Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees

Extended Abstract

Giulia Fanti
ECE Department, CMU
gfanti@andrew.cmu.edu

Shaileshh Bojja Venkatakrishnan
EECS Department, MIT
shaileshh.bv@gmail.com

Surya Bakshi
ECE Department, UIUC
sbakshi3@illinois.edu

Bradley Denby
ECE Department, CMU
bdenby@andrew.cmu.edu

Shruti Bhargava
CS Department, UIUC
bharshruti@gmail.com

Andrew Miller
ECE Department, UIUC
soc1024@illinois.edu

Pramod Viswanath
ECE Department, UIUC
pramodv@illinois.edu

ABSTRACT

Recent work has demonstrated significant anonymity vulnerabilities in Bitcoin's networking stack. In particular, the current mechanism for broadcasting Bitcoin transactions allows third-party observers to link transactions to the IP addresses that originated them. This lays the groundwork for low-cost, large-scale deanonymization attacks. In this work, we present Dandelion++, a first-principles defense against large-scale deanonymization attacks with near-optimal information-theoretic guarantees. Dandelion++ builds upon a recent proposal called Dandelion that exhibited similar goals. However, in this paper, we highlight some simplifying assumptions made in Dandelion, and show how they can lead to serious deanonymization attacks when violated. In contrast, Dandelion++ defends against stronger adversaries that are allowed to disobey protocol. Dandelion++ is lightweight, scalable, and completely interoperable with the existing Bitcoin network. We evaluate it through experiments on Bitcoin's mainnet (i.e., the live Bitcoin network) to demonstrate its interoperability and low broadcast latency overhead.

CCS CONCEPTS

• **Mathematics of computing** → **Probabilistic algorithms**; • **Security and privacy** → *Security protocols*; *Security protocols*;

KEYWORDS

cryptocurrencies; anonymity; P2P networks

This work was supported in part by NSF grant CIF-1705007 and support from Input Output Hong Kong (IOHK), Jump Trading, CME Group, and the Distributed Technologies Research Foundation.

SIGMETRICS'18 Abstracts, June 18–22, 2018, Irvine, CA, USA

ACM ISBN 978-1-4503-5846-0/18/06.
https://doi.org/10.1145/3219617.3219620

ACM Reference Format:
Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. 2018. Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees: Extended Abstract. In *SIGMETRICS'18 Abstracts: ACM SIGMETRICS International Conference on Measurement & Modeling of Computer Systems Abstracts, June 18–22, 2018, Irvine, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3219617.3219620

1 INTRODUCTION

Anonymity is an important property for a financial system; unfortunately, the anonymity protections in Bitcoin and similar cryptocurrencies can be fragile. This is largely for two reasons: (a) Bitcoin users are identified by cryptographic pseudonyms, and (b) all transactions between users (i.e., between their pseudonyms) are logged on a public blockchain. If an attacker can link a user's pseudonym to her true identity, the attacker may be able to learn the user's entire transaction history.

Although researchers have traditionally focused on the privacy implications of having a public blockchain [7, 8, 10], a recent body of work has considered lower-layer vulnerabilities that emerge from Bitcoin's peer-to-peer (P2P) network. For instance, recent studies have demonstrated P2P-layer anonymity vulnerabilities that allow transactions to to be linked to users' IP addresses with accuracies over 30% [2, 6]. Such attacks represent a significant step towards deanonymizing users. Understanding how to patch these vulnerabilities without harming utility remains an open question. The goal of our work is to propose a practical, lightweight modification to Bitcoin's networking stack that provides theoretical anonymity guarantees against the types of attacks demonstrated in [2, 6], and others. We begin with an overview of Bitcoin's P2P network, and explain why it enables deanonymization attacks.

1.1 Bitcoin's P2P Network

Bitcoin nodes are connected over a P2P network of TCP links. This network is used to communicate transactions, the blockchain, and

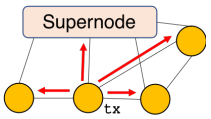

Figure 1: Supernodes can observe relayed transaction propagation metadata to infer which node was the source of a transaction message (tx).

control packets, and it plays a crucial role in maintaining the network's consistency. Each peer is identified by its (IP address, port) combination. Whenever a node generates a transaction, it broadcasts a record of the transaction over the P2P network; critically, transaction messages do not include the sender's IP address—only their pseudonym. Since the network is not fully-connected, transactions are relayed according to epidemic flooding [9]. This ensures that all nodes receive the transaction and can add it to the blockchain. Hence, the broadcasting of transactions enables the network to learn about transactions quickly and reliably.

However, the broadcasting of transactions can also have negative anonymity repercussions. Bitcoin's current broadcast mechanism spreads content isotropically over the graph; this allows adversarial peers who observe the spreading dynamics of a given transaction to infer the source IP of each transaction. For example, in recent attacks [2, 6], researchers launched a supernode (disguised as a regular node) that connected to all P2P nodes (Figure 1) and logged their relayed traffic. This allowed the supernode to observe the spread of each transaction over the network over time, and ultimately infer the source IP. Since transaction messages include the sender's pseudonym, the supernodes were able to *deanonymize* users, or link their pseudonyms to an IP address [2, 6].

There have been recent proposals for mitigating these vulnerabilities, included broadcasting protocols that reduce the symmetry of epidemic flooding. Bitcoin Core [1], the most popular Bitcoin implementation, adopted a protocol called *diffusion*, where each node spreads transactions with independent, exponential delays to its neighbors on the P2P graph. Diffusion is still in use today. However, proposed solutions (including diffusion) tend to be heuristic, and recent work shows that they do not provide sufficient anonymity protection [5]. Other proposed solutions, such as Dandelion [3], offer theoretical anonymity guarantees, but do so under idealistic assumptions that are unlikely to hold in practice. The aim of this work is analyze and modify Dandelion's broadcasting mechanism in order to (a) provide provable anonymity guarantees under *realistic* adversarial and network assumptions, and (b) demonstrate the network's broadcasting robustness or latency. We do this by revisiting the Dandelion system and redesigning it to withstand a variety of practical threats.

1.2 Contributions

The main contributions of this paper are threefold:
(1) We identify key idealistic assumptions made by Dandelion [3], and show how anonymity is degraded when those assumptions are violated. In particular, [3] assumes an honest-but-curious adversary that has limited knowledge of the P2P graph topology and *only observes one transaction per node*. If adversaries are instead malicious and collect more information over time, we show that they are able to weaken the anonymity guarantees of [3] through a combination of attacks, including side information, graph manipulation, black hole, and intersection attacks.
(2) We propose a modified protocol called Dandelion++ that subtly changes most of the implementation choices of Dandelion, from the graph topology to the randomization mechanisms for message forwarding. Mathematically, these (relatively small) algorithmic changes completely change the anonymity analysis by exponentially augmenting the problem state space. Using analytical tools from Galton-Watson trees and random processes over graphs, we show that Dandelion++ offers anonymity gains, both theoretically and in simulation, against stronger adversaries.
(3) We demonstrate the practical feasibility of Dandelion++ by evaluating an implementation on Bitcoin's mainnet (i.e., the live Bitcoin network). We show that Dandelion++ does not increase latency significantly compared to current methods for broadcasting transactions, and it is robust to node failures and misbehavior.

2 MAIN RESULTS

The main results of this paper fall in two categories: first, we examine simplifying assumptions made in [3] and evaluate the anonymity implications of an adversary who violates those assumptions. Second, we propose algorithmic changes that provide robustness against such an adversary, along with associated theoretical and empirical analyses. These attacks and countermeasures are summarized in this section and Table 1. We start with a brief description of Dandelion and its guarantees [3].

2.1 Dandelion [3] and Related Attacks

We adopt the same adversarial model as [3], in which a constant fraction p of network nodes are corrupt and wish to map transactions to users. Anonymity is measured by the expected *precision* and *recall* of this adversary; higher precision and recall imply more successful deanonymization. [3] begins by showing a fundamental lower bound: no protocol can achieve an expected precision below p^2 or an expected recall below p. Our goal (and that of [3]) is to meet this fundamental lower bound.

The original Dandelion protocol propagates transactions in two phases: (i) an anonymity (or *stem*) phase, and (ii) a spreading (or *fluff*) phase. In the anonymity phase, each message is passed to a single, randomly-chosen neighbor in an *anonymity graph* H (this graph can be an overlay of the P2P graph G). This propagation continues for a geometric number of hops with parameter q. Critically, different users forward their transactions along the *same* path in the anonymity graph H, which is chosen as a directed cycle in [3]. In the spreading phase, messages are flooded over the P2P network G via diffusion, just as in today's Bitcoin network. Visually the spreading pattern resembles a dandelion seed head. Dandelion periodically re-randomizes the line graph, so the adversaries' knowledge of the graph is assumed to be limited to their immediate neighborhood.

To analyze Dandelion, [3] makes three idealized assumptions: (1) all nodes obey protocol (including the adversarial nodes), (2) each node generates exactly one transaction, (3) all Bitcoin nodes run Dandelion. Under these assumptions, Dandelion achieves

Table 1: Summary of changes proposed in Dandelion++ [4].

Attack	Effect on Dandelion [3]	Dandelion++	
		Proposed solution	Effect
Graph-learning	Order-level precision increase [3]	4-regular anonymity graph	Limits precision gain
Intersection	Empirical precision increase	Pseudorandom forwarding	Improved robustness
Graph-construction	Empirical precision increase	Non-interactive construction	Reduces precision gain
Black-hole	Transactions do not propagate	Random stem timers	Provides robustness
Partial deployment	Arbitrary recall increase	Blind stem selection	Improves recall

an optimal expected recall of $O(p)$, and an expected precision of $O(p^2 \log p)$, within a logarithmic factor of optimal [3].

None of these assumptions necessarily holds in practice. In fact, we find that when these assumptions are violated, Dandelion may exhibit significantly worse anonymity properties than what the theoretical guarantees in [3] would suggest (depending on implementation choices). For example, attackers can increase their expected precision and/or recall by modifying the anonymity graph structure, running intersection attacks over many transactions, and exploiting the fact that any real deployment of Dandelion would require a gradual rollout. Attackers can also impact the robustness of Dandelion as a spreading mechanism by black-holing transactions, thus preventing them from spreading to the whole network.

2.2 Dandelion++

Dandelion++ aims to prevent these attacks by modifying the algorithmic specification of Dandelion in a principled way. Like Dandelion, Dandelion++ proceeds in asynchronous epochs; each node advances its epoch when its internal clock reaches some threshold (in practice, this will be on the order of 10 minutes). Within an epoch, the main algorithmic components are:

(1) Anonymity Graph: Instead of using a line graph for the anonymity phase (as in Dandelion), Dandelion++ uses a random, approximately-4-regular graph. This quasi-4-regular graph is embedded in the underlying P2P graph by having each node choose (up to) two of its outbound edges, without replacement, uniformly at random as Dandelion++ relays. The choice of Dandelion++ relays should be independent of whether the outbound neighbors support Dandelion++ or not. Each time a node changes epoch, it selects fresh Dandelion++ relays. We want to highlight that there are anonymity tradeoffs associated with using a 4-regular graph versus a line graph; these tradeoffs are discussed in more detail in the main paper and depend on the strength of the adversary.

(2) Transaction Forwarding (own): Every time a node generates a transaction of its own, it forwards the transaction, in stem phase, along the *same* outbound edge in the 4-regular anonymity graph. In Dandelion, nodes were assumed to generate only one transaction, so this behavior is not considered in prior analysis.

(3) Transaction Forwarding (relay): Each time a node receives a stem-phase transaction from another node, it either relays the transaction or diffuses it. The choice to diffuse transactions is pseudorandom, and is computed from a hash of the node's own identity and epoch number. Note that the decision to diffuse does not depend on the transaction itself—in each epoch, a node is either a diffuser or a relay node for *all* relayed transactions. If the node is not a diffuser in this epoch (i.e., it is a relayer), then it relays transactions pseudorandomly; each node maps each of its incoming edges in the anonymity graph to an outbound edge in the anonymity graph (with replacement). This mapping is selected at the beginning of each epoch, and determines how transactions are relayed.

(4) Fail-Safe Mechanism: Each node tracks, for each stem-phase transaction that was sent or relayed, whether the transaction is seen again as a fluff-phase transaction within some random amount of time. If not, the node starts to diffuse the transaction.

These small algorithmic changes completely alter the anonymity analysis by introducing an exponentially-growing state space. For example, moving from a line graph to a 4-regular graph (item (1)) invalidates the exact probability computation in [3], and requires a more complex analysis to understand effects like intersection attacks. We also simulate the proposed mechanisms for all attacks and evaluate anonymity tradeoffs compared to Dandelion.

Our analysis in the full paper [4] shows that Dandelion++ has similar anonymity properties to Dandelion, despite protecting against a stronger adversary. Moreover, we show empirically that Dandelion++ introduces low latency overhead; this is evaluated by running a number of Dandelion++ nodes in the Bitcoin mainnet and measuring the latency associated with using Dandelion++ compared to the current diffusion protocols.

REFERENCES

[1] 2015. Bitcoin Core Commit 5400ef6. https://github.com/bitcoin/bitcoin/commit/5400ef6bcb9d243b2b21697775aa6491115420f3.

[2] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of clients in Bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 15–29.

[3] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. 2017. Dandelion: Redesigning the Bitcoin Network for Anonymity. *POMACS* 1, 1 (2017), 22.

[4] Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Brad Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. 2018. Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees. *POMACS* (2018).

[5] Giulia Fanti and Pramod Viswanath. 2017. Anonymity Properties of the Bitcoin P2P Network. *arXiv preprint arXiv:1703.08761* (2017).

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

[7] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. 2013. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 127–140.

[8] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. 2013. Structure and anonymity of the bitcoin transaction graph. *Future internet* 5, 2 (2013), 237–250.

[9] Larry L Peterson and Bruce S Davie. 2007. *Computer networks: a systems approach*. Elsevier.

[10] Dorit Ron and Adi Shamir. 2013. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*. Springer, 6–24.