An Exploration of Privacy and Anonymity in Bitcoin

CS203: Network and Distributed System Security

Christopher A. Wood
Department of Computer Science
University of California Irvine
Email: woodc1@uci.edu

Chris H. Vu
Department of Computer Science
University of California Irvine
Email: ChrisHVu@gmail.com

***Abstract*—TODO**

I. Introduction

Electronic commerce would benefit greatly from the existence of a complete secure, private, and anonymous form of digital currency that did not rely on trusted third parties or external financial institutions to manage transactions. Indeed, there has been significant research invested in this problem in recent years resulting in various forms of cryptography-based digital payment systems, or cryptocurrencies for short, such as DigiCash [**?**], ecash [**?**], HashCash [**?**], Namecoin [**?**], Peercoin [**?**], Litecoin [**?**], Ripple [**?**], and perhaps the most popular variant, Bitcoin [1]. Each of these schemes offer different tradeoffs of security, privacy, and anonymity, and as such have varying popularity among users. However, it is the distribtued, decentralized nature of Bitcoin that has led to its leading popularity to the general public and research communities.

More specifically, Bitcoin is distinguished from other solutions by the fact that it does not rely on trusted third parties. Specifically, the global and publicly accessible ledger which stores records of financial transactions is maintained by a widely distributed, peer-to-peer network of (untrusted) users. Even though each transaction is linked to the public key of a particular user via a digital signature, rather than their real identity, user privacy and anonymity are still at risk because public keys must be owned by specific users. This is true even if a user has multiple public keys and uses them with caution to deter attackers looking for such links. Consequently, the privacy of Bitcoin is an open problem and illustrates the difficulty in achieving a distributed form of cryptocurrency, i.e., one that does not rely on trusted third parties, and one that provides sufficiently useful characteristics such as privacy and anonymity.

Currently, techniques to address such privacy issues with Bitcoin are rather limited and include the use of Chaumian's entirely independent e-cash system [2], which relies on trusted third parties, or Zerocoin [3], which achieves privacy and anonymity at the protocol-level by working in conjunction with Bitcoin, among others. The former is not ideal for several reasons, the most significant of which is that it directly conflicts with the decentralized nature of Bitcoin. That is, reliance on trusted third parties is generally unfavorable if at all feasible. The latter technique is very young, having only been published in the past year, and is just now starting to gain considerable attention. Of course, there exists other academic efforts to further the cause for Bitcoin user privacy, including studies by Ron and Shamir [5] and Androulaki et al. [4], and we can expect to see similar work publishing in the coming years.

In this work we survey Bitcoin and related forms of cryptocurrency with respect to the security, privacy, and anonymity guarantees provided by each. We assess proposed solutions that have and have not been implemented in practice, and offer critical insight into the open problems and difficulties in achieving complete security, privacy, and anonymity with minimal resource consumption (e.g., bandwidth, computational cycles, etc.). We hope that this survey will motivate continued research on this critical problem that has the potential to change financial instutitions and forms of currency for future generations.

TODO: outline the sections here

II. Bitcoin Overview and Privacy Limitations

TODO: intro

A. Bitcoin Basics

Bitcoin is a distributed, decentralized form of cryptocurrency. Accordingly, this enables all (digitally signed) transactions between two parties to be conducted in a peer-to-peer fashion without the inclusion of a trusted third party, such as a bank or other financial institution. This form of decentralized exchange comes at a price, however, as there must be some way to prevent users from *double spending*, or using the same funds to simultaneously pay multiple parties. Bitcoin achieves this property by relying on its users to construct a history for every transaction that takes place in the system. If a majority of the users accept the validity of a particular transaction, or a set of transactions, the global history of the system is affirmatively updated and "confirmed" via a cryptographic hash digest that all users agree upon. This hash digest, referred to as a hash-based proof-of-work, is what constitutes the validity of the system. By the properties of the underlying hash function, the history of the system cannot be changed without breaking the function (i.e., finding collisions) or re-doing the proof-of-work, which is computationally infeasible for small groups of nodes. Therefore, so long as a majority of the Bitcoin users are honest, the system history is deemed correct and thus all signed transactions are verified, preventing double spending by potentially malicious users participating in direct, peer-to-peer transactions.

Unfortunately, while the above scheme is semantically correct and provides strong guarantees that all financial transactions are valid, there are inherent limitations in the amount of user privacy and anonymity that can be achieved in Bitcoin. In order to adequately define these limitations, we first describe how Bitcoin transactions are generated and how the system history is maintained. For simplicity, consider the scenario in which user A wants to send N BTCs (Bitcoins) to user B. Rather than identify users by name, Bitcoin uses *addresses* that are tied to specific users to use in such transactions. Denote addr_A and addr_B as the addresses of users A and user B used in this transaction. It is often convenient to think of Bitcoin addresses as public keys pk_A and sf_B, and as such there are corresponding private keys, which we denote as sk_A, and sk_B, respectively.

Structurally, a transaction T is a tuple comprised of the *source* transactions which supplied the funds necessary to make this transaction, denoted as source, the (public) address of the recipient, addr_B, the amount of BTCs to send, N, and a digital signature of these three properties, $\mathsf{Sign}_{\mathsf{sk_A}}(\mathsf{source}, \mathsf{pk}_B, N)$. In other words, we have

$$T = (\mathsf{source}, \mathsf{pk}_B, N, \sigma),$$

where $\sigma = \mathsf{Sign}_{\mathsf{sk_A}}(\mathsf{source}, \mathsf{pk}_B, N)$. Note that this signature is embedded in T so that any other Bitcoin user may verify the validity of the content using pk_A. Also note that source need not be a single transaction; user A is free to use multiple transactions in order to fund their transaction to B. In addition to the N BTC transfer from A to B, there is often C BTC amount specified in the transaction for a particular address, where C denotes the amount of change that will be given to this address as a result of the transaction. It is not required that the address to which C is addressed is the same as the address of A, though this often happens in practice. Figure 1 illustrates the input and output relation of our transaction from A to B, and Figure 2 illustrates the steps used in constructing this transaction. Note that, in both cases, source is comprised of two transactions T_1 and T_2, and the resulting transaction is denoted as T_3.

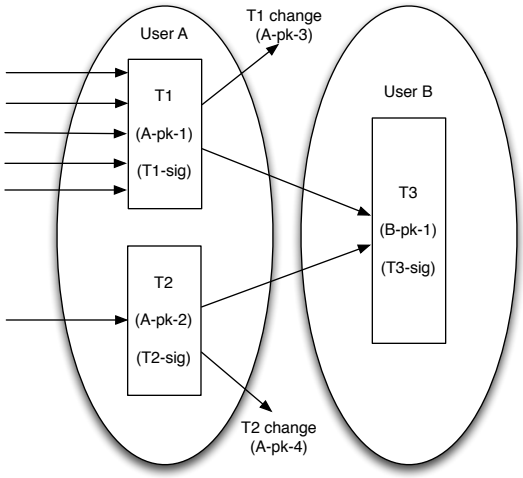

Fig. 1. Visual depiction of the input and output elements of a transaction from user A to user B.

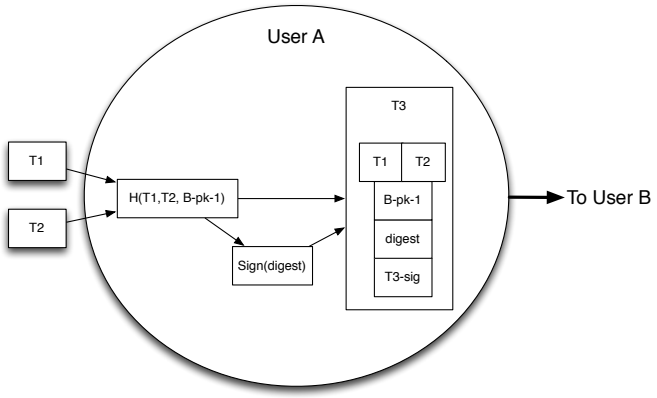

Fig. 2. Visual depction of the steps to create a transaction T_3 from user A to user B using two input transactions, T_1 and T_2.

After a transaction has been created, it is broadcasted in the network. In order to prevent double spending, nodes must confirm this transaction and append it to the chain of accepted transactions in the system's history. This procedure is based on the aforementioned proof-of-work, which works as follows. Bitcoin miners will collect unconfirmed transactions into a buffer, along with the longest chain of system-wide accepted transactions, and compute a Merkle hash of the transactions and digest of the chain. The output digest of this Merkle hash, referred to as the challenge c in the proof-of-work protocol, is then used to find the proof p. Together, c and p have the property that, when concatenated and hashed using a cryptographically strong collision-resistant hash function H, the leading B bits of the output $x = H(c||p)$ are all 0. That is, $x = \{0,1\}^B\{0,1\}^{256-B}$. Given the collision resistant properties of H, finding a valid proof p for the challenge c is comptuationally difficult. Figure 3 illustrates the construction of c and p using a previously confirmed block chain B.

Once a miner finds a proof, it is broadcasted to the other nodes in the network along with the input transactions used by the miner, who can then easily recompute the challenge c and verify the correctness of p. Once verified, this new transaction "block" is appended to the block chain which the miner used in finding the proof. Figure 4 illustrates a snippet of the block chain, where the challenge c is the digest of the previous block and the proof p are embedded in each block. Miners will continually use the longest block chain to gather and verify transaction blocks. Since there is a particular subset of BTCs in each transaction that are paid to the miner who provides the proof-of-work for a block containing that transaction, referred to as the transaction fee, miners are financially incentivized to collect more transactions into a block and continually "mine" for valid proofs-of-work.

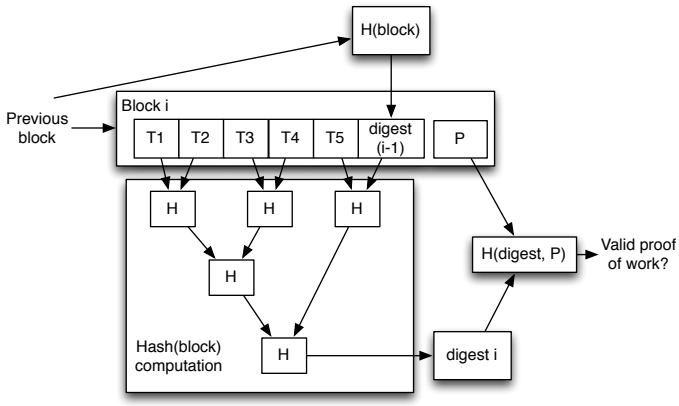

Fig. 3. Proof-of-work computational procedure using the transactions of a block, the digest of the previous block, and the sampled proof p.

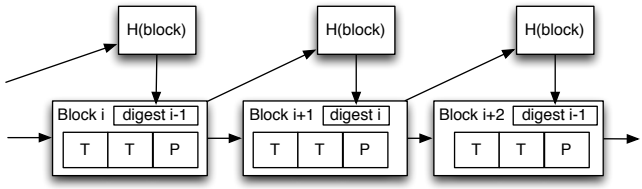

Fig. 4. A snippet of a Bitcoin transaction block chain, illustrating the groupings of transactions into a blocks, the declaration of the proof of the work p, and the digest of the previous block linking the blocks together.

B. Privacy Limitations

TODO

III. Privacy and Anonymity

TODO: definitions, summary of security model and attackers, and goals of the attackers

IV. Attacks, Implications, and Proposed Solutions

- transaction graph attacks
- passive network-layer attacks
- statistical attacks

V. Open Problems

TODO: identify issues in the bitcoin protocol that need to be addressed, and propose some solutions here

VI. Privacy-Preserving Alternatives

TODO: summary of related work and different currency systems (e.g., zerocoin) that achieve privacy

References

[1] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *Consulted* 1 (2008).

[2] David Chaum. Blind Signatures for Untraceable Payments. *Crypto* 82 (1982).

[3] Ian Miers, Christina Garman, Matthew Green, Aviel D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. *IEEE Symposium on Security and Privacy* (2013).

[4] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating User Privacy in Bitcoin. *IACR Cryptology ePrint Archive* 596 (2012).

[5] Dorit Ron and Adi Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. *IACR Cryptology ePrint Archive* 584 (2012).