MProve: A Proof of Reserves Protocol for Monero Exchanges

Arijit Dutta, **Saravanan Vijayakumaran**

Department of Electrical Engineering
Indian Institute of Technology Bombay

IEEE S&B, Stockholm
June 20, 2019

Cryptocurrency Exchanges

- Owning cryptocurrencies = Storing private keys
- Cryptocurrency exchanges
  - Store private keys for customers
  - Allow trading
- Risks for customers
  - Exchanges getting hacked
  - Incompetence, internal fraud, exit scams
  - Fractional reserve exchanges
- Proof of solvency is a possible solution
  - Proof of liabilities
  - **Proof of reserves**

Naive Proof of Reserves for Bitcoin

- Protocol steps
  - Create a transaction Tx which unlocks all owned UTXOs
  - Include a dummy input to make Tx invalid
  - Share Tx with the world.
- Why does it work?
  - Tx proves that exchange owns BTC equal to sum of amounts in unlocked UTXOs
  - Dummy input prevents misuse of Tx
  - Removing the dummy input will invalidate signatures
- Blockstream has released such a tool[1]
- **Drawback:** Privacy is not preserved
  - Exchange may want to reveal its UTXOs

[1] https://blockstream.com/2019/02/04/standardizing-bitcoin-proof-of-reserves/

Provisions Proof of Reserves Protocol

- Proposed by Dagher et al in 2015
- Exchange chooses a set $\mathcal{P}$ of UTXOs from the blockchain
  - It owns a subset $\mathcal{P}_{\text{own}}$ of $\mathcal{P}$. Let $\mathcal{I}_{\text{own}} = \{i \mid P_i \in \mathcal{P}_{\text{own}}\}$.
  - $\mathcal{P}$ plays the role of the anonymity set
  - Each $P_i \in \mathcal{P}$ has an associated amount a_i
- Pedersen commitment to an amount a is given by

$$C(y, a) = yG + aH,$$

  where the dlog of H wrt G is not known and y is a blinding factor
- Exchange creates a Pedersen commitment C_i for each $P_i \in \mathcal{P}$
- It gives a zero-knowledge proof of the following statement

$$C_i = \begin{cases} y_i G + a_i H & \text{if } P_i \in \mathcal{P}_{\text{own}} \\ y_i G & \text{if } P_i \notin \mathcal{P}_{\text{own}} \end{cases}.$$

- Adding all the commitments gives a commitment to the total reserves

$$C_{\text{reserves}} = \sum_{i=1}^{|\mathcal{P}|} C_i = \sum_{i=1}^{|\mathcal{P}|} y_i G + \sum_{i \in \mathcal{I}_{\text{own}}} a_i H.$$

- Solvency is proven via a range proof on $C_{\text{liabilities}} - C_{\text{reserves}}$

Transactions in Monero

- Suppose Alice wants to spend coins from an address P she owns
- Alice assembles a list $\{P_0, P_1, \ldots, P_{n-1}\}$ where $P_j = P$ for exactly one j
- Alice knows x_j such that $P_j = x_j G$
- Key image of P_j is $I = x_j H_p(P_j)$ where H_p is a point-valued hash function
  - Distinct public keys will have distinct key images
- A linkable ring signature over $\{P_0, P_1, \ldots, P_{n-1}\}$ will have the key image I of P_j
  - Signature proves Alice one of the private keys
  - Double spending is detected via duplicate key images
- One cannot say if a Monero address belongs to the UTXO set or not

A fundamental requirement of any proof of reserves protocol for Monero is that it should prove that the key images of the exchange-owned addresses, which contribute to the total reserves commitment $C_{reserves}$, have not appeared on the blockchain.

Some Facts About Commitments

- Suppose C is a Pedersen commitment with amount a and blinding factor x

$$C = xG + aH$$

- One can prove that C is a commitment to the zero amount via a signature with public key C

$$C = xG$$

- If C is a commitment to a non-zero amount a, signature with C as public key will mean dlog of H is known

$$C = xG + aH = yG \implies H = a^{-1}(y - x)G$$

MProve Protocol

- Exchange chooses addresses $\mathcal{P} = (P_1, P_2, \ldots, P_N)$ from the Monero blockchain
- It knows the private keys of $\mathcal{P}_{\text{known}} \subseteq \mathcal{P}$
- For each $P_i \in \mathcal{P}$, it reads commitment C_i

$$C_i = y_i G + a_i H.$$

  For $P_i \in \mathcal{P}_{\text{known}}$, the exchange knows y_i and a_i
- For each $P_i \in \mathcal{P}$, the exchange randomly picks z_i and generates C_i' as

$$C_i' = \begin{cases} z_i G & \text{if } P_i \in \mathcal{P}_{\text{known}}, \\ z_i G + C_i & \text{if } P_i \notin \mathcal{P}_{\text{known}}. \end{cases}$$

- For each $i = 1, 2, \ldots, N$, the exchange publishes a regular ring signature γ_i verifiable by the pair of public keys $(C_i', C_i' - C_i)$
- For each $i = 1, 2, \ldots, N$, the exchange publishes a linkable ring signature σ_i verifiable by the pair of public keys $(P_i, C_i' - C_i)$
- The exchange publishes a commitment C_{reserves} which satisfies the equation

$$C_{\text{reserves}} = \sum_{i=1}^{N} \left(C_i - C_i'\right).$$

MProve Intuition

- Output of an exchange
  - A list of one-time addresses $P_1, P_2, \ldots, P_N$ and commitments $C_1, C_2, \ldots, C_N$.
  - The commitments $C'_1, C'_2, \ldots, C'_N$ created by the exchange.
  - The regular ring signatures γ_i over public keys $(C'_i, C'_i - C_i)$
  - The linkable ring signatures σ_i over public keys $(P_i, C'_i - C_i)$
  - The commitment C_{reserves} to the total reserves

$$C_{\text{reserves}} = \sum_{i=1}^{N} (C_i - C'_i)$$

- When $P_i \notin \mathcal{P}_{\text{known}}$, the exchange has to create σ_i with z_i where $C'_i - C_i = z_i G$
  - This implies $C_i - C'_i$ is a commitment to the zero amount
  - No contribution to C_{reserves}
- When $P_i \in \mathcal{P}_{\text{known}}$, the exchange has to create γ_i with the private key corresponding to either C'_i or $C'_i - C_i$
  - If $C'_i = z_i G$, then $C_i - C'_i$ contributes $a_i H$ to C_{reserves}
  - If $C'_i - C_i = z_i G$, then $C_i - C'_i$ contributes nothing to C_{reserves}
- To avoid zero contribution to C_{reserves}, exchange has to sign with private key of P_i to create σ_i
  - Since σ_i reveals the key image of P_i, exchange cannot use an already spent address

Drawback

- Output of an exchange
  - A list of one-time addresses $P_1, P_2, \ldots, P_N$ and commitments $C_1, C_2, \ldots, C_N$.
  - The commitments $C_1', C_2', \ldots, C_N'$ created by the exchange.
  - The regular ring signatures γ_i over public keys $(C_i', C_i' - C_i)$
  - The linkable ring signatures σ_i over public keys $(P_i, C_i' - C_i)$
  - The commitment C_{reserves} to the total reserves

$$C_{\text{reserves}} = \sum_{i=1}^{N} (C_i - C_i')$$

- When $P_i \in \mathcal{P}_{\text{known}}$, the linkable ring signature contains the key image I_i of P_i
  - A future transaction spending from P_i will contain the same I_i
  - Makes the transaction zero mix-in
  - Ring signature is rendered useless

MProve Simulation Results

$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}_{\text{known}}\rvert$	Proof Size	Generat. Time	Verif. Time	Query Time
1000	100	0.32 MB	0.70 s	0.65 s	0.048 s
1000	500	0.32 MB	0.69 s	0.69 s	0.048 s
1000	900	0.32 MB	0.68 s	0.67 s	0.048 s
10000	1000	3.2 MB	7.01 s	6.76 s	0.087 s
10000	5000	3.2 MB	6.92 s	6.76 s	0.087 s
10000	9000	3.2 MB	6.87 s	6.75 s	0.087 s
100000	10000	32 MB	71.79 s	67.85 s	0.545 s
100000	50000	32 MB	71.13 s	67.83 s	0.545 s
100000	90000	32 MB	70.39 s	67.82 s	0.545 s

Future Directions

- Remove the drawback
- Make the proofs smaller
- Increase the anonymity set
- Ensure that exchanges generate reserves proofs from the same blockchain state
- Better proofs of liabilities

References

- Provisions https://eprint.iacr.org/2015/1008
- MProve https://eprint.iacr.org/2018/1210
- MProve Simulation Code https://github.com/avras/monero/tree/v0.14.0.2-mprove/tests/mprove

Thanks for your attention

Saravanan Vijayakumaran
sarva@ee.iitb.ac.in