Alternative Mining Puzzles

- Essential Puzzle Requirements
- ASIC-Resistant Puzzles
- Proof-of-Useful-Work
- Non-outsourcable Puzzles
- Proof-of-Stake “Virtual Mining”

Puzzles (recap)

Incentive system steers participants

Basic features of Bitcoin’s puzzle
  - The puzzle is difficult to solve, so attacks are costly
  - ... but not too hard, so honest miners are compensated

Q: What other features could a puzzle have?
On today’s menu . . .

Alternative puzzle designs
   Used in practice, and speculative

Variety of possible goals
   ASIC resistance, pool resistance, intrinsic benefits, etc.

Essential security requirements

Alternative Mining Puzzles

- Essential Puzzle Requirements
- ASIC-Resistant Puzzles
- Proof-of-Useful-Work
- Non-outsourceable Puzzles
- Proof-of-Stake “Virtual Mining”
Puzzle Requirements

A puzzle should ...
- be cheap to verify
- have adjustable difficulty
- <other requirements>

- have a chance of winning that is proportional to hashpower
  - Large player get only proportional advantage
  - Even small players get proportional compensation

Bad Puzzle: a sequential Puzzle

Consider a puzzle that takes $N$ steps to solve a “Sequential” Proof of Work
Bad Puzzle: a sequential Puzzle

Problem: fastest miner *always* wins the race!

Solution Found!

Good Puzzle => Weighted Sample

This property is sometimes called *progress free.*
**Alternative Mining Puzzles**

- Essential Puzzle Requirements
- ASIC-Resistant Puzzles
- Proof-of-Useful-Work
- Non-outsourceable Puzzles
- Proof-of-Stake “Virtual Mining”

---

**ASIC Resistance – Why?!**

**Goal:** Ordinary people with idle laptops, PCs, or even mobile phones can mine!

Lower barrier to entry!

**Approach:** Reduce the gap between custom hardware and general purpose equipment.
Memory-hard Puzzles

Premise: the cost and performance of memory is more stable than for processors

![Graph showing the performance of different resources over time]

Example: scrypt (Colin Percival, 2009)

Memory hard hash function (requires large amounts of memory)
⇒ Prevents large-scale parallel attack with limited resources.

Most widely used alternative Bitcoin puzzle (e.g. in LiteCoin)

Also used elsewhere in security (PW-hashing, Tarsnap)

1. Fill memory with random values
2. Read from the memory in random order
**scrypt – Step 1 of 2 (write)**

Input: $X$

$V_1 = H(X)$

$V_2 = H(V_1) = H(H(X))$

$V_3 = H(V_2) = H^3(X)$

$\ldots$

$V_N = H^N(X)$

**scrypt – Step 2 of 2 (read)**

Input: $X$

$A := H^{N+1}(X)$

For $N$ iterations:

$i := A \mod N$

$A := H(A \text{ xor } V_i)$

Output: $A$
**scrypt – Time/Memory Tradeoff**

Q: Why is this memory-hard?
Reduce memory by half, 1.5x the # steps

![Diagram of scrypt function](image)

- Need to access \( V_i \) where \( i \) is even?
  - First, access \( V_{i-1} \)
  - Then, compute \( V_i = H(V_{i-1}) \)

**scrypt – Discussion**

Disadvantages:
Also requires \( N \) steps, \( N \) memory to check

Is it actually **ASIC resistant**?
scrypt ASICs are already available!

[zeusminer.com](http://zeusminer.com/)
**Cookoo Hash Cycles** (John Tromp, 2014)

Example of a memory hard puzzle that’s *cheap to verify*.

- **Input:** $X$
- **For** $i = 1$ to $E$:
  - $a := H_0(X + i)$
  - $b := N + H_1(X + i)$
  - edge($a$ mod $N$, $b$ mod $N$)

  Is there a cycle of size $K$? If so, **Output:** $X$, $K$ edges

---

**Even more Approaches**

- More *complicated hash functions*
  - **X11:** 11 different hash functions combined

- **Moving target**
  - Change the puzzle periodically
**Counter Argument: SHA2 is fine!**

Bitcoin Mining ASICs aren’t changing much.
Big ASICs only marginally more performant than small ones.

![SHA2 Circuit Diagram](image)

**Affordable ASIC**

**Expensive ASIC**

---

**Alternative Mining Puzzles**

- Essential Puzzle Requirements
- ASIC-Resistant Puzzles
- Proof-of-Useful-Work
- Non-outsourcable Puzzles
- Proof-of-Stake “Virtual Mining”
Recovering wasted Work

Recall:

between 150 MW – 900 MW power consumed (as of mid 2014)

Natural Question:

Can we recycle this and do something useful?

Candidates – Needle in a Haystack

Natural choices:
- Protein folding (find a low-energy configuration)
- Search for aliens (find anomalous region of signal)

(These have been successful @Home problems)

Challenges:
- Randomly chosen instances must be hard
Primecoin (Sunny King, 2013)

Puzzle based on finding large prime numbers.

Cunningham chain:

\[ p_1, p_2, \ldots, p_n \] where \( p_{i+1} = 2p_i - 1 \)

each \( p_i \) is large (probable) prime

\( p_1 \) is divisible by \( H(\text{prev} \ || \ mrkl\_root \ || \ \text{nonce}) \)

Primecoin

Many of the largest known Cunningham chains have come from Primecoin miners.

Q: Is this a hard problem?

Q: Is this useful?
Recovering wasted Hardware

Estimate: More than $100M spent on customized Bitcoin mining hardware!

This hardware investment is otherwise useless.

Idea: How about a puzzle where hardware investment is useful, even if the work is wasted?

Permacoin – Mining with Storage (Miller et al., 2014)

Bitcoin

Permacoin

Side effect:
Massively distributed, replicated storage system
**Permacoin**

Assume we have a large file $F$ to store

For simplicity: $F$ is chosen globally, at the beginning, by a trusted dealer

Each user stores a random subset of the file

---

**Storage-based Puzzle**

1. Build a Merkle tree, where each leaf is a segment of the file

2. Generate a public signing key $p_k$, which determines a random subset of file segments

3. Each mining attempt:

   a) Select a random nonce
   b) $h_1 := H(prev || mrkl_root || PK || nonce)$
   c) $h_1$ selects $k$ segments from subset

   d) $h_2 := H(prev || mrkl_root || PK || nonce || F)$
   e) Winner if $h_2 < TARGET$
Proof-of-Storage to Reduce “Honesty” Cost

“Honest” miners validate every transaction

Validation requires the UTXO database ~200MB

Maintaining the UTXO database doesn’t pay

Idea: use Permacoin to reward UTXO storage

Summary

Useful proof-of-work is a natural goal
(while maintaining security requirements)
The benefit must be a pure public good
Viable approaches include storage, prime-finding,
others may be possible
Realized benefit so far has been limited
Alternative Mining Puzzles

- Essential Puzzle Requirements
- ASIC-Resistant Puzzles
- Proof-of-Useful-Work
- Non-outsourceable Puzzles
- Proof-of-Stake “Virtual Mining”

Large Mining Pools are a Threat

Premise: Bitcoin’s core value is decentralization

If power is consolidated in a few large pools, the operators are targets for coercion/hacking

Position: Large pools should be discouraged!
   Analogy to voting: It’s illegal (in US) to sell your vote
Large Mining Pools are a Threat

June 12, 2014
GHash.IO large mining pool crisis

Large Mining Pools are a Threat

Hacking, Distributed

It's Time For a Hard Bitcoin Fork

Ittay Eyal, and Emin Gun Sirer

Friday June 13, 2014 at 02:05 PM

A Bitcoin mining pool, called GHash and operated by an anonymous entity called CEX.io, just reached 51% of total network mining power today. Bitcoin is no longer decentralized. GHash can control Bitcoin transactions.

Is This Really Armageddon?

Yes, it is. GHash is in a position to exercise complete control over which
Large Pools have interesting Dynamics

Observation:
Pool participants don’t trust each other.

Pools only work because the “shares” protocol lets members prove cooperation.
**Standard Bitcoin Mining Pool**

Payout dividing among members

Pool Operator

“shares”: proof that a member is “toeing the line”

Solution found!

**The Vigilante Attack**

Suppose a Vigilante is angry with a large pool

He submits “shares” like normal....

... but if he finds a real solution, discards it

Pool output is reduced, Vigilante loses a little
The Vigilante Attack

- Solution discarded
- "shares": proof that a member is "toeing the line"

Payout dividing among members

Pool Operator

Encouraging the Vigilante (Rewarding Sabotage)

Whoever FINDS a solution spends the reward.

Approach:
- searching for a solution requires SIGNING, not just hashing. (Knowledge of a private key)
- Private key can be used to spend the reward
Encouraging the Vigilante (Rewarding Sabotage)

Solution found!
Take the money and run!
Also: evade detection

Nonoutsorceable Puzzle

Solution:
\[(\text{prev}, \text{mrkl\_root}, \text{nonce}, \text{PK}, s1, s2)\]

such that:
\[\text{H}(\text{prev} \mid\mid \text{PK} \mid\mid \text{nonce} \mid\mid s1) < \text{TARGET}\]
\[
\text{VerifySig}(\text{PK}, t1, \text{prev} \mid\mid \text{nonce})
\]
\[
\text{VerifySig}(\text{PK}, t2, \text{prev} \mid\mid \text{mrkl\_root})
\]
Cryptocurrency Technologies

Alternative Mining Puzzles

Non-outsorceable Puzzles: Concerns

This puzzle discourages all pools including harmless decentralized P2Pools

Other forms of outsourcing? might drive pool members to hosted mining

Alternative Mining Puzzles

• Essential Puzzle Requirements
• ASIC-Resistant Puzzles
• Proof-of-Useful-Work
• Non-outsorceable Puzzles
• Proof-of-Stake “Virtual Mining”
Mining has an unnecessary Step

Proof-of-Work Mining:

Spend money on power and equipment
Find puzzle solutions
Earn mining rewards

Virtual Mining:

Find puzzle solutions
Send money to special address
Earn mining rewards
Spend money on power and equipment

Winners chosen at random by lottery
Benefits of Virtual Mining

Lower overall costs
- No harm to the environment
- Savings distributed to all coin holders

Stakeholder incentives – good stewards?

No ASIC advantage

51% Attack Prevention

- The Bitcoin economy is smaller than the world
- Wealth outside Bitcoin has to move inside
Variations of Virtual Mining

**Proof-of-Stake:** “Stake” of a coin grows over time as long as the coin is unused

**Proof-of-Burn:** mining with a coin destroys it

**Proof-of-Deposit:** can reclaim a coin after some time

**Proof-of-Activity:** any coin might be win (if online)

Open Questions with Virtual Mining

**Q:** Is there any security that can only be gained by consuming “real” resources?

**YES:** Then “waste” is the cost of security

**No:** Then Proof-of-Work mining may go extinct
Conclusion

Many possible design goals for alternative puzzles:
- Prevent ASIC miners from dominating
- Prevent large pools from dominating
- Intrinsic usefulness
- Eliminate the need for mining hardware at all

Best tradeoff is unclear for now

Outlook: alternatives will coexist for the near future