Working Title: Quantitative Privacy Analysis of using Kadcast for Transaction Broadcasting

John Smith
The Thørväld Group
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
jpkumquat@consortium.net

ABSTRACT

When discussing privacy properties of blockchains or Distributed Ledger Technologies, the focus of the analysis is often on the consensus layer of the blockchain. Another major factor for a comprehensive privacy analysis is the network layer, which defines how messages are passed between peers. On this layer many privacy sensitive informations like public IP addresses of users and connections between peers are used and can potentially be recorded by malicious parties. In this report we will conduct a privacy analysis of the structured P2P overlay Kadcast [Rohrer&Tschorsch], which can be used for efficient broadcasting in blockchain based technologies. The focus of our analysis is the usage of Kadcast for transaction broadcasting and the evaluation of an adversary's capability of linking observed transactions to IP addresses. We show that Kadcast is susceptible to network wide deanonymization attacks by botnets, corroborating the hypothesis that its efficient overlay comes at the cost of privacy. To mitigate the impact of such an attack, we propose to [prefix /besseres wort] the broadcasting phase with an anonymity phase, using the Dandelion spreading algorithm [Fanti].

1 INTRODUCTION

So called "Cryptocurrencies" and other blockchain based technologies are gaining increasing attention since the emergence of Bitcoin, both in the academic community and in media [].

These cryptocurrencies, like Bitcoin can be used analogous to fiat money to transfer "coins" from a sender to a receiver. In Bitcoin, new coins can be "mined" by investing computing power. Unlike a fiat money transaction, which is generally handled by a bank, there is no central instance that authorizes Bitcoin transactions. Instead Bitcoin is a peer-to-peer based system that is run publicly on the internet and allows all participants to achieve consensus on a single valid transaction history without the need for a trusted third party. Users can participate in the network via pseudonymous identities and there is no inherent link between a pseudonymous public identifier and a natural person, albeit deanonymization attacks can be performed to potentially link the public key of a user to an IP address or even a real name []. To store the transaction history, Bitcoin uses an append-only, distributed ledger (the so called "blockchain") and consensus on the state of the ledger is periodically reached by the participants of the network via a distributed consensus algorithm. [for detailed explanation refer to nakamoto paper] Since payments and money flow comprise of very privacy sensitive information, and everyone can join the Bitcoin peer-to-peer network and access the transaction history, the privacy of Blockchains has been an active research area since ...[]. There is a large amount of scientific literature [specifically] on the privacy of the consensus layer of blockchains [], describing weaknesses [e.g using heuristics to construct entity graph, ...] and algorithms to mitigate privacy issues [mixing, coinjoin, zk-snarks, ...] [bib].

Yet another important factor for a comprehensive privacy analysis is the network layer, which defines how messages are passed between participants of the network. Because of the open nature of public blockchains, any adversery can join the network and potentially gain access to privacy sensitive information on the networking layer, like the IP addresses of peers they are connected to. The gathered information can then be used to perform deanonymization attacks, e.g. to link transactions to public IP addresses [first described in Koshy].

One of the most important jobs of the network layer in blockchains is to handle broadcast messages, since both new blocks and new transactions are sent as broadcasts through the network. While Bitcoin started with a relatively naive approach of broadcasting new blocks and transactions in the network, by just iteratively spreading the messages to all nearby nodes, the network overlay of blockchains has since been revisited multiple times to optimize the message passing with regards to efficiency and privacy [Koshy, Bojja, Fanti, Rohrer, ...] and is continued to be researched and optimized [].

One [semi-]recently devoloped peer-to-peer overlay that can be used for efficient broadcasting in P2P networks is Kadcast. Kadcast is a Kademlia based network overlay that diverges from the naive and redundant approach of "gossiping", to a structured message propagation, leveraging the bucket logic introduced by Kademlia, and achieving a complete network coverage with minimal messages (excluding neccessary redundancy for stability reasons). [TODO vllt infografik einfügen wie kadcast/bucket logic funktioniert?] The initial application for Kadcast was to broadcast newly mined blocks. Because the direct link between IP addresses and transactions is already broken when a transaction is included in a block, the block propagation was optimized for efficiency only, without any concerns for privacy issues.

However, when using Kadcast to broadcast transactions instead, an adversary that is connected to the network can potentially perform deanonymization attacks and link transactions to the public IP addresses that issued them.

Since the travel path of messages in Kadcast is structured, and message redundancy is reduced to a minimum, the question was raised, what sorts of privacy implications using Kadcast as a broadcasting algorithm for transactions would have [R.].

To start answering this question, we want to use a quantitative approach and perform a deanonymization attack as it could happen in a real world scenario. Because of the lack of historical data, we use a [highly] abstracted network simulation to generate data which then can be used for the attack. The simulation setup, including constraints of our simulation and potential resulting implications

for the attack are described in [Section Simulation Setup].
We further model our adversary and describe the means of our attacker, explaining which simplifications we made to the attacker model, and why we can use such a simplified model without sacrificing much effectiveness of the attack [Ch. atk.model].
We will then shortly elaborate which metrics we applied to evaluate the impact of the attack [Ch. metrics].
In our evalution we will take a closer look at the results of our simulated attacks, and review if the privacy impacts of using Kadcast for transaction broadcasting align with our research hypothesis, that the structured overlay has [negative] effects on the privacy of the network.
In the last step of our evalution, we will try to mitigate the privacy issues we have observed previously by [prefixing /besseres wort] the broadcasts with an anonymity phase as described in [Dandelion].
We conclude our analysis by briefly summarizing our results and giving an outlook on where future research on this topic could be headed from this point on.
TODO Results + Outline

2 SIMULATION SETUP

For our simulation we used the discrete event simulation framework "simpy".
In our setup we do not incorporate network failures or change of any kind, especially we do not have any dropped and corrupted messages or failing nodes.
We run the simulation in two steps, starting with a setup phase, which lets the nodes discover the network and fill their respective buckets and after the network is stabilized we run the broadcast simulation phase.
Because we do not have any dropped messages, we set the redundancy parameter "beta" to 1, as it will always suffice to saturate the network.
We use latencies drawn at random from the iplane latency dataset[ref iplane] for all connections.

3 ATTACKER MODEL

The overall goal of the attacker in our setup is to deanonymize users in the network. More precisely, the attacker wants to map transactions to the public IP addresses that issued them.
The Kadcast protocol comes with inherit security properties that mitigate different kind of attacks and allow us to better define the means of an attacker if we assume those properties hold []. The factor that is most important for mitigating attacks on the network layer, is that participants are not allowed to choose their position in the overlay freely, and it is instead bound to the public IP address of a peer []. The IP to ID mapping is done via a cryptographic hash function we assume to be unbreakable. This makes it considerably harder for attackers to e.g. perform eclipse attacks [] or benefit in other ways from a freely chosen position in the network. [sybil.]
We currently only know of two ways a malicious party that disobeys the protocol could attack or observe the network differently than an "honest-but-curious" attacker.
Those are namely a denial-of-service attack and an attack in which the adversary tries to flood benign nodes with ping messages, in hope of hitting the time frame in which the benign node has at least one unresponsive peer, but has yet to refresh its peer list [eclipsing false friends]. In this case the attacker could potentially insert their own nodes into the peer list of the benign node.
Because of the assumed limited advantages byzantine nodes give an attacker and because our simulation setup doesn't account for any network change and turn, which would be needed to perform a pinging attack as described above, we exclude both of these attack vectors from our evalution and instead focus solely on an "honest-but-curious" attacker.

For our analysis we further assume the adversary does not take the network topology or their overlay-protocol knowledge into account and only uses information recorded by the spies, including aggregated and infered information.
This also means we assume the adversary does not use the broadcasting height attached to each broadcast message for the deanonymization attack. While we do not formally show the impact of not using that information, this assumption can at least be somewhat justified because the broadcasting height can theoratically be hidden or masked to a certain extend (e.g. by sending the same message at different/randomized heights).
Under the specified assumptions we found the most realistic scenario to be a botnet-like adversary that is part of the normal network overlay and is used to deanonymize other users in the network.
We therefore model the attacker as an entity that controls a certain share of nodes in the network (so called "spies"). These spies all behave indistinguishably to benign nodes from an outsider's perspective (i.e. they obey the protocol).
To perform a deanynomization attack, the adversary logs all messages their nodes receive, including timestamp and sender (IP address) for each recorded message. After the collection phase, the attacker feeds the aggregated information up to this point in time into an estimator that computes macro averaged precision and recall values over the whole dataset as defined in [SECTION METRICS].
As an estimator we use a simplistic so called "first-spy-estimator". This estimator maps every observed transaction to the first observed IP address that sent the transaction to any of the spies.

4 ANONYMITY METRICS

The result of each of our deanonymization attacks is a transaction to IP mapping, in which each observed transaction is mapped to exactly on IP address.
Our problem resembles a standard classification problem, as we try to assign each transaction to a class (node). Therefore we can use standard classification metrics to evaluate the effectiveness of our estimator. We decided to compute averaged precision and recall values over the whole mapping, as these are established and tested metrics for our use case [fanti et. al].
The recall value can be interpreted as "probality of detection" and the precision value relates inversely to the average plausible deniability per node. That means, the higher the recall, the more likely it is that a random transaction was correctly classified. The lower the

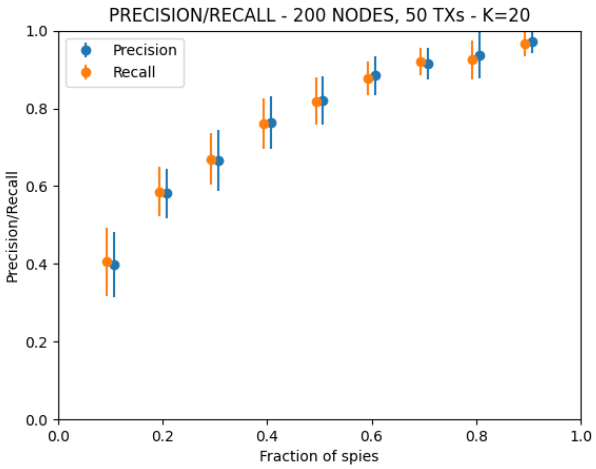

Figure 1: First Spy Estimator Kadcast

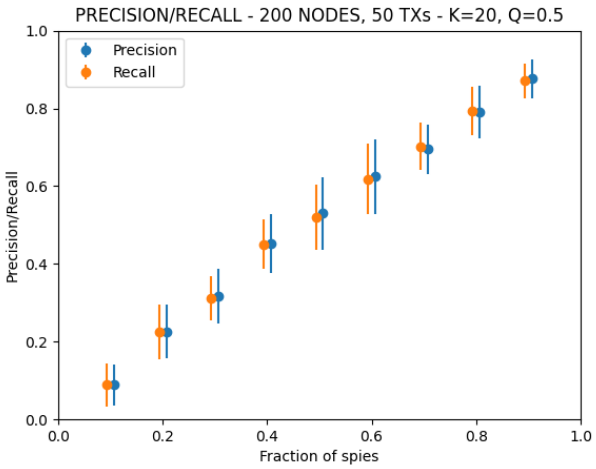

Figure 2: First Spy Estimator Kadcast with Dandelion

precision, the greater the average plausible deniability per node. The values are computed as follows: [hier formel einfügen]

5 EVALUATION

To evaluate the privacy properties of Kadcast we ran multiple simulations of transaction broadcasting with the Kadcast algortihm. Each simulation has the same amount of overall active nodes in the network and overall sent transactions. We run each simulation for different percentages of spies.
The transactions are sent only by benign nodes, and all transactions are randomly distributed among the sending nodes. Not each node does have to send a transaction and some nodes may send multiple transactions.

First, we took a look at how the first spy estimator would perform in a standard setting, using Kadcast with parameters used in "real world scenarios", i.e. a bucket size of 20.

The results can be seen in 1. We see, that the first spy estimator performs relatively well.
Even with an amount of as low as 20% of spies, the attacker can map over half of the sent transactions correctly. This is very far from the best case scenario [TODO which is?].

Just out of curiousity we wanted to test if the bucket size has any meaningful correlation to the performance of the estimator. We therefore repeated our experiment with common bucket sizes, the results can be seen in [TABLE1]. Different k values did not have a positive effect on the anonymity properties.

We further wanted to know if we can improve the anonymity by using dandelion spreading. We therefore implemented a simple version of the dandelion algorithm. It uses the "q" parameter as described in the original dandelion paper, to decide at each hop if the algorithm should stop the anonymity phase and start the broadcasting phase. The q value is the probability to leave the anonymity phase at a hop. We did not build a Hamiltonian circuit as described in the paper, but just used a random (loopfree) path through the graph for each transaction. Therefore our approach is a simplified version of dandelion, a little more akin to regular "diffusion-by-proxy". Nevertheless this approach has some observable positive effects on the anonymity of our network. The result for combining Kadcast with this simple implementation of dandelion, using the standard parameter $q = 0.5$, can be seen in 2. Compared to standard Kadcast spreading this is a massive improvement.
We ran the experiment again with different q values, the results can be seen in [TABLE2].
As Dandelion only adds a constant overhead to spreading [source dand1fanti], we think it would be a meaningful addition to Kadcast, especially if we take the bad results we have observed in our standard Kadcast experiments into consideration.

- auf bucket reihenfolge eingehen - k ändern - bucket reihenfolge ändern - std deviation/x samples/...

6 DISCUSSION

Our results indicate that the privacy properties of kadcast are not optimal.

[DISCUSS RESULTS HERE]

In our analysis we assumed the adversary does not take the network topology into account. It would be interesting to drop this assumption and do further research on what kind of information a less "naive" attacker could exploit.
For example, one could explore if it would be possible for an attacker to infer additional information about the bucket layout of sending nodes, and use this information to create a more sophisticated estimator.
If the attacker could somehow identify some nodes as being far away or close by (according to the XOR distance), they could e.g. weight their obvervations by proximity, as a close node has a higher chance to have a direct link to a node.

To really understand the bounderies of an attack, it would further be of interest to investigate the symmetry properties of the message spreading.
In general, it would be of great benefit if our results were reinforced either by a more realistic simulation that closer resembles real world scenarios, e.g. including failures, or by a formal modal that gives a better understanding of an attackers capabilites.

7 RELATED WORK

The privacy of blockchains is an intensively studied research field. Deanonymization attacks on different layers of the blockchain have been evaluated [], e.g. attacks on the consensus layer that try do deanonymize users by linking different transactions to a single source, even when using different public identifiers for each transaction [].
More recent research adapted general knowledge about P2P networks to the networking layer of blockchain based technologies and evaluates the privacy properties of the network overlays of public blockchains [fanti et. al, ...].

There are two papers on which most of our research is directly based on.
The first is the work on Kadcast by Rohrer & Tschorsch[], who proposed the Kadcast algorithm for broadcasting in P2P networks. While initially proposed for block propagation, we are more interested in using Kadcast for transaction broadcasting and the privacy implications of such an application. In that regard our work is a direct follow up to the questions that were already raised by Rohrer & Tschorsch[], and an attempt to investigate some of the potential privacy problems stated in the original paper.
While Kademlia itself is already leveraged by various blockchain based technologies, including Ethereum [], it isn't commonly used as a broadcasting algorithm. Therefore the privacy properties of such an application are not well understood yet and there exists neither a formal model nor an empirical evaluation of the anonymity of Kadcast.
The second paper which heavily influenced our work is Dandelion(++)[Fanti et. al], from which we drew several ideas, especially on how to assess the privacy of transaction broadcasting on the networking layer. This includes the general idea of the botnet-like adversary, using precision and recall as anonymity metrics and the first spy estimator [TODO vllt noch darauf eingehen, wo die sachen wirklich das erste mal vorgeschlagen wurden]

8 CONCLUSION

The goal of our research was to evaluate which kinds of privacy implications using the Kadcast algorithm for transaction broadcasting in blockchain based technologies would have.
To answer this question we simulated a network which uses Kadcast as a transaction broadcasting algorithm. To evaluate the privacy properties, we further simulated an attack by a botnet-like adversary that tries to link transactions to public IP addresses.
We launch a simple, but effective deanonymization attack using the so called "first-spy estimator", which classifies the first node that forwards a transaction to the botnet as the originator of said transaction.
We have shown that even such a simplistic attack performs relatively well [numbers] [compared to?] in our simulation [how about real world? results applicable?]. Our next step was to try to mitigate the observed privacy issues by combining Kadcast with a simplified version of the Dandelion Spreading algorithm introduced by [Fanti].
The experimental results show the clear trend that combining the Kadcast broadcasting phase with a Dandelion anonymity phase yields privacy benefits.
Since our simulation setup is limited and does not account for any node or message failures, the results of our evalution have to be assessed with caution. While we find it highly unlikely, that real world experiments would completely contradict our results, we made some strong assumptions in the attacker model based on what kind of attacks on Kadcast are currently known and our estimated impact of those attacks. We don't know yet, if there is a realistic attack that e.g. leverages the knowledge an adversary has about the overlay structure, so that they can perform better deanonymization attacks.
An interesting direction for further research on this matter would be to try to enhance the "first-spy estimator", taking into account the characteristic spreading properties of the Kadcast protocol, and maybe even formally prove to what extend an attacker could exploit this. Besides reinforcing our observations with a formal model, it would also be benificial to run a more sophisticated network simulation that incorporates network and node failures (honest or byzantine) and simulates the whole network stack.