future internet

MDPI

Article

A Register Access Control Scheme for SNR System to Counter CPA Attack Based on Malicious User Blacklist

Jia Shi [1,2], **Xuewen Zeng** [1,2] and **Yang Li** [1,2,*]

[1] National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, No. 21, North Fourth Ring Road, Haidian District, Beijing 100190, China; shij@dsp.ac.cn (J.S.); zengxw@dsp.ac.cn (X.Z.)

[2] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, China

[*] Correspondence: liyang@dsp.ac.cn

Abstract: Standalone Name Resolution (SNR) is an essential component of many Information-Centric Networking (ICN) infrastructures that maps and stores the mappings of IDs and locators. The delivery of data can be realized only when the name resolution process is completed correctly. It also makes the SNR become the key target of network attackers. In this paper, our research focuses on the more covert and complex Content Pollution Attack (CPA). By continuously sending invalid content to the network at a low speed, attackers will consume a lot of the resources and time of the SNR system, resulting in a serious increase in the resolution delay of normal users and further cache pollution in ICN. It is difficult to be quickly detected because the characteristics of attack are inconspicuous. To address the challenge, a register access control scheme for an SNR system based on a malicious user blacklist query is proposed. A neighbor voting algorithm is designed to discover possible attacks in the network quickly and build a blacklist of malicious users reasonably. Users on the blacklist will be restricted from accessing the ICN network during the registration phase with the resolution system. Incentives and punishments for network users are introduced to automate responses about the potential malicious behavior reports. Our scheme is more efficient as users do not have to wait for an additional system component to perform operations. In addition, our algorithm can better solve the collusion problem in the voting process when compared with the others. We experimentally evaluate our protocol to demonstrate that the probability of successful collusion attack can be reduced to less than 0.1 when the attacker ratio is 0.5.

Keywords: content pollution attacks; self-certifying naming; decentralization; Standalone Name Resolution (SNR); voting mechanism

check for updates

Citation: Shi, J.; Zeng, X.; Li, Y. A Register Access Control Scheme for SNR System to Counter CPA Attack Based on Malicious User Blacklist. *Future Internet* **2021**, *13*, 262. https://doi.org/10.3390/fi13100262

Academic Editor: Sherali Zeadally

Received: 8 September 2021
Accepted: 9 October 2021
Published: 13 October 2021

1. Introduction

Information-Centric Networking (ICN) is an emerging network architecture for the future network, whose most important characteristic is that it provides support for the identifier (ID) and locator separation. Thus, an infrastructure that maps and stores the mappings of IDs and locators is needed and named as the Name Resolution System (NRS) [1,2]. NRS is an essential component of the ICN infrastructure. The delivery of data or content can be realized only when the name resolution process is completed correctly. Meanwhile, it also become the key target of network attackers, especially for the Standalone Name Resolution (SNR) approach [3,4]. The approach has been adopted by a number of research projects based on the advantages such as being easier to deploy, higher security, and less change to the underlying structure of the network than Name-Based Routing approach (NBR) in two kinds of name resolution approaches of existing ICN architectures, such as DONA [5], MobilityFirst [6], PURSUIT [7], NetInf [8,9], SEANET (on site, elastic, autonomous network) [10], SAIL [11], etc. In the SNR approach, the name resolution process and message routing are decoupled, and it usually uses flat names to look up content's locators (e.g., the IP or NA), and then, the content is routed by the locators. Content publishers and subscribers need to register and authenticate through SNR nodes in order to publish and subscribe content, which makes SNR nodes become important management nodes in the system and the key attack objects of malicious attackers. The two biggest security threats are DDOS attack and Content Pollution Attack (CPA). Compared with the DDOS attack, the CPA attack is a quite covert attack. Attackers disguise as legitimate users to register with the SNR system and continuously send invalid content to the network at a low speed, thus consuming a lot of the resources and time of the SNR system to serve the attackers, resulting in a serious increase in the resolution delay of normal users and even failure to register or to resolve identifiers. However, due to its low-speed characteristic, this attack is difficult to be found and handled quickly by the system. Meanwhile, a large amount of invalid contents injected into the network will further form cache pollution. Therefore, an access control scheme to restrict potential malicious user registration in SNR nodes is necessary. Users are the first to perceive the impact of network attacks. Through the user voting algorithm, network attacks can be found and handled most quickly. Combined with self-certification [5] to authenticate identity and content, we take the immediate revocation of a misbehaving user's self-certifying identifier and his public–private key pairs as primary protection for the safety of SNR nodes and the whole ICN system. To be specific, we propose a decentralized revocation approach by a voting incentive algorithm to reasonably build a blacklist of malicious users and develop a register access control scheme for the SNR system to counter a CPA attack based on the blacklist in this paper. Our scheme can not only solve the content pollution attack of the SNR system but also restrict further form cache pollution. The main contributions are as follows:

1. We introduce a novel scheme to counter content pollution attack from the perspective of SNR system security. We analyze the significant impact of content pollution attacks on the SNR system. As far as we know, the existing content pollution attacks carry out security detection and defense measures on the cache.
2. We give the complete rules of invalid content discovery, reporting and voting revocation process, and the progressive relationship between invalid content revocation and public key revocation, which gives the reasonable process of being identified as blacklist users.
3. We designed a series of rules for network users to automate responses about the potential malicious behavior report and prove the rationality and high reliability of these rules. We prove the robustness of the voting scheme compared with others in different collusion attacker probabilities. Experiments show that with the voting weight continuously increased, the probability of successful collusion attack can be reduced to less than 0.1 when the attacker ratio is 0.5.

The remaining sections of this article are as follows. Section 2 provides a brief introduction of related work on the security and privacy for the SNR system, content pollution attacks in caching, and the distributed voting mechanisms. Section 3 mainly describes the system architecture of our proposed scheme. The basic definition and rules, initialization, and voting procedure of the voting algorithm is introduced in detail in Section 4. Then, in Section 5, we evaluate and analyze the security of our approach. The scenario set up and performance evaluation are demonstrated in Section 6. We conclude the paper and present some future plans in Section 7.

2. Related Work

ICN being a relatively new area of research, most effort has been focused on developing an efficient resolution framework [11–16]. Only few prior studies have explored issues related to security and privacy for the SNR system [17–20]. The authors in [19] mainly focus on the security of the NetInf architecture, analyzing its vulnerability to security attacks in form of data poisoning in the SNR and Denial of Service (DoS). Paper [21] analyzes a potential security threat and proposes an enhancement to address the discovered threat combined with the SNR system. The new enhancement has been formally verified using the formal method approach based on the ID-Based Cryptography (IBC). However, the impact of more covert content pollution attacks on the security of the resolution system has been ignored by researchers. Meanwhile, for content pollution attacks detection, the existing ICN research mainly focuses on caching [22–24]. Most of the current detection algorithms need to manually set thresholds. These methods have poor adaptability to different environments. Paper [25,26] shows that in ICN cache pollution attacks, cache routes are difficult to perceive the existence of attacks. It does not consider the impact of content pollution attacks on the SNR system and the important role of the SNR system in solving such attacks. However, none of these studies considered the problem that when the content pollution attack can be detected, the pollution has formed a certain scale. At the same time, the change of popularity also needs to be carried out through user feedback, but the existing research does not explain how to ensure the reliability of this process, that is, how to ensure that users do not cheat? The distributed voting mechanism research is investigated as follows. Raya et al. [27] take the misbehaving node secluded by neighboring vehicles until the CA issues a centralized revocation for the vehicle and allows vehicles to detect an attacker or malicious user in the neighborhood. Matsumoto et al. proposed a new PKI system with instantaneous automatic response [28]. The system guarantees the credibility of the public key issued by CAs by rewarding the CA that publishes the digital public key correctly, punishing the CA that does misbehave, and rewarding the reporter who does not have the authorized public key. Lu et al. proposed a trust model to improve the credibility of information [29]. The model is based on the direct historical interaction and the indirect view of the sender, and it depends on the reputation of the sender. Based on the Shamir algorithm, paper [30] proposes an immediate public key revocation scheme based on neighbor vehicle voting. Decentralized revocation is more flexible and effective, because it can immediately revoke the privileges of malicious vehicles to protect the privacy and network security.

Inspired by the above distributed management schemes, we design a neighbor voting algorithm to discover possible attacks in the network quickly and build a blacklist of malicious users reasonably for self-certifying named ICN architecture. We introduce a series of novel rules for network users to automate responses at the potential malicious behavior report for adapting ICN infrastructures. Our scheme is more efficient as users do not have to wait for an additional system component to perform operations. We describe our approach in detail in the next section.

3. Basic Definitions and System Framework

In this section, we provide a system model combined with the self-certifying naming ICN structure. User ID is strongly associated with the public key (hash of the public key), so the revocation of the user ID is the revocation of the public key, and the blacklist of the public key equates to the user blacklist. In the following algorithm description, we will use the public key revocation to explain the generation of Public Key Revocation Blacklist (PKBL).

Meanwhile, we choose Standalone Name Resolution (SNR) approach as the routing approach to design the management model. We distribute different initial voting weights according to different grades of the public key. Voting algorithm rules will be described in the next section.

3.1. System Framework

Our reference architecture is based on our funding project SEANET Technology Standardization Research System Development [10], but we believe that our scheme can be adapted to any SNR resolution-based and named network objects on self-certifying ICN architecture such as DONA or NetInf. The system framework design is as follows. Three main parts corresponding to the system are ***Users***, ***SNR nodes,*** and ***Audit Institutions***, as shown in Figure 1.

Users refers to the content publishers and consumers in ICN. Publishers advertise an information item they possess by publishing information about the item's identifier to an SNR node. Consumers request access to an information item by sending a subscription message to the SNR node, which manages a one-to-one or one-to-many mapping relationship on the item's identifier and its network address (NA). A user can be a publisher or a consumer at the same time. Generally, every legitimate user in the system has the following properties:

- Independently generate public and private key, user ID.
- Initiate content query request, initiate name registration request, content registration request.
- Initiate public key or content revocation events, reply the corresponding voting request.

SNR nodes mean the resolution nodes of Standalone Name Resolution (SNR). The SNR domain means each network user in the domain will have the same logical resolution handler (SNR node). In the SNR approach, the name resolution process is decoupled with message routing, and it usually uses flat names to look up content's locators (e.g., IP); then, the content is routed by the locators. The resolution is essentially a lookup service that maps information requests to information advertisements. In addition to normal resolution services, SNR nodes also have the following functions:

- Verify public key and user ID, complete the registration and query request.
- Store PKBL (Public Key Revocation Blacklist), update PKBL, and synchronize malicious PKBL to other SNR nodes.
- Handle error content or key pairs revocation reports in the resolution domain, store and update the SWL (Security Weight List) period.

To the existing SNR nodes of ICN architecture, the resolution node only needs to add the storage and update two lists, PKBL and SWL, without changing the system architecture design, which is easy to deploy and implement.

Audit Institutions are trusted third parties to verify and store the corresponding identity of the users. It is an institution that is responsible for reviewing identity information, issuing the user's initial security weight, and issuing public key certificates. It is a trusted authority confirmation organization by default, such as a distributed blockchain structure or a centralized third-party organization. SNR nodes and user nodes can query the public key registration real identity information of the users from the Audit Institutions when needed.

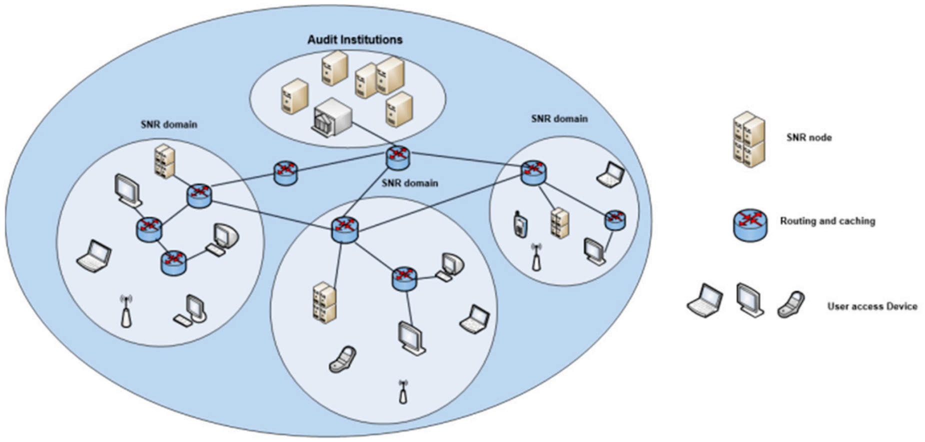

Figure 1. Relationship among user nodes, SNR nodes, and Audit Institutions.

3.2. The Public Key Grades

There are a large amount of users with different identities in the system. Our approach relies on the voting response of legitimate users. In order to reasonably evaluate the credibility of each voting user, we classify the user public key into four grades. At the same time, the credibility of the user becomes higher in high grade of public key, because generating a high-grade public key will be required to complete a more rigorous user verification process during the certificate generation phase by Audit Institutions or a long-term accumulation of positive contributions to network security. It is avoided by malicious users. The public key application objects will be defined as the following. More notably, we will explain in detail the difference and correlation between the grade of the public key and the grade of the digital certificate. The setting rules of the initial security weight value will be introduced in the algorithm in Section 4.

We divide public keys into the following four grades: *Top Grade, Professional Grade, Personal Grade,* and *Basic Grade.* In the public key and certificate generation phase, the user's public key is initially graded based on the scan type of his/her real identity—just like the grade of digital certificates as following:

Top Grade: It is generally used by users with high security requirements such as finance, banking, and e-commerce. Top grade public key certificate generation has the most complex user verification process. Its initial security weight value is also maximum.

Professional Grade: A general enterprise public key is suitable for administration, scientific research institutions, and universities, mailbox, forum, and other large and medium-sized websites. The complexity of the user verification process and initial security weight value are lower than *Top grade* but higher than the *Personal grade* and *Basic grade.*

Personal Grade: It is mainly used for personal users with rich content resources such as bloggers and "we media". The user verification process and initial security weight value is higher than in *Basic grade.*

Basic Grade (quick public service): The public key certificate is issued quickly, with time efficiency and a low security level, aiming at other general users. The user verification process is the easiest, and the initial security weight value is at a minimum.

The security weights will increase or decrease due to the user's contribution to the network security, that is, the user's contribution to the network security events can increase the user's security weight, which may raise the user's public key grade to a higher grade: for example, from *Basic grade* to *Personal grade*. On the contrary, threating behavior to network security can reduce the user's security weight, which may reduce the user's public key grade to a lower grade. It is important to note that the upgrade of the public key only represents the increase in users' credibility in the network and has a higher voting weight in the subsequent voting process. However, its own function cannot be changed. For example, a *Basic grade* user has changed his public key into a *Top grade* by actively participating in network security events for a long time, but he is still just an ordinary user and cannot carry out banking business, but his credibility in the network has become higher.

3.3. Name Registration Process and Content Publish or Query

Users who want to publish contents need to obtain legal public keys and certificates from the Audit Institutions and then send their own user ID (UID) and user's public key to the resolution node. The node verifies the validity of time stamp and the public key, and it checks whether the public key is in the PKBL list, then verifying the validity of the UID by self-certifying naming. The resolution node generates a random number encrypted by the user's public key and sends it to the user. The user uses the private key to decrypt the random number and sends it back to the resolution node, so the node can verify the user identity. After obtaining registration permission, the user gets permission to publish or query content by content ID (CID). Content integrity and publisher authenticity can be checked by self-certification. Figure 2 shows this registration and content publish or query process.

In the next chapter, we will describe how to discover and revoke invalid content and malicious users through voting algorithms.

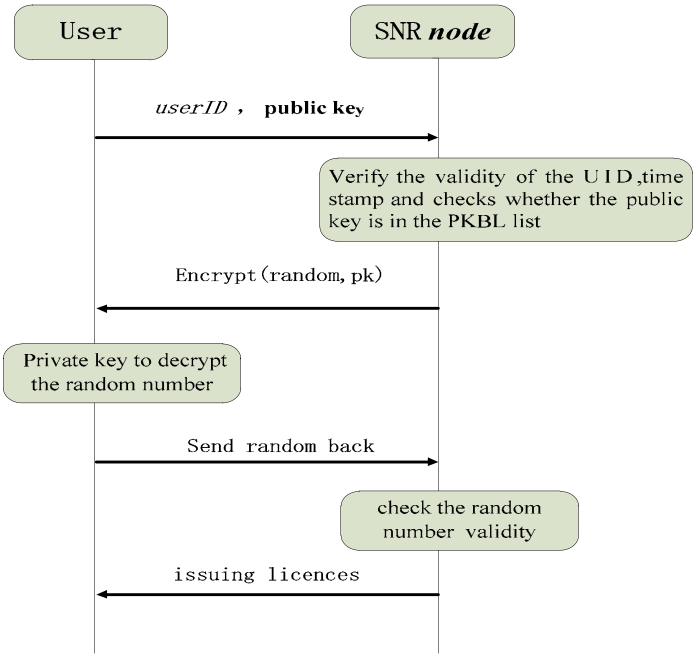
User
SNR *node*
userID , **public key**
Verify the validity of the UID,time stamp and checks whether the public key is in the PKBL list
Encrypt(random, pk)
Private key to decrypt the random number
Send random back
check the random number validity
issuing licences

Figure 2. Name registration process.

4. Voting Algorithm

We explain the voting algorithm, including its basic definition, rules, initialization, and voting procedure in this section.

4.1. Revocation Scheme in Adversary Model

4.1.1. Adversary Model

1. Attackers change contents or send unavailable contents during transmission for content pollution.
2. Other malicious acts of attackers are found by legitimate users and the legitimate users can also initiate a vote for revocation. At this time, although the SNR system has not been attacked, malicious users pose a threat to other parts of the network and should also be blacklisted.

4.1.2. Revocation Scheme

Each UID can register and publish multiple CIDs. Only a small part of the contents of a user having problems is not necessarily a malicious attack. It is obviously unreasonable that revoking the public key of the user results in invalidation of all the contents published by the user. The revocation of CID and UID should be a progressive process. We clarify the revocation hierarchy according to the following rules:

When a user in the network receives a content error, the content error can be initiated as a content revocation event carried out in accordance with the rules of Section 4.2. When the result of successful revocation is generated, it is reported to the SNR. The SNR revokes the content and notifies the user who published the content. The user will be punished by security weight deduction. When the error content numbers associated with the user accumulate to a set value, the user who publishes the error contents is considered to be a malicious user. The SNR revokes the public key of the malicious user and all the content published by the user. Users can report and initiate the revocation request vote event of the malicious user's public key when he finds the malicious behavior.

Before describing the process of the voting algorithm, we first give the rules and basic definition of the algorithm.

4.2. Basic Definition and Rules of the Voting Algorithm

An effective decentralized revocation scheme should have sufficient incentives for users to automate the processing of a revocation report. The scheme needs to clearly address the following issues: how can we better incentivize correct behavior and the reporting of misbehavior, and how can we formally define what it means for a user to behave correctly? What incentives can we offer to users? What mechanisms are necessary for automating the handling reports of misbehavior, and what benefits does automation provide? A clear definition of these problems is given in the following description.

4.2.1. Advantages of User Active Response

Our approach relies on the active response of the users in the domain. The incentives of our strategy for the users are described as follows:

1. The number of allowable registered content items increases with the grades moving up. The relationship between UID and CID is a one-to-many mapping relationship. A user can publish more content when its security weight is at high grades.
2. Users with high security weight will have more credibility with SNR nodes, which will result in a higher response speed. When they initiate voting or participate in voting, the users have the higher voting weight and own the higher reputation of other users in the network. If they receive attacks, they can quickly complete the revocation of the attack's public key.
3. Users with higher security weight enjoy higher tolerance of security misbehavior conducted by themselves. Both the ICN network and the traditional network have their own threshold definition for the occurrence of security attacks based on their own characteristics. However, there are misjudgments in the definition of the threshold. Even if a user enters the scope of the security attack threshold, it may be the normal behavior. In this case, the user with higher security weight can have a higher reputation and error tolerance, in order to avoid unnecessary loss caused by misjudgment.

Next, we give the rules of reward and punishment, giving the mechanisms for automating the handling reports of misbehavior. We distribute different initial security weight according to different grades of the public key. The initial weight value should ensure the effective division of normal users and malicious users under the specified weight calculation rules and content error tolerance of each level. We set up rules for weight increase and decrease first in the following description; then, we give the initialization of security weights.

4.2.2. Rules for Weight Increase and Decrease in Security Weight Value

The increase or decrease in security weights are mainly reflected in the user's contribution to the network security; that is, the user's contribution to the network security events can increase the user's security weight, while threating behavior to the network security can reduce the user's security weight. The rules are as follows:

1. When the user participates in the voting process or reports misbehaving and malicious behavior to the network actively, the security weight is increased by one at one time.
2. When publishing invalid or other prohibited content, the user's security weight decreases with the number of times. The first time security weight is reduced by one, the second by two, and so on (the number of times is recorded by SNR nodes).
3. If there are verified attacks on other users or SNR nodes in the network, the first-time attacking user's security weight is halved and warned, and the second time, the public key is directly revoked.

4.2.3. The Initialization of Security Weight

The allocation principle is based on the user public key grades. The higher the grade, the higher the security coefficient representing the user, and the higher the security weight in the security event. The initial security weight value is set to SW_1, SW_2, SW_3, and SW_4

for Basic Grade, Personal Grade, Professional Grade, and Top Grade, respectively. SWL (Security Weight List) is a list of the security weight values of all nodes in a domain. The four grades correspond to different voting weights in the voting algorithm, which are VW_1, VW_2, VW_3, and VW_4 for the Basic Grade, Personal Grade, Professional Grade, and Top Grade, respectively. The proportion of votes weight VW_1, VW_2, VW_3, and VW_4 is 1:2:3:4. The specific value can be increased proportionally as needed. For four grades of users, the initial security weight value increases according to the grade, and the setting standard is unified and public in the whole network by Audit Institutions. At the same time, the voting weight value VW_i of each grade corresponds to the user's security weight value SW_i one by one. We can derive the value of VW_i from the value of SW_i according to the published rules. For example, when SW_1 = 100, SW_2 = 150, SW_3 = 200, SW_4 = 250, and VW_1 = 1, VW_2 = 2, VW_3 = 3, VW_4 = 4, if a user's security weight is 156, we can infer that his voting weight is 1, or if his SW_i *is* 203, then his voting weight is VW_2; i.e., 2. With this mapping rule, we will not store the VW_i values of all user nodes and reduce the system overhead. The setting standard of a specific SW_i value is limited by the following conditions. We assume that ET_i is the number of content error tolerance; according to the rules for security weight increase and decrease, the initial weight value should be set as the following according to the rules above:

$$SW_i - 1 - 2 - \cdots - ET_i \leq 0 \Rightarrow \frac{(1 + ET_i) * ET_i}{2} \geq SW_i. \tag{1}$$

When the number of content errors reaches ET_i, the weight SW_i is reduced to 0, which is considered as a malicious user, and the public key is revoked. For example, we set SW_1 = 100; then, ET_i = 14, SW_1 is reduced to 0 and meets the Formula (1) conditions. It means that SW_1 level users are only allowed to publish less than 14 invalid or incorrect contents; otherwise, they will be considered as malicious users. If the security weight of the higher grades has been reduced to the lower grades, its public key grades will also be degraded. For example, if the *Top-Grade* user's security weight value is reduced to SW_3 due to multiple misbehaving, then its public key grade is reduced to Professional Grade. On the contrary, if the security weight is increased to a higher grade due to good performance, the public key will also be upgraded. The subsequent SW_i *value* update is based on the table generated during the voting procedure, which is called RVWL (Revocation Voting Weight List). It is an array with each element in the form of three-tuple composed of <UID_x, SW_{xnew}, α_x > from all voting users. It is generated and updated by the initiator of the revocation within the valid voting time. After the revocation is successful, it is sent to the SNR node for publication. SW_{xnew} is the new security weight due to actively reporting misbehavior and malicious behavior to the network, and α_x is the voting coefficient of users. In the next part of the voting procedure, we will specifically explain the meaning and calculation method of the parameters.

Meanwhile, the primary feature of the ICN network is to ensure the fast and effective content search, which means to find the content nearby. When the publisher of a certain content is malicious, the biggest impact will apply on other users and routing caching nodes in the same resolution domain. Therefore, the algorithm synchronizes the revocation list and treats the resolution domain as a basic unit, and we design the security event level and blacklist synchronization time to reduce the overhead.

4.2.4. Security Event Level and Blacklist Synchronization Time

The main reason of event-level classification for different public key revocation is that if the update period of the Public Key Revocation Blacklist (PKBL) is too long, the security of cross-domain authentication will be affected. However, synchronizing the PKBL, every revocation will cause a lot of unnecessary overhead. We classify security events as three grades—primary, intermediate, and advanced—to define the synchronization time.

1. Users need to revoke their public key due to their own reasons, such as suspected key disclosure, and the revocation request is initiated by the user. The security event is primary and the synchronization time is the set PKBL synchronization cycle time.
2. When a user publishes invalid content many times, the security weight is reduced to the set threshold, and the public key revocation request is initiated by the SNR nodes; then, the security event is intermediate, and the synchronization delay of the PKBL blacklist is half of the primary event.
3. The malicious behavior of the user is reported by other nodes, and the revocation is initiated by the attacked node. The security event is advanced. After the revocation of the public key in the domain is completed, the SNR node directly synchronizes the list to the whole network.

4.3. Voting Procedure

According to the rules made in the previous description, we introduce the voting algorithm and explain the main voting procedure for decentralized revocation. Users who have obtained a legal identity certificate through Audit Institutions and registered through the resolution nodes can initialize the revocation event. They need to generate a revocation request according to the following rules and send it to the local SNR node, which will publish it to other registered users in the domain (domain means SNR domain). After the revocation event is initiated, other neighboring users in the domain vote for the event. Each public key valid user in the domain has the voting right, and the voting weight is scaled to its key grades. The revocation threshold is set to th_{cs} for a given SNR domain. The accumulative threshold value th_c is calculated by Formula (2):

$$th_c = \sum\nolimits_{k \in R} VW_k \times \alpha_k \tag{2}$$

where the users' voting coefficient α_k can be 0, 1, -1. Among them, 0 represents abstention, 1 represents consent, and -1 represents opposition. k is the voting user in R, and R is the set of all users voting for revocation information in the domain.

When the revocation is completed, it is reported to the SNR node in the domain. If it is a public key revocation, the first time, the attacking user's security weight is halved and warned, and the next time, the public key is directly revoked. The SNR node stores it in its own PKBL list, synchronizing it to other resolution nodes within a certain period according to the blacklist synchronization time. If it is content revocation, the SNR node deletes the revoked content and supervises the users who publish the error content to accept the punishment of security weight reduction. The revocation process is described in the following subsection.

4.3.1. Initiated Revocation Request

The revocation message of publisher E_i contains the following: revocation event serial number CN_i, revocation public key Pub_r, or revocation content $Cont_r$ of *publisher* E_r, E_i*'s* public key Pub_i, E_i*'s* UID_i, revocation reason M_i, security weight SW_i, valid time stamp to vote on revocation message T_i, and signature Sig_i of revocation message by E_i. The generated revocation information M_{ir} is as follows:

$$Sig_i = SIG(H(CN_i, M_i, UID_i, \alpha_i, SW_i, Pub_i, Pub_r/Cont_r, T_i)). \tag{3}$$

Then, the initial threshold value th_c is calculated by Formula (4), and VW_i is the voting weight of E_i:

$$th_c = VW_i \times \alpha_i \tag{4}$$

$$M_{ir} = (Sig_i, CN_i, M_i, th_c, UID_i, \alpha_i, SW_i, Pub_i, Pub_r/Cont_r, T_i) \tag{5}$$

$$SW_{inew} = SW_i + 1 \tag{6}$$

SIG (●) is the signature function. *H* (●) is the hash function. After generating message M_{ir}, user E_i still needs to add its UID_i, new security weight SW_{inew}, and α_{i} to RVWL. The user E_i publishes the generated revocation information to the local resolution handler through the publish–subscribe mode. All users in the domain who have subscribed to participate in secure voting events can receive this publication information and choose whether to vote.

4.3.2. Vote Accumulation Stage

When the neighbor subscriber E_x subscribes and receives the revocation request packet, it first verifies the signature and valid time stamp T_i of the revocation request packet from E_i. If the signature is incorrect or the valid time expired, E_x discards the message. Otherwise, E_x extracts the revocation reason to judge and vote, generates its own voting information VM_{xr} as Formula (8), and send the generated voting information to the revocation initiator E_i. After successfully receiving and counting the voting information, E_i calculates the latest accumulative voting threshold th_{cnew} as Formula (9), the latest security weight SW_{xnew} of the successful voting user as Formula (10) and it adds UID_x, SW_{xnew}, and α_x to the RVWL.

$$Sig_x = SIG(H(CN_i, UID_x, \alpha_x, Pub_x, SW_x)) \tag{7}$$

$$VM_{xr} = (Sig_x, CN_i, SW_x, \alpha_x, UID_x, Pub_x) \tag{8}$$

$$th_{cnew} = VW_i \times \alpha_i + VW_x \times \alpha_x \tag{9}$$

$$SW_{xnew} = SW_x + 1 \tag{10}$$

E_i continues to record new voting messages according to the above rules until the th_c reaches the set threshold th_{cs}, which is as shown in Formula (11)

$$\textstyle\sum_{k\in R} VW_k \times \alpha_k \geq th_{cs} \tag{11}$$

where th_{cs} is the setting threshold value for the revocation algorithm. If th_{cs} is not reached within the valid time stamp T_i, the revocation event fails.

4.3.3. Synchronization Revocation Result

When the voting weight th_c reaches the set threshold th_{cs}, revocation result with the RVWL is generated and sent directly to the SNR node in the domain. The results information will be announced in a publish–subscribe mode to voting nodes within an expiration time, and all users participating in voting can verify the results and question unreasonable points to ensure that E_i will not falsify the counting results.

4.3.4. Synchronization Blacklist

If there are no questions with the voting process, the SNR node updates the stored SWL according to the RVWL generated in this round and recalculates the SWL of the punished *publisher* E_r according to the rules of Sections 4.2 and 4.3. It will also add the revoked public key to the PKBL or delete the wrong CID of *publisher* E_r. The nodes participating in the voting can query the security weight increase in their current round of voting through the effective UID information. When the PKBL update cycle is reached, the stored PKBL is sent to other SNR nodes for the whole network to broadcast the update.

The pseudo code of the voting procedure on each user and the E_i for voting information statistics is described as the following Algorithm 1 and Algorithm 2:

Algorithm 1 Voting procedure of neighbor subscribes E_x

```
Input: $M_{ir}$
Output: A voting message $VM_{xr}$
Verify user $E_i$, Check validity of time stamp $T_i$
if invalid then
    Discard the message
else
    Verify $Sig_i$
    if invalid then
        Discard the message
    else
        compute the voting information $VM_{xr}$ and send to $E_i$
    endif
endif
```

Algorithm 2 Voting information statistics of publisher E_i

```
Input: $VM_{xr}$, $th_{cs}$
Output: A voting success or fail message, A voting list RVWL1
Verify $Sig_x$
if invalid then
    Discard the message
else
    Count the voting information
    if the setting threshold $th_{cs}$ is reached, then
        Notify the SNR to deal with $Pub_r$ (revoke or halved) or revoke the error content $Cont_r$
    else
        Continue until the $th_{cs}$ is reached or the time stamp $T_i$ is expired
    endif
endif
```

4.4. Threshold Setting Standard

For the proper setting of threshold value th_{cs}, we should consider the following constraints. For this algorithm, collusion attack is a key security issue. If there are unsafe nodes in the voting users, colluding to vote and the malicious revocation of other users' public keys will cause great security problems. Assuming that the total number of legitimated honest users in the domain is r and the total number of malicious collusion users is k, the voting weight calculation model of users with collusion attack is as follows:

$$th_c = \sum_{i=1}^{4}\sum_{j=1}^{V_i} VW_{ij}\alpha_{ij} + \sum_{i=1}^{4}\sum_{g=1}^{U_i} VW_{ig}\alpha_{ig} \text{ and } \left(\sum_{i=1}^{4} V_i \leq r, \sum_{i=1}^{4} U_i \leq k\right). \quad (12)$$

V_i is the number of legal users in grade i, VW_{ij} is the voting weight corresponding to each honest user, α_{ij} is the honest user's voting coefficient, and U_i is the number of malicious users in grade i. VW_{ig} is the voting weight corresponding to each malicious user, and α_{ig} is the malicious user's voting coefficient.

There are two purposes that collusion attackers want to achieve through conspiracy: one is to prevent the revocation by voting against it; the other is to cause the user's public key or contents to be revoked by mistake by approval voting actively. For the first case, the attacker and his accomplices use the voting rights obtained in the previous stage to vote, so that the accumulative voting threshold th_c cannot reach the approval threshold th_{cs} required to revoke. In order to achieve the attack target, the conspirators in the domain will do their best to vote against it, which means the number of opponents voting in the domain is close to the upper limit k. If the number of legitimate voters in the domain responding to the revocation information cannot reach the set threshold after offsetting the negative vote of the attacker, the collusion attack is considered successful, and the accumulative value th_c

would satisfy Formula (13). In this case, the voting coefficient α_{ig} is -1. At the same time, the threshold value th_{cs} should also be less than the th_{max} as Formula (14); otherwise, the revocation event cannot be completed even if all users in the domain vote.

$$\sum_{i=1}^{4}\sum_{j=1}^{V_i} VW_{ij}\alpha_{ij} + \sum_{i=1}^{4}\sum_{g=1}^{U_i} VW_{ig}\alpha_{ig} = th_c < th_{cs} \tag{13}$$

$$th_{max} = \sum_{i=1}^{4}\sum_{j=1}^{V_i} VW_{ij}\alpha_{ij} + \sum_{i=1}^{4}\sum_{g=1}^{U_i} VW_{ig}\alpha_{ig} \text{ and } \left(\sum_{i=1}^{4} V_i = r, \sum_{i=1}^{4} U_i = k\right) \tag{14}$$

For the second case, the attacker and his conspirators also use the voting right obtained in the previous period to vote so that the accumulative voting threshold value cannot avoid reaching th_{cs}, thus accelerating the revocation process, which should not be revoked. In this case, the collusion attack is considered successful and the threshold value th_c would satisfy Formula (15). In this case, the voting coefficient α_{ig} is 1.

$$\sum_{i=1}^{4}\sum_{j=1}^{V_i} VW_{ij}\alpha_{ij} + \sum_{i=1}^{4}\sum_{g=1}^{U_i} VW_{ig}\alpha_{ig} = th_c > th_{cs} \tag{15}$$

5. Security Analysis

The security analysis of this algorithm is limited to the possibility of network attack and the security of the algorithm when malicious user nodes exist in the network. The security of other cryptography algorithms used in the scheme is not discussed. The cipher algorithm with a highly secure coefficient has been selected by default.

5.1. Security of the Voting Scheme

For this algorithm, publishers cheating and collusion attacks are two key security issues. Firstly, we solve publisher cheating by setting public supervision of the successful revocation results. The results information will be announced in a publish–subscribe mode to ensure that publishers will not falsify the counting results. Secondly, we also analyze and limit the threshold th_{cs} to avoid collusion attack threat in Section 4.4. Meanwhile, the private key signature ensures the unforgeability of user identity. Therefore, the robustness of the voting scheme is proved.

5.2. Collusion Attack and Independent Vote

The threshold based on collusion attack has been discussed in detail in Section 4.4, and it is not covered here.

At the same time, each voter independently receives information from the revocation information initiator through the publish–subscribe mode and votes independently. Except for the deliberate collusion attack, voters will not be affected by other voters and judge the credibility of the revocation information independently, which enables each voting user to make a fair judgment on this voting event.

5.3. Malicious User Mobility

By setting the time threshold period of the revocation PKBL blacklist update and th_{cs} threshold discussed in Section 4.4 for security events, the user can no longer attack other domains after the key is revoked, even if it moves and tries to access within other SNR domains.

5.4. Revocation Information Forged

The revocation information needs to be signed by the initiator. It is proved by cryptography that the revocation signature cannot be forged maliciously by the attacker. The reasons for the user's public key revocation can be divided into two categories. One is its own reasons, such as the private key being stolen, and another is because of security threats to other nodes.

For the first reason, the purpose of the attacker is to steal the user's identity by stealing the private key, so as to carry out attack activities in the network. At this time, although the attacker can forge the user signature and initiate public key revocation information, this behavior has no benefit to the attacker, but it can help the user revoke their already insecure public keys more quickly.

For the second reason, the initiator public key of revocation information is within the validity period and has not been embezzled, so the signature of revocation information is valid.

From the above analysis, we can be sure that when the revocation information appears in the network, it is a valid revocation information.

5.5. Defense against Common Attacks of ICN

The advantage of the voting algorithm is that it can establish the security ecological system for a decentralized environment. At the same time, the automatic incentive mechanism makes our model have better ability to resist CPA attacks. When the attacker steals the private key, forges the identity, and publishes invalid or error content to attack the cache, our key revocation scheme can quickly discover the attack behavior by automated reporting and revoke the attacker's valid identity in the network so as to ensure the security of the system.

6. Performance Evaluation

In order to further evaluate the approach we designed, we use the simulation platform to simulate the above scheme and then analyze and evaluate the performance of the scheme. The experimental environment of simulation is configured as the following: Intel i7-4790 CPU@3.60 ghz (8 CPU cores), memory 4096 MB, system model Dell OptiPlex 9020; the selected network simulation software is *OMNET* software, and it uses a modular open-source multi-protocol network simulation platform. It supports the functions of a wireless communication network and wired communication network modeling, protocol simulation modeling, queuing network modeling, multi-processor and distributed hardware system modeling, hardware system modeling, and it evaluates the performance of the complex software system. In this paper, the simulation of network topology is built based on the *OMNET* simulation platform by importing different scenarios and topology types supported by *OMNET*. The experiment simulated six random topologies under inet-flat type, selected 10 groups of topology generation parameters to generate topology, and tested the number of users—100, 200, 300, and 500, respectively. We assume that the transmission delay of the link is 10 ms and the packet loss rate is 0.5% [31]. The average latency of the SNR node is 10 ms [16]. The *OMNET* presets a random value of response time between 0 and 5 s for each user in the cc-module. The validity of time stamp T_i of the revocation message is set to 10 s. The expiration time of publishing the revocation result is set to 5 s. We conducted the simulation in 10 rounds for each user value to obtain the average revocation delay and number of voters. The number of users at different grades also selects a truncnormal distribution in each test. With the increase in threshold $th_{cs,}$ the total revocation delay and the number of needed votes changed, and the average values of test parameters under different topologies are shown in Sections 6.2 and 6.3.

6.1. Communication Overhead

The communication overhead is the additional communication overhead caused by the increment of revocation messages and signature sizes. Tables 1 and 2 show the additional message sizes in bytes for the algorithm according to Formulas (5) and (8).

Table 1. Revocation message sizes (bytes).

CN_i	M_i	th_c	UID_i	Pub_i	$Pub_r/Contr$	T_i	Sig_i	α_i	SW_i
4	64	1	32	64	64	2	64	1	1

Table 2. Voting message sizes (bytes).

CN_i	SW_x	UID_x	Pub_x	α_x	Sig_x
4	1	32	64	1	64

Therefore, the total cost of a one-time revocation voting message is 4 + 64 + 1 + 32 + 64 + 64 + 2 + 64 + 1 + 1 = 297 bytes. The communication cost of the voting process is 4 + 1 + 32 + 64 + 1 + 64 = 166 bytes. The communication cost of RVWL containing *UID*, SW_{xnew} (1 byte), and α_x is n × (32 + 1 + 1) = 34 × n bytes. Therefore, the communication overhead of the whole event is (34 + 166 + 297) × n = n × 497 bytes where n is the total number of voters, which are in Section 6.3.

6.2. Average Revocation Delay

Figure 3 shows the average revocation delay with a different number of users. The delay model of the revocation event is as shown in Formula (16):

$$T_{delay} = T_{voting} + T_{publish} + T_{resolution} \tag{16}$$

where T_{delay} is the average revocation delay of each voting process of the algorithm, T_{voting} is the time of vote accumulation stage, $T_{publish}$ is the expiration time for publishing the revocation result, and $T_{resolution}$ is the average latency when publishers generated revocation information and submitted it to the local SNR node.

From Figure 3, it can be seen that with the increase in users in the domain, the total revocation delay gradually decreases, because with the increase in the number of users, the density of active users becomes larger, so the total voting delay decreases.

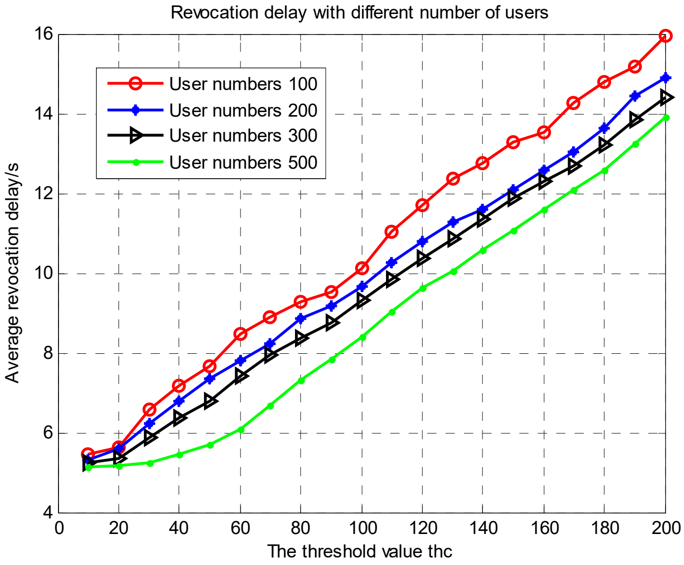

Figure 3. Revocation delay with different numbers of users.

6.3. Average Number of Users Needed to Vote

The number of average voting users is shown in Figure 4. Under the same threshold th_{cs}, the total number of voting users is basically the same with different numbers of users in the domain. This is because the number of voting users is only related to the public key grades of voting users. The higher the user's public key grades in the domain, the fewer users that are needed to vote for reaching the preset revocation threshold in an SNR domain. The total number of users in the domain and the distribution density of users do not affect the number of votes because no matter how many users there are in the domain, in order to reach the same threshold, the number of votes required is independent of the total number of users.

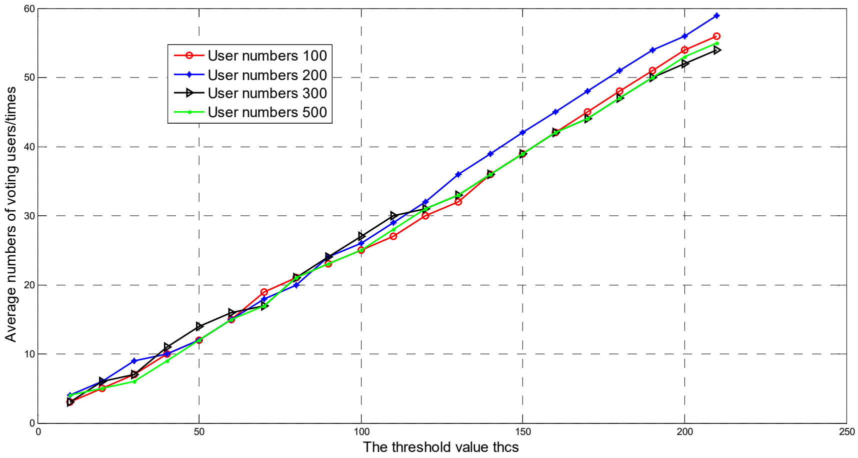

Figure 4. The number of voting users.

6.4. *Selection of Threshold Value th_{cs}*

Figure 5 calculates the probability of revocation success of different users under different threshold conditions, which provides a reference for setting threshold th_{cs}. We count the probability curve of voting schemes when the number of users is 100, 200, 300, and 500 with the C program. We assume that the value of votes VW_1, VW_2, VW_3, and VW_4 is 1, 2, 3, and 4. The voting success probability is defined as the proportion of the number of successful voting combinations, each of which reaches th_{cs}, to the number of all possible voting combinations that also include those failing to reach th_{cs}. Among them, voting user combination is the combination of the number of users at four different grades responding to each voting event. For example, when the threshold value is set to 100 and the total number of users is 200, a possible voting combination can be 30 users at the Basic Grade, 10 users of the Personal Grade, 10 users of the Professional Grade, and 5 users of the Top Grade. However, if only 10 Basic Grade users and 10 Personal Grade users respond to this vote, this voting event fails. The probability curve shows the proportion of changes with the increase in the preset threshold th_{cs}. The numbers of users in the domain and the threshold value th_{cs} are the two important factors affecting the probability of successful revocation. The revocation success probability decreases with the increment of the threshold value $th_{cs.}$ With the increase in the threshold th_{cs}, the number of approval votes required increases, while the number of users with high grades is a small proportion, so the probability of successful vote revocation decreases. In contrast, revocation success probability increases with the increment of user numbers, because the density of the high grades also increases.

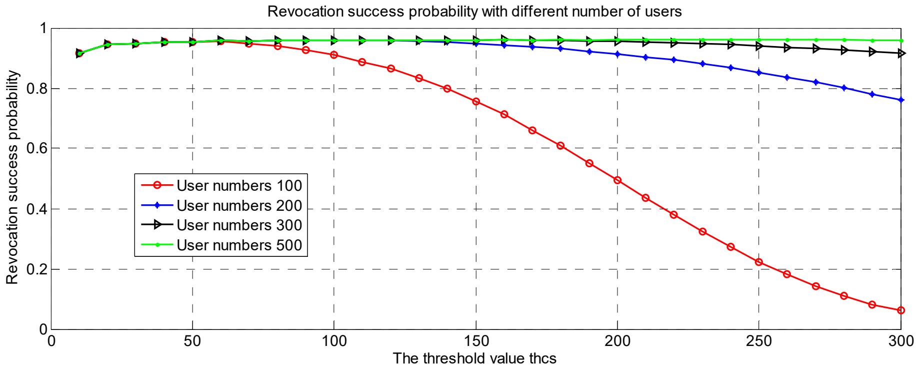

Figure 5. Revocation success probability with different numbers of users.

Meanwhile, according to the analysis in Section 4.4, we also give the setting standard of threshold th_{cs} in different conspirator ratios and users. The probability of successful attack is defined as the probability of revocation failure when there is a certain proportion of attackers. Figure 6 analyzes the collusion attack of condition one (voting against revocation) in Section 4.4. The threshold value th_{cs} satisfies Formula (13). The conspirator ratio is 0.1–0.5, and the number of users is 100, 200, 300, and 500 and the vote weights VW_1, VW_2, VW_3, and VW_4 is 1, 2, 3, and 4, respectively. The four figures represent the attack success probability with different user numbers. Figure 7 analyzes the collusion attack of condition two (approval voting) in Section 4.4. The threshold value th_{cs} satisfies Formula (15). Figure 8 analyzes the change curve of the successful attack probability in Formulas (13) and (15) when the vote weights VW_1, VW_2, VW_3, and VW_4 respectively increase from 1, 2, 3, and 4 to 10, 20, 30, and 40 and the conspirator ratios are 0.1, 0.3, and 0.5 with 100 users. We can have the following conclusions from the above figures:

1. As the number of users in the domain increases, the threshold th_{cs} for the lowest successful attack probability also increases.
2. When the number of users in the domain is constant, the greater the voting weight, and the lower the probability of successful attack.

That means we can reduce the success probability of an attack and improve the robustness of the system by setting the voting weight and threshold. It can be seen from Figure 8 that when the attacker ratio is 0.1 and the values of votes VW_1, VW_2, VW_3, and VW_4 are 2, 4, 6, and 8, the attack success probability has been reduced to 1%. When the attacker ratio is 0.5, the probability of a successful attack can be reduced to less than 0.1 with the voting weight continuously increased, which also reflects the robustness of the system. Even if the system is attacked by half of the total users in the domain, it still maintains a 90% probability of correct revocation. At the same time, due to the limitation of crypto puzzles, it is quite difficult for attackers to occupy 50% of total users, which will consume a lot of computing power. Therefore, our scheme can resist collusion attacks very well and ensure system security.

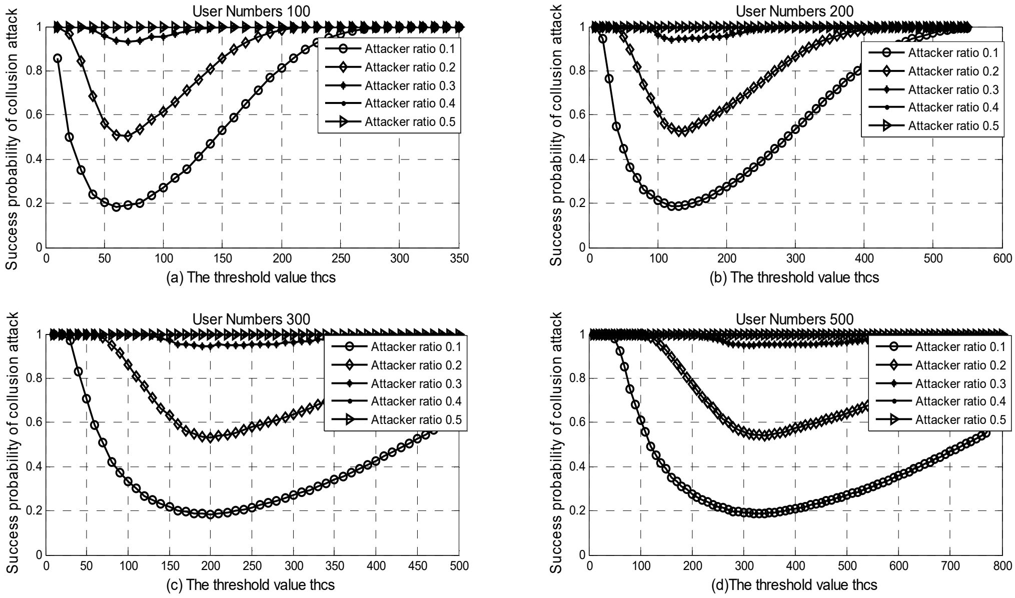

Figure 6. Collusion attacks to prevent the revocation of a malicious public key. Panels (**a**–**d**) show the success probability fluctuation of this kind of collusion attack when user number is 100, 200, 300, and 500.

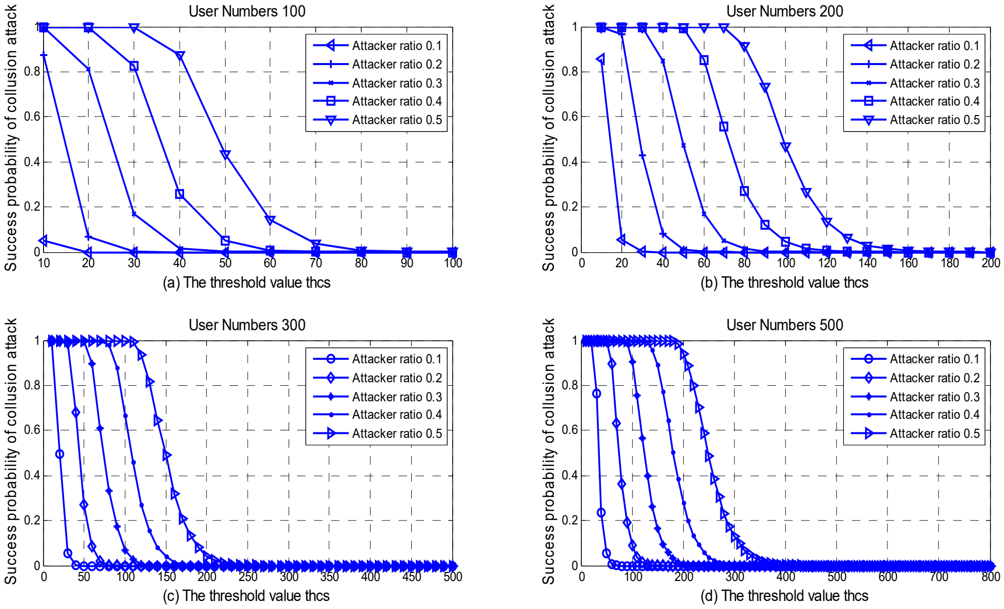

Figure 7. Collusion attacks to error revocation of the legitimate user's public key. Panels (**a**–**d**) show the success probability fluctuation of this kind of collusion attack when user number is 100, 200, 300, and 500.

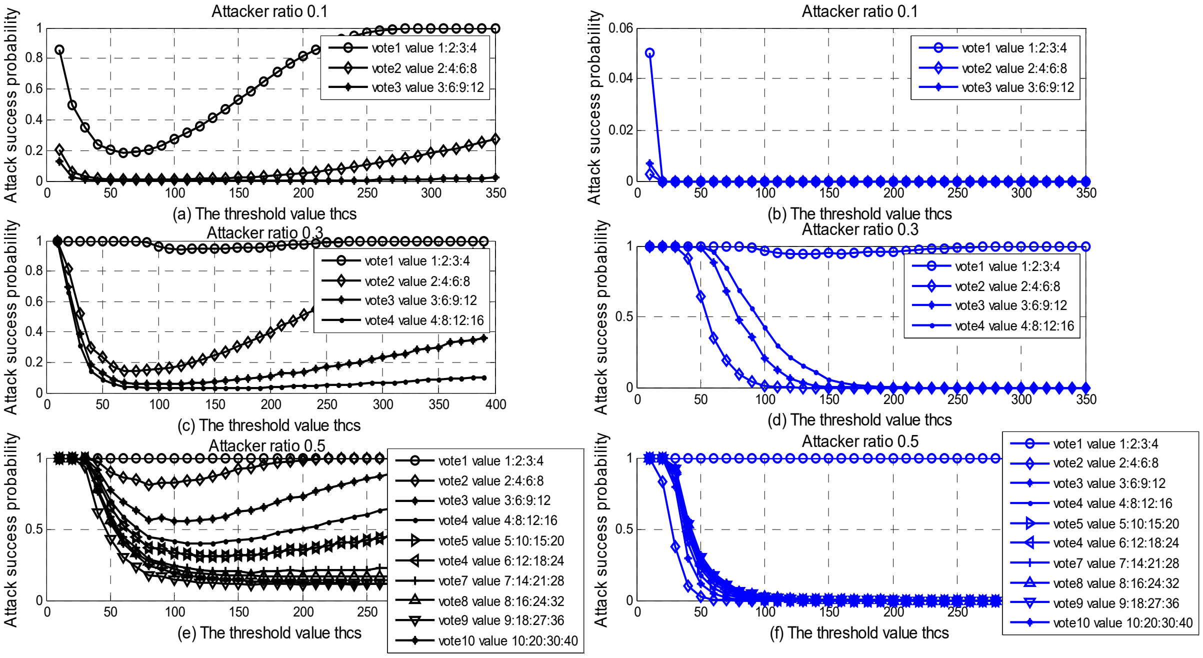

Figure 8. The change curve of the attack probability with increase in VW. Panels (**a**–**f**) analyze the change curve of the successful attack probability in Formulas (13) and (15) when the vote weight VW_1, VW_2, VW_3, and VW_4 increases from 1, 2, 3, and 4 to 10, 20, 30, and 40 and the conspirator ratio is 0.1, 0.3, and 0.5 with 100 users.

7. Conclusions

Based on self-certifying naming, this paper discusses the problems of CPA attack and its solution. Through the user voting algorithm, we build a complete malicious user discovery strategy, which is used to access control of the name resolution system, so as to alleviate the CPA attack against the system and further protect the whole network from large-area cache pollution attacks. Meanwhile, in the decentralized revocation algorithm, the users can revoke any malicious or misbehaving attackers within their communication range. Decentralized revocation is more efficient, as the users do not need to wait for an additional system component to take action, and they can preserve their privacy and network security by revoking the privileges of a malicious user straightaway. The key synchronous revocation list is also clearly defined in the paper. The relevant performance parameters are tested by using the simulation platform. The results show a significant reduction in attack success probability and revocation failure rate when the appropriate threshold is selected, which reflects the robustness of the decentralized system and provides reference data for engineering implementation.

In the next step, we hope to implement the scheme in the experimental ICN project, such as SEANet [10]. We will get more reliable data through the deployment of the actual network system, so as to analyze the adaptability of the scheme for the actual application scenarios and further improve the scheme.

Author Contributions: Conceptualization, J.S. and Y.L.; Data curation, J.S.; Formal analysis, J.S.; Funding acquisition, Y.L. and X.Z.; Investigation, J.S.; Methodology, J.S., Y.L. and X.Z.; Software, J.S.; Supervision, X.Z.; Validation, J.S.; Writing—original draft, J.S.; Writing—review and editing, Y.L. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by SEANET Technology Standardization Research System Development (Project No. XDC02070100). This research was funded by Frontier Basic Research Project of Institute of Acoustics of Chinese Academy of Sciences (Contract No: QYJC201902).

Data Availability Statement: Not applicable, the study does not report any data.

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

References

1. Xylomenos, G.; Ververidis, C.N.; Siris, V.A.; Fotiou, N.; Tsilopoulos, C.; Vasilakos, X.; Katsaros, K.V.; Polyzos, G.C. A survey of information-centric networking research. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 1021049. [CrossRef]
2. D'Ambrosio, M.; Dannewitz, C.; Karl, H.; Vercellone, V. MDHT: A hierarchical name resolution service for information-centric networks. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, Toronto, ON, Canada, 19 August 2011; pp. 7–12.
3. Dong, L.; Wang, G. A hybrid approach for name resolution and producer selection in information centric network. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; p. 57580.
4. Chen, Z.; Meng, H.W.; Guan, Z. Research on intrinsic security in future internet architecture. *J. Cyber Secur.* **2016**, *1*, 10–13.
5. Koponen, T.; Chawla, M.; Chun, B.-G.; Ermolinskiy, A.; Kim, K.H.; Shenker, S.; Stoica, I. A data-oriented (and beyond) network architecture. In Proceedings of the ACM SIGCOMM, Kyoto, Japan, 27–31 August 2007.
6. Raychaudhuri, D.; Nagaraja, K.; Venkataramani, A. MobilityFirst: A Robust and Trustworthy Mobility- Centric Architecture for the Future Internet. *ACM SIGMobile Mob. Comput. Commun. Rev.* **2012**, *16*, 2–13. [CrossRef]
7. Fotiou, N.; Nikander, P.; Trossen, D.; Polyzos, G. Developing information networking further: From PSIRP to PURSUIT. In Proceedings of the International Conference on Broadband Communications, Networks and Systems, Athens, Greece, 25–27 October 2010; pp. 1–13.
8. 4WARD: Web Site (2010). Available online: http://www.4ward-project.eu (accessed on 8 October 2021).
9. Ohlman, B.; Karl, H.; Ahlgren, B.; Farrell, S.; Dannewitz, C.; Kutscher, D. Network of Information (NetInf)—An information-centric networking architecture. *Comput. Commun.* **2013**, *36*, 721–735.
10. Wang, J.; Cheng, G.; You, J.; Sun, P. SEANet: Architecture and Technologies of an On-site, Elastic, Autonomous Network. *J. Netw. New Media Technol.* **2020**, *9*, 1–8. (In Chinese)
11. Edwall, T. Scalable and Adaptive Internet Solutions (Sail). 2011. Available online: https://sail-project.eu/wp-content/uploads/2011/02/SAIL-project-summary.pdf (accessed on 8 October 2021).
12. Louati, W.; Ben-Ameur, W.; Zeghlache, D. A bottleneck-free tree-based name resolution system for Information-Centric Networking. *Comput. Netw.* **2015**, *91*, 341–355. [CrossRef]
13. Barakabitze, A.A.; Xiaoheng, T.; Tan, G. A Survey on Naming, Name Resolution and Data Routing in Information Centric Networking (ICN). *Int. J. Adv. Res. Comput. Commun. Eng.* **2014**, *3*, 8322–8330. [CrossRef]
14. Sevilla, S.; Mahadevan, P.; Garcia-Luna-Aceves, J.J. FERN: A unifying framework for name resolution across heterogeneous architectures. *Comput. Commun.* **2015**, *56*, 124. [CrossRef]
15. Hong, J.; Chun, W.; Jung, H. A flat-name based routing scheme for information-centric networking. In Proceedings of the 17th International Conference on Advanced Communication Technology (ICACT), Pyeonhchang, Korea, 1–3 July 2015.

16. Liao, Y.; Sheng, Y.; Wang, J. A deterministic latency name resolution framework using network partitioning for 5G-ICN integration. *Int. J. Innov. Comput. Inf. Control.* **2019**, *15*, 1865–1880.
17. Loo, J.; Aiash, M. Challenges and solutions for secure information centric networks: A case study of the netinf architecture. *J. Netw. Comput. Appl.* **2014**, *50*, 6472. [CrossRef]
18. Pentikousis, K.; Ohlman, B. *Information-Centric Networking: Evaluation Methodology*; Technical Report; Internet Draft; Pentikousis, K., Ohlman, B., Davies, E., Spirou, S., Boggia, G., Mahadevan, P., Eds.; 2013; Available online: https://datatracker.ietf.org/doc/draft-irtf-icnrg-evaluation-methodology/00/ (accessed on 8 October 2021).
19. Loo, J.; Aiash, M. An integrated authentication and authorization approach for the network of information architecture. *J. Netw. Comput. Appl.* **2014**, *50*, 7379.
20. Aiash, M.; Loo, J. A formally verified access control mechanism for information centric networks. In Proceedings of the 12th International Conference on Security and Cryptography, Colmar, France, 20–22 July 2015; Volume 4, pp. 377–383.
21. Gouge, J.; Seetharam, A.; Roy, S. On the scalability and effectiveness of a cache pollution-based DoS attack in information centric networks. In Proceedings of the International Conference on Computing, Networking and Communications, Kauai, HI, USA, 15–18 February 2016; pp. 1–5.
22. Xie, M.; Widjaja, I.; Wang, H. Enhancing cache robustness for content-centric networking. In Proceedings of the IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2426–2434.
23. Zhu, Y.; Shi, J.; Gong, P.; Cao, Q.; Su, D. Collaborative detection mechanism for low-rate cache pollution attack in named data networking. *J. Beijing Univ. Posts Telecommun.* **2015**, *38*, 44–48.
24. Conti, M.; Gasti, P.; Teoli, M. A lightweight mechanism for detection of cache pollution attacks in Named Data Networking. *Comput. Netw.* **2013**, *57*, 3178–3191. [CrossRef]
25. Li, M.; Sun, Y.; Lu, H.; Maharjan, S.; Tian, Z. Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. *IEEE Internet J.* **2020**, *7*, 6266–6278. [CrossRef]
26. Yao, L.; Fan, Z.; Deng, J.; Fan, X.; Wu, G. Detection and defense of cache pollution attacks using clustering in named data networks. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 1310–1321. [CrossRef]
27. Raya, M.; Papadimitratos, P.; Aad, I.; Jungels, D.; Hubaux, J.-P. Eviction of misbehaving and faulty nodes in vehicular networks. *IEEE J. Sel. Area. Commun.* **2007**, *25*, 1557–1568. [CrossRef]
28. Matsumoto, S.; Reischuk, R.M. IKP: Turning a PKI around with decentralized automated incentives. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 410–426.
29. Lu, Z.; Wang, Q.; Qu, G.; Liu, Z. BARS: A Blockchain-Based Anonymous Reputation System for Trust Management in VANETs. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy In Computing And Communications, New York, NY, USA, 1–3 August 2018; pp. 98–103.
30. Asghar, M.; Pan, L.; Doss, R. An efficient voting based decentralized revocation protocol for vehicular ad hoc networks. *Digit. Commun. Netw.* **2020**, *6*, 422–432. [CrossRef]
31. Song, Y.; Ni, H.; Zhu, X. Analytical Modeling of Optimal Chunk Size forEfficient Transmission in Infor mation-Centric Networking. *Int. J. Innov. Comput. Inf. Control.* **2020**, *16*, 1511–1524.