Report on Anonymity of Bitcoin Network

Siqi Liu
515021910324
Shanghai Jiao Tong University, China

ABSTRACT

Bitcoin have surged in popularity over the last decade for its fairness and transparency as a financial system, which is realized by decentralized technology. It enjoys a public perception of being a 'privacy-preserving' financial system. Cryptocurrencies publish users' entire transaction histories in plaintext, albeit under a pseudonym which is for transaction validation. In reality, however, there are many deanonymization attacks studied by the researchers that exploit weaknesses in the Bitcoin network's peer-to-peer (P2P) networking protocols. The attacker links the public key to its originating source's IP addresses, which may cause serious privacy problems.

In this report, firstly I focus on the open problem from the first work[1], which is about how to **break the symmetry** of the spreading protocols in order to weaken deanonymization attacks. Then I made some work on breaking the symmetry. Finally, I revise my model based on the second reference work [2], which redesigned the P2P network, providing strong, provable anonymity guarantees and effective methods to analyze the anonymity property and deanonymization attacks.

KEYWORDS

Cryptocurrencies, Bitcoin, Peer-to-Peer Networks, Symmetry of spreading protocols,Privacy

1 INTRODUCTION

Cryptocurrencies exhibit two key properties: egalitarianism and transparency. In this context, egalitarianism means that no single party wields disproportionate power over the network's operation. This diffusion of power is achieved by asking other network nodes (e.g., other Bitcoin users) to validate transactions, instead of the traditional method of using a centralized authority for this purpose. Moreover, all transactions and communications are managed over a fully-distributed, peer-to-peer (P2P) network.

Cryptocurrencies are transparent in the sense that all transactions are verified and recorded with cryptographic integrity guarantees; this prevents fraudulent activity like double-spending of money. Transparency is achieved through a combination of clever cryptographic protocols and the publication of transactions in a ledger known as a blockchain. This blockchain serves as a public record of every financial transaction in the network.

In this report, I firstly study the anonymity of Bitcoin Network based on the work[1] which teacher Fu assigned. The first work consider new adversarial models and spreading mechanisms and theoretically prove that Bitcoin's networking protocols (both pre- and post-2015) offer poor anonymity properties. Then, my work is mainly based on the open problem from the first work, which is about how to break the symmetry of the spreading protocol in order to weaken deanonymization attacks. To start with, I will introduce some policy about the Bitcoin network below.

1.1 Bitcoin Primer

Bitcoin represents each user and each unit of Bitcoin currency by a public-private key pair. A user possesses a coin by knowing its private key. Any time a user A wishes to transfer her coin m to B, it generates a signed transaction message, which states that A (denoted by her public key) transmitted m (denoted by its public key) to B (denoted by his public key). This transaction message is broadcast to the rest of the whole network (other nodes), which help validate transactions and race to append the transaction to a global ledger known as the blockchain.

1.2 Bitcoin message propagation

To understand the mechanics of broadcasting, note that cryptocurrencies can be abstracted into two layers: the application layer and the network layer.The application layer handles tasks like transaction management, blockchain processing, and mining. Nodes are identified by their public keys in the application layer.The network layer handles communication between nodes, where nodes are identified by their IP addresses. In fact, the node's IP address and public key should remain unlinkable for privacy reasons. Bitcoin's peer-to-peer broadcast of transactions and blocks is based on some spreading protocols like Trickle and Diffusion, which is studied well by many researchers.

1.3 Anonymity in the Bitcoin P2P network

Recently, several deanonymization attacks exploit Bitcoin's transaction flooding protocols. To be specific, if an attacker can infer the IP address that initiated a transaction broadcast, then the attacker can also link the IP address to the associated user's Bitcoin pseudonym, which could cause a serious privacy problem.

Paper Structure. I begin by focusing on the work [1] in section II to demonstrate some lightspots the paper proposed, where the open problem about the symmetry of spreading protocols is given. Then I propose a model in spreading protocols in Section III. In section IV I analyze the idea of the second work[2], which provides some also considerably useful references.

2 PROBLEM STATEMENT

2.1 Bitcoin Network Model

2.1.1 Network model. The authors of the first work model the P2P network of servers as a graph *G(V,E)*, where V is the set of all server nodes and E is the set of edges, or connections, between them. In practice, each server node is represented by a (IP address, port) tuple. Since regular trees are a natural class of graphs to study. In their theoretical analysis, they model G as a d-regular tree.

2.1.2 Spreading Protocols. Until 2015, the Bitcoin network used a gossip-like trickle broadcasting protocol. However, in the wake of various anonymity attacks, the reference Bitcoin implementation changed its networking stack to use a different broadcasting protocol known as diffusion.

Trickle spreading is a gossip-based flooding protocol. In this spreading process, Each message source or relay randomly orders its neighbors who have not yet seen the message. It then transmits the message to its neighbors which has a link to the source or the relay, according to the order. We can model this spreading protocol by assuming a discrete-time system.

Diffusion spreading is a continuous-time system, in which each source or relay node transmits the message to each of its uninfected neighbors with an independent, exponential delay of rate λ.

2.1.3 Adversarial Model. The author introduce an **eavesdropper adversary**. It is a supernode that connects to most of the servers in the Bitcoin network. The supernode can

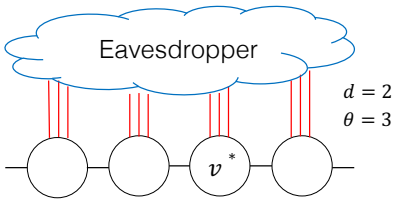

Figure 1: The eavesdropper adversary establishes θ links (shown in red) to each server. Honest servers are connected in a d-regular tree topology (edges shown in black).

make multiple connections to each honest server, with each connection coming from a different (IP address, port). The eavesdropper adversary model is illustrated in Figure 1.

Source Estimation. The adversary's goal is as follows: given the observed noisy timestamps τ (up to estimation time t) and the server graph G, find an estimator M(τ ,G) that correctly identifies the true source. The metric of success for the adversary is **probability of detection**. Meanwhile, the authors mention two estimator, which is the **first-timestamp estimator** and the **maximum-likelihood (ML) estimator**. However, as we can see in the second work [2] (in Section IV), **the probability of detection cannot exactly describe the abilities of a adversary** and [2] proposes **precision** and **recall** to evaluate the abilities of a adversary.

2.2 Contributions of the first work

The authors consider new adversarial models named 'eavesdropper adversary' and spreading mechanisms that have not been previously studied in the source-finding literature and they theoretically prove that Bitcoin's networking protocols (both pre- and post-2015) offer poor anonymity properties.

Analysis of Trickle (Pre-2015) and Diffusion (Post-2015). By in-depth study of two kinds of estimators, the authors analyze the probability of both spreading protocols, Trickle and Diffusion. The table 1 below illustrate the summary of probability of detection.

2.3 Open Problem

In the end of this work, the author mention several open problems. I curiously pay attention to one of the problem, which is about breaking the symmetry of the spreading protocols to prevent deanonymization attacks. The authors state that a key reason that deanonymization is currently possible is because of the symmetry of current spreading protocols. That is to say, diffusion and trickle both propagate content over the underlying graph in all directions at roughly the same rate. This symmetry enables powerful centrality-based attacks. Thus, a natural solution is to break the symmetry of diffusion and trickle.

Although it is hard to directly design a new spreading protocol, after researching on many papers and techniques, I try to propose a model which to some extend breaks the symmetry of the spreading protocols, and I will illustrate some details of the model in the later section.

3 MY MODEL

It is clearly that diffusion and trickle has a common characteristic by the knowledge from Section II, and that is to say both of the protocols spreads in the network in a symmetrical ways. In trickle, each message source or relay randomly orders its uninfected neighbors and then transmits the message to its neighbors according to the ordering. In diffusion, each source or relay node transmits the message to each of its uninfected neighbors with an independent, exponential delay of rate λ.

In this section, I try to break the symmetry of the spreading protocols and offer an idea by making the source firstly choose an uninfected neighbor randomly and then transmit the message to the chosen neighbor. It is called one hop. Then the neighbor who has received the message repeat the same action and it also choose an uninfected neighbor randomly and transmit the message to it. It is called the second hop. After K hops like this, the final node who receives the message begins spreading the message to the whole network by diffusion. Note that there some variables we can control to study our model, such as the total hops K and the way one node choose the next neighbor.

Table 1: Summary of probability of detection results on a network of honest servers in a d-regular tree topology. The adversary has θ connections to each honest server.

		Trickle	Diffusion
First-Timestamp	All θ	$\frac{\theta}{d\log 2}\left[\mathrm{Ei}(2^d\log\rho)-\mathrm{Ei}\left(\log\rho\right)\right]$	$\frac{\theta}{d-2}\log\left(\frac{d+\theta-2}{\theta}\right)$
	$\theta=1$	$\frac{\log(d)}{d\log(2)}+o\left(\frac{\log d}{d}\right)$	$\frac{\log(d-1)}{(d-2)}$
Maximum-Likelihood	All θ	$1-\frac{d}{2(\theta+d)}$	$1-d\left(1-I_{1/2}\left(\frac{1}{d-2},1+\frac{1}{d-2}\right)\right)$
	$\theta=1$	$1-\frac{d}{2(d+1)}$	

3.1 total hops K.

We can control total hops K dynamically according to the **complexity of the P2P network** (C), the **danger level** (μ) which can be evaluated by the possibility of detection, as well as the **transmitting time (t)** in which the source transmits the message to the whole network. I find intuitively that the more hops the nodes choose, the safer the network is and the spreading pattern is less likely learned by the adversary. However, this kind of phenomenon currently lack a exact proof yet.

$$K \propto \frac{C * \mu^2}{\sqrt{t}} \tag{1}$$

Note that the exponent over **danger level** (μ) is bigger than **complexity of the P2P network** (C) for the consideration that K in this way is more dynamical and more defensive against attacks, while the complexity of the network will not change critically over a certain time. Meanwhile, I choose $\sqrt{t}$ in the denominator as a punishment of K, in order to make a tradeoff between the K (assuming more hops leads to safer) and the total transmitting time.

What's more, the exponent over each variable in equation (1) is obtained theoretically and it needs more further work to test and validate.

3.2 the way choosing the next node.

For a certain node, there are many ways of choosing the next node. In this report, I make nodes transmit the message to the next node randomly, in order to provide a more random and mixing spreading pattern, which is hard to learn by adversaries. Moreover, we can take some references from the work [2], which refers to several Bitcoin network stacks, one of which is called 'Diffusion-by-Proxy', and in that protocols the authors explain some defects of offering too many paths to adversaries.

Diffusion-by-Proxy is a method to break the symmetry of the spreading protocols, in which for every transaction, the source node chooses a peer uniformly at random from the pool of **all nodes**. It transmits the message to that node, who then broadcasts the message. More generally, the network could forward each message a few hops (each hop choosing a new node at random) before diffusing it. My model differs from 'Diffusion-by-Proxy' in that the hops in my model is node-by-node, that is to say in every hop the node chooses one of its neighbor randomly and then transmit the message to it but not chooses from all of the nodes in the network.

The second work has stated that although 'Diffusion-by-Proxy' might seem like it should prevent attacks because the graph is so dynamic, that intuition actually turns out to be false. Intuitively, this statement holds because each node delivers its own message to the adversary with probability p, and few other nodes report to the adversary over the same edge. So even though diffusion-by-proxy breaks the symme try of diffusion, it also provides many paths for messages to reach the adversary. Since there are many total paths to the adversary, each path sees (relatively) less traffic, which in turn reduces the amount of mixing that happens.

However, in my model, the node chooses the next node from its neighbors, which means that the path to transmit the message is relatively directional, unlike to the large amount of the path provided by 'Diffusion-by-Proxy'. Thus, it might be better than 'Diffusion-by-Proxy'.

4 IMPROVEMENT ON THE BITCOIN NETWORK

4.1 Contributions of the second work

In the second work [2], the authors propose a simple networking policy called Dandelion, which achieves nearly-optimal anonymity guarantees at minimal cost to the network's utility.

4.2 Model

Dandelion consists of two phases. In the first phase, each transaction is propagated on a random line; that is, each relay passes the message to exactly one (random) node for a random number of hops. In the second phase, the message is broadcast as fast as possible using diffusion. Dandelion has two key constraints: (a) in the first phase, all transactions from all sources should propagate over the same line, and (b) the adversary should not be able to learn the structure of the line beyond the adversarial nodes' immediate neighbors.

Meanwhile, the authors offer many lightspots about Bitcoin P2P network, including precision and recall together rather than only the probability of detection to measure a broadcasting scheme's anonymity.

$$\textbf{Precision} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Positives}|}$$

$$\textbf{Recall} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Negatives}|}$$

Precision can be interpreted as the probability that a randomly-selected item with label 1 is correct, whereas recall can be interpreted as the probability that a randomly-selected data item from class 1 is correctly classified. Adapting this terminology to their problem, they have a multiclass classification problem; each server is a class, and each transaction is to be classified.

Dandelion has a great ability, however, it also faces a number of practical considerations, like how to construct the underlying line graph and how to prevent graph leakage.

4.2.1 Constructing a line graph. In this section, the author offers two method to construct a line graph. One is [5] and the other is to use Bitcoin's current networking strategy to approximate a line. That is to say we can approximate a line by asking each server to create one outgoing connection at random. We can further refine this protocol by having each server, prior to making a connection, contact k nodes and connect to the node with the smallest in-degree.

4.2.2 Preventing graph leakage. Another challenge associated with Dandelion is that it assumes the graph G is a line whose structure is unknown to the adversary. However, lines can be learned over time.

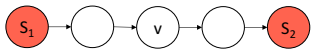

Figure 2: The adversary can easily learn line graphs.

Assume that the adversary can reliably infer the diffusion source v. Since s2 did not receive the message before the spreading phase began, and v was the source of the spreading phase, the adversary learns that v lies between s1 and s2. In this way, the adversary learns the internal, honest nodes of G at a rate proportional to the creation of new transactions; by learning this graph, the adversary's expected per-node precision grows to p.

4.3 Reflects on my model

As is stated in sections above, Dandelion and 'Diffusion-by-Proxy' have been proposed to break the symmetry of the spreading protocols, as well as the model I proposed. However, there are some significant differences among them. Dandelion make the spreading from different sources over the same line before diffusion phase; 'Diffusion-by-Proxy' chooses few proxies from the pool of all nodes randomly before diffusion; My model chooses proxies node-by-node, in which every sources or relay nodes only chooses proxies from its neighbors. In this way, Dandelion may have the best behavior, for which it break the symmetry of the spreading protocols more effectively. Meanwhile, while the effects might be the same between 'Diffusion-by-Proxy' and my model at the first glance, however, my model tends to have a exact direction to spread the message, in case of offering a large number of paths to adversaries, rather reducing the mixing of patterns in 'Diffusion-by-Proxy'.

Inspired by Dandelion, I make some revises on my model.

4.3.1 Strengthen the directivity. I make the node choose the next proxy according to the independent probability α of each its neighbors, rather than randomly choose the next nodes from its neighbors. In this way, Each node stores the information about the transmit probability α of its neighbors, and transmit the message to them by probability. For each neighbor, the nth transmit probability α is given below.

$$\alpha(n) = 1.1 * \alpha(n-1), n > 1 \tag{2}$$

The process goes like this. To begin with (n=1), we consider a certain node and each of its neighbors' transmit probability α is uniform. At the first round, the nodes transmit the message to its neighbors on equal probability. At the second round transmit (n=2), the probability α of the neighbor who relays the message at the first round rises from $\alpha(1)$ to $\alpha(2) = 1.1 * \alpha(1)$, while the rest of the neighbors' probability decrease by the same amount. Thus, the transmit continue hop K times and then the final nodes enter the diffusion phase. The design of the probability is to strengthen the directivity of the spreading, in which the node who relays more times has a higher possibility to transmit the neighbors in the next round. However, The probability cannot increase over a long time. Thus, we reset the probability after N round, where N is a controllable constant determined by the network and adversaries and is worthy a further study.

The design of probability can to some extend strengthen the directivity of the spreading based on the model in Section II.

5 CONCLUSION

In this report, I analyze the work of [1] [2], and illustrate some comprehensive knowledge about the Bitcoin network and its anonymity. It is clear that the anonymity of the Bitcoin network has been not protected well and many attacks occur due to the symmetry of the spreading protocols. Thus, Dandelion is designed to break the symmetry of the spreading. Moreover, I propose a model based on the two work, which has a strong directivity to break the symmetry. Further work is needed to test the model in its feasibility and its trade off between the time-consuming and the anonymity protection.

Acknowledgements

I would like to extend earnest thanks to teacher Wang, teacher Fu and all of other teachers as well as some graduates of the laboratory who give me help in this project.

REFERENCES

[1] Giulia Fanti, Pramod Viswanath, Anonymity Properties of the Bitcoin P2P Network, arXiv:1703.08761 [cs.CR], 26 Mar 2017.

[2] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, Pramod Viswanath, Dandelion: Redesigning the Bitcoin Network for Anonymity, arXiv:1701.04439 [cs.CR], 16 Jan 2017.

[3] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin p2p network. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pages 15-29. ACM, 2014.

[4] Philip Koshy, Diana Koshy, and Patrick McDaniel. An analysis of anonymity in bitcoin using p2p network traffic. In International Conference on Financial Cryptography and Data Security, pages 469-485. Springer, 2014.

[5] J. Kim and R. Srikant. Peer-to-peer streaming over dynamic random hamilton cycles. In Information Theory and Applications Workshop (ITA), 2012, pages 415-419. IEEE, 2012.