REM: Resource Efficient Mining for Blockchains

Fan Zhang, Ittay Eyal, Robert Escriva, Ari Juels, Robbert van Renesse

Vancouver, Canada
The Cryptocurrency Vision

Originally

• Satoshi Nakamoto’s Bitcoin (’08-’09)
• Decentralized currency
The Cryptocurrency Vision

Originally

- Satoshi Nakamoto’s Bitcoin (’08-’09)
- Decentralized currency
Fintech Blockchain / DLT Vision

• Bank to bank transactions (money, securities)
• Smart contracts infrastructure
• Security structuring
• Insurance
• Provenance (supply chain, art, fair trade)
• IoT micropayments
Towards a Fintech blockchain

<table>
<thead>
<tr>
<th>Reality</th>
<th>Fintech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probabilistic guarantees</td>
<td>Hard requirements</td>
</tr>
<tr>
<td>Handful tx/sec</td>
<td>Thousands tx/sec</td>
</tr>
<tr>
<td>Minutes/hours for confirmation</td>
<td>Seconds for confirmation</td>
</tr>
<tr>
<td>Problematic resource consumption</td>
<td>No “waste”</td>
</tr>
</tbody>
</table>

13 September 2017 USENIX Security 2017
PoW: Proof of Waste?

Block proves (statistically) real-world waste

- Capital expenditure
- Operational expenditure

Attacker must similarly waste resources

13 September 2017
USENIX Security 2017

https://digiconomist.net/bitcoin-energy-consumption

16 TWh!

1.4 million household
Environment-Friendly Alternatives in other settings

Permissioned system (BFT)
  • Centralized

Proof of Stake
  • needs a good solution for “nothing-at-stake”

Proof of Storage (Space)
  • consumes storage instead of computation
Achieve the robustness of PoW without the waste?
Proof of Useful Work (PoUW): Repurpose innately useful work as mining effort
Software Guard eXtension

Integrity

Other software and even OS cannot tamper with control flow.

Confidentiality

Other software and even OS can learn nothing about the internal state*.

* Modulo side channels
SGX: remote attestation

Group Signature

\[ \text{Sig} \left[ \text{SK}_{\text{sgx}, \text{fingerprint}} \right] \]

Only known to SGX

Remote entity

Untrusted Application Code

Untrusted Operating System & Hypervisor

Trusted Processor

Untrusted Hardware
SGX-backed blockchain: A new security model

• Permissionless
  • Anyone can join

• Partially decentralized
  • SGX works as advertised
  • Intel manages the group signature
Related: Proof of Elapsed Time (PoET)

• Simulate PoW by sleeping 😴.
• Consensus in partially decentralized model
• (ideally) low mining cost + offhand mining
Unaddressed challenges in PoET

Mining power not proportional to CPU value

The *Stale Chips Problem*:
- The equilibrium is to mine using old, useless devices
- Build dedicated farms

High mining cost (contrary to the original intent)
Intel’s PoET

Individual CPUs can be compromised

The *Broken Chips Problem*

Intel proposes a simple statistical test. But
1. What is the adversary’s advantage?
2. What is the cost of this test?
Proof of Useful Work

- Replace the hash calculation in PoW with “useful” mining work.
- Each unit of useful work grants a Bernoulli test.
- Similar exponential block time.
Meter the useful work

- Count CPU instructions
- Why?
  - A representative (although not perfect) metric
  - Can be done in a trustworthy way (i.e. w/o trusting OS etc.)
  - Switching to better options (if any) doesn’t change REM.
Secure Instruction Counting

- Arbitrary (malicious) programs
- Publicly verifiable
- Dynamic + static program analysis

- Enforcing $W \oplus X$ code permission
- Enforcing single-threaded enclaves
- Details in the paper
If any success, block header.

PoUW Enclave

Eval[$P'$]

$n$ instructions

Simulate $n$ Bernoulli tests

If any success

Yield similar exponential block interval.

Result of $P$
Public Verifiability

Two things to verify:

- **Validity of PoUW**
- **Compliance**
  - i.e. $P'$ is correctly instrumented
  - Requires the code of $P'$

- ✗ Put code on chain
- ✗ Predefined $P'$
- ✓ Arbitrary $P'$
Hierarchical Attestation

Compliance Checker

Alice’s Program  Bob’s Program  Carols’s Program

Validity + Compliance

PoW:
compliant:

13 September 2017  USENIX Security 2017
SGX might not be perfect!

• Individual CPU might be broken
• -> Can forge PoUW at will
• “Broken chip problem”

Picture source: https://www.forbes.com/sites/susanadams/2015/12/02/how-to-get-paid-to-do-nothing-5/#3fbee0b14eaa
Implicit PKI in SGX

Intel manages the signature group

Broken SGX CPUs cannot forge identities
Tolerating Compromised SGX CPUs

• Adversarial Model:
  • may forge PoUW at will
  • cannot forge identities

• Mitigation: statistical test
  • “If a miner is way too lucky, her block shall not be accepted.”
  • Devised rigorous framework
Advantage: adv revenue / honest revenue

Adversarial Advantage
(1 is optimal)
Cost: probability of false rejection

$\text{Waste} (P_{\text{stat}})$ vs. $t \text{ [days]}$

- Red line: Waste ($y_1$)
- Blue line: Rejected Blocks ($y_2$)

False Rejection (0 is optimal)
Performance of REM

![Performance Chart]

- Protein Folding: REM 6.5%, SGX, Native
- SVM: REM 14.4%, SGX, Native
- zlib: REM 5.8%, SGX, Native
- SHA3: REM 10.8%, SGX, Native
Conclusion

• PoUW: a **proof of useful work** scheme that avoids waste
• REM: a PoUW-based blockchain
  • Efficient: up to 15% overhead relative to native linux programs
• **Broken chip problem**: rigorous framework and effective policies.