## Our Contributions in a Nutshell

### Protocol for **view divergence**

<table>
<thead>
<tr>
<th></th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrychowicz et al, CRYPTO 2015</td>
<td>$\theta(N)$</td>
</tr>
<tr>
<td><strong>Our contribution</strong></td>
<td>$\theta(\ln N / \ln \ln N)$</td>
</tr>
</tbody>
</table>
Permissionless Distributed System

- N honest nodes
- Nodes join the system without permission
  - No central authority
  - Set of nodes and N are not known

bitcoin
Sybil Attack

Sybil Nodes

Controls

Controls
Computational Puzzle

- Non-trivial computation
  - E.g., reversing a hash function
    - Given $y$, find any $x$ such that: $\text{hash}(x) = y$
- Challenge $\rightarrow$ Solution

- Adversary has limited computational power
View divergence
View Divergence

- View divergence **breaks** the basis of many protocols
- Protocols in distributed algorithms traditionally are **permissioned** and requires same views
  - “Authenticated algorithms for byzantine agreement” (Dolev et. al, 1983)
  - “The byzantine general problem” (Lamport et. al, 1982)
  - “Protocols for secure computations” (Yao, 1982)
- Overlay protocols requires same view for bootstrapping
  - “Towards a scalable and robust DHT” (Awerbuch et al, 2009)
  - “Highly dynamic distributed computing with byzantine failures” (Guerraoui et. al, 2009)
Agree on a final, common view
## Our Contributions

- Recall $N = \text{number of honest nodes}$

<table>
<thead>
<tr>
<th></th>
<th>Running time</th>
<th>Total communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrychowicz et al, CRYPTO 2015</td>
<td>$\theta(N)$</td>
<td>$\theta(N^2)$</td>
</tr>
<tr>
<td>Katz et al, 2014</td>
<td>$\theta(N)$</td>
<td>$\theta(N^2)$</td>
</tr>
<tr>
<td><strong>Our contribution</strong></td>
<td>$\theta(\ln N / \ln \ln N)$</td>
<td>$\theta(N \ln^2 N / \ln \ln N)$</td>
</tr>
</tbody>
</table>
Our Contributions

<table>
<thead>
<tr>
<th>State-of-the-art</th>
<th>$\theta(N)$</th>
<th>$\theta(N^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our contribution</strong></td>
<td>$\theta(\ln N / \ln \ln N)$</td>
<td>$\theta(N \ln^2 N / \ln \ln N)$</td>
</tr>
</tbody>
</table>

- Alleviates bottleneck issue
  - Many security protocols have polylog complexity
    - “Towards a scalable and robust DHT” (Awerbuch et al, 2009)
    - “Highly dynamic distributed computing with byzantine failures” (Guerraoui et. al, 2009)
  - The overhead of previous $\theta(N)$ view reconciliation protocols would have been the bottleneck!
On View Divergence in BitCoin

- BitCoin does not solve view divergence
- E.g., Eclipse attack
  - “Eclipse attacks on bitcoins peer-to-peer network” (Heilman et. al, 2015)
- Our protocol together with existing overlay protocols would prevent such an attack on BitCoin!
Our Approach

- Existing protocols are deterministic
- Randomization
  - Has $\delta$ error, similar to many security protocols
    - 256-bit AES: attacker has at least $2^{256}$ probability of guessing the key correctly
  - Our complexity scales with $\log (1/\delta)$
Our Approach

- RandomizedViewReconcile (RVR)
- RVR uses randomization to obtain better performance
  - Utilize computational puzzles to elect a leader probabilistically
    - Traditionally puzzles used only to challenge computational power limitation of adversary
  - Randomized sampling and gossipping
Some Challenges

- How to handle malicious leader, missing leader, multiple leaders?
- How to spread leader’s proposal efficiently?
- No common estimate on N: How to determine when the protocol should finish?
- All results were proven, details in the paper
Conclusions

RVR solves view divergence with probability $1 - \delta$. RVR has a time complexity of $\Theta(\frac{\ln N}{\ln \ln N} \frac{1}{\ln \frac{1}{\delta}})$ and communication complexity of $\Theta(N \ln \frac{N}{\delta})$.

- We presented the first view reconciliation protocol with polylog(N) time complexity
  - Previously known protocol has $\theta(N)$ tc
- Bridges many existing permissioned security protocols to work under the permissionless settings