Digital signatures for interoperability and privacy in blockchains

Fadi Barbàra

2022

Some words about me...

Few words about me

- Last Year PhD student at UNITO
- Working on Interoperability and Privacy
- Using mainly Digital Signatures

Some words about me...

Few words about me

- Last Year PhD student at UNITO
- Working on Interoperability and Privacy
- Using mainly Digital Signatures

I like questions! Please, interrupt me

Motivation

- Currently the idea of privacy and interoperability are linked to Zero-Knowledge proofs. Examples are:
  - ZK-sync – interoperability (between layers)
  - Zendoo [GKO20] – cross-chain interoperability
  - Zcash – privacy in coins
- But zero-knowledge isn't the only way to achieve those goals. Other ways:
  - Digital Signature Schemes
  - Digital Signature Schemes Variations
- Goal of the talk is to present those cases

Summary

In this talk we will see:

- Part I
  - Definitions about Privacy in blockchain
  - Definitions about Interoperability
  - Notation and basic definitions of Digital Signature Scheme

Summary

In this talk we will see:

- Part II

```
for i in {multisig,threshold,ring}; do
        echo definition $i
        echo application_to_privacy $i
        echo application_to_interoperability $i
    done
```

Privacy

- Privacy in blockchain is a vast topic, I gave a talk on that[1]
- Three kinds of on-chain privacy:
  - Amount hiding: the confidentiality of a party's transaction amount
  - Unlinkability:
    - After a successful Bitcoin mixing transaction, honest participants' input and output addresses must be unlinkable [RMK14]
    - For any two transactions, it should be impossible to prove that they were sent to the same person [KFTS17, VS13]
  - Untraceability:
    - Given a transaction input, the real output being redeemed in it should be anonymous among a set of other outputs [KFTS17].
    - For each incoming transaction all possible senders are equiprobable [VS13]

[1] You can find it at `https://www.youtube.com/watch?v=7xsvxA54vIE`

Privacy

- Three kinds of on-chain privacy:
  - ~~Amount hiding~~ (use Pedersen Commitments)
  - Unlinkability (use indistinguishability)
  - Untraceability (use indistinguishability)

Interoperability

We can identify three types of interoperability in the blockchain world

- Cross-chain interoperability – blockchain-to-blockchain (B2B)
- Blockchain $\leftrightarrow$ Layer interoperability – blockchain-to-layer (B2L)
- Blockchain $\leftrightarrow$ Reality interoperability – blockchain-to-reality (B2R)

Basically interoperability is:

> *I give you something in a blockchain, I get something else in return in*
>
> - *Another blockchain*
> - *Another layer*
> - *Real life*

Transaction Models

There are mainly two blockchain-transaction models: UTXO and Account models. The differences are:

UTXO:

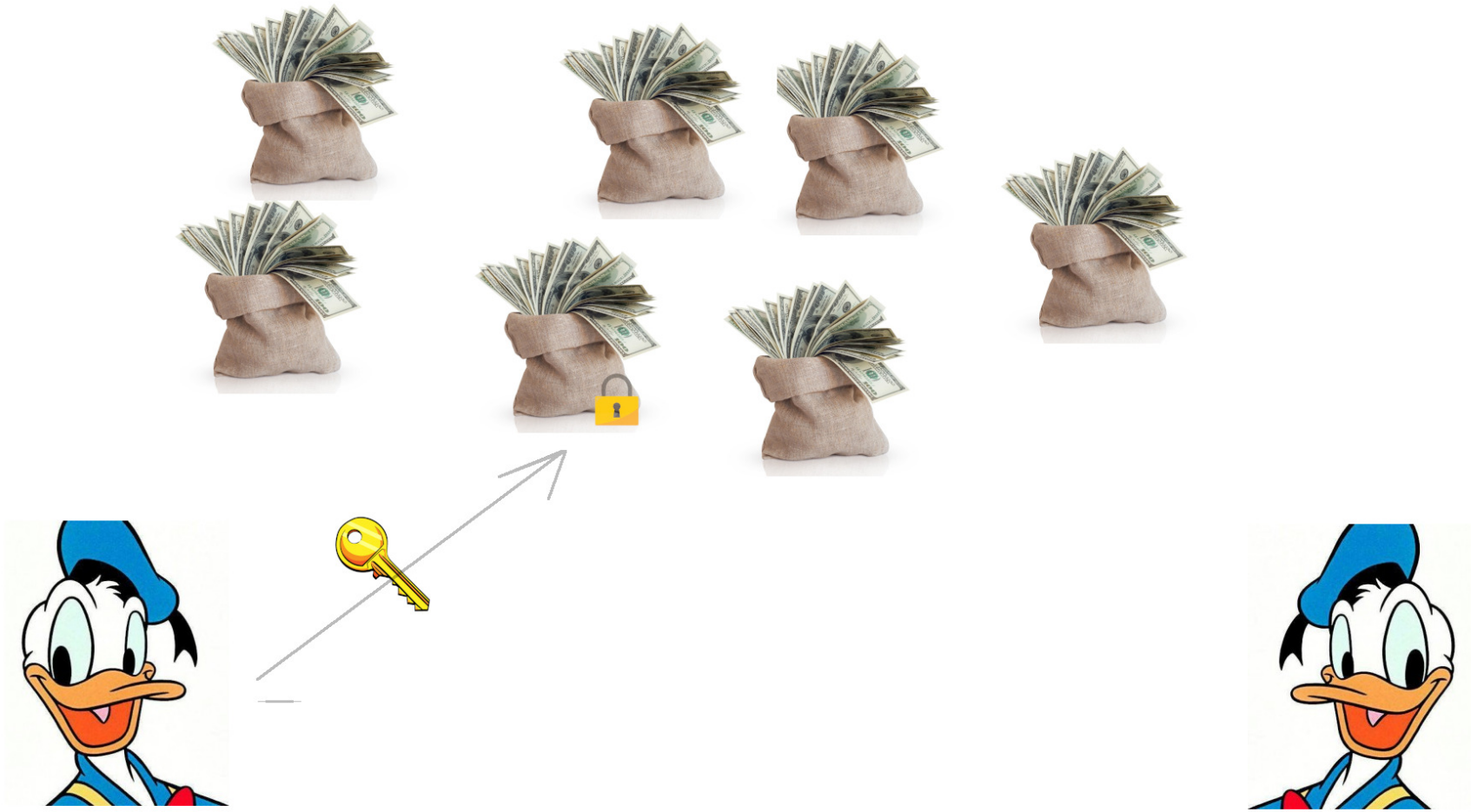
- Coins = reference to previously unspent transaction outputs
- Transactions can have multiple inputs to match output
- Balance of user is sum of all UTXOs
- Example: *Bitcoin,Monero, Mimblewimble-based coins*

Account:

- User balance in global state
- Transactions validation is against total balance
- Example: *Ethereum*

UTXO visual

User *A* can move funds with a key



UTXO visual

User B has the key to move the funds from user A

Account visual

User A sends funds to User B

Account visual

User B receives funds from A

Digital signature scheme

A digital signature scheme $(\text{Gen}, \text{Sign}, \text{Ver})$ is a tuple of algorithms such that:

- $(sk, pk) \leftarrow \text{Gen}(1^k)$ is the key-generation algorithm
- $\sigma \leftarrow \text{Sign}(sk, m)$ is the signing algorithm
- $0|1 \leftarrow \text{Ver}(pk, \sigma, m)$ is the verification algorithm

Any digital signature scheme can have variations adding other routines, but those three routines are always present.

Examples

Many digital signature schemes used in blockchain

- ECDSA – ex Bitcoin, Ethereum
- Schnorr – Bitcoin
- EdDSA – Tezos, Monero
- BLS – Ethereum 2.0
- ...

Multisig

First way to put multiple signatures on a message. While not formally defined, we can assume $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{MultiVer})$ where

- $\Sigma = ((pk_1, \sigma_1), (pk_2, \sigma_2), (pk_3, \sigma_3), \ldots)$ set of couples
- $0|1 \leftarrow \mathsf{MultiVer}(\Sigma, m)$ is the verification algorithm s.t.

$$\mathsf{MultiVer}(\Sigma, m) = \begin{cases} 1 & \text{if } \textit{all } \sigma_i \text{ are valid w.r.t } pk_i \text{ and } m \\ 0 & \text{otherwise} \end{cases}$$

Multisig – Uses

Creating the logic for multisig is easy in theory. In practice it depends on the transaction model used:

- Multisignatures can be used natively in UTXO systems: instead of checking that one signature is valid, it checks multiple signatures. The logic is not difficult to extend.
- This does not apply to account-models. In this case a user moves the funds directly. It is not possible to add a signature to specific funds if not through a smart contract.
  - e.g. Gnosis Smart Contract

Multisig and Privacy

- Actually *loss* of privacy
  - leak number of owner/parties involved
  - leak who makes the transaction
    - e.g. Bitmex has a 3-4 (P2SH) multisig
- If there is a flow it's even worse
  - e.g. you understand hierarchy of organization
- Remember *pseudoanonymity* $\neq$ *privacy*

Multisig and Interoperability

- Exchange committee
  - BTC-Liquid [DPW+17]
  - BTC-ETH (WBTC+BitGo) [Bit18]
- Where you CAN'T find
  - BSC-ETH bridge

Threshold signatures

A threshold signature scheme enables n parties to share the power to issue digital signatures under a single public key [GG18] In particular:

- $(sk_i, pk_1, \ldots, pk_n) \leftarrow \mathrm{ThGen}(1^k, P_1, \ldots, P_n)$ is a distributed key generation algorithm
- $\sigma \leftarrow \mathrm{ThSign}(sk_1, \ldots, sk_n, m)$ is a distributed signing algorithm
- $\mathrm{Ver}(\sigma, pk_1, \ldots, pk_n, m)$ is a verification algorithm

Note Ver() isn't "distributed": any user can verify a signature independently

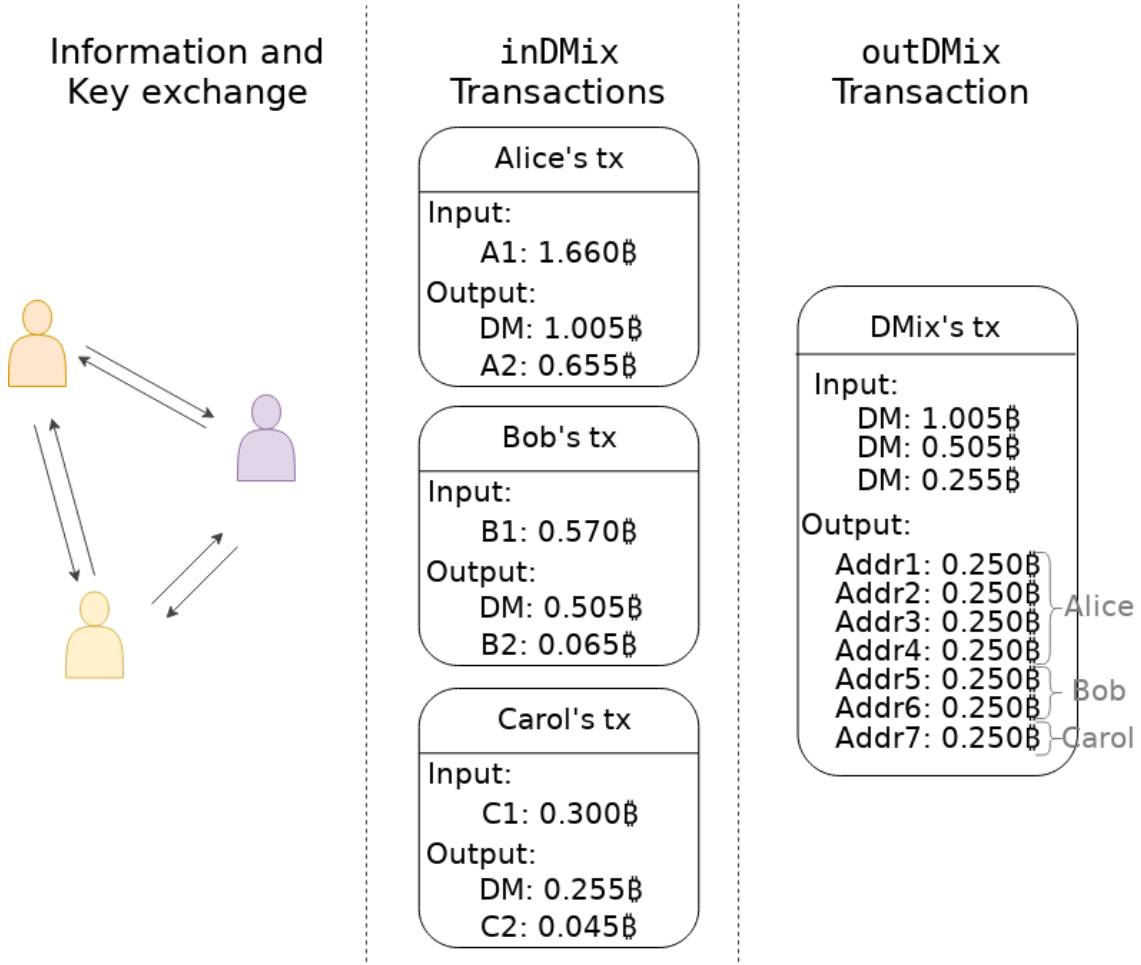
Threshold signatures

- A threshold signatures scheme is generally based on MPC. Consequently, it is also possible to aggregate signatures of different messages, as seen in:
  - Schnorr [NRS20]
  - BLS [Edg]
  - ECDSA [GG18]
- Since signature management is done off-chain, the signature method is based only on the signature scheme used
  - This is different from the Multisig case

Threshold signatures and Privacy

- Only one signature – regardless number of participants
- Mixers
  - DMix (BTC) [BS20] – Unlinkability but not Untraceability
  - ShareLock (ETH) [SS20] – Unlinkability but not Untraceability

DMix


Information and Key exchange
inDMix Transactions
outDMix Transaction
Alice's tx
Input:
A1: 1.660฿
Output:
DM: 1.005฿
A2: 0.655฿
Bob's tx
Input:
B1: 0.570฿
Output:
DM: 0.505฿
B2: 0.065฿
Carol's tx
Input:
C1: 0.300฿
Output:
DM: 0.255฿
C2: 0.045฿
DMix's tx
Input:
DM: 1.005฿
DM: 0.505฿
DM: 0.255฿
Output:
Addr1: 0.250฿
Addr2: 0.250฿
Addr3: 0.250฿
Addr4: 0.250฿
Addr5: 0.250฿
Addr6: 0.250฿
Addr7: 0.250฿
Alice
Bob
Carol

Threshold Signatures and Interoperability

- Multi-HTLC [BS]
- In principle any multisignature can be transformed into a threshold signature protocol

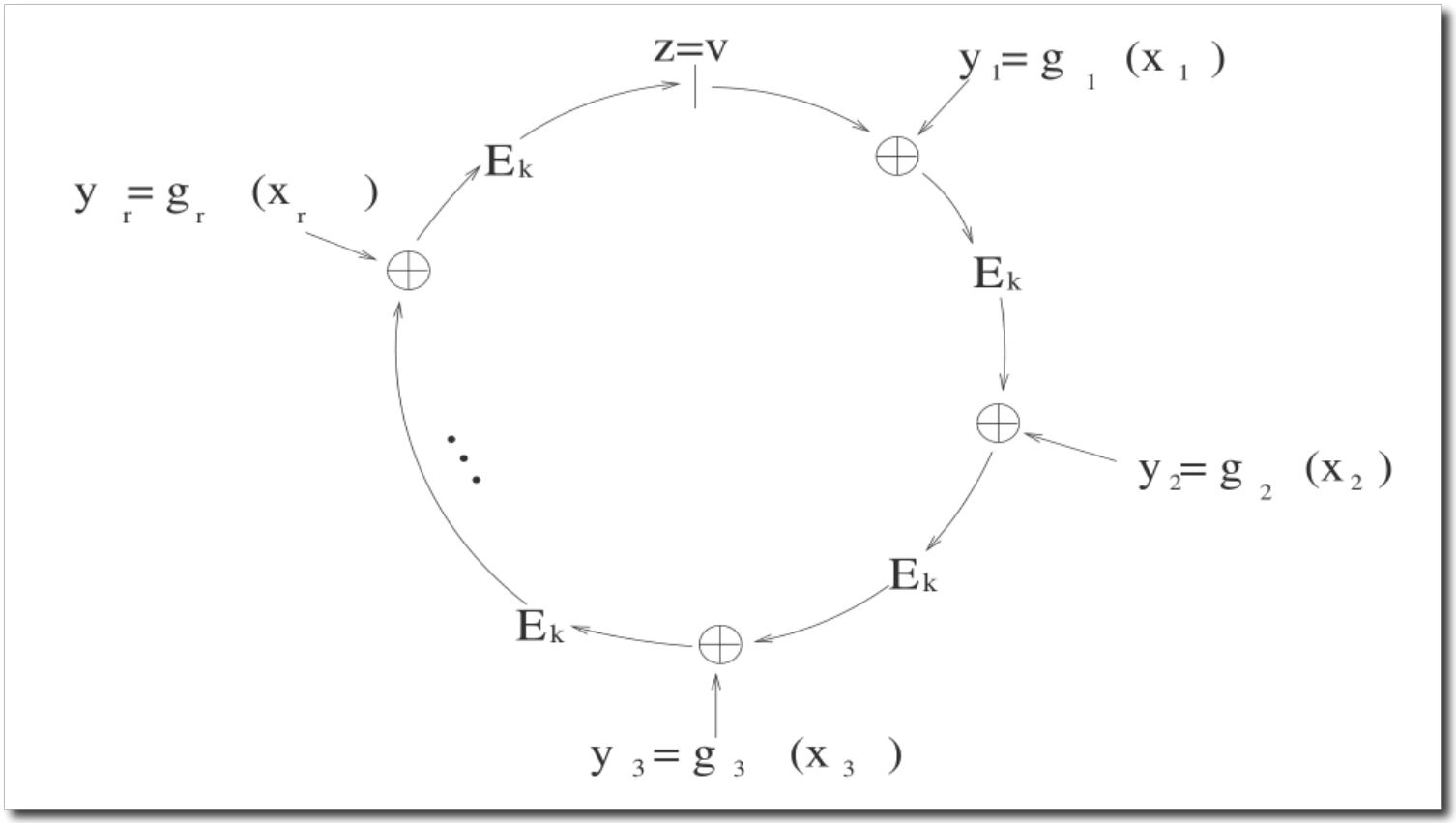
Ring Signatures

- Ring signatures are a particular case of group signatures [CvH91]. The goal is to specify a set of $n+1$ possible signers without revealing which member actually produced the signature [RST01]
- The algorithms would be
  - $\text{Gen}(1^k)$
  - $\sigma \leftarrow \text{RingSign}(sk, pk, pk_1, \ldots, pk_n, m)$ a different signing algorithm
  - $0|1 \leftarrow \text{RingVer}(\sigma, pk, pk_1, \ldots, pk_n, m)$ is a verification algorithm

Ring signatures

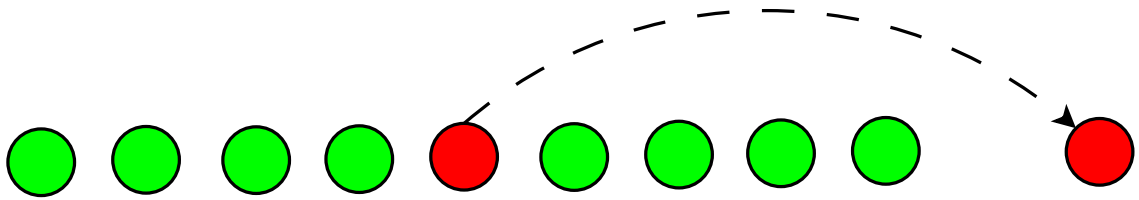
- A ring signature scheme has to use *existing* keys to be practically used in the blockchain. This has been seen for example in:
  - EdDSA [VS13]
  - Schnorr [NRS20]
  - ECDSA [FLM21]
- Since signature management is done off-chain, the signature method is based only on the signature scheme used
  - This is different from the Multisig case

Ring Signatures

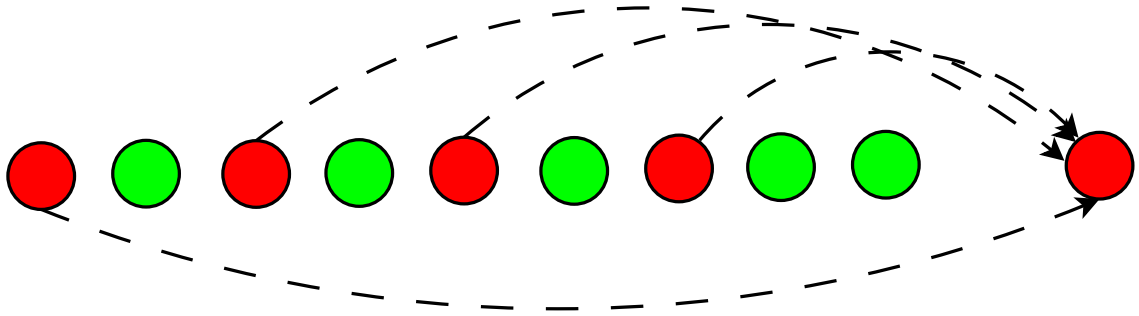


Ring signature transactions

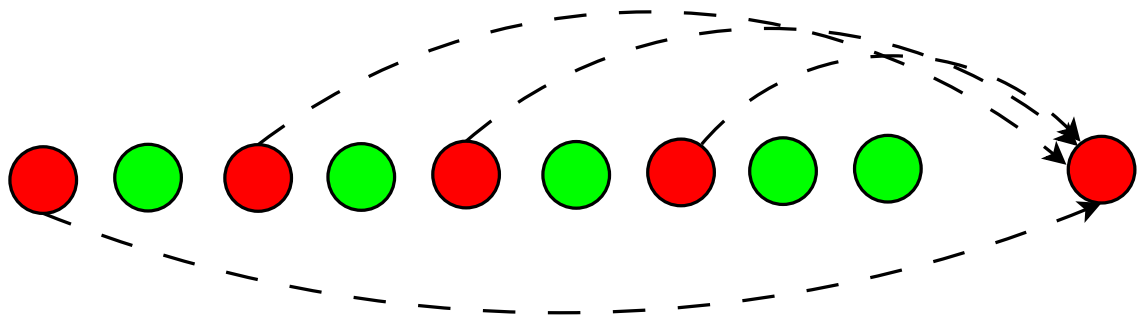
Bitcoin transaction:



Monero transaction:



Ring Signature and Privacy



- Untraceability: Given a transaction input, the real output being redeemed in it should be anonymous among a set of other outputs [KFTS17].
  - Achieved because inputs used in the transaction are indistinguishable
- Linkability: to prove that the private key was never been used to sign any other message with any other ring

ring Signature and Interoperability

- BTC-XMR atomic swap [Gug20]
- XMR-L2 interoperability [MBL+20]

Conclusions

- Digital Signature Variations rocks!
  - Improve Privacy
  - Improve Interoperability
  - (Improve Scalability)

Contacts

- Email: `fadi.barbara@unito.it`
- Telegram: `@fadibarbara`


BitGo.
Wbtc brings bitcoin to ethereum, 2018.


Fadi Barbàra and Claudio Schifanella.
[submitted] mp-htlc: Enabling blockchain interoperability through a multiparty implementation of the htlc.
Concurr. Comput. Pract. Exp.


Fadi Barbàra and Claudio Schifanella.
Dmix: decentralized mixer for unlinkability.
In *2nd Conference on Blockchain Research & Applications for Innovative Networks and Services, BRAINS 2020, Paris, France, September 28-30, 2020*, pages 1–8. IEEE, 2020.


David Chaum and Eugène van Heyst.
Group signatures.
In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 257–265, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.


Johnny Dilley, Andrew Poelstra, Jonathan Wilkins, Marta Piekarska, Ben Gorlick, and Mark Friedenbach.
Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks.
arXiv:1612.05491 [cs], January 2017.


Ben Edgington.
Bls12-381 for the rest of us.
Online.


Armando Faz-Hernández, Watson Ladd, and Deepak Maram.
Zkattest: Ring and group signatures on top of existing ECDSA keys.
IACR Cryptol. ePrint Arch., page 1183, 2021.


Rosario Gennaro and Steven Goldfeder.
Fast multiparty threshold ECDSA with fast trustless setup.
In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON,*

Canada, October 15-19, 2018, pages 1179–1194, address, 2018. organization, ACM.

Alberto Garoffolo, Dmytro Kaidalov, and Roman Oliynykov.
Zendoo: a zk-snark verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains.
In *40th IEEE International Conference on Distributed Computing Systems, ICDCS 2020, Singapore, November 29 - December 1, 2020*, pages 1257–1262. IEEE, 2020.

Joël Gugger.
Bitcoin-monero cross-chain atomic swap.
IACR Cryptol. ePrint Arch., page 1126, 2020.

Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena.
A traceability analysis of monero's blockchain.
In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017,*

Proceedings, Part II, volume 10493 of *Lecture Notes in Computer Science*, pages 153–173. Springer, 2017.

Pedro Moreno-Sanchez, Arthur Blue, Duc Viet Le, Sarang Noether, Brandon Goodell, and Aniket Kate.
DLSAG: non-interactive refund transactions for interoperable payment channels in monero.
In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers*, volume 12059 of *Lecture Notes in Computer Science*, pages 325–345. Springer, 2020.

Jonas Nick, Tim Ruffing, and Yannick Seurin.
Musig2: Simple two-round schnorr multi-signatures.
IACR Cryptol. ePrint Arch., 2020:1261, 2020.

Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate.
CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin.

In Mirosław Kutyłowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014*, volume 8713, pages 345–364. Springer International Publishing, Cham, 2014.

Ronald L. Rivest, Adi Shamir, and Yael Tauman.
How to leak a secret.
In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

Omer Shlomovits and István András Seres.
Sharelock: Mixing for cryptocurrencies from multiparty ECDSA.
IACR Cryptol. ePrint Arch., page 563, 2020.

Nicolas Van Saberhagen.
Cryptonote v 2.0, 2013.