Generic Privacy Preserving Private Permissioned Blockchains

Frédéric A. Hayek
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne
LIMOS
Clermont-Ferrand, France
frederic.hayek@uca.fr
0000-0003-1083-0625

Mirko Koscina
be ys Pay
Clermont-Ferrand, France
0000-0003-4158-6952

Pascal Lafourcade
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne
LIMOS
Clermont-Ferrand, France
0000-0002-4459-511X

Charles Olivier-Anclin
be ys Pay
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne
LIMOS
Clermont-Ferrand, France
0000-0002-9365-3259

***Abstract*—Private permissioned blockchains are becoming gradually more sought-after. Such systems are reachable by authorized users, and tend to be completely transparent to whoever is allowed to interact with the blockchain. In this paper, we mitigate the latter. Authorized users can now stay unlinked to the transaction they propose in the blockchain while being authenticated before being allowed to interact. As a first contribution, we developed a consensus algorithm for private permissioned blockchains based on Hyperledger Fabric and the Practical Byzantine Fault Tolerance consensus. Building on this blockchain, five additional variations achieving various client-wise privacy preserving levels are proposed. These different protocols allow for different use cases and levels of privacy control and sometimes its revocation by an authority. All our protocols guarantee the unlinkability of transactions to their issuers achieving anonymity or pseudonymity. Miners can also inherit some of the above privacy preserving settings. Naturally, we maintain liveness and safety of the system and its data.**

***Index Terms*—Privacy, Byzantine Fault, Blockchain, Signature.**

I. Introduction

Blockchains are replicated synchronized databases shared across a trustless network. They implement either a Nakamoto consensus, meaning having miners compete to create the next block with a longest-chain-win rule[1] or a BFT consensus. Privacy of blockchains is either nonexistant or characterized by pseudonymity or anonymity. Pseudonymity is having entities identified by pseudonyms, but not necessarily being able to link the pseudonyms to the identities behind them. A good example of pseudonymity is Bitcoin [2], where users are identified by their public keys, and it is generally infeasible to trace the public key to a real world identity. Note that in Bitcoin a user can create as many pseudonyms as they wish. Anonymity, on the other hand, is when it is infeasible to link anything. One such example is Monero's usage of ring signatures [3] in order to anonymize the sender's identity that is hidden among a set of other identities.

For distributed ledgers, we consider private/public to reflect user restriction, and permissioned/permissionless to reflect mining restriction. Private distributed ledgers must naturally restrict their usage to legitimate users only, and permissioned distributed ledgers must restrict their mining to legitimate miners only. These restrictions seem, at first, incompatible with privacy, giving rise to a dilemma: how to restrict private distributed ledgers usage while ensuring user privacy? And how to restrict permissioned distributed ledgers mining to legitimate miners while ensuring miner privacy?

Most known privacy works on blockchains aim at improving public and permissionless blockchains, such as Monero (built on Cryptonote [4]), Zerocoin [5], Zcash [6]. Identity confidentiality improvements of private or permissioned blockchains is usually complex or application-specific (thus not general-purpose) [7]–[12].

[1]Length here does not necessarily mean number of blocks (as it did it at Bitcoin's conception). Length can for example denote difficulty in the current Bitcoin protocol, or weight in Ethereum's GHOST algorithm implementation [1].

II. Our Blockchain SignCons

Our generic SignCons blockchain is highly inspired by Hyperledger Fabric's modular blockchain framework [13], [14]. It comprises: (1) a finite set Clients of authorized entities, (2) a finite set Miners of authorized miners divided into two categories: (2a) a set Endor of entities who endorse transactions of cardinal TotEnd and (2b) a set Order of entities who achieve consensus of cardinality TotOrd. The process is outlined in Figure 1. It consists of six steps described hereafter:

(1) **Transaction Proposal.** The user signs their operation and sends it to the endorsers, using algorithm $TxProp$.
(2) **Transaction Endorsement.** Each endorser peer verifies the transaction, and if valid, signs it (as a sign of endorsement) and sends it back to the user, with $TxEndors$.
(3) **Broadcasting to Consensus.** The client collects the endorsements and when enough are received, sends them to the orderers, with $TxBrodOrd$.
(4) **Block Proposal.** The orderer leader, upon receiving enough endorsed transactions, creates a block and proposes it to the other orderers, with $TxComOrd$.
(5) **Block Preparation.** The other orderers, upon receiving the block from the leader, check it, and validate it by signing and broadcasting it to the other orderers, with $Prepare$.

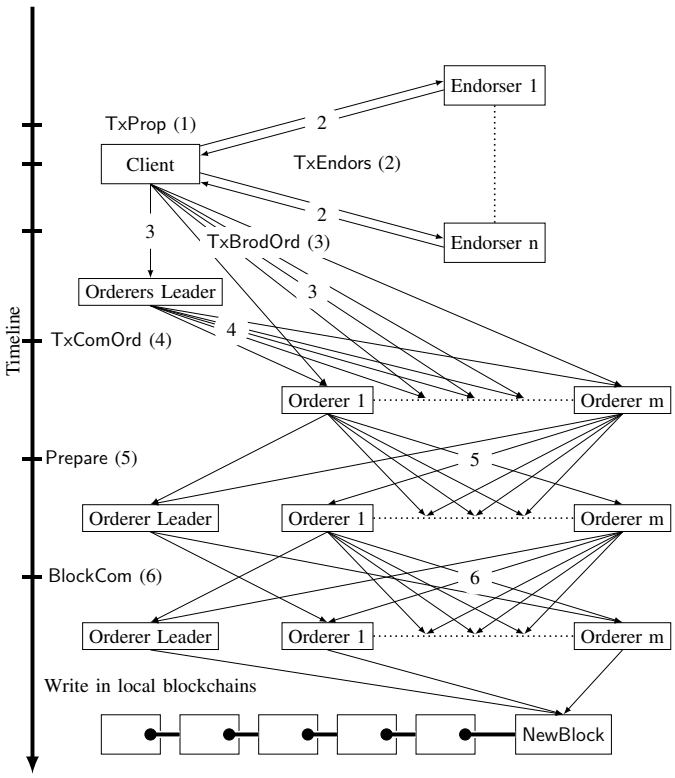

Fig. 1: Transaction Flow of our BFT Consensus.

(6) **Appending Block to Blockchain.** Each orderer, upon receiving more than $\frac{2\mathsf{TotOrd}}{3}$ block validations, then each orderer appends the new block to their local version of the blockchain, and broadcasts the new block to users, with $BlockCom$.

The orderers' consensus is highly inspired from PBFT [15], to which we have added endorsers and modified the messages' contents: steps 3 through 6 in Figure 1. The orderers mimic the view change protocol described in [15] for orderer leader's update in order to ensure safety and liveness. Note that what we call leader, they call *primary*. To quickly summarize the view change: if the leader is inactive or misbehaving, then another orderer can initiate a view change; it stops confirming new blocks, and proposes to the set of orderers to change leaders (as per a predetermined schedule). When enough of orderers reply positively, the view change happens.

Network Model: We assume an asynchronous distributed system where nodes are connected by a network who may fail to deliver messages, delay them, duplicate them or deliver them out of order. We allow for the adversary to coordinate faulty nodes, delay communication, or delay correct nodes but not indefinitely.

III. Privacy Preserving Blockchains

By only modifying signature schemes (and their verification) in the blockchain introduced in Section II, we can get different privacy properties for users, endorsers and orderers. Privacy preserving settings can be achieved independently for any of the defined roles.

A. User Privacy

All constructions for user privacy replace the regular signature scheme in (1) **Transaction Proposal** with a different kind of signature.

1) User Anonymity:

a) Based on Blind Signatures: Suppose the existence of a trusted authority. Let us change the signature scheme in (1) **Transaction Proposal** to a blind signature [16] as follows: the user authenticates himself to the authority which verifies that the user has the right to transact on the blockchain. If that is the case, the authority blind signs the user's transaction. And the transaction's issuer is considered "authenticated" by endorsers (who must naturally verify a blind signature instead of a regular signature scheme). This variant of SignCons is called BlindCons.

b) Based on Group Signatures: Suppose the existence of a trusted authority. Let us change the signature scheme to a group signature [17] with the authority as group manager. This variant of our scheme, called GroupCons, enables any registered user to sign hiding amongst the group of authorized users; and it allows the authority, if need be, to reveal a message's signer. We assume that authorized users are registered with the authority and that a public record of all of them is available. Hence, all keys are generated through a protocol with the authority and registered in the Clients record. Compared to the blind signature construction, group signatures also limit the computational load on the authority as it no longer needs to execute its part of the blind signature protocol for each new transaction, and only needs to generate the group once.

c) Based on Ring Signatures: Let us change the signature scheme to a ring signature scheme [3] where the ring consists of (potentially all the) authorized users. In this variant, RingCons, no authority is required and transactions are unlinkable to their issuers. Let $U = u_1, \ldots, u_n$ be the set of authorized users. Suppose u_1 wants to make a transaction: u_1 signs the transaction using a ring signature in the name of U (or in the name of a subset of U if U is too big). That way the endorsers verifiy the ring signature and can thus know that it was indeed someone of U that produced the signature without being able to know who.

2) User Pseudonymity:

a) Based on Linkable Group Signatures: With a trusted authority's presence, we can use linkable group signatures [18], and achieve variant LinkGroupCons. This permits linking of all transactions signed by the same user together, but not to link them back to their issuer. The authority remains able to revoke the issuer's privacy if need be.

b) Based on Linkable Ring Signatures: We can use linkable ring signatures [19], and achieve variant LinkRingCons. This permits linking of all transactions signed by the same user together, but not to link them back to their issuer. And it is impossible to revoke the issuer's privacy.

B. Endorser and Orderer Pseudonymity

Anonymity for endorsers and orderers is not desired, since we need to count the endorsements and $prepare$ messages.

But it remains possible to have pseudonymity for endorsers and orderers. Using linkable group signatures and linkable ring signatures helps us keep the endorsers' and the orderers' privacy, while at the same time restricting just enough of the excessive anonymity that is brought by blind, group and ring signatures. Thus, for endorsers, we can replace the signature scheme in (2) **Transaction Endorsement** with linkable group signatures or linkable ring signatures, which we call respectively EndGroupCons and EndRingCons. For orderers, we can identically replace the signature scheme in (5) **Block Preparation** and yield variants OrdGroupCons and OrdRingCons. Note that an extra step needs to be added after verifying the signature: the verifier must also check if this signature can be linked to another signature of the same content (transaction or block) before taking it into account: if it can be linked then it must be discarded.

IV. Protocol Properties

Our constructions satisfy *Safety*, *Liveness* and *Unforgeability* of a block along with privacy preserving settings, namely, *Anonymity* or *Pseudonymity*. This is described in Theorem 1.

a) Safety: A protocol is said to be *consistent* if it ensures that a transaction generated by a valid user stays immutable in the blockchain.

b) Liveness: The liveness property means that a consensus protocol ensures that if an honest client submits a valid transaction, a new block is later appended to the chain with the transaction in it. Hence, the protocol must ensure that the blockchain grows if valid clients generate valid transactions.

c) Unforgeability: An adversary against the *unforgeability* of a protocol tries to overstep the validation process of a transaction in order to engrave a transaction in the blockchain without obtaining the full transaction acceptance from the endorsers and the orderers.

d) Pseudonymity: An entity $\mathcal{E}$ and a witness w are said to be *linked* in a group G's perspective, if it is possible for G to infer that $\mathcal{E}$ produced w based on the available information to G. *Pseudonymity* of an entity holds when $\mathcal{E}$ cannot be linked to the witnesses $w_1, \ldots, w_k$ it has produced, but this property does not prevent from linking the witnesses to each other.

e) Anonymity: When linking w_i and w_j is hard for all $1 \leq i < j \leq k$, it is considered as a stronger privacy preserving property called *anonymity*.

Theorem 1 (Informal)**.** *All our blockchain protocols achieve safety, liveness and unforgeability with any possible combination of privacy properties for clients, endorsers and orderers.*

Proof Sketch: Safety and Liveness are both inherited from PBFT, assuming that the adversary cannot delay correct nodes indefinitely. We model and prove the other security properties using game based formalism and reductions. In general, we consider a security experiment where a PPT challenger $\mathcal{C}$ interacts with a PPT adversary $\mathcal{A}$. The adversary simulates the behaviour of a malicious entity, while the challenger runs the rest of the system honestly. We show that the security of our protocols also depends on the secure primitives used for instantiating it. Any secure signature could instantiate our blockchains.

	Revoke User	Authority No	Authority Inactive	Privacy Level
BlindCons				Ano.
GroupCons	✓		✓	Ano.
RingCons		✓	✓	Ano.
LinkGroupCons	✓		✓	Pseu.
LinkRingCons		✓	✓	Pseu.

TABLE I: Users' Privacy Preserving Protocols. Ano. : Anonymous; Pseu. : Pseudonymous.

	Revoke Endorser	Authority Not Needed	Privacy Level
(Ord) EndGroupCons	✓		Pseu.
(Ord) EndRingCons		✓	Pseu.

TABLE II: Orderers' and Endorsers' Protocols.

Our constructions are summed up in Tables I and II. The algorithms, the formalism as well as full argumentation and proofs can be found in the technical report [20].

References

[1] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *FC*, 2015.
[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[3] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *ASIACRYPT*, 2001.
[4] N. Van Saberhagen, "Cryptonote v 2.0," 2013.
[5] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *IEEE S & P*. IEEE, 2013.
[6] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification," *GitHub: San Francisco, CA, USA*, 2016.
[7] T. Hardjono and A. Pentland, "Verifiable anonymous identities and access control in permissioned blockchains," *CoRR*, vol. abs/1903.04584, 2019.
[8] J. Chen, "Hybrid blockchain and pseudonymous authentication for secure and trusted iot networks," *ACM SIGBED Review*, 2018.
[9] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: Challenges and directions," *IEEE Security & Privacy*, 2018.
[10] A. Jivanyan, "Lelantus: Towards confidentiality and anonymity of blockchain transactions from standard assumptions." *IACR ePrint.*, 2019.
[11] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, 2019.
[12] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy-preserving reputation system," in *IFIP SEC*, 2016.
[13] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *EuroSys conference*, 2018.
[14] L. Foundation, "Hyperledger." https://www.hyperledger.org/, 2019.
[15] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, 1999.
[16] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, 2000.
[17] D. Chaum and E. v. Heyst, "Group signatures," in *EUROCRYPT*. Springer, 1991.
[18] T. Nakanishi, T. Fujiwara, and H. Watanabe, "A linkable group signature and its application to secret voting," *Trans. of Information Processing Society of Japan*, 1999.
[19] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Information Security and Privacy*, H. Wang, J. Pieprzyk, and V. Varadharajan, Eds. Springer, 2004.
[20] F. A. Hayek, M. Koscina, P. Lafourcade, and C. Olivier-Anclin, "Generic privacy preserving private permissioned blockchains," https://hal.uca.fr/hal-03906880, 2022.