Analyzing Bitcoin Security

Philippe Camacho
philippe.camacho@dreamlab.net
Bitcoin matters
An open question (until 2008)

Is it possible to create (digital) money without a centralized authority?
Who wants some satoshis?

- What kinds of problems are hard to solve when building a decentralized digital cash system?
P2P Network

- TCP/IP
- No authentication
- 8 outgoing connections
- Up to 117 incoming connections
- Hardcoded IP addresses + DNS seeders to get first list of peers
- Probabilistic algorithm to choose peers
- Specific data structure to store peers list
- Gossip protocol to broadcast transactions
Bitcoin addresses

3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v

3J5KeQSVUBEs3v2vEEkZDBtPLWqLTuZPuD
Bitcoin transactions

- Conceptually very simple
- In practice quite complicated (more on this later)
Double spending attack

3NDQz8rZ3CnmsiBGrATk8SCpDXF2sAUiuM

1BTC

3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v

3J5KeQSVBUEs3v2vEEkZDBtPLWqLTuZPuD

3NDQz8rZ3CnmsiBGrATk8SCpDXF2sAUiuM
How does Bitcoin prevent the double-spending attack?

- **Idea**
  - Have participants of the network vote to establish the “official” ordered list of transactions
  - Check the validity of each transaction with this ledger

- **Challenge**
  - We are in an open network => Sybil attack is always possible
Consensus

Instead of voting with your IP, vote with your CPU
The Blockchain
Who wants some satoshis?

- Who will extend the next block?
  Or how to agree in a fair way on the participant that will extend the chain?
Proof of Work [Back2002]

Find value $x$ so that the output begins with 3 zeros.

The «only way» to compute this value so that the output starts with $n$ zeros is to try at random around $2^n$ times.
On the limits of the Random Oracle

- Approximate Bitcoin Mining
  \[\text{LH2015, VDR2015}\]

- Patent pending AsicBoost.com

- Enable to increase profitability of miners by 20\%\textasciitilde30\%
in 1 slide
Who wants some satoshis?

- What happens if two miners produce a block at almost the same time?
The longest chain rule

Idea:
the longest chain represents the accumulated computational power of the network

Main chain that will be extended by the miners
Who wants some satoshis?

- How are bitcoins created?
- Why would people spend their computational power to protect the network?
Incentive

- Each block mined that ends up in the main chain will be awarded with 12.5 BTC (*)

- Hence the metaphor «Mining»

(*) It was 50 BTC at the beginning, halving every 210000 blocks
What is the probability that the monkey, sooner or later, will fall off the cliff?

$P[Left] = 1 - \alpha = \beta$

$P[Right] = \alpha$

We call this probability $P_1$

We have that $P_k = P_1^k$

**Theorem:** If $\alpha > \beta$ then $P_1 = 1$

If $\alpha < \beta$ then $P_1 = \frac{\alpha}{\beta}$ and $P_k = \frac{\alpha}{\beta}$
51% attack [Nakamoto2008]

I will try to catch up with 3 blocks and rewrite the history of transactions
51% attack

$1 - \alpha = \beta$

$\alpha$

Computational power of the adversary

What is the probability that the adversary catches up with $k$ blocks, sooner or later?

$k$ is the number of confirmations

We have that $P_k = P_1^k = (\frac{\alpha}{\beta})^k$

Decreases exponentially fast in $k$
Who wants some satoshis?

- What are some implicit assumptions in the previous analysis?
Selfish Mining Attack [ES2014]

- **Idea:** The attacker will mine his blocks *privately* and release them at the right time so that honest miners *waste their computational power*. 
Selfish Mining Attack

This block is «lost» for the honest miners

Honest miners keep mining on the new longest chain

This block is «lost» for the honest miners
Selfish Mining Attack

**State 0:** only a single public chain
Selfish Mining Attack

**State 1:** Adversary manages to mine a block. The block is kept *private.*
Selfish Mining Attack

**State 2:** Adversary manages to mine a block. The block is kept *private.*
Selfish Mining Attack

**State 2:** Honest miners find a block

In this situation the private chain is published and the honest miners loose their block.
State 0: After releasing the private chain, back to state 0.

New head of the public chain.
Selfish Mining Attack

**State 1:** Adversary manages to mine a block. The block is kept *private*.
Selfish Mining Attack

**State 0':** Honest miners and adversary's chain are competing

In this situation release the private block and hope the honest miners will mine on top of it.
Selfish Mining Attack

\( \alpha: \) Adversary’s computational power

\( \gamma: \) Portion of honest miners that will mine on top of adversary’s block

Selfish Mining Attack

- Now we can compute the relative gain of the adversary

\[ R_A = \frac{r_a}{r_a + r_h} = \frac{\alpha(1 - \alpha)^2(4\alpha + \gamma(1 - 2\alpha)) - \alpha^3}{1 - \alpha(1 + (2 - \alpha)\alpha)} \]
Who wants some satoshis?

- If everything were “fine”, how much should $R_a$ be equal to?

\[ R_A = \frac{r_a}{r_a + r_h} = ??? \]
Selfish Mining Attack

- **Results:**
  - $\alpha > \frac{1}{4}$:
    - the selfish mining strategy is more profitable than the honest strategy
  - Depending on $\gamma$ this can be worse
    (i.e. the selfish mining strategy is always profitable)

- **What is the problem?**
  - If miners are rational then they will prefer to join the adversary’s pool => soon the adversary’s pool will be
Eclipse Attack [HKZG2015]

The attacker surrounds the victim in the P2P network so that it can filter his view on the events.
Eclipse Attack

- Mainly an implementation problem
  - It is possible to populate the tables of peers of the victim

- But with huge consequences as this attack can be used to leverage others
  - Selfish mining
  - 51%
  - Double spending
### Transaction Malleability

**Step 1:**
Compute the unsigned transaction

**Step 2:**
Compute the signature of the transaction

**Step 3:**
Put the signature inside the transaction

**Step 4:**
Compute the hash of the signed transaction => this is the transaction ID

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>01 00 00 00</td>
</tr>
<tr>
<td>input count</td>
<td>01</td>
</tr>
<tr>
<td>previous output hash</td>
<td>48 4d 40 d4 5b 9e a0 d6 52 fc a8 25 8a b7 ca a4 25 41 eb 52 97 58 57 f9 6f b5 0c d7 32 c8 b4 81</td>
</tr>
<tr>
<td>previous output index</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>script length</td>
<td></td>
</tr>
<tr>
<td>scriptSig</td>
<td><em>script containing signature</em></td>
</tr>
<tr>
<td>sequence</td>
<td>ff ff ff ff</td>
</tr>
<tr>
<td>output count</td>
<td>01</td>
</tr>
<tr>
<td>value</td>
<td>62 64 01 00 00 00 00 00</td>
</tr>
<tr>
<td>script length</td>
<td></td>
</tr>
<tr>
<td>scriptPubKey</td>
<td><em>script containing destination address</em></td>
</tr>
<tr>
<td>block lock time</td>
<td>00 00 00 00</td>
</tr>
</tbody>
</table>

Transaction Malleability

**Problem:** signature algorithm is probabilistic (ECDSA) => very easy to create «identical» transactions with different hashes

**Step 1:** Compute the unsigned transaction

**Step 2:** Compute the signature of the transaction

**Step 3:** Put the signature inside the transaction

**Step 4:** Compute the hash of the signed transaction => this is the transaction ID

Privacy with Bitcoin

«Standard» user id is replaced by a random looking sequence.

Bitcoin address
31uEbMgunupShBVTewXjtqbBv5MndwfXhb
However Bitcoin is not totally anonymous

Anonymity = Pseudonymity + Unlinkability
Improving Anonymity with mixers
Other initiatives: Zerocash [BCG+2014]

+ Uses of near to practical «universal» zero-knowledge proofs (ZK-SNARKs)
+ Provides a much higher level of anonymity than mixers
- Requires to change bitcoin source code
- Requires a trusted setup
Bitcoin Backbone protocol [GKL2014]

- **Purpose:** models the problem that occurs when the time of mining a block becomes small
- **Security model:** synchronous setting (*)

(*) Asynchronous setting is even more complex and analyzed in [PSS2016].
Bitcoin Backbone protocol

- **Common prefix property:**
  - Let $f$ be the expected blocks mined per network synchronization round
  - If $\beta > \lambda \alpha$ where $\lambda > 1$ and $\lambda^2 - f\lambda + 1 \geq 0$
    then two honest participants will have the same chain if $k$ blocks are pruned (i.e. the probability that it does not happen drops exponentially in $k$)
Chain quality property:

- if $\beta > \lambda \alpha$ where $\lambda > 1$ then the ratio of blocks in the chain of any honest player that are contributed by honest players is at least $\left(1 - \frac{1}{\lambda}\right)$
Open problems

- Anonymity
- Selfish Mining
- Alternatives to PoW
- Scalability
- Avoiding centralization in mining
- ASIC resistance proof of work
- Useful proof of work
- ...

Thank you!
References

References


