Deanonymization in the Bitcoin P2P Network

Review 2

Name: Ashwath Venkataraman UFID: 5198-9461

Bitcoin has well established its prowess as privacy preserving financial system. However it is prone to securities vulnerabilities, specifically an attacker learning of the user's identity and as a consequence the user's transaction history. Bitcoin protocol is associated with the anonymity implications where the attacker uses a "**supernode**" to listen to transaction traffic, and use estimators to get the source IP addresses of each node. The so called "**eavesdropper adversary**" was able to associate IP addresses to pseudonyms with an accuracy up to 30%. Two types of flooding protocols are discussed and analyzed in this paper; trickle and diffusion spreading.

In **trickle** spreading which a gossip based protocol, source nodes choose random nodes uniformly at random (every 200ms). This is a variation of the round-robin gossip protocol, where 'd' neighbors of a node receive messages within 'd' timesteps. **Diffusion** spreading transmits messages to its peers with an independent exponential delay. The adversary tries to identify the source node, its port and IP address. The probability of detection is based on two estimator variants; First-timestamp estimator and maximum-likelihood estimator. **First-timestamp estimator (FT)** tells the first node to report to the attacker and is quite simple to implement. **Maximum-likelihood (ML)** estimator depends upon possible orderings of all observed timestamps.

In trickle propagation, the lower-bound of the FT estimator is identified, since for a large degree (that does not include source node), simultaneous reporting is uncommon, and hence probability of detection of the FT estimator is close to the lower bound. It is observed from the analysis that higher the degree, higher is the likelihood that a node reports to the attacker before the source.

In ML estimator, the set of all feasible orderings of the timestamps coming from each node as source, that satisfy rules of trickle spreading is taken, based on a proposed estimator scheme called **timestamp rumor centrality.** A simplified version called ball centrality is used that approaches optimal probabilities as 't' increases. It is observed that with an ML estimator, increasing degree would not reduce detection probability, and the adversary gains whatever it wants at small times t.

In diffusion based protocol, the FT estimator is similar to that of trickle propagation, and does not provide any order level gains on anonymity (the probability of detections approaches zero as degree of the tree increase). For ML estimator a lower bound is computed by analyzing a scheme called **centrality based estimator**. The number of nodes that reports to the adversary are counted by the estimator from each of the adjacent sub-trees of the nodes. An arbitrary node is chosen, from which the candidate node is equal in each subtree. It is observed from the analysis that (up to degree 5), that neither FT nor centrality based estimator, outperform each other.

The paper concludes by saying that both have near similar probabilities, with constants for trickle being ½ and for diffusion around 0.30. The results are represented in an asymptotic sense for degree d regular trees. Diffusion is not much of an improvement from the trickle based protocol.

Comments, Views, Suggestions

This paper aims to press on the fact that both the flooding protocols - trickle and diffusion spreading offer poor anonymity with respect to first-timestamp estimator and maximum-likelihood estimator.

The possible pitfalls of this paper might be related to the capabilities of the adversaries. This goes to say that there could be more evidence on what the attacker capabilities are in terms of the computational powers and whether it has access to public information. It is just defined as "supernode" which is quite abstract and more information on adversary would help relate more.

Results are computed 'asymptotically' in degree d. Asymptotic analysis may not be very convenient in many cases, as it is often not true in practice since in our case we use snapshots of a real Bitcoin network. The same could be said for 'empirical' probability detection values.

The network model assumes probability detection analysis for d 'regular' trees where 'd' is the degree. In practice a Bitcoin network creates a random topology of nodes, so this scenario might not work out for a real time Bitcoin P2P network.

A possible counter-measure would be to properly provide IP address level security like hiding, mixing or using VPNs to mask addresses properly to prevent hijacking or identification by adversaries.