Zero-Knowledge Proofs in Blockchains
(Blokzincirlerde Sıfır Bilgi İspatları)

2. Uusal Blokzincir Çalıştayı, İstanbul, 2019
26 Eylül 2019

Dr. Muhammed Ali BİNGÖL
muhammedali.bingol@tubitak.gov.tr
Privacy against who?

Decentralized, no trusted server

Central Trusted Authority

Centralized: Reveal amount, sender/receiver info to the bank

De-centralized: Reveal amount, sender/receiver info to everyone
Transaction amounts available in the clear

Everyone can see the payer, payee, and value

Business implications:

- Company pays employees in Bitcoin.
  ⇒ all salaries are public

- Public supply chain prices:
  - How much does Ford pay its supplier for tires?

Problem: Every transaction ever made is recorded forever
Bitcoin is neither confidential nor anonymous. Transactions are linkable and can be potentially de-anonymized.

### Anonymity vs Pseudonymity

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
</tr>
<tr>
<td>Fee Rate</td>
</tr>
<tr>
<td>Received Time</td>
</tr>
<tr>
<td>Mined Time</td>
</tr>
<tr>
<td>Included in Block</td>
</tr>
</tbody>
</table>

### Details

<table>
<thead>
<tr>
<th>Transaction Hash</th>
<th>Amount</th>
<th>Block Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2561b292ed4878bb28478a8caff1f99a01faeb9c5a906715fa595cac0e8d1d8</td>
<td>0.53333328 BTC</td>
<td>mined Apr 10, 2017 12:38:00 AM</td>
</tr>
<tr>
<td>16k4365rzdecPKGwJDNnBEkXj696MbChwx</td>
<td>0.01031593 BTC (U)</td>
<td></td>
</tr>
<tr>
<td>1Bsh4KD9ZJT4dJcoo75SsuS1jvmtVmREb7</td>
<td>1.47877788 BTC</td>
<td>2 BTC (S)</td>
</tr>
<tr>
<td>1AFLhD4ELG2uZmFxfdXCyGUNqCqD58B7u</td>
<td>2.01031593 BTC</td>
<td></td>
</tr>
</tbody>
</table>

Bitcoin only offers **pseudo-anonymity**. Transactions are linkable and can be potentially de-anonymized.
Transaction Flow Graph [Maxwell 2016]

Pseudonymity cannot provide Anonymity!!

- Transaction graph is still public

[Reid Martin 11] [Barber Boyen Shi Uzun 12] [Ron Shamir 12] [Ron Shamir 13] [Meiklejohn Pomarole Jordan Levchenko McCoy Voelker Savage 13] [Ron Shamir 14]
### Transaction Details

<table>
<thead>
<tr>
<th>Blockchain</th>
<th>Bitcoin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Transfer</td>
</tr>
<tr>
<td>Amount</td>
<td>94,504 <strong>BTC</strong> ($1,018,147,900 USD)</td>
</tr>
<tr>
<td>Timestamp</td>
<td>2 weeks 6 days ago (Fri, 06 Sep 2019 03:30:05 UTC)</td>
</tr>
<tr>
<td>Hash</td>
<td>4410c8d14ff9f87ceed1d65cb58e7c7b2422b2d7529afc675208ce2ce09ed7d</td>
</tr>
<tr>
<td></td>
<td>View transaction in blockchain.info</td>
</tr>
</tbody>
</table>
| From             | Unknown  
Multiple Addresses |
| To               | Unknown  
37XuVSEpWWtrkfmvWzegTHQt7BdktSKU |
|                  | View address in blockchain.info |
Confidentiality and Anonymity

1. Confidentiality: hiding the transferred amounts
2. Sender Anonymity: hiding the identities of the sender / the transaction origins
3. Receiver Anonymity: hiding the recipients identity
Bitcoin is not anonymous....what is next?

Option 1: minting/burning, mixers/tumblers compatible with Bitcoin

Option 2: New coin based on Zero Knowledge proofs
Zero-Knowledge Proofs

Sıfır Bilgi İspatları
Zero-Knowledge Proofs [Goldwasser-Micali-Rackoff’85]

Every statement that has a classical proof (in NP) has zero-knowledge interactive proof, if one-way functions exists. [Goldreich-Micali-Wigderson’91]

- There exists a ZK proof system for the NP-complete graph colouring problem with three colours.


http://web.mit.edu/~ezyang/Public/graph/svg.html
Secure Computation vs. Communication

- Secure Communication
  - Symmetric-Key Cryptography
    - Block Ciphers
    - Stream Ciphers
    - Hash Functions
  - Public-Key Cryptography
    - Asymmetric Encryption
    - Signature Schemes
  - Access Control
  - Etc.

- Secure Computation
  - Secure Multi-party Computation
  - Zero-Knowledge Protocols
    - Fiat-Shamir Protocol
    - Schnorr Proofs
    - Zk-Snarks
    - Zk-Starks
    - Bulletproofs
    - Sigma Bulletproofs etc.
  - Private Function Evaluation
  - Homomorphic Schemes
  - Etc.
Can we achieve full privacy with ZKPs?

ZKPs ≠ privacy

ZKPs == honest computation

\[ f(x) = y + \text{proof} \]
There are four common statement types, though the following is not an exhaustive list:

- An equality statement (the subject’s bank account balance is equal to $x$), or non-equality statement.
- An inequality statement (the subject’s bank account balance exceeds $x$).
- A range statement (the subject’s bank account balance is within interval $[a,b]$), or out-of-range statement.
- A membership statement (the subject is on the client list of bank $X$), or non-membership statement.
Alice has two cups each containing $x \in [0, n)$ marbles. She wants to prove to Bob that both contain the same number without revealing $x$.

Alice prepares 10 pairs of buckets, both buckets in the $i^{th}$ pair containing a random number $R_i \in [0, N)$ of marbles.

Bob chooses one of the pairs at random, and inspects the other 9 pairs to ensure that each pair indeed contains an identical number of marbles.
Alice has two cups each containing $x \in [0, n)$ marbles. She wants to prove to Bob that both contain the same number without revealing $x$.

Alice prepares 10 pairs of buckets, both buckets in the $i^{th}$ pair containing a random number $R_i \in [0, N)$ of marbles.
Alice has two cups each containing \( x \in [0, n) \) marbles.

She wants to prove to Bob that both contain the same number without revealing \( x \).

Alice pours the marbles from the first cup to the first bucket, and from the second cup to the second bucket.

Both contain \( R_i \in_r [0, N) \) marbles.
Alice has two cups each containing \( x \in [0, n) \) marbles. She wants to prove to Bob that both contain the same number without revealing \( x \).

Alice pours the marbles from the first cup to the first bucket, and from the second cup to the second bucket.
Alice has two cups each containing $x \in [0, n)$ marbles. She wants to prove to Bob that both contain the same number without revealing $x$.

Alice pours the marbles from the first cup to the first bucket, and from the second cup to the second bucket.

Bob accepts the proof if both buckets contain the same number of marbles.

**Soundness:** If the cups contain a different number of marbles, Bob rejects with prob $\geq 0.9$.

**Zero Knowledge:** The number $x + R_i$ Bob sees is distributed $n/N$ close to the uniform distribution on $(0, N)$. (Other 9 numbers are independent of $X$)
What is the success probability?

99,99..9 %
Properties of ZKP

- **Completeness:**
  - if the statement is true, the honest verifier will be convinced of this fact by an honest prover.

- **Soundness:**
  - if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.
Properties of ZKP

- **Zero-knowledge:**
  - if the statement is true, no verifier learns anything other than the fact that the statement is true.

Formalized by showing that every verifier has some *simulator* that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the verifier in question.
# Zero-Knowledge Proof Schemes

<table>
<thead>
<tr>
<th>Classical Schnorr Proofs</th>
<th><strong>C P Schnorr</strong> [1989] Efficient identification and signatures for smart cards, Crypto '89</th>
</tr>
</thead>
<tbody>
<tr>
<td>zk-STARKS</td>
<td><strong>E Ben-Sasson, I Bentov, Y Horesh, M Riabzev</strong> [2018] Scalable, transparent, and post-quantum secure computational integrity. e-print 2018/046</td>
</tr>
</tbody>
</table>
Simple ZK proof - Schnorr’s Protocol

I know the secret key $x$ and the public key $y = g^x$

I know a public key $y = g^x$ and $g$

1. Pick $r \in \mathbb{Z}_p$

   $a = g^r$

2. Pick $c \in \mathbb{Z}_p$

   $c$

3. Compute

   $z = r + cx \mod p$

   $z$

   Accept iff $g^z = Ry^c$

I know Alice has a secret key for the public key $(y, g)$
Variant: Non-Interactive ZK (NIZK)

1. Pick \( r \in \mathbb{Z}_p \), compute \( R = g^r \)

2. Pick \( c = \text{Hash}(R, y, g) \)

3. Compute \( z = r + cx \mod p \)

Accept iff \( c = \text{Hash}(R, y, g) \)

I know Alice has a secret key for the public key \((y, g)\)

Using Blockchains

I know a public key \( y = g^x \) and \( g \)

I know the secret key \( x \) and the public key \( y = g^x \)

Common Reference String

Maintained by Trusted Party or PKI
The **amount confidentiality** is provided by using **Pedersen commitment**

- The correctness (= balance) of the input and output amount is guaranteed by the **additive homomorphic** property of using Pedersen commitment.
- But we still need to ensure that for every transaction amount $M$:
  
  $$0 \leq M < \text{max}$$

- We need a (compact) **zero-knowledge range proof** for all transaction amount $M$!

They use inner product argument (Bulletproof)

- Represent each amount $M$ as a binary vector $(a_1, a_2, \ldots, a_n)$
- showed in ZK that $M = (a_1, a_2, \ldots, a_n) \cdot (1, 2, 4, 8, \ldots, 2^{k-1})$
- $0 \leq M < 2^k$
Commitment Schemes

Example:
- Alice and Bob must agree who will clean tonight
- They are at their offices. Each tosses a coin & they call:
  - If tosses are the same, then Alice cleans
  - If tosses are different, then Bob cleans
- Who talks first?
Commitment Schemes

Alice and Bob toss
- Alice talks first
  Bob says he tossed the same value
- Bob talks first
  Alice says she tossed the opposite value

How can we avoid this?
Commitment Schemes

- Commitment: an envelope with a strange seal
  - Alice talks first
  - **Commit phase**: she hides toss in envelope, gives it to Bob
  - Bob reveals toss
  - **Reveal phase**: Alice tells Bob how to unseal envelope
Commitment Schemes

Properties:

- **Hiding**: The content of the envelope is not visible
  
  Bob doesn’t know anything about Alice’s toss

- **Binding**: Alice can’t change the content in the envelope
  
  Alice can’t cheat after getting Bob’s toss
Pedersen Commitments

- **Setup**: \( G_p^* = \langle g \rangle \), prime field, \( h = g^s \in G_p^* \backslash \{1\} \), \( s \) unknown
- **Commitment of input value** \( x \in \{0,1\} \):
  - Choose random witness \( w \leftarrow_R \{1, \ldots, p-1\} \)
  - Compute \( \text{Commit}(x, w) = g^w h^x = g^w g^{xs} = g^{w+xs} \)
  - **Binding**: Alice can’t change the content in the envelope?
  - **Hiding**: The content of the envelope is not visible?

\[ a = \text{Commit}(x, w) \quad \text{for} \quad x \in \{0,1\} \]

Check \( a = \text{Commit}(x, w) \)
Confidential Transactions

Summary

Size 1110 (bytes)
Fee Rate 0.0016173243243244 BTC per kB
Received Time Apr 10, 2017 12:38:00 AM
Mined Time Apr 10, 2017 12:38:00 AM
Included in Block 0000000000000001f0115cca585646832b337404032c88539ce2995e799e5c

Details

Sum of inputs $\geq$ Sum of outputs?
Outputs positive?

Pedersen commitment:
Commit($x;r$) = $g^x h^r$
Bulletproofs

Use Bulletproofs for more efficient range proofs only and not for privacy directly
Proving that a number is within a range

\[ v \in [0, 2^n) \]

**Zero Knowledge about the Inner Product of Two Vectors**

Any number can be represented as inner product of two vectors.

5 equals inner product of 2 vectors [1, 0, 1] and [2², 2¹, 2⁰]
This is also how binary works

$$101_{\text{binary}} = 5_{\text{decimal}} \text{ since } 1(2^2) + 0(2^1) + 1(2^0)$$

$$v = <a, 2^n>$$

Example:
$$v = 5 \text{ and we wanted to prove that } 5 \text{ is in range of } 0 \text{ to } 2^n \text{ without showing } 5$$

$$v \in [0, 2^n)$$
Concrete Range Proof using bit commitments

\[ c_i = commit(b_i, r_i) \land x = \sum_{i=0}^{n-1} b_i \times 2^i \land b_i \in [0,1] \]

\[ x = (b_0, \ldots, b_{n-1}), b_i \in [0,1] \]
\[ r_i \leftarrow \mathbb{Z}_q \forall i \in [0, n-1] \]
\[ c_i = commit(b_i; r_i) \forall i \in [0, n-1] \]
Research Directions

- Bandwidth Efficiency
- Computation Efficiency
- Round Efficiency
Full Scheme of the Bulletproofs

https://github.com/dalek-cryptography/bulletproofs

zkSNARK construction via QAP and Linear PCPs

- Computation
- Algebraic Circuit
- R1CS (Rank-1 Constraint System)
- QAP (Quadratic arithmetic program)
- Linear PCP (probabilistically checkable proof)
- zk-SNARK
Properties of Zk-Snakrs

• Efficiency:
  – 288 byte **proof per transactions** (128-bit security)
  – <6 ms to **verify a proof**
  – <1 min to create for $2^{64}$ coins; asymptotically: log(#coins)
  – 896MB “system parameters” (fixed throughout system lifetime).

• **Trust in initial generation of system parameters (once).**

• Crypto assumptions:
  – Pairing-based elliptic-curve crypto
  – Less common: Knowledge of Exponent
    [Boneh Boyen 04] [Gennaro 04] [Groth 10] ...
  – Properties of SHA256, encryption and signature schemes
<table>
<thead>
<tr>
<th>Proof System</th>
<th>Schnorr $\Sigma$-Protocol</th>
<th>Zk-SNARKs</th>
<th>STARKs</th>
<th>Bulletproofs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proof Size</td>
<td>Long</td>
<td>Very Short</td>
<td>Shortish</td>
<td>Shortish</td>
</tr>
<tr>
<td>Prover</td>
<td>Linear</td>
<td>FFTs (memory req.)</td>
<td>FFT (Big memory req.)</td>
<td>Multiexp.</td>
</tr>
<tr>
<td>Verifier</td>
<td>Linear</td>
<td>Efficient</td>
<td>Efficient</td>
<td>Linear</td>
</tr>
<tr>
<td>Trusted Setup</td>
<td>No</td>
<td>Required</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Practical</td>
<td>Yes</td>
<td>Yes</td>
<td>Not Quite</td>
<td>Yes</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Dlog + RO</td>
<td>Pairing + KoE</td>
<td>RO</td>
<td>Dlog + RO</td>
</tr>
<tr>
<td>Quantum Resistancy</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Dr. Muhammed Ali BİNGÖL

TÜBİTAK BİLGEM
National Research
Institute of Electronics and Cryptology
Blockchain Research Lab.

muhammedali.bingol@tubitak.gov.tr
T: +90 262 648 1702