Privacy in blockchain: a survey

Sung-Shine Lee
University of Waterloo
s469lee@uwaterloo.ca

ABSTRACT

As proposed by Satoshi Nakamoto, blockchain is a public distributed ledger that is trustable by verifying cryptographically. In a blockchain design, cryptographic tools such as hash functions provide integrity and the decentralized design provide availability. However, often misunderstood to be untraceable, account information on Bitcoin, the original blockchain design, is public and can be viewed by anyone. To improve privacy, researchers have proposed solutions that are either compatible to the existing blockchain or a novel blockchain mechanism. On the other hand, securing the data and preserving the privacy of participants of smart contracts is essential for the adoption of decentralized applications. This paper summarizes the techniques and ideas used on the blockchain system to preserve privacy of participants.

KEYWORDS

Blockchain, Privacy, Security, Mixer, Smart Contract

1 INTRODUCTION

Since Satoshi proposed Bitcoin[26], the underlying technology blockchain, a decentralized public ledger, has demonstrated its capability of being a valid currency and its potential to shape future industries with its decentralized paradigm. The blockchain is maintained by trustless participants called miners running a distributed consensus protocol. For the system to be secure, it is required that no adversary can control a significant portion of the resource (such as computational power[26] or crypto assets[20] defined by the protocol. When running the protocol, miners will not only collect transaction information but also execute scripts that are specified in the transaction and update the blockchain accordingly.

The system can be viewed as a service that offers integrity and availability in a decentralized fashion. However, Bitcoin, often being misunderstood as an anonymous payment system, does not provide confidentiality due to its design nature. Specifically, to make a transaction, one has to broadcast all information about the transaction including the sender, the receiver, the amount, and other metadata. The information is then collected by miners who will publish it permanently on the blockchain. In this case, pseudonymity is offered by the system since accounts are represented by public keys thus not directly linked to the user identity. However, the whole transaction history of a given account is visible to the public.

While a user can have multiple accounts, it was shown by [30] to be possible to link the accounts under the same identity by analyzing their transaction history. Moreover, by matching on-chain transaction history with off-chain data, an adversary can connect accounts to the person in the real world[18]. There are several companies such as Chainalysis that provide tools to analyze and track account activities on the blockchain. Although it seems beneficial to track the currency and thus preventing its usage in the black market, these tools can be used to spy on regular users and thus raise serious privacy issues. Various proposals have been introduced to counteract the loss of privacy.

These techniques mainly fall into two categories:

(1) Add-on Services. In a blockchain that is not privacy-preserving and does not plan to update the entire network, developers proposed add-on services that are compatible with the blockchain. These schemes often require additional participants such as a facilitator or other peers.
(2) Novel blockchain design. Modifying the blockchain system directly to support obfuscation of transaction removes the need for additional participants.

The design of Bitcoin has also inspired other blockchain-based cryptocurrencies that are not only used for financial purposes. While the scripting language on Bitcoin is not Turing-complete and limited in usage, Ethereum[12], proposed by Vitalik in 2013, has designed the scripting language in Ethereum to be Turing-complete. Users of the network are able to run arbitrary programs, also known as smart contracts, on the blockchain, creating decentralized applications.

Despite the potential of smart contracts, note that current implementations of the technologies lack privacy. Similar to making a transaction in Bitcoin, when a user tries to invoke a smart contract, all inputs and the smart contract itself are broadcasted to the network. The lack of privacy hinders the cryptocurrency technology from being adopted by essential but privacy-sensitive applications such as those in finance or health.

2 METHODOLOGY

In this survey, I briefly summarize the main techniques that either are widely used or showed promising potential. The survey started from the white papers of privacy centered cryptocurrencies such as Zcash[19], Monero[33], and DASH[16] which led me to the academic papers of privacy-preserving techniques. Cryptocurrencies news sources[4] and developers forums [6][2] are also used to observe which techniques are gaining traction and identify those that are stirring discussion among the developers.

3 CRYPTOGRPHIC TOOLS

3.1 Zero-Knowledge Proofs

Zero-Knowledge proof is a method to prove the possession of a knowledge without revealing the actual knowledge. The zk-SNARK[8], zero-knowledge Succinct Non-Interactive Argument of Knowledge, is a variant that uses Elliptic Curve Cryptography. It is popularized by ZCash[19] and had been adopted by Ethereum lately in the Metropolis(Byzantium) update[3].

One of the criticism of the mechanism is that computation of zk-SNARK requires a circuit that is generated by a group of trusted third party. Multi-party computation(MPC) is used to generate the circuit so that no member has the whole set of parameters. As long as at least one of the members in the group discards the secret, then the system is secure. Because of the heavy computational cost in the MPC process, the parameter of ZCash involved only 6 people to setup which led to serious doubts[5]. Using a recent method[10] that scales the MPC process to hundreds of people in the process, all participants have to collude to break the system, thus reducing the risk significantly.

Ben-Sasson et al. proposed zk-STARK[7], an alternative zero-knowledge system, that removes the need of trusted third party completely. Compared to zk-SNARK, zk-STARK uses hash function only so that it is resistant against quantum computers and safer since it relies on simpler cryptographic assumptions. Unfortunately, while the size of its proofs is asymptotically efficient, in practice they are around 1,000 folds larger than those of zk-SNARK.

Bulletproof[11], authors include Dan Boneh and Greg Maxwell, is another recent advancement in the family of zk-system that also don't require a trusted setup. Although part of the solution is based on Discrete Logarithm problem which is not resistant against quantum computers, it produces efficient proofs that is logarithmic with respect to the witness size. The paper has also shown that it is a competitive technique in applications such as confidential transaction.

3.2 Ring Signatures

Group signature[14], first proposed by Chaum and van Eyst, is a cryptographic tool that allows members of a group to create the digital signature of the group under the supervision of a trusted third party. Rivest et al. presented Ring signature[29], a scheme that has the same ability but removes the need of the trusted third party. One of the important subsequent works is the traceable ring signature [17] which can detect and reveal the identity of the member who signed two messages with identical tags.

3.3 Blind signature

Chaum [13] presented a method to have a message signed but not revealed to the signer. Before the message is sent to the signer, it is passed into a function that blinds the message with a secret parameter b. Then, the signer signs and returns the blinded message. Finally, the blinded message can be unblinded by proving the secret parameter b.

3.4 Pedersen Commitments

Commitments allow parties to commit to secret values without revealing the secrets. Once the commitment is made, other parties can detect when the committed party try to revoke or lie about the commitment. Pedersen commitments [27] are a special form of commitments that is additive homomorphic.

4 ADD-ON SERVICES

4.1 One-time address

Since transaction history is traceable through the address, an observer cannot infer anything from a newly generated address that has no previous records. Therefore, using a new address whenever receiving funds help mitigate the information leakage by hiding the identity of the receiver. Only when the receiver spends the coin, an observer can try to infer the identity.

4.1.1 Naive implementation. The simplest way of doing this is to have the fund receiver to generate a new address and transmit it to the sender securely. A reasonable example would be a website that generates the key pair in the back-end and display the public-key to the sender while stores or sends the private key to the fund receiver.

4.1.2 Stealth Address. Using Elliptic Curve Diffie-Hellman, Peter Todd proposed stealth addresses[1] which enables the sender to generate one-time address for the receiver, removing the need of communication. First, the receiver generates a parent key pair and publishes the parent public key as its stealth address. The sender will then randomly sample a nonce and generate a one-time address for the receiver with the published stealth address. The transaction is then published with the nonce. To receive a fund, the receiver has to scan the blockchain to try to generate a one-time secret key whenever a transaction with a stealth address and nonce is detected.

4.2 Mixer

One way to make tracking difficult is through mixing: a service that redistributes the coins among multiple users with unlinkable pattern. The service will first receive the fund from a user and mix the coins with those from others; It will then split the fund into several transactions that returns it to different addresses that belongs to the user. The anonymity of the service relies on the number of users and cryptocurrencies that are mixing.

4.2.1 Naive Mixer. The naive implementation of this scheme is to have a centralized entity to handle the funds; however, the trusted third party required by the scheme introduces severe risks. The coins are still traceable to the service provider if they decided to log the mapping of fund redistribution. Most importantly, nothing prevents the service provider to steal the funds. To alleviate the logging problem, users can mix the funds through several service providers so that the funds are only traceable if all the mixer services collude.

4.2.2 Peer-to-Peer Mixer. To remove the trusted third party, users can cooperate and mix the coin among themselves. Coinjoin[23], proposed by Gregory Maxwell, uses the technique that takes advantage of the design of Bitcoin which allows multiple inputs and multiple outputs in the same transaction. The downside of Coinjoin is that it still requires the node that is in charge of mixing to be trusted. CoinShuffle[31] by Ruffing et al. eliminates the existence of such node at the cost of communication overhead. Both of the proposals suffer from the fact that the coinjoin transaction needs to be signed by all participants of the mixing process and there might not be enough participants at a specific time to achieve anonymity.

4.2.3 Accountable Mixer. While there are already protocols that prevent theft, the idea of having the mixer to be accountable is noteworthy. The mixer proposed in Mixcoin[9] can generate a digitally signed warranty to the client so that the client can prove that when the mixer steals the coin. The subsequent work, BlindCoin[32], uses blind signature to make it impossible for the mixer to trace the input and output of a user.

4.3 Payment Hub

5 NOVEL BLOCKCHAIN DESIGN

5.1 Zerocoin & Zerocash

Zerocoin extends the blockchain scripting language and introduces an e-cash system within the blockchain of a cryptocurrency. The mint operation allows users to convert the underlying currency that is unprotected to a sender-protected e-cash coin using a randomly generated secret. The user can spend the e-cash coin when presented with a zero-knowledge proof of the coin exists in the available list and that the user possesses the secret that mints the coin. The anonymity is based on the amount of available minted coins in the e-cash system. Therefore it can be viewed as a decentralized mixer on the blockchain. However, the Zerocoin protocol's functionalities are limited in 3 ways: (1) Zerocoin has fixed denomination, and therefore it is inconvenient to use. (2) When users spend the coin, it converts into the underlying cryptocurrency.(3) Only the information about the sender is protected, the receiver and the amount are still public.

Zerocash, the subsequent work of Zerocoin, has improved on the drawbacks by introducing a form of zero-knowledge proof called zk-SNARK. Instead of receiving the underlying currencies, participants can accept the e-cash directly. This retains privacy of the participants; for an observer, the amount and the receiver of the transaction is hidden.

5.2 CryptoNote

Cryptonote[33] is a proposal by Nicolas van Saberhagen. Before a transaction is made, the sender will first find other accounts that have the same amount of funds and construct a ring signature with such group. The potential double-spending problem due to the obfuscation of sender is solved by using a form of traceable ring signature that embeds the key image, the hash of the sender's private key, as its tag. The traceable ring signature technology then is able to detect when the entity is trying to double spend the coin.

6 PRIVACY OF SMART CONTRACT

While smart contracts can be seen as programs that provides integrity and availability and is well suited for financial applications, their potential is currently being hindered by the lack of privacy. Current design allows every participant in the network to see the person who invoked the smart contract, the inputs of the program, and the content of the smart contract.

6.1 Enigma

Enigma[34], a project by a team at MIT, is a system that builds on top of blockchain to store and process data securely. The key problems that Enigma tries to solve are scalability and the privacy of data storage and computation. The authors observe that although the computations and data-storage on blockchain are decentralized, it is not distributed. The redundancy is introduced to ensure the blockchain system behaves correctly, however, not all data storage and computation needs to be done on the blockchain. With this insight, Enigma separates the main data storage and private computation from the blockchain.

The Enigma system includes the following three components:

- Public ledger: the blockchain serves as a public ledger that performs public computation; records proofs of correct computation; manages access-control and identities; and stores references to off-chain data. The cryptocurrency that works on the blockchain is also used to reward honest and beneficial behavior and punish malicious activities. All participants of the network that perform computation or data storage have to make deposits so that the system could punish malicious behaviors by confiscating them.
- Off-chain storage: Private data is encrypted and stored inside the off-chain storage using distributed hash table (DHT) that uses a variation of Kademlia DHT protocol[24] to provide secure communication channels between nodes.
- Secure multi-party computation (MPC): Enigma uses SPDZ[15], a MPC protocol that ensures correctness in the existence of malicious parties, to compile the computation process into a circuit. The circuit can take encrypted inputs and produce an encrypted output which only the private key holder can decrypt.

Currently, Enigma is a closed source project and the implementation details of the Enigma system has not been released.

6.2 Hawk

Ahmed et al. [21] observed that the execution of a smart contract was public since the execution message and its inputs have to propagate through the network; therefore, they proposed a framework named Hawk that ensures privacy for smart contracts. In addition, a formal model is created and used to verify the cryptographic protocols within the Hawk framework.

Traditionally, a smart contract is fully public and is being executed by all miners on the network who collect the execution message. In the Hawk framework, the execution is facilitated by introducing a manager, a minimum trusted third-party, that executes the code in private and submits only the result on the blockchain. The authors highlighted that the manager is not to be thought of as a trusted third-party; instead, it is important that the execution process will not be affected even if the manager deviates from the protocol.

In the Hawk framework, a smart contract is composed of a private section and a public section. According to the contract content, the compiler will generate three different programs that will run cryptographic protocols between users, the manager, and the blockchain. During the process, users will first commit an encrypted value to the manager. After the manager collects all commits, users will then reveal their input values to the manager with a proof. Finally, the manager executes the code and submits only the result to the public blockchain. The authors suggest that the steps above ensure input independent privacy so that it is not possible for a

user to learn othersâĂŹ input before committing even when they cooperate with a malicious manager. In other words, the manager is only being trusted to keep these inputs private after users reveal their value.

In addition to the proposed Hawk system, one of the salient contributions of this paper is to provide the first formal model to examine the performance and security of a smart contract system. Along with the theoretical results and the general framework, Ahmed et al. implemented several example Hawk programs and performance evaluation on the following metrics: on-chain computation time, the size of on-chain public parameters, the cost and time for manager computation, and user computation.

7 OPEN PROBLEMS & FUTURE RESEARCH DIRECTIONS

7.1 cryptographic tools

While Bulletproof identified confidential transactions as one of its applications, it is interesting to see whether the technique can be used to construct protocols that hide the sender and the recipient more efficiently.

With quantum computer just around the corner, it might be important to identify quantum resistant cryptographic primitives that provide the same functionalities.

7.2 Add-on Services & Novel Blockchain Design

There are several empirical research that investigate the anonymity provided by ZCash[28] and Monero [22] [25]. However, the anonymity that ZCash and Monero provided is not on the network level. While it is recommended to use Tor on the official site of ZCash to avoid IP linkability, it would be interesting to understand the percentage of users who followed the instructions to achieve anonymity in practice.

7.3 Smart Contract

While Hawk and Enigma have provided an initial idea of how to discuss privacy issues of smart contract system, it is far from complete. Both systems focus on privacy of the input, data, and computation of the smart contract. However, the obfuscation of the smart contract itself is not discussed. Moreover, the identities of the participants are not hidden.

REFERENCES

[1] [n. d.]. [Bitcoin-development] Stealth Addresses. ([n. d.]).
[2] [n. d.]. bitcoin forum. ([n. d.]). https://bitcointalk.org
[3] [n. d.]. Byzantium HF Announcement. ([n. d.]). https://blog.ethereum.org/2017/10/12/byzantium-hf-announcement/
[4] [n. d.]. Coindesk. ([n. d.]). https://www.coindesk.com
[5] [n. d.]. Zcash and the Art of Security Theater. ([n. d.]). https://www.coindesk.com/defending-zcash-blockchain-art-security-theater/
[6] [n. d.]. zCash Forum. ([n. d.]). https://forum.z.cash
[7] Eli Ben-Sasson, Iddo Bentov, Ynon Horesh, and Michael Riabzev. 2017. Scalable, transparent, and post-quantum secure computational integrity. *Manuscript.(2017). Slides at https://people. eecs. berkeley. edu/˜ alexch/docs/pcpip_bensasson. pdf* (2017).
[8] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. 2012. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 326–349.
[9] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. 2014. Mixcoin: Anonymity for Bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*. Springer, 486–504.
[10] Sean Bowe, Ariel Gabizon, and Ian Miers. 2017. Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model. (2017).
[11] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. [n. d.]. Bulletproofs: Short Proofs for Confidential Transactions and More. ([n. d.]).
[12] Vitalik Buterin et al. 2013. Ethereum white paper. *GitHub repository* (2013).
[13] David Chaum. 1983. Blind signatures for untraceable payments. In *Advances in cryptology*. Springer, 199–203.
[14] David Chaum and Eugène Van Heyst. 1991. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 257–265.
[15] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P Smart. 2013. Practical covertly secure MPC for dishonest majority–or: breaking the SPDZ limits. In *European Symposium on Research in Computer Security*. Springer, 1–18.
[16] Evan Duffield and Daniel Diaz. 2014. Dash: A PrivacyCentric CryptoCurrency. (2014).
[17] Eiichiro Fujisaki and Koutarou Suzuki. 2007. Traceable ring signature. In *International Workshop on Public Key Cryptography*. Springer, 181–200.
[18] Steven Goldfeder, Harry Kalodner, Dillon Reisman, and Arvind Narayanan. 2017. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *arXiv preprint arXiv:1708.04748* (2017).
[19] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. 2016. *Zcash protocol specification*. Technical Report. Tech. rep. 2016-1.10. Zerocoin Electric Coin Company.
[20] Sunny King and Scott Nadal. 2012. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* 19 (2012).
[21] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 839–858.
[22] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. 2017. A traceability analysis of moneroâĂŹs blockchain. In *European Symposium on Research in Computer Security*. Springer, 153–173.
[23] Greg Maxwell. 2013. CoinJoin: Bitcoin privacy for the real world. In *Post on Bitcoin forum*.
[24] Petar Maymounkov and David Mazieres. 2002. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*. Springer, 53–65.
[25] Andrew Miller, Malte Möser, Kevin Lee, and Arvind Narayanan. 2017. An empirical analysis of linkability in the Monero blockchain. *arXiv preprint arXiv:1704.04299* (2017).
[26] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
[27] Torben Pryds Pedersen. 1991. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual International Cryptology Conference*. Springer, 129–140.
[28] Jeffrey Quesnelle. 2017. On the linkability of Zcash transactions. *arXiv preprint arXiv:1712.01210* (2017).
[29] Ronald L Rivest, Adi Shamir, and Yael Tauman. 2001. How to leak a secret. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 552–565.
[30] 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.
[31] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. 2014. CoinShuffle: Practical decentralized coin mixing for Bitcoin. In *European Symposium on Research in Computer Security*. Springer, 345–364.
[32] Luke Valenta and Brendan Rowan. 2015. Blindcoin: Blinded, accountable mixes for bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 112–126.
[33] Nicolas Van Saberhagen. 2013. Cryptonote v 2. 0. (2013).
[34] Guy Zyskind, Oz Nathan, and Alex Pentland. 2015. Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471* (2015).