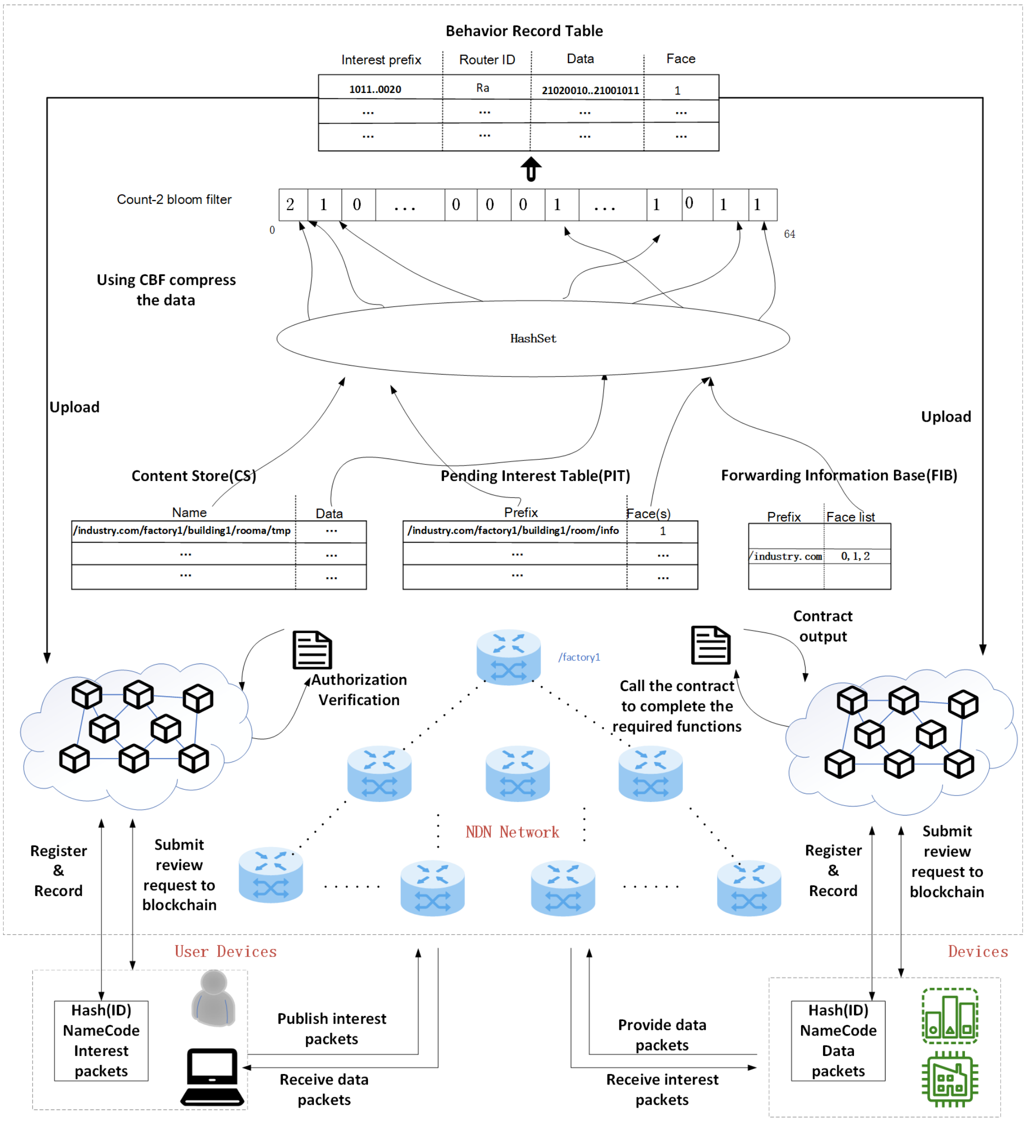

Figure 2. Audit Scheme Diagram.

4.1. Privacy Protection Behavior Audit Scheme

Our research focuses on privacy protection behavior audit schemes for identifying and protecting against malicious node behavior. Currently, there are two main approaches for this: identity authentication and access control mechanisms in traditional networks and monitoring abnormal traffic and feedback from attacked nodes. Access control relies on rules like attributes or roles to determine if a device's behavior is permitted. Fine-grained access control can ensure security, but it can also increase the message transmission overhead for each node. To detect malicious behavior through abnormal behavior monitoring, we need to establish detection standards, ensure the credibility of behavior auditors, maintain privacy during the audit process, and establish an appropriate feedback mechansm. In this section, we analyze the authentication stage, behavior audit stage, feedback stage and present corresponding solutions using smart contracts.

4.1.1. Authentication Stage

The authentication phase plays a crucial role in enabling legitimate user access to blockchain data and facilitating lightweight and efficient application submissions while mitigating the risk of massive uploads and potential DOS attacks. To ensure authentication, all devices must register on the blockchain. When a device seeks a network connection, it undergoes authentication using its registration information stored on the blockchain. Once authenticated, the device verifies the integrity of key information hashes to detect potential intrusions.

When legally verifying users, we mainly consider the user's identity and their recent behavior: device administrators, production security officers, and supervisory department personnel can access the chain, device administrators can register and prohibit devices, production security officers and supervisors can use the audit function to audit and ban devices, but they cannot perform registration operations and cannot view the device registry. The permissions of users with different identities are recorded in the identity permission table on the blockchain.

4.1.2. Behavior Audit Stage

The behavior audit stage aims to identify and address abnormal behavior in a lightweight and efficient manner. Consumers compress their interest and data packets using a specified method before requesting the blockchain to invoke smart contracts for on-chain behavior auditing. The Bloom filter compresses data during auditing, and verification can be performed quickly through the AND operation. This significantly reduces the time required for the audit. The process is illustrated in Figure 3.

During the audit, the hash map is involved, and excessive memory access can result in additional time and space consumption. To address this, we improve the computer memory access principle. The elements are mapped to k bits randomly selected from the bit array, and k bits are selected in a word instead of the entire bit array. The length of the CBF is set as a machine word length, and each piece of data can be completely retrieved after accessing the memory twice at most. This greatly reduces the consumption of reading data from memory. After the behavior record is taken out, verification is efficient since the record is stored as a Bloom filter. Behavior is deemed abnormal if the result of the AND operation is 0.

Assuming the consumer's interest packet is I_i, the corresponding reply data packet is D_i, the forwarding port is f, and the hash function set is *hashset* (with the process of using $hash(s, o)$ where s is the hash function set, and o is the object being hashed), such as Figure 3.

The router searches for CS and PIT based on rules when the consumer sends an Interest packet request I_i. The router then replies or forwards I_i through the cached data, and only when the router responds to the Interest packet is the reply data packet D_i, the Interest packet I_i, and the forwarding port f stored in the behavior record table. To store the data, we map (I_i, D_i) to the Bloom filter using the hash function set hashset to obtain the value $C2BF(I_i, D_i)$. We then store the value $C2BF(I_i, D_i)$ of the Bloom filter in the table. These data are uploaded to the blockchain as a basis for auditing. Please see Algorithm 1 for the compression algorithm.

When auditing, the consumer triggers an audit contract C1 and provides the hash value of the received data packet I_i' and the sent interest packet D_i'. The audit contract then matches the interest packet in the latest block, finds the record that matches the interest packet in the record $C2BF(I_i', D_i')$, and obtains the route of the interest packet and the corresponding data packets within a specific time period.

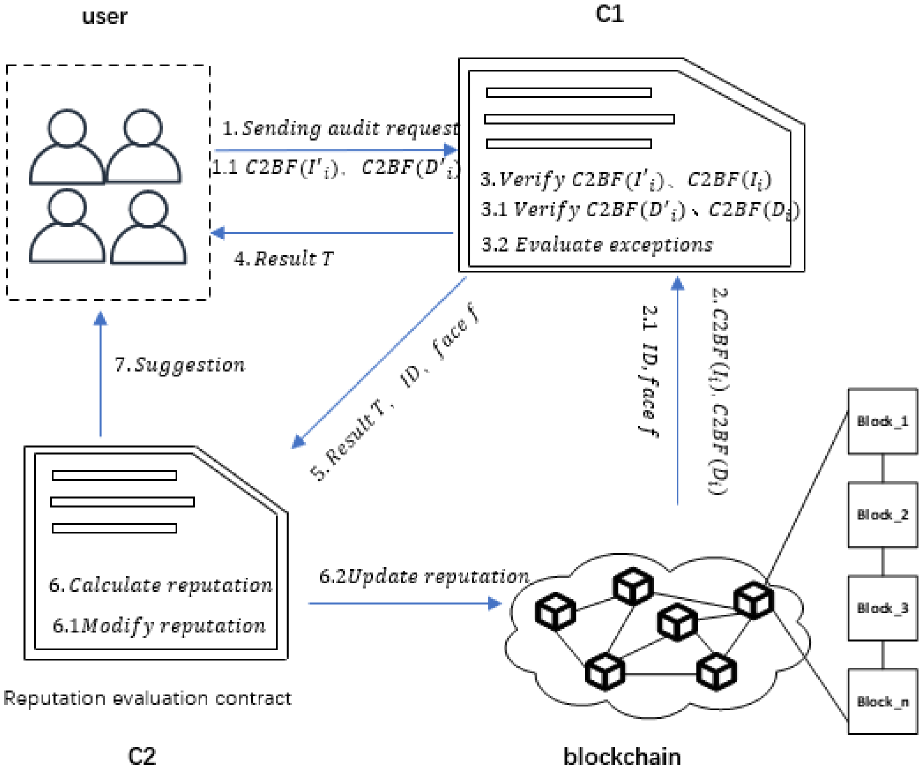

Figure 3. Contract interaction diagram in the review phase.

Algorithm 1 Bloom filter data compression algorithm

```
Input: input Behavior record $(I_i, D_i)$, Bloom filter length $l$, random number $seed$
Output: output Compressed behavior record $C2BF(I_i, D_i)$
Initialize $CBF, byte[l]bits1, byte[l]bits2$
Add the set random number seed to the hash function generator to generate the required hash function set $\{G_1, G_2, G_3 \ldots G_t\}$
$functionHashGenerator[seed]$
$returnHashGroup = \{G_1, G_2, G_3 \ldots G_t\}\}$
$functionaddBits(I_i, D_i)$
$\{G_1, G_2, G_3 \ldots G_t\} \leftarrow I_i$
$\{G_1, G_2, G_3 \ldots G_t\} \leftarrow D_i$
for $j = 0; j \leq t; j++$ do
    $G_t \leftarrow I_i, G_t \leftarrow D_i$
    $bits1[G_j(I_i)\%l]++$
    $bits2[G_j(D_i)\%l]++$
$C2BF(I_i, D_i) \leftarrow bits1, bits2$
return $C2BF(I_i, D_i)$
```

By comparing $C2BF(D_i)$ with $C2BF(D_i')$, the behavior of all devices involved and the sender of the information can be determined. If $C2BF(D_i') = C2BF(D_i)$, there is a violation on the device, and the corresponding routing ID and interface will be returned. If $C2BF(D_i') \neq C2BF(D_i)$ or $C2BF(I_i') \neq C2BF(I_i)$, the application is doubtful. Please refer to Algorithm 2 for the comparison algorithm.

Algorithm 2 Comparison of Behavioral Records

```
Input: Abnormal record $C2BF(I_i', D_i')$
Output: record verification result $T$
Get the latest stored behavior data from the chain $I_i = byte[]bits1, D_i = byte[]bits2$
$C2BF(I_i', D_i') \rightarrow byte[]bits1', byte[]bites2'$
$T = true$
if $bits1'.length \neq bits1.length$ then
    return $T = false$
else if $bits2'.length \neq bits2.length$ then
    return $T = false$
for $j = 0; j \leq bits1'.length; j++$ do
    if $bits1'[j] \neq bits1[j]$ then
        return $T = false$
for $j = 0; j \leq bits1'.length; j++$ do
    if $bits2'[j] \neq bits2[j]$ then
        return $T = false$
return $T$
```

4.1.3. Feedback Stage

The feedback phase aims to address exceptions and make security adjustments promptly. The review results are sent back to the request initiator and the reputation calculation contract C2, which computes the reputation value and identifies potential risks for timely defense measures.

In terms of routes, the blockchain assigns an initial reputation value during registration, and this value increases over time only when the route serves credible content. After the content consumer verifies the content, an update is sent to the blockchain network to notify the served cache storage of any increase or decrease in the reputation value. If a device's reputation score drops below a certain threshold, it will be denied access to the blockchain network.

Regarding content publishers, multiple review requests submitted simultaneously may indicate malicious content publishers. Thus, the audit contract will be triggered to verify the content's validity when a certain node exceeds the transaction threshold.

4.2. Data Compression Scheme Based on Bloom Filter

In the previous section, we discussed the use of blockchain to monitor device behavior during NDN identification resolution and transmission. The blockchain employs a consensus mechanism and cryptography technology to establish internal security mechanisms that possess the characteristics of non-repudiation, openness, and transparency, thereby addressing trust and credit challenges faced by society. However, the blockchain's consensus characteristics result in limited throughput and block size, necessitating an efficient compression mechanism to ensure regulatory behavior information is stored effectively on the blockchain in the industrial internet. One such mechanism is the data compression scheme based on the Bloom filter.

4.2.1. Forwarding Behavior Table

To ensure accurate and effective supervision, it is crucial to determine how data should be recorded and what type of data should be recorded. In the NDN architecture, naming content is the main building block, with each data packet requiring a name. Similar to the TCP/IP system, each device has an IP address [27]. In the NDN system, each route stores three data structures necessary for data transmission: Forwarding Information Table (FIB), Pending Interest Table (PIT), and Content Store (CS).

The FIB contains routing information for requests to producers and contains the outgoing interface and next-hop neighbor list for each request, pointing the corresponding Interest to the path of the matching data. The FIB can be populated manually using NDN's

Open Shortest Path First protocol or OSPFN and/or automatically by the NDN routing protocol in the control plane using the Link State Routing Protocol or NLSR. The PIT stores all pending interests that have been retweeted but not yet fulfilled and contains a list of incoming interfaces for each interest, allowing each packet to be routed back to the consumer. Finally, CS caches the data and stores the most recent data for each name prefix.

Forwarding data are the main content of forwarding behavior and serves as a basis for judging whether there is forged or modified data in the route. Additionally, it is necessary to reflect the responsible subject of this forwarding behavior in the records, as without this information, the behavior cannot be linked to the attacker. The same subject can perform multiple different actions at different times or at the same time (such as responding to different Interest packets), but these actions finally forward the data to different subjects according to different prefixes. Therefore, we need to record the corresponding name prefixes to deal with various attacks against the prefixes while being able to quickly locate the specific device being attacked to improve overall security.

In summary, this article defines the attributes of the routing behavior record table as interest packets, routing IDs, forwarding data, and forwarding interfaces. The routing ID specifies the subject of the behavior, the interest packet is used to determine the direction of the behavior, the forwarded data are the content of the behavior, and the forwarding interface. The information in the behavior record form will be uploaded to the blockchain for behavior auditing.

4.2.2. Data Compression Scheme Using Improved Bloom Filter

In the previous section, we discussed the routing behavior record table and determined what data to record on the blockchain. In this section, we focus on the storage aspect and propose a solution using an improved Bloom filter. Bloom filters have long been used in network security [28]. In virus scanning, Bloom filters can be used to store signatures of known viruses and check whether a file is infected or not by comparing the hash value of the file with the filter. You can monitor network traffic and check whether the target address is in the filter to detect worms, and filter malicious packets to identify malicious or phishing URLs, etc.

Storing device behavior information in the behavior record table is necessary for auditing purposes, but the list of information increases linearly with time. In the NDN network, the interest packet contains the name that determines the data request, and the name prefix of the interest packet increases in size as the request travels across domains. Similarly, the reply data takes up significant space.

The Bloom filter is a compact probabilistic set representation used to determine if an element is in a set. It is a simple and efficient data structure with a time complexity of $O(k) \approx O(1)$ and takes up minimal space. The Bloom filter consists of a bit array initialized to 0, and an independent hash function maps the input value to the bit array. Let the size of the Bloom filter be m, and $S = \{s_1, s_2, s_3, \cdots, s_n\}$ is the set of elements to be inserted into the Bloom filter, where m is a prime number to avoid hash collisions. We choose k hash functions, $h_i(x)$, $(1 \leq i \leq k)$, to map $x \in S$ into the Bloom filter. To determine if an element y belongs to the set S, we check each bit of $h_i(y)$. If not all bits are 1, the element y is not in the set.

To compress the recorded device behavior data, we use a counting Bloom filter instead of a standard Bloom filter since the latter does not support dynamic deletion. The counting Bloom filter expands each bit of the standard Bloom filter into a small counter that adds 1 to the corresponding k (k is the number of hash functions) counters during insertion and decreases the values of the corresponding k counters during deletion. However, this approach occupies several times the storage space, and we alleviate this issue by limiting the size of its counter to 4 bits, which has been shown to be sufficient for most applications, according to analysis. Given a counting Bloom filter with n elements, k hash functions, and

m counters, the probability that the $i-thcounter$ is incremented by j times is a binomial random variable, as shown in Equation (1).

$$P(c(i)=j)=\left(\begin{array}{c}nk\\ j\end{array}\right)\left(\frac{1}{m}\right)^{j}\left(1-\frac{1}{m}\right)^{nk-j} \tag{1}$$

The right side of the equation can be interpreted as follows: j selections are made out of nk hash operations, the $i-th$ counter is selected j times, and it is not selected $nk-j$ times. Therefore, the probability that the value of the $i-th$ counter is greater than j can be defined as the probability that the counter is at least j, which can be calculated as $mP(c(i)\geq j)$. Using Stirling's formula Equation (2), we can simplify this expression to obtain Equation (4).

$$n!=\sqrt{2\pi n}\left(\frac{n}{e}\right)^{n} \tag{2}$$

$$P(c(i)\geq j)\leq\left(\begin{array}{c}nk\\ j\end{array}\right)\left(\frac{1}{m}\right)^{j} \tag{3}$$

$$\begin{aligned}
Pc(i)\geq j &\leq \frac{(nk)!}{j!(nk-j)!}\left(\frac{1}{m}\right)^{j}\\
&\approx \frac{\sqrt{2\pi nk}\left(\frac{nk}{e}\right)^{nk}}{\left(\sqrt{2\pi j}\left(\frac{j}{e}\right)^{j}\right)\cdot\left(\sqrt{2\pi(nk-j)}\left(\frac{nk-j}{e}\right)^{nk-j}\right)}\left(\frac{1}{m}\right)^{j}\\
&\leq \frac{(nk)^{nk}}{\sqrt{2\pi j}\sqrt{2\pi(nk-j)}j^{j}(nk-j)^{nk-j}}\left(\frac{1}{m}\right)^{j}\\
&\leq \frac{(nk)^{nk}}{\sqrt{2\pi j}j^{j}(nk)^{nk-j}}\left(\frac{1}{m}\right)^{j}\\
&\leq \frac{(nk)^{j}}{\sqrt{2\pi j}}\left(\frac{1}{m}\right)^{j}
\end{aligned} \tag{4}$$

To explore the relationship between j and k, we utilize the Lagrangian interpolation method for simplification. This method requires multiple sample points as data to observe the relationship between two variables. By using k and j as variables, we can obtain corresponding sample points by inputting them and then derive the relationship between j and k. This relationship is expressed as Formula (5).

$$C(nk,j)=\sum_{\mathrm{i}=0}^{nk}f(i)L(i,j) \tag{5}$$

$$L(i,j)=\prod_{k=0}^{nk,k\neq i}\frac{(x-x_{k})}{(x_{i}-x_{k})} \tag{6}$$

To explore the relationship between j and k further, we simplify using the Lagrangian interpolation method, which requires multiple sample points to observe the relationship between two variables. By taking k and j as variables and using their corresponding sample points, we obtain the relationship between j and k as shown in Formula (5), where $f(i)$ represents the value of the i-th element, $L(i,j)$ represents the Lagrangian interpolation polynomial shown in Formula (6), and $x, x_k, and\ x_i$ represents the number of extracted elements, k elements, and i elements, respectively. It can be deduced that $P(c(i)\geq j)$ takes the minimum value when $k=j$.

In addition to ensuring a low false positive rate, the number of hash functions is also crucial, as it determines the number of calculations required, and the hash process may consume significant time. Thus, choosing a large k value is not recommended.

Moreover, the length of the Bloom filter is proportional to the space it occupies, directly affecting the insertion and deletion operations. To improve space compression rate and reading efficiency, we design the length of the Bloom filter as a multiple of the word length. This approach takes advantage of the computer's reading performance, thereby enhancing the Bloom filter's operation efficiency. The word length is the computer's unit to read information.

5. Safety Analysis and Experimental Evaluation

In this section, we analyze the security of our proposed scheme and evaluate its resilience against Interest flood, black hole, and content poisoning attacks. We also conduct experiments to verify the effectiveness and performance of our scheme against these attacks.

5.1. Security Analysis

To address security issues in the analysis and transmission of industrial Internet identification in NDN networks, this paper proposes a behavior monitoring scheme based on the blockchain. The scheme comprises a Bloom filter compression method based on limited counter size, a single memory access on-chain behavior audit method, and a device reputation evaluation scheme. In this section, we analyze and prove the effectiveness of our scheme in detecting and defending against the following attacks while also protecting data privacy

(1) Flood attack: The flooding attack is initiated by the attacker from the consumer side. By sending a large number of interest packet requests in a short period of time, the network load increases, or the PIT table overflows, and other users' normal requests cannot be responded to. Its characteristic is that the number of interest requests from the same interface increases sharply in a short period of time, and it is quite different from normal interest packets, and most of them are meaningless requests. On the chain, the transaction volume from the device has increased dramatically (because every behavior record is a transaction), and the gas consumption is abnormal. When this happens, it indicates that the network is suffering from an Interest flood attack, and the device under attack can be identified by looking up the device name in the record. At this time, the message requester submits $C2BF(I_i)$ because he cannot obtain the corresponding reply. At this time, two situations will occur. There is no relevant information about the interest packet in the blockchain, that is, $C2BF(I_i') \neq C2BF(I_i)$, there is $C2BF(I_i') = C2BF(I_i)$, but no reply is received. In both cases, the corresponding attacked route can be found. In both cases, the corresponding attacked route can be found. The first one is its adjacent router, and the second one records the route connected to the last forwarding port face. Two types record the route of the last forwarding port face connection.

(2) Black Hole Attack: A black hole attack occurs when an attacker discards all received information of a specific prefix during data transmission. Consumer requests that cannot be responded to will submit a review request to the blockchain. If $C2BF(I_i') = C2BF(I_i)$, but $C2BF(D_i)$ was not obtained after forwarding, it is considered that the I_i is suffering from a black hole attack. Since the interest packet and the data packet in the record are stored in pairs, when the data of the reply packet matching the prefix in the record is all 0, it is considered that the I_i suffers from a black hole attack. At this time, the attacker can be identified based on the RouterID.

(3) Content Poisoning Attack: In a content poisoning attack, the data received by the consumer is abnormal because the attacker replies with abnormal content. Since the behavior information is compressed by the hash function, when the data changes, $C2BF(I_i') = C2BF(I_i)$, but $C2BF(D_i') \neq C2BF(D_i)$. Therefore, we can find specific attackers by comparing the interest packets provided by consumers with the hashes of the content.

(4) Privacy Protection: Since only the device name is stored on the blockchain after hash mapping, $I_i \neq C2BF(I_i)$ and $D_i \neq C2BF(D_i)$. Additionally, due to the characteristics of the hash function, $C2BF(I_i) \nRightarrow I_i$ and $C2BF(D_i) \nRightarrow D_i$. The hash value cannot restore the original data, making it safe to disclose on the blockchain.

(5) Formal Analysis Of Smart Contracts: The use of formal methods is a mathematical technique for modeling, designing, and testing software and hardware systems to ensure they are built correctly, which is suitable for ensuring the security of smart contracts [29]. We employ the formal modeling method introduced in [30] to verify the execution environment's integrity and effectiveness of the smart contract behavior. To simulate the execution of the audit contract, we model the interaction between user behavior and the audit contract. The audit parameter (InterestPacket, DataPacket) is utilized, and the audit function is invoked through Audit Call. The simulation demonstrates that the user achieves a 100% success rate when interacting with the audit contract.

5.2. Experimental Evaluation

In this section, we verify the effectiveness and efficiency of our proposed scheme in a simulated environment. To accomplish this, we utilized NDNSIM to simulate a realistic NDN network environment and constructed a blockchain network for testing. The test machine used had an Ubuntu system, a 4-core CPU, and 4 GB of memory. Compared to Ethereum, FISCO BCOS offers significant advantages in performance, privacy protection, multi-chain support, and enterprise-level features. It excels in handling high-throughput transactions with minimal latency, providing encrypted data storage and transmission, and ensuring participant privacy and data security through identity anonymity processing. FISCO BCOS exhibits higher stability, reliability, and security, making it well-suited for practical applications in industries like finance and supply chain.

To demonstrate the efficiency and effectiveness of our scheme, we conducted a comparative analysis of the two crucial factors that impact efficiency: gas consumption by contract operation and audit time. The experiment utilized a hash set of four hash functions for mapping. The standard Bloom filter length was set at 64, and our scheme used a total of eight Bloom filters, with each counter's maximum value set at 256.

We compared our scheme with a scheme using a standard Bloom filter and a scheme without Bloom filter compression.

Figure 4 shows the gas consumption of uploading behavior data in the three schemes as the number of input characters varies, with a Bloom filter length of 64. It is evident that our scheme is not sensitive to changes in the number of input characters, as consumption remains stable at around 11,000 gas. Conversely, the gas consumption of the direct upload data scheme increases significantly with the number of characters, from 11,000 gas equivalent to our scheme, to more than double that value due to the string expansion mechanism when the limit is exceeded, resulting in a piecewise increase. There is little difference between the two schemes when the character length is less than 60, but our scheme has a clear advantage when it exceeds 60. Compared to the standard Bloom filter scheme, our scheme is almost the same, with a gas value difference of approximately 500.

We conducted tests on the gas consumption and audit time of different audit schemes. In Figures 5 and 6, we show the results of the experiment, which audited the latest 200 pieces of data, with the size of the data increasing from 32 characters to 512 characters. The tests were carried out considering various attacks and persistent attacks. The audit time was obtained by taking the average of multiple tests.

During auditing, direct comparison of strings is not possible, and the hash value needs to be compared after being hashed by a hash function. The standard Bloom filter scheme maps the data to a bit array, and then compares the value of the Bloom filter byte arrays, each time according to the number of bytes. On the other hand, our proposed scheme maps the data to the counter, and the length of the Bloom filter is fixed, so the number of comparisons is relatively fixed, resulting in a more stable gas and audit time less affected by the amount of data.

As the length of the string increases, the cost and duration of hashing on the chain will be higher, so the gas spent and audit time are positively correlated with the length of the string. However, when the data are less than 64, there is not much difference, and the cost of directly auditing the string is lower.

Compared with the standard Bloom filter scheme, our proposed scheme does not require conversion into a byte array twice. Additionally, the average number of comparisons is smaller, making it faster and consuming less gas. Compared with the 64-bit standard Bloom filter, our proposed scheme consumes about 1500 less gas each time, and the average audit time is 5 ms less each time.

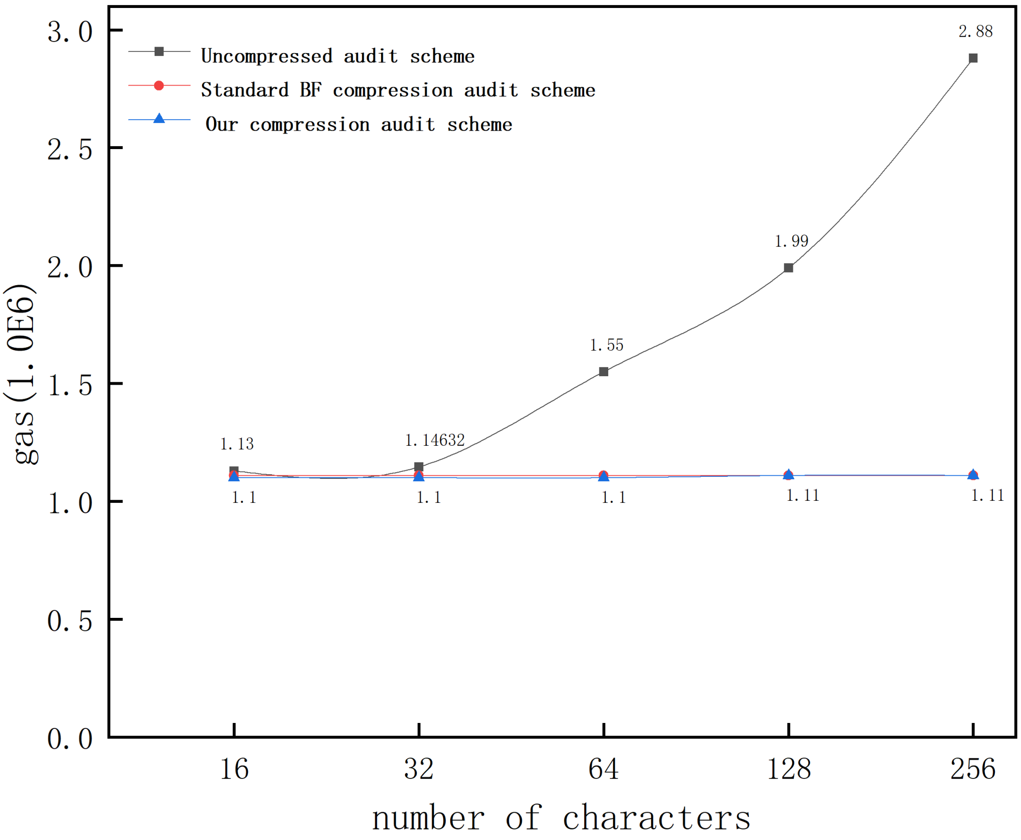

Figure 4. Comparison insert gas consumption of three schemes with different numbers of characters.

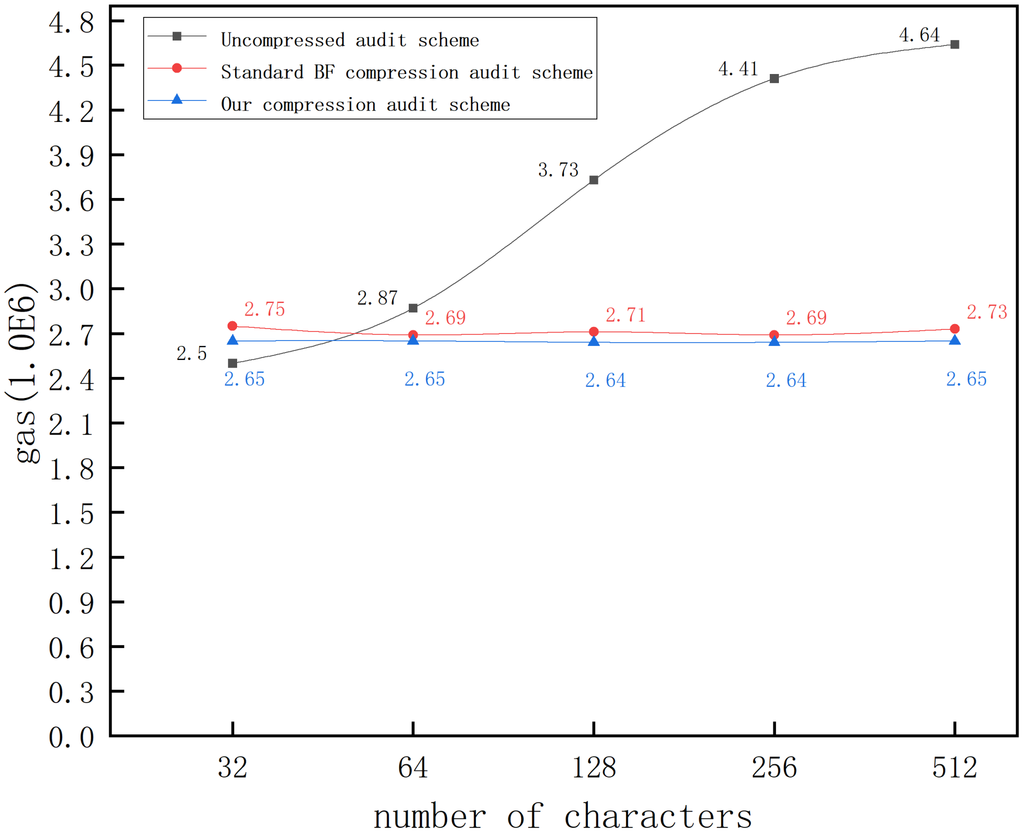

Figure 5. Comparison of audit gas consumption of three schemes with different number of characters.

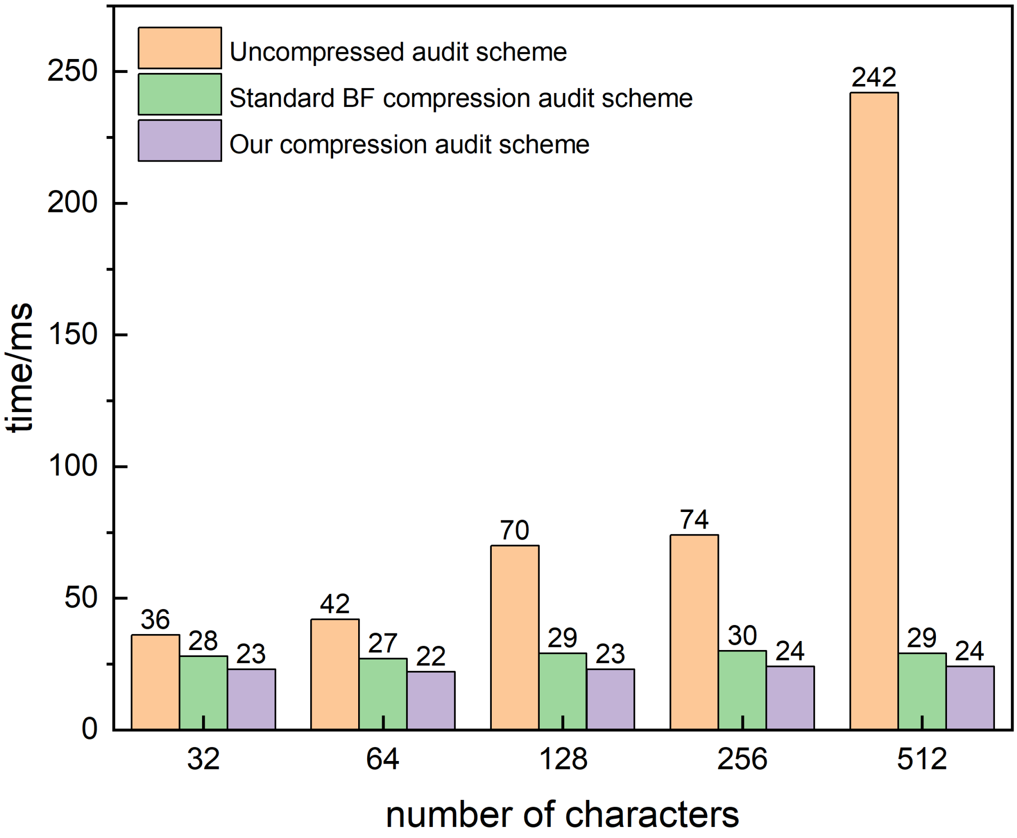

Figure 6. Comparison of audit time of three schemes with different numbers of characters.

6. Conclusions and Future Directions

We propose a lightweight behavior auditing scheme based on blockchain to address security issues in the process of NDN industrial Internet identification resolution and transmission. Our solution records node behavior using blockchain technology and conducts credible audits through smart contracts, ensuring the security of the audit process. Furthermore, we use an improved Bloom filter to compress data for efficient storage and auditing. Our experiments confirm that this scheme greatly reduces auditing consumption while maintaining effectiveness against various attacks.

Moving forward, we suggest several research directions. To mitigate sensitive information leakage and reduce storage consumption, this paper proposes a variant Bloom filter data processing method. However, further investigation is required to determine the appropriate counter size for different length marks, which can enhance efficiency and reduce false positive rates. The selection of the number of hash functions in the hash function set also warrants consideration. Future research should explore the optimal relationship between efficiency, counters, and hash functions in different scenarios. Additionally, the data stored in the behavior record table holds significant importance. Although the current solution optimizes uplink data size through a compression scheme, there is still room to eliminate data redundancy and improve audit efficiency without compromising the audit results. Resolving this issue is a crucial aspect to be addressed.

Author Contributions: Methodology, Y.H. and Y.M.; Software, Z.Z.; Validation, Q.H.; Formal analysis, C.W.; Investigation, K.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the R&D Program of Beijing Municipal Education Commission under Grant KM202010009010, in part by the Beijing Municipal Natural Science Foundation under Grant M21029, in part by the National Natural Science Foundation of China under Grant 62272007.

Data Availability Statement: Not applicable.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hail, M.A. IoT-NDN: An IoT architecture via named data netwoking (NDN). In Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bali, Indonesia, 1–3 July 2019; pp. 74–80.
2. Aboodi, A.; Wan, T.C.; Sodhy, G.C. Survey on the Incorporation of NDN/CCN in IoT. *IEEE Access* **2019**, *7*, 71827–71858. [CrossRef]
3. Zhang, L.; Estrin, D.; Burke, J.; Jacobson, V.; Thornton, J.D.; Smetters, D.K.; Zhang, B.; Tsudik, G.; Claffy, K.; Krioukov, D.; et al. *Named Data Networking (NDN) Project*; Technical Report NDN-0001; Xerox Palo Alto Research Center-PARC: Palo Alto, CA, USA, 2010; Volume 157, p. 158.
4. Buragohain, M.; Nandi, S. Demystifying security on NDN: A survey of existing attacks and open research challenges. In *The "Essence" of Network Security: An End-to-End Panorama*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 241–261.
5. Chatterjee, T.; Ruj, S.; Bit, S.D. Security issues in named data networks. *Computer* **2018**, *51*, 66–75. [CrossRef]
6. Kumar, N.; Singh, A.K.; Aleem, A.; Srivastava, S. Security attacks in named data networking: A review and research directions. *J. Comput. Sci. Technol.* **2019**, *34*, 1319–1350. [CrossRef]
7. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. Blockchain technology overview. *arXiv* **2019**, arXiv:1906.11078.
8. Luo, L.; Guo, D.; Ma, R.T. Optimizing bloom filter: Challenges, solutions, and comparisons. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1912–1949. [CrossRef]
9. Enguehard, M.; Droms, R.; Rossi, D. On the cost of secure association of information centricthings. In Proceedings of the 3rd ACM Conference on Information-Centric Networking, Kyoto, Japan, 26–28 September 2016; pp. 207–208.
10. Compagno, A.; Conti, M.; Droms, R. Onboardicng: A secure protocol for on-boarding iot devices in icn. In Proceedings of the 3rd ACM Conference on Information-Centric Networking, Kyoto, Japan, 26–28 September 2016; pp. 166–175.
11. Mick, T.; Tourani, R.; Misra, S. LASeR: Lightweight authentication and secured routing for NDN IoT in smart cities. *IEEE Internet Things J.* **2017**, *5*, 755–764. [CrossRef]
12. Kar, P.; Misra, S.; Mandal, A.K.; Wang, H. SOS: NDN Based Service-Oriented Game-Theoretic Efficient Security Scheme for IoT Networks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 3197–3208. [CrossRef]

13. Huang, H.; Wu, Y.; Xiao, F.; Malekian, R. An efficient signature scheme based on mobile edge computing in the NDN-IoT environment. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 1108–1120. [CrossRef]
14. Qu, D.; Lv, G.; Qu, S.; Shen, H.; Yang, Y.; Heng, Z. An effective and lightweight countermeasure scheme to multiple network attacks in NDN. *IEEE/ACM Trans. Netw.* **2021**, *30*, 515–528. [CrossRef]
15. DiBenedetto, S.; Papadopoulos, C. Mitigating poisoned content with forwarding strategy. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 164–169.
16. Yang, N.; Chen, K.; Wang, M. SmartDetour: Defending blackhole and content poisoning attacks in IoT NDN networks. *IEEE Internet Things J.* **2021**, *8*, 12119–12136. [CrossRef]
17. Kim, D.; Bi, J.; Vasilakos, A.V.; Yeom, I. Security of cached content in NDN. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2933–2944. [CrossRef]
18. Lei, K.; Fang, J.; Zhang, Q.; Lou, J.; Du, M.; Huang, J.; Wang, J.; Xu, K. Blockchain-based cache poisoning security protection and privacy-aware access control in NDN vehicular edge computing networks. *J. Grid Comput.* **2020**, *18*, 593–613. [CrossRef]
19. Alsamhi, S.H.; Almalki, F.A.; Afghah, F.; Hawbani, A.; Shvetsov, A.V.; Lee, B.; Song, H. Drones' edge intelligence over smart environments in B5G: Blockchain and federated learning synergy. *IEEE Trans. Green Commun. Netw.* **2021**, *6*, 295–312 [CrossRef]
20. Alsamhi, S.H.; Shvetsov, A.V.; Shvetsova, S.V.; Hawbani, A.; Guizani, M.; Alhartomi, M.A.; Ma, O. Blockchain-Empowered Security and Energy Efficiency of Drone Swarm Consensus for Environment Exploration. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 328–338. [CrossRef]
21. Myrzashova, R.; Alsamhi, S.H.; Shvetsov, A.V.; Hawbani, A.; Wei, X. Blockchain Meets Federated Learning in Healthcare: A Systematic Review with Challenges and Opportunities. *IEEE Internet Things J.* **2023**. [CrossRef]
22. Nour, B.; Sharif, K.; Li, F.; Moungla, H.; Liu, Y. A unified hybrid information-centric naming scheme for IoT applications. *Comput. Commun.* **2020**, *150*, 103–114. [CrossRef]
23. Simon, W.A.; Ray, V.; Levisse, A.; Ansaloni, G.; Zapater, M.; Atienza, D. Exact neural networks from inexact multipliers via fibonacci weight encoding. In Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 5–9 December 2021; pp. 805–810.
24. Jeet, R.; Arun Raj Kumar, P. A survey on interest packet flooding attacks and its countermeasures in named data networking. *Int. J. Inf. Secur.* **2022**, *21*, 1163–1187. [CrossRef]
25. Gündoğan, C.; Amsüss, C.; Schmidt, T.C.; Wählisch, M. Content Object Security in the Internet of Things: Challenges, Prospects, and Emerging Solutions. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 538–553. [CrossRef]
26. Anjum, A.; Olufowobi, H. Towards Mitigating Blackhole Attack in NDN-Enabled IoT. In Proceedings of the 2023 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 6–8 January 2023; pp. 1–6.
27. Sabir, Z.; Amine, A. NDN vs TCP/IP: Which One Is the Best Suitable for Connected Vehicles? In *Recent Advances in Mathematics and Technology, Proceedings of the First International Conference on Technology, Engineering, and Mathematics, Kenitra, Morocco, 26–27 March 2018*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 151–159.
28. Geravand, S.; Ahmadi, M. Bloom filter applications in network security: A state-of-the-art survey. *Comput. Netw.* **2013**, *57*, 4047–4064. [CrossRef]
29. Krichen, M.; Lahami, M.; Al-Haija, Q.A. Formal Methods for the Verification of Smart Contracts: A Review. In Proceedings of the 2022 15th International Conference on Security of Information and Networks (SIN), Sousse, Tunisia, 11–13 November 2022; pp. 1–8. [CrossRef]
30. Abdellatif, T.; Brousmiche, K.L. Formal Verification of Smart Contracts Based on Users and Blockchain Behaviors Models. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–5. [CrossRef]

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.