2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application

2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application

A Two-Stage Deanonymization Attack Towards Bitcoin Hidden Service Nodes

Yue Gao*†‡, Jinqiao Shi*§, Xuebin Wang*†‡, Ruisheng Shi§, Can Zhao*†‡ and Chenglong Li¶
* Institute of Information Engineering Chinese Academy of Sciences
† National Engineering Laboratory for Information Security Technologies
‡ School of Cyber Security, University of Chinese Academy of Sciences
§ Beijing University of Post and Telecommunications
¶ Institute for Network Sciences and Cyberspace, Tsinghua University

***Abstract*—With the increasing popularity of Bitcoin, considerable attention has been paid to its privacy and anonymity. To protect users' privacy better from the P2P network perspective, Bitcoin encourages users to connect to the network through the anonymity network Tor. In this paper, we manage to prove that the combining of Bitcoin and Tor is easier to be exploited to deanonymize Bitcoin users. Specifically, we propose a two-stage deanonymization attack towards Bitcoin hidden service nodes, intending to reveal the IP addresses of the Bitcoin hidden services and identify their transactions. The experiments show that the node-level adversary and the gateway-level adversary both can correlate the IP address and the onion address of Bitcoin hidden services by shaping the traffic. On the premise of successful location deanonymization, the adversaries can identify the transactions from the target Bitcoin hidden service by delaying the packets. We also provide the theoretical analysis of the success probability and the countermeasures to mitigate this kind of attack.**

***Index Terms*—Bitcoin, Tor, Hidden Service, Deanonymization.**

I. Introduction

Bitcoin was first proposed by Nakamoto in 2008 [1], which has attracted wide attention since then. One of the reasons Bitcoin is attractive is that it is known as an anonymous currency, which can avoid censorship. The address of Bitcoin is generated based on public key encryption algorithm, which has no direct relationship with the users' real identity. To some extent, the pseudonym mechanism protects the privacy of users. However, it is known that these pseudo-anonymous transactions are also abused in illegal activities where the criminals try to obscure their money traces, like money laundering, drug dealing and so on.

On the one hand, some research manages to deanonymize Bitcoin users. Previous attacks can be divided into two categories, aiming at identifying addresses that belong to the same user or revealing the relationship between addresses and real-world identities, such as IP addresses. The former is necessary for deanonymization, because criminals may create large numbers of addresses to make transactions for hiding themselves better. Normal users can also create new addresses with the consideration of privacy. But to achieve the ultimate goal of deanonymization, a further step needs to be taken to link these addresses to real-world identities. One way to map addresses and thus entities to identities is by gathering information from side channels. For example, addresses exposed in news reports or publicly declared by users on social networks. A more rigorous way is exploiting the information revealed on the P2P network to correlate transactions to their originator's IP addresses.

On the other hand, multiple techniques have been proposed to address the Bitcoin privacy problem. Mixing services and alternative cryptocurrencies such as Dash and Litecoin are used to resist the deanonymization that is based on address clustering. The introduction of anonymous networks, such as Tor [2], aims to resist the attacks on the P2P network. Tor hidden services mechanism protects the IP addresses of Bitcoin nodes from exposure when accepting the incoming connections.

In this paper, we try to analyze whether the introduction of Tor hidden services protects the anonymity of bitcoin users better. The ultimate goal of deanonymization is to reveal the relationship between bitcoin transactions and real-world identifiers. In previous attack scenarios without Tor, attackers only need to associate transactions and the users' bitcoin node. In our model, two problems need to be solved, discovering the IP address of Bitcoin hidden services and then recognizing their transactions.

Our contributions. We propose a two-stage deanonymization attack towards Bitcoin hidden service nodes. To the best of our knowledge, it is the first deanonymization attack towards Bitcoin hidden service nodes. The key insight of the attack might also be applicable to hidden service nodes of other cryptocurrencies derived from Bitcoin, or Bitcoin hidden nodes over other anonymity networks, like I2P.

The crucial idea of the attack is that a malicious Bitcoin node can establish a connection with honest Bitcoin hidden service nodes and embed a signal through a specific behavior pattern, so that the collusive Tor guard node or local gateway can extract the onion address from the signal. When it is confirmed that a connection is compromised, the attacker can delay the messages on other connections of the target Bitcoin hidden service node to ensure that the malicious super node

Corresponding author: Xuebin Wang, Email:wangxuebin@iie.ac.cn

978-1-6654-9457-1/21/$31.00 ©2021 IEEE
DOI 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00095

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.

can receive the transactions from the target earliest.

The experiments show that the node-level adversary and the gateway-level adversary both can correlate the IP address and the onion address of Bitcoin hidden services with the accuracy of 93.80% and 97.75% respectively. On the premise of successful location deanonymization, the attacker can identify the transactions from the target Bitcoin hidden service with the approximate probabilities of transactions identification 90% and 80% respectively when the delay time is set to 5 seconds.

Roadmap. The rest of this paper is organized as follows. Section II gives a brief background of the Bitcoin network, Tor network, and the combining of Bitcoin and Tor. Section III describes how an attacker deanonymizes the Bitcoin hidden service nodes, including the threat model, the basic idea, and the details of the attack. In Section IV, we design and implement several experiments to prove the feasibility of our attack, and give the theoretical analysis of the deanonymization. In Section V we give some further discussion, about the mitigation of the attack, the limitations and future work, and the ethical concerns of our research. Section VI provides an overview of related work. Finally, in Section VII we conclude our paper.

II. Background

A. Bitcoin Network

1) Bitcoin P2P network: Bitcoin is managed by a fully distributed unstructured P2P network. According to whether or not accepting incoming connections, Bitcoin nodes can be divided into reachable nodes and unreachable nodes (nodes behind NAT or firewalls). At the time of writing, there are over 10 thousand reachable Bitcoin nodes. By default, each Bitcoin node maintains 8 outgoing connections. The maximum number of all connections is 125.

After joining the network and establishing connections, each Bitcoin node keeps the copy of the blockchain locally, which plays a role as a public ledger. When a transaction or block is generated, it is broadcasted to the whole network. After a block is verified by the proof-of-work mechanism, all transactions in the block are considered valid. Each Bitcoin node can be either the creator of a transaction or just a forwarder. Next, we will introduce the transaction propagation mechanism of the Bitcoin network in detail.

2) Transaction propagation: Bitcoin nodes propagate transactions through flooding, that is, each node will send its own transactions or received transactions to all of its neighbor nodes. To reduce the network traffic burden, the process of propagation can be divided into three steps (Fig. 1). Firstly, node A which has a new transaction advertises this fact to its neighbor B with an INV (inventory) message containing the transaction hash only. Secondly, if node B does not yet know the transaction announced in the INV message, it replies with a GETDATA message to request the entire transaction. Thirdly, the TX messages are sent by node A to respond to a GETDATA message. Upon receiving the complete transaction, node B sends the transaction to its neighbor C in the same manner.

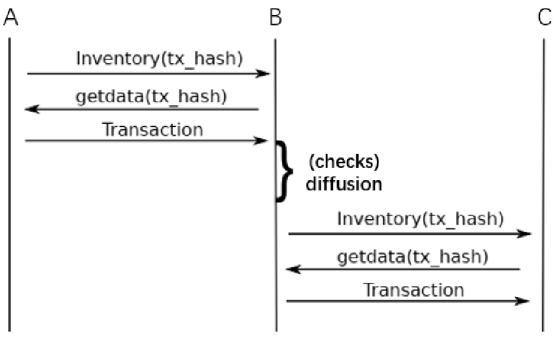


Fig. 1. Transaction propagation in Bitcoin.

Since broadcasting transactions by sending them to all neighbors as soon as possible harms privacy, Bitcoin nodes introduce random waiting times when sending transactions. The current propagation protocol in Bitcoin is known as diffusion spreading protocol, which spreads transactions with independent exponential delays. Besides, Bitcoin encourages user nodes to connect to Bitcoin network through TOR to protect privacy better.

B. Tor Network

1) Tor overview: The second generation onion router (Tor) is the most popular low-latency anonymity network with over 6,000 volunteer routers. Tor is based on onion routing, which provides both client anonymity and server anonymity (known as Hidden Service, HS). As shown in Fig. 2, when Alice tries to connect to Bob, it first chooses three Tor relays, guard, middle and exit. Then it builds a circuit and negotiates keys with each one of them. Before sending a message, the user encrypts it with the negotiated keys. When receiving the message, each hop in the circuit strips off its layer of encryption. In this way, the message reaches Bob anonymously. Besides, to avoid attacks based on the size of packets, Tor reformats traffic into constant sized (514 bytes) segments called cells.

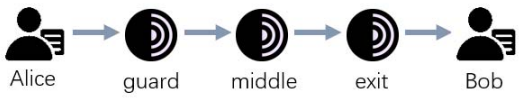


Fig. 2. Tor network communication.

2) Hidden service: Tor also allows server anonymity through hidden services mechanism, which supports users to offer all kinds of TCP services without revealing their IP addresses. The mechanism of the hidden service is described in [3]. As shown in Fig. 3, the circuit between a Tor client and a hidden service is composed of six Tor relays, guard, middle and rend-point relays of the client, guard, middle and exit of the hidden service. Since the guard node knows the IP address of the hidden service, it is chosen from a small set and doesn't change for a long period (about 120 days).

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.

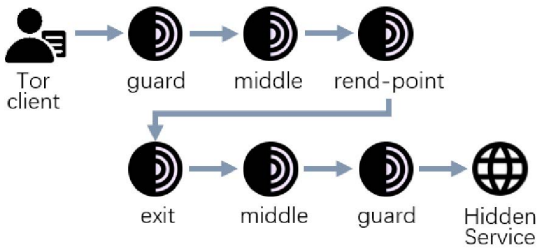


Fig. 3. Tor Hidden Service.

C. Bitcoin over Tor

There are two ways for nodes to connect to the Bitcoin network over Tor. One is through Tor proxy, the other is to run Bitcoin node as Tor hidden service.

1) Bitcoin nodes via Tor proxies: Bitcoin nodes can connect to the Bitcoin network simply over Tor proxy to protect their privacy better. This kind of node can actively establish a connection with normal Bitcoin nodes or those nodes acting as Tor hidden services. For privacy reasons, such nodes do not open TCP port (default 8333) to accept incoming connections, that is, they will only get a maximum of 8 total connections.

2) Bitcoin nodes as Tor hidden services: Any TCP-based service can be made available as a Tor hidden service. In Bitcoin, Onioncat address format is a way to represent an onion address of Bitcoin hidden service node as an IPv6 address: the first 6 bytes of an OnionCat address are fixed and set to FD87:D87E:EB43 and the other 10 bytes are the hex version of the onion address (i.e. base32 decoded onion address after removing the ".onion" part). When a bitcoin node acts as a hidden service, it accepts incoming connections from bitcoin nodes acting as Tor clients (i.e., bitcoin nodes using Tor proxies).

III. Our Approach

In this section, we propose a two-stage deanonymization attack towards Bitcoin hidden service nodes. We describe the threat models, the basic ideas, and the details of the attack. Although we focus on Bitcoin hidden service nodes, our method is also applicable to Bitcoin nodes that access the network through Tor proxy.

A. Threat Model

According to the capabilities of the adversaries, we divide them into node-level adversaries and gateway-level adversaries. The threat models are shown in Fig. 4(a) and Fig. 4(b) respectively. In both threat models, the attacker needs to control a malicious Bitcoin node to establish connections with Bitcoin hidden service nodes and embed the signals. This node also plays the role of a super node, who is responsible for establishing connections with other bitcoin nodes in the P2P network and recording the details of transactions.

For a node-level adversary, the attacker needs to deploy a certain number of nodes in the Tor network to have a certain probability of being selected by the Bitcoin hidden service node as the guard node. This kind of threat model is similar to previous deanonymization attacks in Tor network [22]–[24].

For a gateway-level adversary, we assume a local gateway located at the edge of the Tor network, which can observe and control the traffic between the Bitcoin hidden service nodes and their guard nodes.

B. Basic Idea

The proposed deanonymization attack is split into two stages, location deanonymization and transaction deanonymization.

According to the protocol of Tor, only the guard relay or the local gateway knows the IP addresses of the Bitcoin hidden service nodes. However, they know nothing about the relationship between the onion addresses of the node and their IP addresses. We expect to shape the traffic from the malicious bitcoin node to the Bitcoin hidden service nodes, to deliver the onion address to the guard relay or the local gateway. Once the signal is detected and the onion address is extracted, it can be concluded that the Bitcoin hidden service node is at one end of the connection, thus exposing its IP address.

Once the embedded signal is detected, the malicious guard node or local gateway can start the second stage of the attack. Through the analysis of the Bitcoin transaction propagation mechanism, it can be seen that the randomness of the diffusion mechanism makes transaction creators not always be the first to report the transaction to the super node. To address this problem, we utilize the malicious guard node or local gateway to delay forwarding the messages of the target Bitcoin hidden service node. As a result, if the adversary receives any transaction from the target node earliest, the target node could be identified as the creator of the transaction.

C. Stage 1: Location Deanonymization

The onion addresses of Bitcoin hidden service nodes can be obtained through the bitcoin nodes crawler or third-party data sources, like *bitnodes.io* [6]. A malicious Bitcoin node can connect to any Bitcoin hidden service node with their onion addresses.

Node-level adversary: After the connection has been established, the onion address is modulated by the malicious Bitcoin node to the number of the Bitcoin heartbeat PING packets in each time window (6 seconds). For each window, two PING packets represent signal 1 and one PING packet is signal 0. At the time of writing, there are about five thousand Bitcoin hidden service nodes in the current Bitcoin network [6]. By numbering their onion addresses from 0, the identifiers can be represented as 12-bit binary numbers. The collusive guard relay records [the circuit ID, the previous-hop IP address, the cell direction, the cell timestamp] to detect the signal by counting the number of incoming Tor cells in each time window.

Gateway-level adversary: By contrast, the collusive local gateway can only see a TCP connection between the Bitcoin hidden service node and its guard node, but there is no way to distinguish packets between the various Tor circuits. That's to

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.

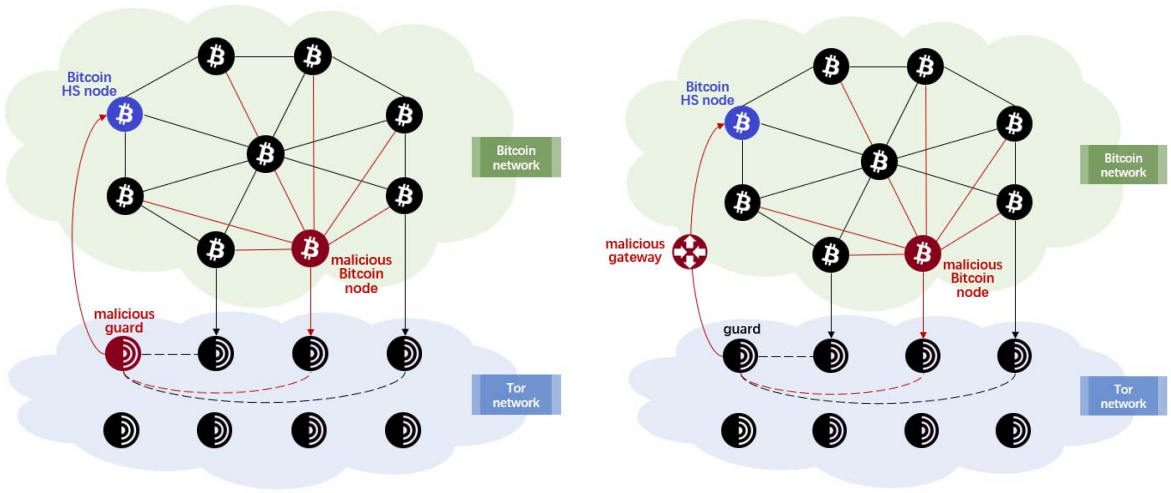


Fig. 4. (a) The threat model for the node-level adversary. (b) The threat model for the gateway-level adversary.

say, the packets-counting-based signal embedding method is no longer suitable for gateway-level adversary. Luckily, we noticed two facts that are conducive to signals to embed. Firstly, to speed up the synchronization process of historical blockchain data, Bitcoin nodes newly added to the network rarely choose Bitcoin hidden service nodes as their neighbor nodes. Secondly, as the size of a Tor cell is smaller than the MTU of IP packets, multiple cells can merge into one single packet. If an unsynced Bitcoin node chooses a Bitcoin hidden service node to synchronize historical data, it will increase the ratio of multiple-cell packets sent by the Bitcoin hidden service node. Therefore, we modulate the onion addresses to the ratio of multiple-cell packets. Specifically, we change the ratio by controlling whether the malicious Bitcoin node synchronizes data with the Bitcoin hidden service node. In each time window (20 seconds), if the ratio of multiple-cell observed by the collusive local gateway in the first 10 seconds is less than that in the next 10 seconds, it represents 1, otherwise it is 0. The collusive local gateway needs to observe all connections to Tor guard nodes, whose IP addresses can be obtained in Tor's consensus [7]. The details of records include [the IP addresses, the packet direction, the packet timestamp, the packet size], to count the ratio of multiple-cell packets in each time window.

D. Stage 2: Transaction Identification

Once the embedded signal is detected on a Tor circuit or a TCP connection, the second stage of the attack starts. The goal of transaction identification is to improve the probability that the Bitcoin hidden service node tells his own transactions to the super node earliest.

Node-level adversary: When the collusive guard node detects a signal on a circuit, it delays forwarding messages on all other circuits (except the collusion circuit) of the target Bitcoin hidden service node, including incoming messages and outgoing messages. Recall that Bitcoin transaction propagation consists of three messages, i.e., INV, GETDATA and TRANSACTION. Therefore, if the attacker delays each message by d seconds, the time for the neighbors to inform the transaction to the super node will be delayed by $3d$ seconds in total.

Gateway-level adversary: The challenge for the collusive local gateway is still its inability to distinguish between Tor circuits. We control the collusive local gateway to delay only the incoming messages on the TCP connection. As a result, the target hidden service node can send all INV messages and TRANSACTION messages as usual, but the GETDATA messages sent from its neighbors will be delayed. The innocent neighbors will not forward the transaction until it receives the complete transaction, i.e. the TRANSACTION message. But for the super node, it distinguishes transactions only based on the time of receiving the INV message. If the attacker delays each incoming message by d seconds, the time for the neighbors to inform the transaction to the super node will be delayed by d seconds in total.

IV. Experiments and Evaluation

In this section, we conduct several experiments of the deanonymization attack in the Tor network and Bitcoin network. In addition, we conducted a theoretical analysis to evaluate the effect of deanonymization.

A. Location Deanonymization as Node-Level Adversary

Experiment setup: To validate the feasibility of our location deanonymization method as a node-level adversary, we conduct the experiments in the living Tor network and Bitcoin network. Specifically, a modified Bitcoin node (version 0.21.0) was run as a malicious bitcoin client who can connect to the target bitcoin hidden service node and embed the signals. In the Tor network, we run a Tor relay based on a modified Tor (version 0.4.4.6) as the collusive guard node, which can detect

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.

TABLE I
LOCATION DEANONYMIZATION AS NODE-LEVEL ADVERSARY.

	Signal Detection			Address Extraction
NO.	Precision	Recall	F1-Score	Accuracy
1	100%	75.44%	91.30%	93.20%
2	100%	88.00%	93.62%	94.00%
3	100%	87.50%	94.74%	95.00%
4	100%	82.00%	92.47%	93.00%
5	100%	89.00%	94.18%	94.50%
Avg	100%	84.29%	93.26%	93.80%

and recognize our signals. Besides, an innocent Bitcoin node was run as the target bitcoin hidden service node.

Experiment result: During the experiment period, we did five repeated groups of experiments totally. In each group, we controlled the malicious Bitcoin node to embed signals 100 times on 100 circuits, which form our foreground data set. For background data, we randomly chose 100 circuits owned by other IPs at the collusive guard node.

In order to evaluate the performance of location deanonymization in more detail, we split it into two tasks. The first is to identify whether the circuit is embedded with a signal or not, which can be regarded as a binary classification problem. The second is to extract the onion address of the Bitcoin hidden service node from the signal, which can be regarded as a multi-class classification problem. The details of experiment results are shown in Table I. The average results indicate that the attacker can detect the signal with 100% precision, 84.29% recall, and 93.26% f1-score. And the accuracy of addresses extraction is 93.80% on average.

Catch probability analysis: One of the prerequisites for our method to succeed is that one of the controlled nodes is selected as the guard relay by the Bitcoin hidden service node. In this part, we provide the analysis of the catch probability [23].

The catch probability depends on the bandwidth and flags of Tor relays. According to the flags assigned by the Tor authorities, Tor relays can be divided into four types: relays with only guard flag (*guard-only, g*), relays with only exit flag (*exit-only, e*), relays with both guard and exit flags (*guard-exit, ge*), and relays without guard nor exit flags (*ge-none, n*).

Suppose that a Tor relay with the bandwidth B' belongs to the type $T \in \{g, e, ge, n\}$, then the probability of the relay selected as the node on the circuit with position $pos \in \{guard, middle, exit\}$ is $P_{pos}(B', T)$. And the $P_{pos}(B', T)$ can be calculated as:

$$\begin{aligned} P_{pos}(B', T) &= \frac{B' \cdot W_{pos}^{T}}{B_*} \\ B_* &= B_g \cdot W_{pos}^{g} + B_e \cdot W_{pos}^{e} + B_{ge} \cdot W_{pos}^{ge} + B_n \cdot W_{pos}^{n} \end{aligned} \tag{1}$$

Where B_g, B_e, B_{ge}, B_n are the total bandwidth of each type of relays in the whole Tor network. The weights W_{pos}^{T} are given in Table II.

TABLE II
BANDWIDTH-WEIGHTS

Position \ Flag	guard-only	exit-only	guard-exit	none
guard	1.0	0.0	w_e	0.0
middle	w_g	w_e	w_{ge}	1.0
exit	0.0	1.0	w_g	0.0

Assuming that the controlled relays have the same bandwidth b, and the numbers of controlled relays with different types are $k_c = \{k_g, k_e, k_{ge}, k_n\}$. So the bandwidth of controlled relays can be represented as $k_c b = \{k_g b, k_e b, k_{ge} b, k_n b\}$.

The catch probability of the controlled relays caught as guard node is $P_g(k_c b)$. We ignore the increase of B_* caused by relays deployed by the attacker since $k_c b$ is far less than B_*.

$$P_g(k_c b) = \frac{k_g + k_{ge} w_e}{B_g + B_{ge} w_e} b \tag{2}$$

Since $w_e < 0$, all controlled nodes should be deployed as guard-only nodes to ensure the full utilization of bandwidth resources. In this case, Equation 2 can be rewritten as:

$$P_g(k_c b) = \frac{k_g \cdot b}{B_g + B_{ge} w_e} \tag{3}$$

According to the consensus of July 23, 2021, the value of $B_g + B_{ge} w_e$ is about $53,775$ MB/s. When the bandwidth of the controlled node is 20 MB/s and the number of controlled nodes is 50, the catch probability is approximately 1.9%. According to statistics on bitnodes.io [6], the Bitcoin nodes explorer, the current number of Bitcoin hidden service nodes is about 5 thousand. Thus over 100 nodes are expected to be caught by our controlled guard nodes. The attacker can increase the catch probability by increasing the total number and bandwidth of controlled nodes, and extending the attack time to wait for the Bitcoin hidden service nodes to change their guard nodes.

B. Transaction Identification as Node-Level Adversary

Experiment setup: We conducted small-scale experiments of transaction identification in the real-world Bitcoin network and large-scale experiments in the Bitcoin test network (testnet3). Testnet3 is an alternative Bitcoin blockchain that developers use for testing. The transaction identification experiments were carried out on testnet without using real bitcoins or worrying about breaking the main chain. Compared with the location deanonymization, in the second stage the malicious Bitcoin node is not only responsible for embedding the signal, but also needs to play the role of a supernode. Besides, an innocent Bitcoin hidden service node was deployed to create and relay transactions normally. The Bitcoin sent on testnet3 was freely taken from the faucet [5].

Experiment result: In order to observe the impact of different delay times on the effect of transaction identification,

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.

TABLE III
TRANSACTION IDENTIFICATION AS NODE-LEVEL ADVERSARY.

Delay	0s	1s	2s	3s	4s	5s
Probability	53.8%	73.6%	83.1%	89.5%	94.7%	96.3%

we conducted a total of 6 groups of experiments in testnet3. The added delay time of each group was set to 0, 1, 2, 3, 4, and 5 seconds respectively. For each group, 300 transactions were created and broadcast to the whole network by the target Bitcoin hidden service node, which composes the target transactions set.

As shown in Table III, as the delay time increases, the probability of identification increases significantly. When the delay is 5 seconds, the attacker can identify the transactions of the target Bitcoin hidden service node with a probability of 96.3%. In addition, we created 10 transactions in the real-world Bitcoin network and set the delay to 5, of which 9 transactions were successfully identified. It indicates that the proposed deanonymization method also works well in real-world Bitcoin network.

Analysis: Now, we will give the analysis of the relationship between the delay time and the success probability of transactions identification. In Bitcoin, each node sends INV message to each of its neighbors with an independent, exponential delay of rate λ. By deeply investigate the implementation of Bitcoin client, we noticed that for the outgoing connections, $\lambda_{out} = 2.5$ (seconds), but for the incoming connections, $\lambda_{in} = 5$ (seconds). In addition, when a neighbor node receives an INV message from an incoming connection, it will wait 2 seconds before sending the GETDATA message, but if it is from an outgoing connection, the GETDATA message will be sent directly.

Firstly, we give the lower bound of the success probability. Suppose that the delay added by the malicious guard node on an innocent circuit is $3d$, and the random delay added by the target node when sending the transaction to the attacker is Y. If $Y < 3d$, then the attacker must receive the transaction from the target node earliest. This is a sufficient and unnecessary condition. Since $Y \sim Exp(\lambda_{in})$, the lower bound of the success probability P_{lower} can be written as:

$$P_{lower} = P(Y < 3d) = F_Y(3d) = 1 - e^{-3d\lambda_{in}} \quad (4)$$

Then, we give the upper bound of the success probability. By default, each Bitcoin node can establish 8 outgoing connections. For each outgoing connections, the total delay of the INV message from the target source node to its outgoing neighbors and then to the attacker is $X = T_1 + T_2 + 3d + 2$, where $T_1 \sim Exp(\lambda_{out})$ and $T_2 \sim Exp(\lambda_{in})$. If the attacker receive the transaction from the target node earliest, then $Y < min(X_1, X_2, ...X_8)$. This is a necessary and insufficient condition. Let $Z = min(X_1, X_2, ...X_8)$, the upper bound of the success probability P_{upper} can be written as:

$$P_{upper} = P(Y < min(X_1, X_2, ...X_8)) = \int F_Y(z) f_Z(z) dy \quad (5)$$

TABLE IV
LOCATION DEANONYMIZATION AS GATEWAY-LEVEL ADVERSARY.

	Signal Detection			Address Extraction
NO.	Precision	Recall	F1-Score	Accuracy
1	94.12%	96.00%	95.05%	98.50%
2	96.81%	91.00%	93.82%	95.50%
3	93.70%	90.00%	91.81%	98.50%
4	94.94%	94.00%	94.47%	98.50%
Avg	94.89%	92.75%	93.79%	97.75%

It is worth noting that our analysis of the upper bound and lower bound has no connection with the network size, so it is equally applicable to testnet and mainnet. The values of P_{lower}, P_{upper} and the experimental success probabilities are depicted in Fig. 5(a). It can be seen that the experimental results align closely with the upper bound, since during our experiment, the average number of neighbors of the target Bitcoin hidden service node is slightly greater than 8. The probabilities of transaction identification are positively correlated with d. There is a compromise between the stealthiness of attacks and the probability of identification.

C. Location Deanonymization as Gateway-Level Adversary

Experiment setup: The experiment setup for location deanonymization as the gateway-level adversary is similar to that as the node-level adversary, which has been described in Section IV-A. We still deployed a malicious guard node, but we just grab all encrypted TCP/IP packets at this node to act like a gateway-level adversary.

Experiment result: During the experiment period, we did four repeat groups of experiments totally. In each group, we controlled the malicious Bitcoin node to embed signals 100 times, and the details of experiment results are shown in Table IV. The average results indicate that the attacker can detect the signal with 94.89% precision, 92.75% recall, and 93.80% f1-score. And the accuracy of addresses extraction is 97.75% on average.

D. Transaction Identification as Gateway-Level Adversary

Experiment setup: We also conducted large-scale experiments in Bitcoin test network (testnet3) and small-scale experiments in the real-world Bitcoin network to validate the feasibility of transaction identification as the gateway-level adversary. Compared with the node-level adversary, the gateway-level adversary only delays the incoming packets on the TCP connection.

Experiment result: In order to observe the impact of different delay times on the effect of transaction identification, we conducted a total of 6 groups of experiments in testnet3. The added delay time of each group was set to 0, 1, 2, 3, 4, and 5 seconds respectively. For each group, 200 transactions were created and broadcast to the whole network by the target Bitcoin hidden service node, which composes the target transactions set. The details of the experiment results are described

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.

TABLE V
TRANSACTION IDENTIFICATION AS GATEWAY-LEVEL ADVERSARY.

Delay	0s	1s	2s	3s	4s	5s
Probability	45.5%	59.5%	69%	73.5%	78.0%	81.0%

in Table V. When the delay is set to 5 seconds, the attacker can identify the transactions of the target Bitcoin hidden service node with a probability of 81.0%. In addition, we created 10 transactions in the real-world Bitcoin network and set the delay to 5, of which 7 transactions were successfully identified.

The theoretical analysis of the identification probability of the gateway-level adversary is similar to that of the node-level adversary. When the delay added by the local gateway is three times that by the guard node, the same identification probability can be obtained. The values of P_{lower}, P_{upper} and the experimental success probabilities for the gateway adversary are depicted in Fig. 5(b).

V. DISCUSSION

A. Mitigation of the Attack.

Bitcoin hidden service nodes can determine whether there is a malicious behavior of its guard relay or gateway by some detection schemas. For instance, detect the frequency of received messages to prevent packets-counting-based signal embedding attacks, detect the interval between INV messages and GETDATA messages to prevent malicious delay forwarding and so on.

In addition, the proposed deanonymization attack is towards Bitcoin hidden nodes over Tor. Except for Tor network, there are some different routing schemes that can protect user's privacy at the network level, such as, Dandelion [25] and its improvement Dandelion++ [26].

B. Limitations and Future Work.

The proposed deanonymization attack can be utilized by guard nodes to check whether the hidden services they serve are Bitcoin nodes, or utilized by the gateway to check whether there is a Bitcoin node in its network, and if so, they can deanonymize the node and identify its transaction. But limited by resources, it's impossible for the adversary to deanonymize all Bitcoin hidden service nodes.

In the first stage of the attack, we utilize signal embedding for location deanonymization. It is worth noting that we chose this embedding and modulating schema because it is easy to implement, and it is enough to show the feasibility of our method. But in fact, it can also be replaced by other more robust methods.

In the second stage of the attack, to save bandwidth resources, the super node only receives transactions and does not forward transactions. In practice, super nodes can forward transactions normally, due to concealment considerations. We delayed all messages in the experiments, maybe some work can be done to distinguish the types of Bitcoin messages, and then only delay the transaction, so that the attack can be carried out in a more stealth way.

C. Ethical Considerations.

In order to evaluate the performance of our two-stage deanonymization attack towards Bitcoin hidden service nodes, we conducted several experiments in the living Tor network and Bitcoin network. All the experiments on the living network are performed in a responsible manner. The Bitcoin hidden service node and their transactions used for deanonymization were all generated by ourselves. Additionally, we securely delete all collected data after statistically analyzing them, only publish aggregated statistics about the collected data.

VI. RELATED WORK

The anonymity and privacy of cryptocurrencies, like Bitcoin and Ethereum, have attracted accumulating attention and research. Some studies try to enhance the anonymity of cryptocurrencies [14]–[18], [25], [26], while others manage to deanonymize users from various perspectives. The deanonymization attacks can be divided into two categories, aiming at identifying addresses that belong to the same user [8], [9], [19], [20], or revealing the relationship between addresses and real-world identities. The analysis of P2P network provides the possibility to establish the association, by correlating transactions to their originators' IP addresses.

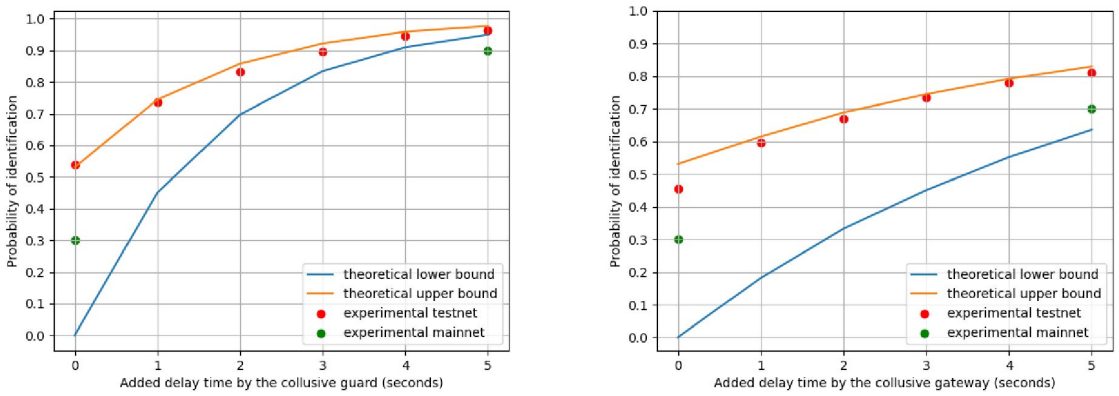
The deanonymization attack on blockchain P2P network was first proposed by Dan Kaminsky during the 2011 Black Hat conference [4], which has been applied in [10]. In 2015, the Bitcoin community responded to these attacks by changing the network's flooding mechanism to a different protocol, known as diffusion. However, by theoretically analyzing the anonymity properties of Bitcoin in [12], the author proved that the diffusion protocols also offer poor anonymity. To better protect the privacy of users, Bitcoin encourages nodes to connect to the network over Tor.

In 2015, Biryukov [21] showed that combining Tor and Bitcoin creates an attack vector for the man-in-the-middle attacks. Moreover, the author also shows that an attacker can fingerprint users and recognize them and learn their IP address when they decide to connect to the Bitcoin network directly. To the best of our knowledge, this is currently the only research that analyzes the security of the mechanism of combining Tor and Bitcoin. However, the attack technique proposed focuses more on the availability of Bitcoin nodes. Although the author also proposed an IP address exposure method based on "address cookie", the prerequisite for the success of this attack is that the Bitcoin hidden service nodes will directly connect to the Bitcoin network in some cases. But for a privacy-conscious Bitcoin user, he may not connect to the Bitcoin network directly, which makes the attack not work. In addition, this method can only deanonymize the location of Bitcoin hidden service nodes, but can not identify their transactions.

VII. CONCLUSION

In this paper, we study the state of anonymity of Bitcoin hidden service nodes. Specifically, we describe and implement a two-stage deanonymization attack towards Bitcoin hidden

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 5. (a) The comparison of the identification probability of the node-level adversary, theoretical and experimental. (b) The comparison of the identification probability of the gateway-level adversary, theoretical and experimental.

service nodes, with the goal to reveal the IP addresses of the Bitcoin hidden services and identify their transactions. In detail, the experiments show that the node-level adversary and the gateway-level adversary both can correlate the IP address and the onion address of Bitcoin hidden services by shaping the traffic, with the accuracy of 93.80% and 97.75% respectively. On the premise of successful location deanonymization, the attacker can identify the transactions from the target Bitcoin hidden service with the approximate probabilities of transactions identification 90% and 80% respectively when the delay time is set to 5 seconds. The experiment results and the theoretical analysis prove that the combining of Bitcoin and Tor is easy to be exploited to deanonymize Bitcoin users.

ACKNOWLEDGMENT

This work was supported by the Key Research and Development Program for Guangdong Province under Grant 2019B010137003, the Beijing Natural Science Foundation under Grant M21037, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDC02040400.

REFERENCES

[1] Nakamoto, Satoshi. “Bitcoin: A peer-to-peer electronic cash system.” Decentralized Business Review (2008): 21260.
[2] “Tor Project — Anonymity Online”. [Online]. Available: https://www.torproject.org/.
[3] “Tor specification.” [Online]. Available: https://gitweb.torproject.org.
[4] Kaminsky D. Black ops of TCP/IP 2011[J]. Black Hat USA, 2011: 44.
[5] “Yet Another Bitcoin Testnet Faucet! Bech32!”. [Online]. Available: https://testnet-faucet.mempool.co/.
[6] “Global Bitcoin Nodes Distribution - Bitnodes”. [Online]. Available: https://bitnodes.io/.
[7] “Index of /recent/relay-descriptors/consensuses”. [Online]. Available: https://collector.torproject.org/recent/relay-descriptors/consensuses/.
[8] Victor F. Address clustering heuristics for Ethereum[C]//International Conference on Financial Cryptography and Data Security. Springer, Cham, 2020: 617-633.
[9] Béres F, Seres I A, Benczúr A A, et al. Blockchain is Watching You: Profiling and Deanonymizing Ethereum Users[J]. arXiv preprint arXiv:2005.14051, 2020.
[10] Koshy P, Koshy D, McDaniel P. An analysis of anonymity in bitcoin using p2p network traffic[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014: 469-485.
[11] Biryukov A, Khovratovich D, Pustogarov I. Deanonymisation of clients in Bitcoin P2P network[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 15-29.
[12] Fanti, Giulia, and Pramod Viswanath. “Deanonymization in the bitcoin P2P network.” Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017.
[13] Biryukov, Alex, and Sergei Tikhomirov. “Deanonymization and linkability of cryptocurrency transactions based on network analysis.” 2019 IEEE European Symposium on Security and Privacy. IEEE, 2019.
[14] Bonneau, Joseph, et al. “Mixcoin: Anonymity for bitcoin with accountable mixes.” International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014.
[15] Maxwell, Gregory. “CoinJoin: Bitcoin privacy for the real world.” Post on Bitcoin forum. 2013.
[16] Ruffing, Tim, Pedro Moreno-Sanchez, and Aniket Kate. “Coinshuffle: Practical decentralized coin mixing for bitcoin.” European Symposium on Research in Computer Security. Springer, Cham, 2014.
[17] Meiklejohn, Sarah, and Rebekah Mercer. “Möbius: Trustless tumbling for transaction privacy.” (2018): 881-881.
[18] Seres, István András, et al. “Mixeth: efficient, trustless coin mixing service for ethereum.” International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
[19] Androulaki, Elli, et al. “Evaluating user privacy in bitcoin.” International conference on financial cryptography and data security. Springer, Berlin, Heidelberg, 2013.
[20] Reid, Fergal, and Martin Harrigan. “An analysis of anonymity in the bitcoin system.” Security and privacy in social networks. Springer, New York, NY, 2013. 197-223.
[21] Biryukov, Alex, and Ivan Pustogarov. “Bitcoin over Tor isn’t a good idea.” 2015 IEEE Symposium on Security and Privacy. IEEE, 2015.
[22] Overlier, Lasse, and Paul Syverson. “Locating hidden servers.” 2006 IEEE Symposium on Security and Privacy. IEEE, 2006.
[23] Chen, Muqian, et al. “Signalcookie: Discovering guard relays of hidden services in parallel.” 2019 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2019.
[24] Ling, Zhen, et al. “Protocol-level hidden server discovery.” 2013 Proceedings IEEE INFOCOM. IEEE, 2013.
[25] Bojja Venkatakrishnan, Shaileshh, Giulia Fanti, and Pramod Viswanath. “Dandelion: Redesigning the bitcoin network for anonymity.” Proceedings of the ACM on Measurement and Analysis of Computing Systems 1.1 (2017): 1-34.
[26] Fanti, Giulia, et al. “Dandelion++ lightweight cryptocurrency networking with formal anonymity guarantees.” Proceedings of the ACM on Measurement and Analysis of Computing Systems 2.2 (2018): 1-35.

Authorized licensed use limited to: Concordia University Library. Downloaded on March 16,2024 at 21:06:11 UTC from IEEE Xplore. Restrictions apply.