Hindawi
Security and Communication Networks
Volume 2021, Article ID 2067137, 17 pages
https://doi.org/10.1155/2021/2067137

Research Article

A Privacy Protection Method of Lightweight Nodes in Blockchain

Lin Ge[1,2] **and Tao Jiang**[3,4]

[1]*School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou Henan 450046, China*
[2]*Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin 300300, China*
[3]*National Digital Switching System Engineering and Technological Research Center, Zhengzhou Henan 45002, China*
[4]*National Engineering Laboratory for Mobile Network Security, Beijing 100876, China*

Correspondence should be addressed to Lin Ge; lingesnow@126.com

Received 6 May 2021; Revised 14 June 2021; Accepted 30 June 2021; Published 16 July 2021

Academic Editor: Irshad Azeem

Aiming at the privacy protection of lightweight nodes based on Bloom filters in blockchain, this paper proposes a new privacy protection method. Considering the superimposition effect of query information, node and Bloom filter are regarded as the two parties of the game. A privacy protection mechanism based on the mixed strategy Nash equilibrium is proposed to judge the information query. On this basis, a Bloom filter privacy protection algorithm is proposed when the probability of information query and privacy, not being leaked, is less than the node privacy protection. It is based on variable factor disturbance, adjusting the number of bits' set to 1 in the Bloom filter to improve the privacy protection performance in different scenarios. The experiment uses Bitcoin transaction data from 2009 to 2019 as the test data to verify the effectiveness, reliability, and superiority of the method.

1. Introduction

In recent years, the research and application of blockchain technology have shown explosive growth. It is considered to be the fifth disruptive innovation in the computing paradigm after mainframes, personal computers, the Internet, and mobile/social networks. Blockchain technology is highly transparent, decentralized, detrusted, and collectively maintained. It can be enabling decentralized credit-based interactions in distributed networks where nodes do not need to trust each other [1–4]. Blockchain-based applications are becoming increasingly widespread and cover a wide range of fields, including financial services, reputation systems, the Internet of Things, and security services [5–10].

However, blockchain nodes require a certain amount of disk space to download complete ledger information. And as the scale of blockchain usage grows, verifying the correctness of broadcast blocks and transactions will also bring considerable overhead to nodes. Such problems will become more acute when users use resource-constrained devices, such as mobile devices, smartphones, sensor devices, embedded systems, and virtual private servers [11, 12]. Nakamoto proposed a Simplified Payment Verification (SPV) lightweight node [13], which was extended to use Bloom filters to receive node-related transactions. With the continuous expansion of blockchain applications, SPV nodes have become the most common form of blockchain nodes [14–17].

The SPV node does not store the entire blockchain, nor does it verify all transactions in the system. It only receives a subset of transactions filtered by the connected full nodes. This request for specific data reveals transaction information related to the node unintentionally. A third party in the network can associate related transactions with users by tracking the data requested by the SPV node, thereby posing a threat to privacy and security [18–20]. For example, in digital currency applications, analyzing transactional information can provide access to a user's transaction patterns and enable the deduction of user identity and location information; in financial applications, if a significant amount

of transactional information is obtained, it is possible to both trace individual account transaction details that contain user privacy information and analyze macrofinancial trends that include core company data; in energy industry applications, blockchain is often used to enable peer-to-peer energy exchange where transaction data can potentially reveal sensitive information such as energy transmission [21–26]. With the development of IoT and sensor technology, privacy protection research for lightweight nodes of blockchain has become the focus and difficult research.

2. Related Work

Bloom filter and its application in privacy protection is always a hot research topic. Mullin [27, 28], respectively, proposed the calculation method of the false positive rate of Bloom filter; Bianchi et al. [29] quantified the privacy properties of Bloom filters, but when attackers can access multiple Bloom filters from the same entity, their analysis cannot solve the privacy problem; Yeh et al. [30] proposed a dual-level Bloom filter mechanism to solve privacy and performance efficiency issues. In the first level, unrelated IP lists are excluded, and the second level further determines the relevance of the desired lists. The issues' universality needs to be further verified and improved.

Blockchain privacy protection methods are mainly around data distortion, data encryption, and restricted release [31]. CoinParty [32] adopts a secure multiparty computation protocol to implement an improved scheme that can guarantee the effectiveness of the mixing-coin process in the case of a malicious operation or failure of some hybrid nodes. The lightning network [33] enables secure out-of-chain transactions. In the lightning network, the majority of the transaction details between the users are implemented offline. Bao et al. [34] present Lockmix, providing users with a mixed service by using blind signature and multisignature schemes to prevent attackers from linking the input addresses to the output addresses. Ruffing et al. [35] design ValueShuffle, a mixing protocol that ensures the anonymity of mixing participants and confidentiality of payment values by combining CoinJoin with Confidential Transactions and additionally Stealth Addresses. Liu et al. [36] propose an unlinkable coin mixing scheme that uses ring signatures with elliptic curve digital signature algorithms (ECDSA) to hide coin transfers between addresses.

The privacy protection of lightweight nodes in the blockchain has gradually expanded in recent years. Nojima and Kadobayashi [37] proposed a cryptographic security privacy protection Bloom filter protocol based on the blind signature, but the additional computational load will be generated on the SPV node. Gervais et al. [18] analyzed the privacy protection effect of Bloom filters in lightweight nodes and provided a solution. Their solution is to set the Bloom filter in accordance with different analysis results. Every time the SPV node needs to query the address, it needs to reset the Bloom filter. Their solution lacks systemicity, completeness, and flexibility. Kanemura et al. [38] proposed a privacy-preserving Bloom filter design for Bitcoin's SPV client based on γ-Deniability, but it is necessary to obtain the number of unique Bitcoin addresses that have appeared from the last checkpoint. Qin et al. [39] designed SPV protocol to replace the Bloom filter by using Private Information Retrieval (PIR) to create a fully private and high performance query, but the client needs to obtain the available block headers independently and query the PIR database in an appropriate order. The occupation of bandwidth resources needs to be further optimized, and the proposed protocol is static that cannot reflect the latest state of the blockchain in real time. Henry et al. [40] proposed to make PIR scheme for blockchain transactions to address the problem of fetching transactions privately, which is suitable and efficient for blockchain transactions. Jiang et al. [41] propose a privacy-preserving thin client authentication scheme (PTAS) that uses the concept of PIR to enable thin clients to function properly as full-node users while protecting their privacy. Li et al. [42] design a d-differentially private mechanism based on trusted hardware to secure queries from SPV clients so that semihonest adversaries cannot acquire the real access pattern. Niu et al. [43] propose an efficient transactional query scheme for privacy-preserving lightweight clients running Intel Software Guard Extensions (SGX) enclave on the full node, using a secure enclave to serve transactional query requests from lightweight clients. Le et al. [44] propose a two-tree oblivious random access memory (ORAM) construction to protect SPV clients' requests from a potentially malicious server. Zhou et al. [45] proposed a privacy-preserving two-factor user authentication protocol for the SPV nodes in the Bitcoin network that meets all the security requirements and has provable security.

Game theory was proposed by von Neumann and Morgenstern [46], and in the 1950s, Nash proposed the Nash equilibrium theory [47]. In the strategy combination at the Nash equilibrium point, when the strategy of others does not change, the corresponding benefit is the best. At this time, each rational participant will not or cannot change his strategy in order not to reduce his own benefit. Based on this, this paper proposes a new privacy protection method for lightweight nodes in blockchain and conducts experimental verification. The main contributions are as follows:

(1) Propose a privacy protection mechanism based on mixed strategy Nash equilibrium, combined with historical query records and the privacy protection threshold of SPV nodes, to discriminate the address information that needs to be inserted into the Bloom filter for query.

(2) A privacy protection algorithm based on the variable factor adjustment of the Bloom filter is proposed. Through the disturbance of the variable factor, it increases the number of bits setting to 1 in the Bloom filter, thereby improving its privacy protection performance.

(3) The blockchain Bitcoin ledger data is used as the test data set to verify the feasibility, reliability, and effectiveness of the method proposed in this paper.

The remainder of the paper is organized as follows. Section 3 establishes the system model and attack model; Section 4 proposes a privacy protection mechanism based on the mixed strategy Nash equilibrium and a privacy protection algorithm based on the variable factor disturbance of the Bloom filter; Section 5 uses the Bitcoin ledger's data as data set to verify the method; Section 6 summarizes the full text.

3. SPV Node Privacy Protection Model Based on Bloom Filter

To reduce the communication load running on resource-constrained devices, the SPV node does not store complete ledger information, nor does it verify all transactions in the system. It constructs Bloom filters by embedding all transaction addresses it has used. The Bloom filter completing the initial handshake agreement is outsourced to the full node [13]. Whenever a full node receives transaction information, it will first check whether its input/output address matches the Bloom filter of the SPV node. If it matches, the full node will forward the received transaction to the SPV node, as shown in Figure 1.

It is assumed that an attacker can eavesdrop on the communication link to obtain one or more Bloom filters related to the SPV node. The attacker can also obtain the parameters used to create the Bloom filter (for example, the target false positive rate, the number of hash functions, the number of bits). Since the blockchain is an open network system that is distributed and jointly maintained, the attacker can also access all address/transaction information appearing in the blockchain and their respective execution order, as shown in Figure 2.

4. SPV Node Privacy Protection Method

SPV nodes need to query transaction data related to their addresses, and the transaction data in the blockchain is updated in real time. As the number of additions increases, even if a single query is within the tolerable range of SPV node's privacy leakage, superimposing the address information obtained multiple times may exceed the tolerance of the SPV node to privacy leakage. This paper regards the SPV node and the Bloom filter as the two parties of the game. According to the game theory, the Bloom filter is not hijacked and will not leak privacy/may be hijacked and the privacy is leaked & adding addresses directly/not adding addresses directly is adopted by both parties. In this processing, analyze the strategy and the corresponding benefits the two parties can be obtained and establish a game mechanism as the basis for realizing the privacy protection of SPV nodes. On this basis, a variable factor adjustment algorithm is used to enhance the privacy protection performance of the Bloom filter, thereby further ensuring the security of the address information query service required by the SPV node.

This paper considers the superimposition effect of each query of address information and records the insertion operation of the address information by the Bloom filter through feedback. When the address needs to be inserted again, it will combine the previous records to calculate the corresponding benefits in different states. The game is played based on these gains to obtain the Nash equilibrium so as to obtain the probability that the privacy is not leaked when the address information is inserted at this time. At the same time, the SPV node sets a threshold according to its own tolerance for privacy leakage. The larger the threshold, the lower the user's tolerance and the stronger the awareness of privacy protection. Compare this threshold with the probability of privacy not being leaked. If the probability value is greater than the threshold, this address information will be added to the Bloom filter. Else the Bloom filter will be disturbed by a variable factor to adjust. The method's flowchart is shown in Figure 3.

4.1. SPV Node Privacy Protection Mechanism Based on Mixed Strategy Nash Equilibrium. In this paper, the SPV node and Bloom filter are regarded as the two parties of the game. It is assumed that both parties of the game are rational, and their decision making is in order. However, the strategy adopted by the first decision maker cannot be observed by the latter. Simultaneous decision making is a static game. The game strategies adopted by both sides are natural. For SPV nodes, the strategies that can be adopted are "add addresses directly" or "not add addresses directly"; for Bloom filters, the game strategies that can be adopted are "not hijacked or "hijacked." Different game strategies adopted by the two parties in the game will bring different benefits. When calculating the benefits of both parties, the influence of the private information that the Bloom filter has obtained in the past must also be considered. Based on benefits, there may be Nash equilibrium in the game, and the probability that the Bloom filter is not hijacked can be obtained through the Nash equilibrium. SPV nodes can set a threshold in the privacy protection policy according to their tolerance for privacy leakage. Only when the probability of not hijacked is higher than the threshold is allowed to add addresses. After each address addition is completed, the SPV node records the addition. It is a factor in the benefit calculation of the same Bloom filter in the subsequent game. The more addresses are added, the more private information may be obtained according to the superposition effect, and the greater the probability of leaking user's privacy. Therefore, the corresponding benefits of the same Bloom filter in different times of address addition are different. The probability that it is not hijacked should also decrease until it is lower than the threshold set by the SPV node. At that time, the Bloom filter will no longer add an address.

The combination states of Bloom filter and SPV node are State1 (not hijacked, add address directly), State2 (hijacked, add address directly), State3 (hijacked, not add address directly), and State4 (not hijacked, not add address directly).

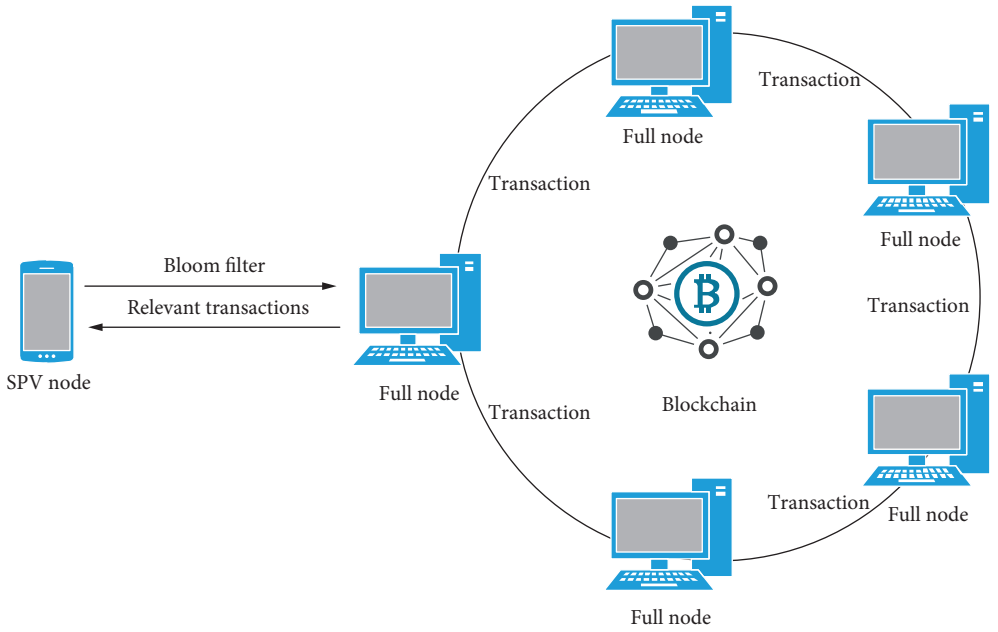

Figure 1: Schematic diagram of SPV node.

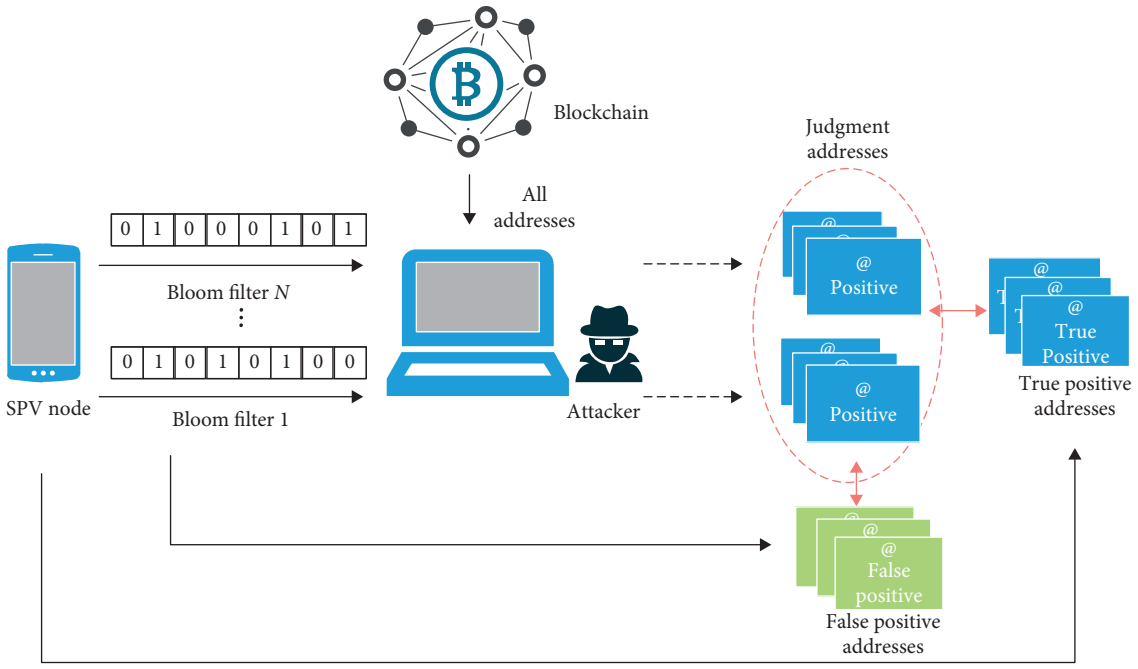

Figure 2: SPV node attack model based on Bloom filter.

Theorem 1. *There is a Nash equilibrium in the game between the SPV node and Bloom filter.*

Proof. SPV node and Bloom filter are the two parties of the game, and their state set is a finite set, so it is a finite strategic game. According to the existence theorem of Nash equilibrium, that is, every finite strategic game has at least one Nash equilibrium [48].

Suppose the discounted value of the effect of inserting address information on the SPV node is η, and the

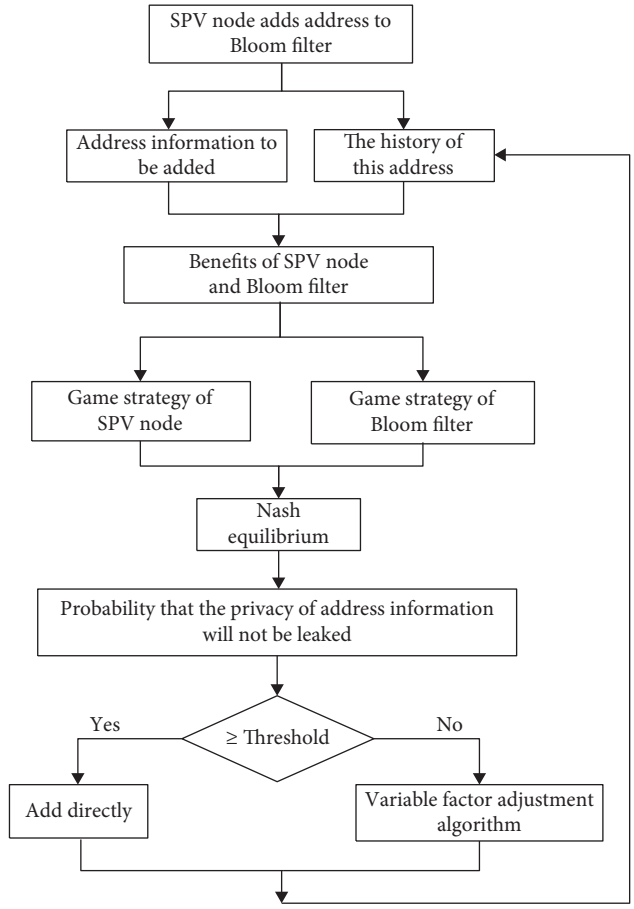

Figure 3: Privacy protection method of SPV node.

discounted value of the effect on Bloom filter is μ. Define the corresponding benefits when the two sides of the game adopt different strategies.

Definition 1. S_earning_State1 represents the earning of the SPV node when the Bloom filter is not hijacked and the SPV node adds the address directly.

This earning can be seen as the benefit that the SPV node obtains by allowing the addition of addresses to the Bloom filter in a safe state. Assuming that every time an address is inserted into the Bloom filter, the SPV node's earning is s_earning_State1; then, the nth time the address is inserted:

$$\begin{aligned} S\_\text{earning\_State1} &= \left(s\_\text{earning\_State1} + \cdots + s\_\text{earning\_State1} \times \eta^{n-1}\right) \\ &= s\_\text{earning\_State1} \times \frac{1-\eta^n}{1-\eta}. \end{aligned} \tag{1}$$

Definition 2. B_earning_State1 represents the earning of the Bloom filter when the Bloom filter is not hijacked and the SPV node adds the address directly.

This earning can be seen as the benefit of the Bloom filter that obtains the address added by the SPV node in a safe state and realizes its service function. Assuming that every time an

address is inserted into the Bloom filter, the Bloom filter's earning is b_earning_State1, then the nth time the address is inserted:

$$B\_\text{earning\_State1} = \left(b\_\text{earning\_State1} + \cdots + b\_\text{earning\_State1} \times \mu^{n-1}\right) = b\_\text{earning\_State1} \times \frac{1-\mu^n}{1-\mu}. \tag{2}$$

Definition 3. S_loss_State2 represents the loss of the SPV node when the Bloom filter is hijacked, and the SPV node adds the address directly.

This loss can be seen as the loss of privacy information leakage due to the SPV node allowing to add addresses to the Bloom filter that in an insecure state. Assuming that each time an address is inserted into the Bloom filter, the SPV node's loss is s_loss_State2, then the nth time the address is inserted:

$$S\_\text{loss\_State2} = \left(s\_\text{loss\_State2} + \cdots + s\_\text{loss\_State2} \times \eta^{n-1}\right) = s\_\text{loss\_State2} \times \frac{1-\eta^n}{1-\eta}. \tag{3}$$

Definition 4. B_earning_State2 represents the earning of the Bloom filter when the Bloom filter is hijacked and the SPV node adds the address directly.

This earning can be regarded as the benefit that the Bloom filter obtains address information exceeding the level of privacy protection that the SPV node can tolerate. Assuming that every time an address is inserted into the Bloom filter, the Bloom filter's earning is b_earning_State2, then the nth time the address is inserted:

$$B\_\text{earning\_State2} = \left(b\_\text{earning\_State2} + \cdots + b\_\text{earning\_State2} \times \mu^{n-1}\right) = b\_\text{earning\_State2} \times \frac{1-\mu^n}{1-\mu}. \tag{4}$$

Definition 5. S_earning_State3 represents the earning of the SPV node when the Bloom filter is hijacked and the SPV node does not add the address directly.

This earning can be seen as the benefit that the SPV node has not added an address to the Bloom filter which in an insecure state and protected its own private information successfully. Assuming that each time an address is inserted into the Bloom filter, the SPV's earning is s_earning_State3, then the nth time the address is inserted:

$$S\_\text{earning\_State3} = s\_\text{earning\_State3} + \cdots + s\_\text{earning\_State3} \times \eta^{n-1} = s\_\text{earning\_State3} \times \frac{1-\eta^n}{1-\eta}. \tag{5}$$

Definition 6. B_loss_State3 represents the loss of the Bloom filter when the Bloom filter is hijacked and the SPV node does not add the address directly.

The loss can be regarded as the loss caused by the Bloom filter not realizing its own service function. Assuming that every time an address is inserted into the Bloom filter, the Bloom filter's loss is b_loss_State3; then, the nth time the address is inserted:

$$B\_\text{loss\_State3} = b\_\text{loss\_State3} + \cdots + b\_\text{loss\_State3} \times \mu^{n-1} = b\_\text{loss\_State3} \times \frac{1-\mu^n}{1-\mu}. \tag{6}$$

Definition 7. S_loss_State4 represents the loss of the SPV node when the Bloom filter is not hijacked and the SPV node does not add the address directly.

The loss can be regarded as the loss caused by the SPV node, which does not add an address to the Bloom filter in a safe state so that the required service cannot be achieved. Assuming that every time an address is inserted into the Bloom filter, the SPV node's loss is s_loss_State4; then, the nth time the address is inserted:

$$S\_\text{loss\_State4} = s\_\text{loss\_State4} + \cdots + s\_\text{loss\_State4} \times \eta^{n-1} = s\_\text{loss\_State4} \times \frac{1-\eta^n}{1-\eta}. \tag{7}$$

Definition 8. B_earning_State4 represents the earning of the Bloom filter when the Bloom filter is not hijacked and the SPV node does not add the address directly.

The earning can be regarded as the benefit of the Bloom filter failing to realize its own service function in a safe state, and the value is 0. The earning of the SPV node in Definition 1 is also the loss of the SPV node in Definition 7, namely S_loss_State4 = S_earning_State1. Based on the above analysis of earnings and losses, the game matrix of both parties is shown in Table 1.

Theorem 2. *There is no pure strategy Nash equilibrium in the game matrix of SPV node and Bloom filter; the mixed strategy Nash equilibrium needs to be calculated.*

Proof. From the perspective of the Bloom filter,

(1) When the SPV node selects "add the address directly," the Bloom filter can obtain greater benefits in the "hijacked" state, that is, B_earning_State2> B_earning_State1

(2) When the SPV node selects "not add the address directly," the Bloom filter can obtain greater benefits in the "not hijacked" state, that is, 0>-B_loss_State3

From the perspective of SPV nodes:

(3) When the Bloom filter is in the "not hijacked" state, the SPV node chooses "add the address directly" to be greater than the benefit obtained by "not add the address directly," that is, S_earning_State1>-S_earning_State1

(4) When the Bloom filter is in the "hijacked" state, the SPV node chooses "not add the address directly" to be greater than the benefit obtained by "add the address directly," that is, S_earning_State3>-S_loss_State2

Complete

Assuming that the probability of SPV node "add the address directly" is x, then the probability of "not add the address directly" is $1-x$, and the mixed strategy probability matrix of SPV node is $P_s[x, 1-x]$; assuming that the probability of the Bloom filter "not hijacked" is y, then the probability of "hijacked" is $1-y$, and the mixed strategy probability matrix of the Bloom filter is $P_b[y, 1-y]$. The earning matrix of the SPV node and the Bloom filter are represented by M_S and $M_{B,}$ respectively. The earning E_S of the SPV node is shown in the following equation:

$$S\_\text{loss\_State4} = s\_\text{loss\_State4} + \cdots + s\_\text{loss\_State4} \times \eta^{n-1} = s\_\text{loss\_State4} \times \frac{1-\eta^n}{1-\eta}. \tag{8}$$

E_S takes the derivative of x to get the following:

$$\frac{dE_S}{dx} = 2 \times y \times S\_\text{earning\_State1} - (S\_\text{loss\_State2} + S\_\text{earning\_State3}) \times (1-y). \tag{9}$$

Let the above formula be equal to 0, combined with the related definitions above, the probability y that the Bloom filter is not hijacked is obtained as follows:

$$y = \left(\frac{s\_\text{loss\_State2} + s\_\text{earning\_State3}}{2 \times s\_\text{earning\_State1} + s\_\text{loss\_State2} + s\_\text{earning\_State3}} \right). \tag{10}$$

Table 1: Game matrix of privacy protection mechanism.

SPV node	Bloom filter	
	Not hijacked	Hijacked
Add the address directly	[S_earning_State1, B_earning_State1]	[−S_loss_State2, B_earning_State2]
Not add the address directly	[−S_loss_State4, B_earning_State4]	[S_earning_State3, −B_loss_State3]

4.2. Bloom Filter Privacy Protection Algorithm Based on Variable Factor Disturbance. In order to improve the privacy protection performance when querying address information through Bloom filters, this section proposes a Bloom filter privacy protection algorithm based on variable factor disturbance.

Bloom filter is a space-saving probabilistic data structure, mainly used to test the membership of elements [49, 50]. In the blockchain, it represents a collection of address elements $A = \{@_1, @_2, \ldots, @_m\}$, a total of m address elements are mapped to a bit vector V of length n through k mutually independent hash functions, and the Bloom filter B expresses the overall address information through the vector V.

Let p represent the probability that any bit in the vector V is 0, and the probability of being 1 is $1 - p$. In this paper, the perturbation factor α is added to perturb the probability to adjust the privacy protection performance provided by the Bloom filter. Among them, α is the probability value which represents the probability that any bit that has been set to 0 is still set to 0 after random disturbance. Assuming that the value of the hash function obeys a uniform distribution, when all the inserted address elements are mapped, the probability that any bit is 0:

$$\begin{aligned} p' &= \alpha\left(1 - \frac{1}{n}\right)^{km} \\ &\approx \alpha e^{-\left(\frac{km}{n}\right)} \\ &= \alpha p. \end{aligned} \tag{11}$$

When an element that does not belong to the inserted address set A is misjudged as true, the corresponding position of the element in the vector V is set to 1. That is, the false positive rate is

$$\begin{aligned} P_f &= (1 - p')^k \\ &\approx (1 - \alpha p)^k \\ &= \left(1 - \alpha e^{-(km/n)}\right)^k \\ &= \exp\left[k \ln\left(1 - e^{\ln \alpha - (km/n)}\right)\right]. \end{aligned} \tag{12}$$

Given the target false positive rate P_t of the Bloom filter, when the number of bits n of the Bloom filter and the number of hash functions k is constant, the number of inserted address elements of the Bloom filter m can be calculated by the following equation:

$$m = -\left(\frac{n}{k}\right)\ln\left(\frac{1 - P_t^{1/k}}{\alpha}\right). \tag{13}$$

It can be seen from (13) that the more the hash function k, the more the number of bits that can be mapped in the address element, the more element information can be expressed, and the false positive rate may decrease. However, as the number of bits set to 1 increases, the false positive rate may also increase. Therefore, in (12), let $g(k) = k \ln(1 - e^{\ln \alpha - (km/n)})$, and function $g(k)$ and k reach the minimum at the same time and take the derivative of k for $g(k)$ to obtain the following:

$$\frac{dg(k)}{dk} = \ln\left(1 - e^{\ln \alpha - (km/n)}\right) + \frac{km}{n}\left(\frac{e^{\ln \alpha - (km/n)}}{1 - e^{\ln \alpha - (km/n)}}\right). \tag{14}$$

Let $dg(k)/dk = 0$, solve for $k_{\min}$ to get $e^{\ln \alpha - (km/n)} = 1/2$. When $k = k_{\min}$, $g(k)$ reaches the minimum value:

$$k_{\min} = \frac{n}{m}\ln(2\alpha). \tag{15}$$

The value range of the added privacy protection variable factor α can be further determined $\alpha \in [1/2e^{m/n}, 1]$. When the value of α is 1, it means that the probability distribution of the hash function is not disturbed. At this time, the privacy protection performance is the effect provided by the initialization of the Bloom filter.

According to equation (12) of false positive rate, the calculation formula of the number of bits n of the Bloom filter can be obtained:

$$n = -\frac{km}{\ln\left(\left(1 - P_t^{1/k}\right)/\alpha\right)}. \tag{16}$$

The function $n(k)$ and k can reach the minimum at the same time. To obtain the minimum number of bits n of the Bloom filter, take the derivative of k to obtain the following:

$$\frac{dn(k)}{dk} = \frac{m\ln\left(\left(1 - P_t^{1/k}\right)/\alpha\right) - m\left(1/\left(1 - P_t^{1/k}\right)\right)P_t^{1/k}\ln\left(P_t^{1/k}\right)}{\left[\ln\left(\left(1 - P_t^{1/k}\right)/\alpha\right)\right]^2}, \tag{17}$$

Let $dn(k)/dk = 0$, get $(1 - P_t^{1/k})\ln(1 - P_t^{1/k}) = P_t^{1/k}\ln P_t^{1/k} + (1 - P_t^{1/k})\ln \alpha$, let $P_t^{1/k} = x$; by derivation on the left and right sides of the above formula separately, get $\ln x/\alpha + \ln(1 - x) = -2$. Derivative to get $x = 1/2$, that is, $P_t^{1/k} = 1/2$; combining (15), get the number of bits of Bloom filter:

$$n = -\frac{m \ln P_t}{\ln 2 \ln(2\alpha)}. \tag{18}$$

Among all the positive values that the attacker correctly guessed, the probability of the j true positive values matching the Bloom filter is $P_h(j)$, as shown in the following equation:

$$P_h(j) = \frac{N}{N+F} \cdot \frac{N-1}{(N+F)-1} \cdots \frac{N-j+1}{(N+F)-j+1} = \prod_{i=0}^{j-1} \frac{N-i}{(N+F)-i} = \prod_{i=0}^{j-1} \frac{N-i}{N+|R-N|P_f(m)-i}. \quad (19)$$

N represents the number of addresses inserted into the Bloom filter, R represents the number of all existing addresses in the network, and F represents the number of all false positive addresses, but the attacker does not know. Therefore, the attacker can correctly guess the probability of all addresses inserted into Bloom filter B as follows:

$$P_h(N) = \frac{N!F!}{(N+F)!} = \prod_{i=0}^{N-1} \frac{N-i}{N+F-i} = \prod_{i=0}^{N-1} \frac{N-i}{N+|R-N|P_f(m)-i}. \quad (20)$$

$P_h(j)$ is used to represent the privacy degree provided by the Bloom filter. The higher the privacy degree, the worse the privacy protection performance it can provide, and the lower the value of the privacy degree, the better the privacy performance it can provide.

When an attacker obtains more than two Bloom filters belonging to the same SPV node, the common elements of different filters can be obtained by calculating the intersection between each pair of Bloom filters. Given b Bloom filters belonging to the same SPV node, the attacker can estimate the number of elements inserted in each filter by formula (13). Suppose that Bloom filters $B_1, \ldots, B_b$ are sorted in ascending order by the number of insertable elements. According to (21), the more Bloom filters can be obtained by an attacker, the smaller error in the classification of the real address, and the larger $P_h(.)$ value, the worse the privacy protection effect provided by the Bloom filter.

$$P_h(j) \approx \prod_{i=0}^{j-1} \frac{N-i}{N-i+|R|\prod_{\forall j} P_f\left(m_j\right)}. \quad (21)$$

4.3. Working Flow of the Method

4.3.1. Insertion of Addresses

(a) Calculate the game benefit of SPV node and Bloom filter separately with the address information to be inserted

(b) Determine the probability y that the address information will not be leaked through Nash equilibrium

(c) Select the threshold β according to the privacy tolerance of the SPV node

(d) Compare y with the threshold β in Table 2 to determine whether this piece of address information can be directly added to the Bloom filter. If it can be added directly, ignore step (c)

(e) Add this process as a historical record to the historical information

4.3.2. The Generation of Address Elements

(a) Calculate the k hash addresses of address element @1, that is, $h_1(@_1), h_2(@_1) \ldots h_k(@_1)$

(b) Set the k positions corresponding to the Bloom filter, that is, $B[h_1(@_1)] = B[h_2(@_1)] \cdots B[h_k(@_1)]$, as shown in Figure 4

4.3.3. The Perturbation of Variable Factors. According to different scenarios of the Bloom filter, the variable factor α is selected. As shown in Figure 5, the bit setting to 0 is still set to 0 randomly. Among them, scenario 1 is that the number of inserted addresses matches the capacity of the Bloom filter, scenario 2 is that part of the address capacity is reserved when the Bloom filter is initialized, scenario 3 is the expansion when the number of inserted addresses exceeds the maximum capacity of the Bloom filter, and scenario 4 is multiple Bloom filter, the detailed introduction of each scene will be carried out in Section 5.2;

4.3.4. Query of Addresses

(a) Calculate the k hash addresses corresponding to $@x$, that is $h_1(@_x), h_2(@_x) \ldots h_k(@_x)$.

(b) Check the k corresponding positions of the Bloom filter vector; that is, $B[h_1(@_x)] = B[h_2(@_x)] \cdots B[h_k(@_x)]$, whether they are all 1, as shown in Figure 6. As long as one of the bits is 0, $@x$ is not in the query set; if all corresponding bits are all 1, then $@x$ may be in the query set, but it may not be the real query result, and a false positive may occur at this time.

Table 2: The relationship between privacy leakage tolerance and threshold.

Level	Privacy leakage tolerance	Threshold β
0	Extremely high	[0, 0.2]
1	High	(0.2, 0.4]
2	Medium	(0.4, 0.6]
3	Low	(0.6, 0.8]
4	Extremely low	(0.8, 1]

5. Experimental Verification

5.1. Experimental Settings. Bitcoin is the most successful blockchain application so far. This paper selects transaction data in Bitcoin as experimental data. Using the parser in [51], we analyzed the genesis block from 2009 to the end of December 2019 and collected nearly 29 million different addresses [52]. In the experiment, a Bitcoin wallet is constructed based on the standard Bitcoinj library [53]. In the current implementation of the SPV node, when the Bitcoin address is inserted into the Bloom filter, the address and its public key hash will be added at the same time, so $m = 2N$. Bitcoin developers believe that a false positive rate of 0.1% can provide a better privacy effect [54], so the target false positive rate in this paper is 0.1% as a reference. The blue line represents the experimental data with the variable factor adjustment algorithm ($\alpha = 0.75$) and the red line represents the experimental data without the variable factor adjustment algorithm.

5.2. Privacy Protection Performance of Bloom Filter and the Comparison

5.2.1. Privacy Protection Performance of Single Bloom Filter

Scenario 1. The number of inserted addresses matches the Bloom filter capacity.

Set the total capacity E from 2 to 500 and the corresponding number of inserted addresses from 1 to 250. The probability of guessing all the inserted addresses is shown in Figure 7, and the probability of guessing anyone inserted address is shown in Figure 8. It can be seen that the probability of privacy leakage is close to 0. The difference between the algorithm with and without variable factor adjustment is not significant, and both can obtain the best privacy protection effect.

Scenario 2. Reserve part of the address capacity when the Bloom filter is initialized.

Addresses are inserted sequentially when the Bloom filter is initialized, and the privacy degree P_h is recorded. Set the total capacity E as a fixed value, and take 100 and 200, respectively, as shown in Figures 9 and 10. It can be seen from the figure that with variable factor adjustment, the value of P_h approaches 0, and the privacy protection effect is far better than without variable factor adjustment.

Scenario 3. Expand when the number of inserted addresses exceeds the maximum capacity of the Bloom filter.

Take the initial capacity $E = 100$. When the number of inserted addresses N is greater than 50, the Bloom filter needs to be expanded. The expanded capacity is the initial capacity, as shown in Figures 11 and 12. It can be seen from the figure that with the variable factor adjustment, the value of P_h approaches 0, and the privacy protection effect is far better than without the variable factor adjustment.

5.2.2. Privacy Protection Performance of Multiple Bloom Filters. The privacy protection values that can be provided by adding filters are shown in Table 3. Among them, the target false positive rate is, respectively, 0.05% and 0.1%. The numbers of addresses that can be inserted into the 5 Bloom filters are 50, 100, 200, 500, and 1000, respectively. The maximum capacity of the Bloom filter is set to exactly match the number of inserted addresses. The value of b, respectively, represents the number of Bloom filters acquired by the attacker. It can be seen from Table 3 that the more the number of filters obtained by the attacker, the more serious the privacy degree leakage.

The $P_h(.)$ values obtained by the methods of capacity expansion, addition, and disturbance are shown in Table 4. Among them, the target false positive rate is 0.1%, and the number of inserted addresses is 50, 100, 150, 200, and 250, respectively. It can be seen from the table that there is little difference in the privacy degree between the expansion and the addition of filters. The expansion method has a slight advantage. The degree of privacy protection under the two methods reveals almost all private information with probability 1 when the number of inserted addresses is small. The $P_h(.)$ value generated by the disturbance method is much lower than the first two methods, which can produce a better privacy protection effect. Among them, the best effect is when the disturbance factor is closing to 1/2.

5.3. The Effectiveness of the Mixed Strategy Nash Equilibrium Privacy Protection Mechanism and the Comparison. This section uses scenario 2 as an example for testing. The mixed strategy Nash equilibrium privacy protection mechanism is compared and analyzed with the situation without it. In the experiment, the maximum capacity of the Bloom filter is 200. Each time 50 addresses are randomly selected to insert the Bloom filter from 100 addresses. Repeat 50 times to observe the probability that at least one of the inserted addresses can be guessed by the attacker. Repeat this experiment 100 times and take the average of the 100 experimental results as the final experimental result, as shown in Figures 13–16, where the gray line indicates that no privacy protection mechanism is added, the dark blue line indicates the Gervaisy et al.'s [18] solution, and the remaining lines indicate that privacy protection mechanism is added. Level 0-4 respectively corresponds to the privacy leakage tolerance in Table 2.

It can be seen from Figure 13 that as the number of times of address insertion increases, the probability of privacy leakage increases with or without adding a privacy protection mechanism. This is due to the superimposing effect

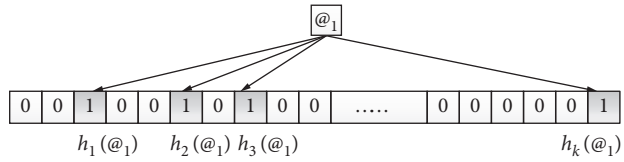

Figure 4: Insertion of address elements.

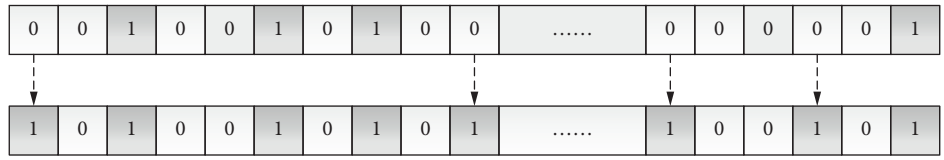

Figure 5: Perturbation of variable factors.

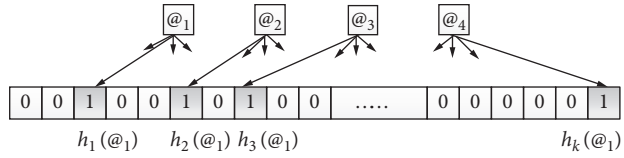

Figure 6: Query of address elements.

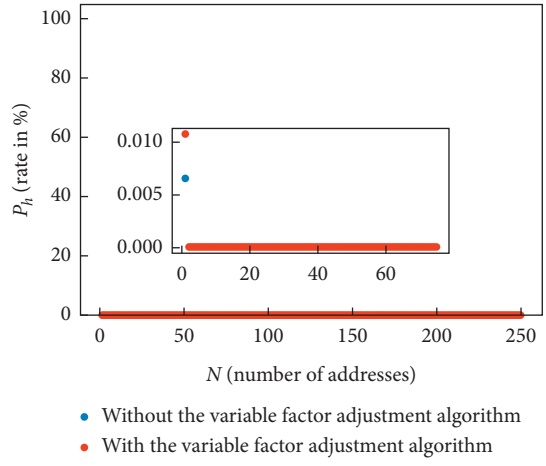

Figure 7: P_h (N) when the number of inserted addresses matches the capacity of Bloom's filter.

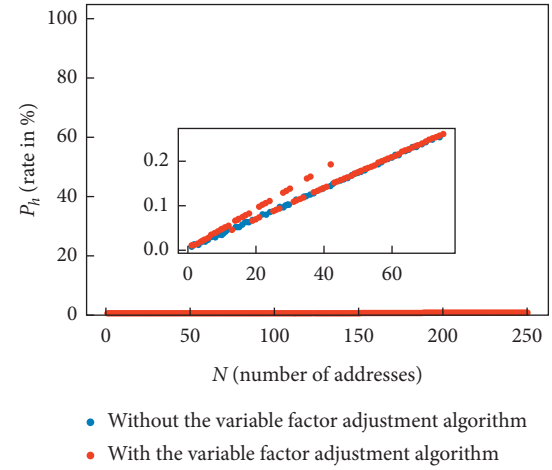

Figure 8: $P_h(1)$ when the number of inserted addresses matches the capacity of Bloom's filter.

of private information as the number of insertions increases. With the increase in the number of insertions, the privacy information leakage of SPV nodes also increases. Even if some privacy information itself does not exceed the tolerance of SPV nodes for privacy leakage, the combined privacy information disclosed may exceed the tolerance of the node. In comparison, the privacy leakage probability of adding a privacy protection mechanism is always lower than that of not adding it. It can be seen from the figure that when the set privacy tolerance is lower, the probability of privacy leakage also decreases significantly. When the privacy tolerance is the lowest, the privacy leakage probability is close to Gervaisy et al.'s solution, and both have good privacy protection performance.

It can be seen from Figure 14 that as the number of insertions increases, the effectiveness of the adding and not adding privacy protection mechanism are decreasing. That is, as the number of insertions increases, the attacker may obtain the privacy information of the SPV node has also increased. It can be seen from the figure that after adding the privacy protection mechanism, the downward trend has slowed down.

Figure 15 shows the comparison of the time required to return query results. It can be seen from the figure that as the degree of privacy tolerance decreases, the return time of the query result becomes larger. This is because when the SPV node has a low tolerance for privacy leakage, the probability that the inserted address information which will not be

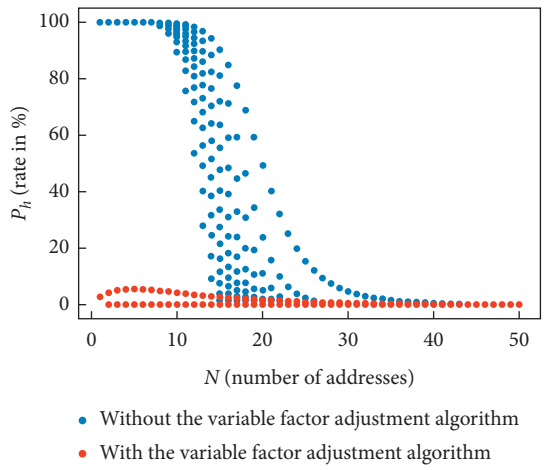

Figure 9: $P_h(.)$ for $E = 100$ and N increased to 50.

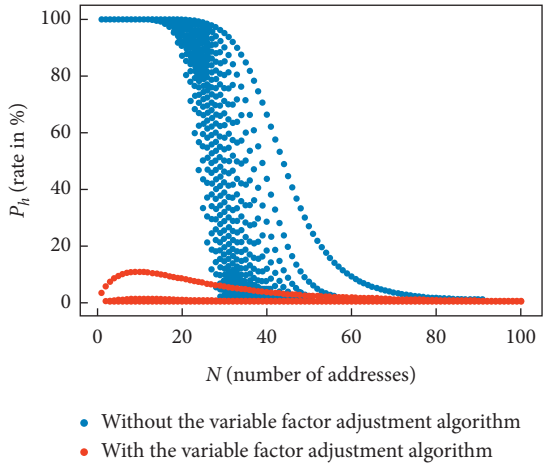

Figure 10: $P_h(.)$ for $E = 200$ and N increased to 100.

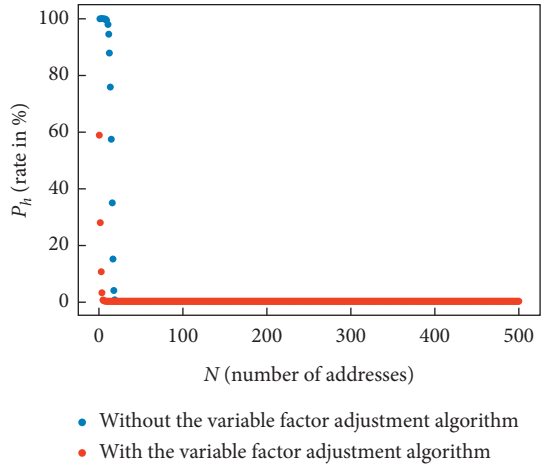

Figure 11: P_h (N) when the expansion is fixed.

leaked is easily higher than the privacy leakage tolerance of the node, and the Bloom filter needs to be adjusted by a variable factor.

Figure 16 shows the comparison of average bandwidth cost per query when the number of queries reaches 10, 20, 30, 40, and 50, respectively. It can be seen from the figures that the

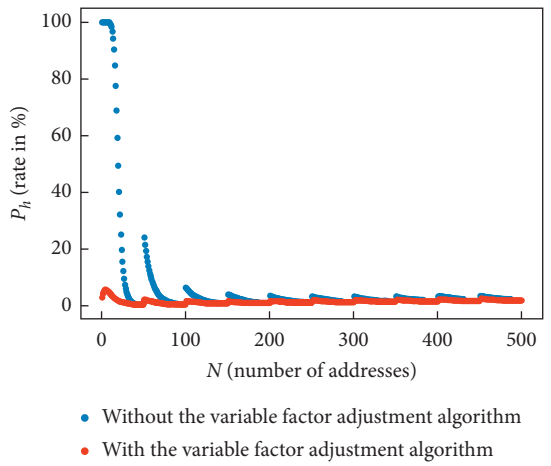

Figure 12: P_h (1) when the expansion is fixed.

Table 3: P_h(.) with additional filter disturbance factor $\alpha = 0.51$.

Target false positive rate	$P_t = 0.05\%$		$P_t = 0.1\%$	
Privacy degree	$P_h(1)$	P_h (N)	P_h (1)	P_h (N)
$b = 1$	0.0035	0	0.0017	0
$b = 2$	0.8754	0	0.6371	0
$b = 3$	0.9999	0.9843	0.9994	0.8803
$b = 4$	1	1	1	0.9999
$b = 5$	1	1	1	1

Table 4: P_h (.) values using expansion, addition, and perturbation, respectively.

Number of inserted addresses	Privacy degree	Expansion	Addition	$\alpha = 0.51$	$\alpha = 0.6$	$\alpha = 1$
$N = 50$	$P_h(1)$	0.0017	0.0017	0.0018	0.0018	0.0017
	P_h (25)	0	0	0	0	0
	P_h (50)	0	0	0	0	0
$N = 100$	P_h (1)	0.2673	0.6371	0.0018	0.0048	0.2676
	P_h (25)	0	0	0	0	0
	P_h (50)	0	0	0	0	0
$N = 150$	P_h (1)	0.9238	0.9994	0.0018	0.007	0.9239
	P_h (25)	0.0697	0.9809	0	0	0.0699
	P_h (50)	0	0.8814	0	0	0
$N = 200$	P_h (1)	0.9938	1	0.0018	0.0085	0.9938
	P_h (25)	0.8094	1	0	0	0.8096
	P_h (50)	0.2645	0.9999	0	0	0.2648
$N = 250$	P_h (1)	0.9992	1	0.0018	0.0096	0.9992
	P_h (25)	0.9734	1	0	0	0.9735
	P_h (50)	0.8385	1	0	0	0.8386

mechanism proposed in this paper is better than the solution of Gervais et al. in terms of average bandwidth cost. This is because the false alarm rate P_f of the Bloom filter constructed by the solution of Gervais et al. is close to the target false alarm rate. While the mechanism of this paper appropriately reduces the false alarm rate while ensuring privacy performance, the number of addresses feedback by the full node is reduced, thereby reducing the bandwidth cost.

In addition to the storage space required to initialize the Bloom filter and the pregeneration of N Bitcoin addresses, the algorithm proposed in this paper does not generate additional overhead on the SPV node. When the Bloom filter is initialized, the SPV node needs to be stored locally: the number of addresses embedded in the file manager (4 bytes); the target false positive rate P_t (8 bytes); the value of α selected by the user (8 bytes); Bloom filter flag (2 bytes) and insertion address. The SPV node can add a pointer to the ECKey class of Bitcoinj to link each Bitcoin address to the corresponding Bloom filter. The size of the pointer is about 2 bytes/address. Therefore, the storage required to initialize each Bloom filter is approximately $2N + 22$ bytes. The method proposed in this paper requires only a small amount

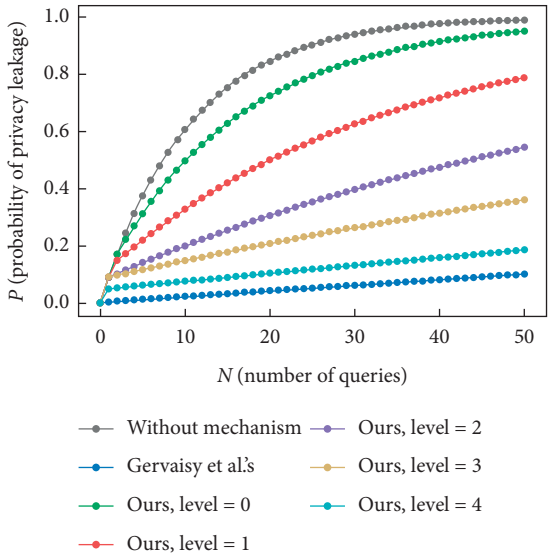

Figure 13: Comparison of privacy leakage probability.

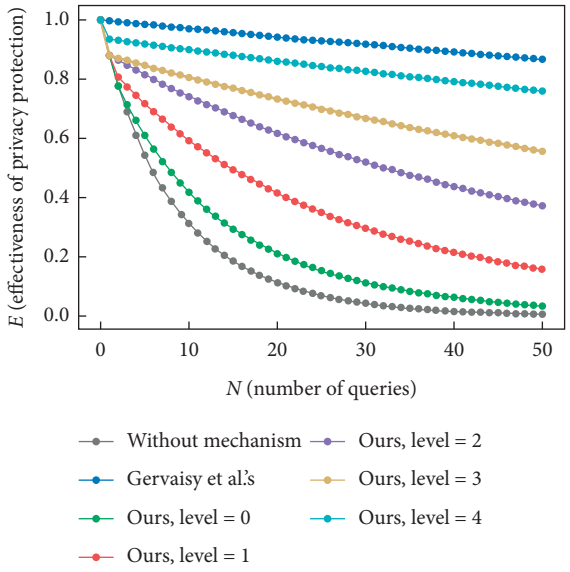

Figure 14: Comparison of the effectiveness of privacy protection mechanisms.

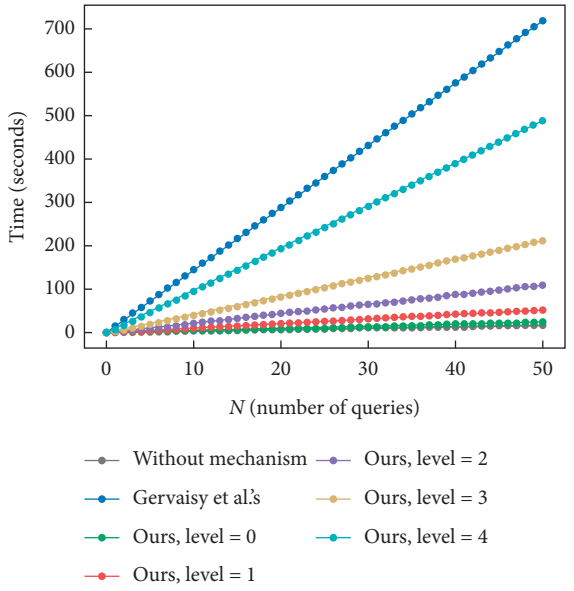

Figure 15: Comparison of the time required to return query results.

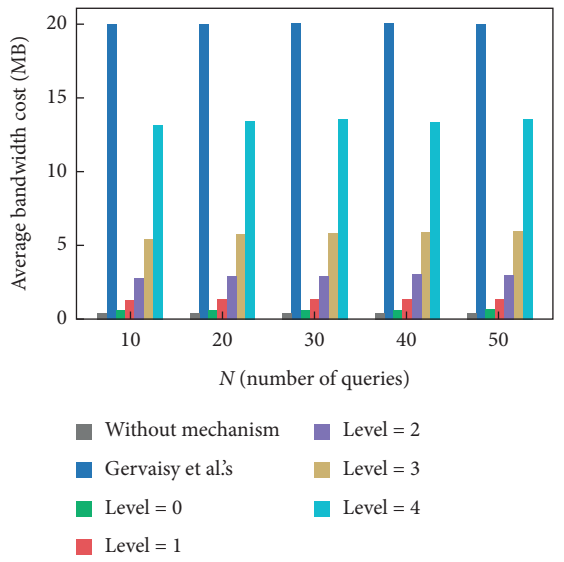

Figure 16: Comparison of the average bandwidth cost per query.

of modification to the existing SPV node, and such overhead can be tolerated in the implementation of the existing SPV node.

6. Conclusions

In blockchain applications, lightweight nodes with limited resources mainly use Bloom filters to obtain relevant information. This kind of request for specific data unintentionally reveals the node's private information. In response to such problems, this paper proposes a new privacy protection method for lightweight nodes in blockchain. Aiming at the superimposition effect of multiple information queries that may exceed the node's tolerance for privacy leakage, a privacy protection mechanism based on mixed strategy Nash equilibrium is proposed. Based on this, in the case that the probability of information query and privacy not being leaked is less than the node privacy protection, a Bloom filter privacy protection algorithm based on variable factor is proposed. Experimental results show that the method proposed in this paper is feasible, effective, reliable, and superior. It can be deployed and applied on existing lightweight nodes.

With the development of IoT and sensor technology, lightweight nodes have become more and more. Resource consumption is the most important issue in the privacy protection of lightweight nodes. The method in this paper will be further optimized in the actual deployment and application in the next step to save node resource consumption. The current privacy protection threshold is set by human experience. The future work will quantify the user's privacy requirements and the node's computing and storage resources to generate the corresponding privacy protection threshold.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Science and Technology Planning Project of Henan Province (project no. 192102210283), Key Scientific Research Projects of Colleges and Universities in Henan Province (project no. 20A520040), Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (project no. CAAC-ITRB-201707), and National Natural Science Foundation of China (project no. U1904119).

References

[1] S. Melanie, "Preface," in *Blockchain: Blueprint for a New Economy,* O'Reilly, Sebastopol, CA, USA, 2015.

[2] K. Panetta, "Top trends in the Gartner hype cycle for emerging technologies," 2017, https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/.

[3] Z. Zheng, S. Xie, H. Dai, and H. Wang, "An overview of blockchain technology: architecture, consensus, and future trends," in *Proceedings of the IEEE International Congress on Big Data*, pp. 557–564, Boston, MA, USA, December 2017.

[4] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: beyond bitcoin," *Applied Innovation*, vol. 2, pp. 6–10, 2016.

[5] P. Treleaven, R. Gendal Brown, and D. Yang, "Blockchain technology in finance," *Computer*, vol. 50, no. 9, pp. 14–17, 2017.

[6] R. Dennis and G. Owen, "Rep on the block: a next generation reputation system based on the blockchain," in *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions*, pp. 131–138, London, UK, December 2015.

[7] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[8] X. Wang, X. Zha, W. Ni et al., "Survey on blockchain for internet of things," *Computer Communications*, vol. 136, pp. 10–29, 2019.

[9] S. Homayoun, A. Dehghantanha, R. M. Parizi, and K.-K.-R. Choo, "A blockchain-based framework for detecting malicious mobile applications in app stores," in *Proceedings of the IEEE Canadian Conference of Electrical and Computer Engineering*, pp. 1–4, Edmonton, Canada, May 2019.

[10] S. Singh and N. Singh, "Blockchain: future of financial and cyber security," in *Proceedings of the 2nd International Conference on Contemporary Computing and Informatics*, pp. 463–467, Greater, Noida, India, December 2016.

[11] S. Han, Z. Xu, and L. Chen, "Jupiter: a blockchain platform for mobile devices," in *Proceedings of the IEEE 34th International Conference on Data Engineering*, pp. 1649–1652, Paris, France, April 2018.

[12] Y. Ren, Y. Liu, S. Ji, A. K. Sangaiah, and J. Wang, "Incentive mechanism of data storage based on blockchain for wireless sensor networks," *Mobile Information Systems*, vol. 2018, Article ID 6874158, 10 pages, 2018.

[13] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pdf.

[14] M. Hearn and M. Corallo, "Connection bloom filtering," 2012, https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki.

[15] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2017.

[16] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, O'Reilly, Sebastopol, CA, USA, 2nd edition, 2017.

[17] P. Franco, *Understanding Bitcoin: Cryptography, Engineering and Economics*, Wiley, New York, NY, USA, 2014.

[18] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, "On the privacy provisions of bloom filters in lightweight bitcoin clients," in *Proceedings of the ACM Annual Computer Security Applications Conference*, pp. 326–335, New Orleans, LA, USA, November 2014.

[19] X. Li, Y. Niu, L. Wei, C. Zhang, and N. Yu, "Overview on privacy protection in bitcoin," *Journal of Cryptologic Research*, vol. 6, no. 2, pp. 133–149, 2019.

[20] J. Bernal Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. Torres Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[21] L. Zhu, F. Gao, M. Shen et al., "Survey on privacy preserving techniques for blockchain technology," *Journal of Computer Research and Development*, vol. 54, no. 10, pp. 2170–2186, 2017.

[22] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[23] Z. Guan, G. Si, X. Zhang et al., "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[24] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[25] F. Gao, L. Zhu, M. Shen, K. Sharif, Z. Wan, and K. Ren, "A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks," *IEEE Network*, vol. 32, no. 6, pp. 184–192, 2018.

[26] T. T. Thwin and S. Vasupongayya, "Blockchain-based access control model to preserve privacy for personal health record systems," *Security and Communication Networks*, vol. 2019, Article ID 8315614, 15 pages, 2019.

[27] J. K. Mullin, "A second look at bloom filters," *Communications of the ACM*, vol. 26, no. 8, pp. 570-571, 1983.

[28] K. Christensen, A. Roginsky, and M. Jimeno, "A new analysis of the false positive rate of a bloom filter," *Information Processing Letters*, vol. 110, no. 21, pp. 944–949, 2010.

[29] G. Bianchi, L. Bracciale, and P. Loreti, ""Better than nothing" privacy with bloom filters: to what extent?" in *Proceedings of the 2012 International Conference on Privacy in Statistical Databases*, pp. 348–363, Palermo, Italy, September 2012.

[30] L.-Y. Yeh, P. J. Lu, S.-H. Huang, and J.-L. Huang, "SOChain: a privacy-preserving DDoS data exchange service over SOC consortium blockchain," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1487–1500, 2020.

[31] L.-H. Zhu, B.-K. Zheng, M. Shen, F. Gao, H.-Y. Li, and K.-X. Shi, "Data security and privacy in bitcoin system: a survey," *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 843–862, 2020.

[32] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, K. Wehrle, and " CoinParty, "Secure multiparty mixing of bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pp. 75–86, San Antonio, TX, USA, March 2015.

[33] J. Poon and T. Dryja, "The Bitcoin lightning network: scalable off-chain instant payments," 2019, http://lightning.network/lightning-network-paper.pdf.

[34] Z. Bao, W. Shi, S. Kumari, Z.-y. Kong, and C.-M. Chen, "Lockmix: a secure and privacy-preserving mix service for Bitcoin anonymity," *International Journal of Information Security*, vol. 19, no. 3, pp. 311–321, 2020.

[35] T. Ruffing and P. Moreno-Sanchez, "ValueShuffle: mixing confidential transactions for comprehensive transaction privacy in bitcoin," in *Proceedings of the Financial Cryptography and Data Security*, pp. 133–154, Sileme, Malta, April 2017.

[36] Y. Liu, X. Liu, C. Tang, J. Wang, and L. Zhang, "Unlinkable coin mixing scheme for transaction privacy enhancement of bitcoin," *IEEE Access*, vol. 6, pp. 23261–23270, 2018.

[37] R. Nojima and Y. Kadobayashi, "Cryptographically secure bloom-filters," *Transactions on Data Privacy*, vol. 2, no. 2, pp. 131–139, 2009.

[38] K. Kanemura, K. Toyoda, and T. Ohtsuki, "Design of privacy-preserving mobile bitcoin client based on γ-deniability enabled bloom filter," in *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, pp. 1–6, Montreal, Canada, October 2017.

[39] K. Qin, H. Hadass, A. Gervais, and J. Reardon, "Applying private information retrieval to lightweight bitcoin clients," in *Proceedings of the Crypto Valley Conference on Blockchain Technology*, pp. 60–72, Rotkreuz, Switzerland, June 2019.

[40] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: challenges and directions," *IEEE Security & Privacy*, vol. 16, no. 4, pp. 38–45, 2018.

[41] W. Jiang, H. Li, G. Xu et al., "PTAS: privacy-preserving thin-client authentication scheme in blockchain-based PKI," *Future Generation Computer Systems*, vol. 96, pp. 185–195, Jul. 2019.

[42] X. Li, Z. Yang, L. Wei, and C. Zhang, "Protecting access privacy for Bitcoin lightweight client using trusted hardware," in *Proceedings of the IEEE/CIC International Conference on Communications*, pp. 706–711, Changchun, China, August 2019.

[43] Y. Niu, C. Zhang, L. Wei, Y. Xie, X. Zhang, and Y. Fang, "An efficient query scheme for privacy-preserving lightweight bitcoin client with Intel SGX," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Waikolola, HI, USA, December 2019.

[44] D. V. Le, L. T. Hurtado, A. Ahmad, M. Minaei, B. Lee, and A. Kate, "A tale of two trees: one writes, and other reads," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 2, pp. 519–536, 2020.

[45] L. Zhou, C. Ge, and C. Su, "A privacy preserving two-factor authentication protocol for the bitcoin SPV nodes," *Science China Information Sciences*, vol. 63, no. 3, Article ID 130103, 2020.

[46] J. V. Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, USA, 1944.

[47] J. F. Nash Jr., "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48-49, 1950.

[48] Y. F. Luo, *Game Theory*, Tsinghua University Press, Beijing Jiao tong University Press, Beijing, China, 2007.

[49] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.
[50] K. Xie, J.-G. Wen, D.-F. Zhang, and G.-G. Xie, "Bloom filter query algorithm," *Journal of Software*, vol. 20, no. 1, pp. 96–108, 2009.
[51] Bitcoin Blockchain Parser, https://github.com/znort987/blockparser.
[52] Bitcoin Core, https://bitcoincore.org/bin/bitcoin-core-0.19.0.1/.
[53] Bitcoin Wallet, https://play.google.com/store/apps/details?id=de.schildbach.wallet.
[54] J. Bitcoin, "Privacy assumptions," 2014, https://github.com/bitcoinj/bitcoinj/blob/ee2a91010e5cf66299684160d6a48a108ff2299b/core/src/main/java/com/google/bitcoin/core/PeerGroup.java#L250.