Beating Sybil with Resource Burning

Diksha Gupta
Ph.D. Candidate
University of New Mexico, USA
Beating Sybil with Resource Burning

Joint work with

Prof. Jared Saia

Prof. Maxwell Young
How would you judge the difference?
The Sybil Attack

John R. Douceur
Microsoft Research
johndo@microsoft.com

"One can have, some claim, as many electronic personas as one has time and energy to create."

-- Judith S. Donath [12]

Abstract — Large-scale peer-to-peer systems face security threats from faulty or hostile remote computing elements. To resist these threats, many systems employ redundancy. However, if a single faulty entity can present multiple identities, it can control a substantial fraction of the system, thereby undermining this redundancy. One approach to preventing these "Sybil attacks" is to have a trusted agency certify identities. This paper shows that, without a logically centralized entity, Sybil attacks are always possible except under unrealistic assumptions of coordination among entities.

If the local entity has no direct physical knowledge of remote entities, it perceives them only as informational abstractions that we call identities. The system must ensure that distinct identities refer to distinct entities; otherwise, when the local entity selects a subset of identities to redundantly perform a remote operation, it can be duped into selecting a single remote entity multiple times, thereby defeating the redundancy. We term the forging of multiple identities a Sybil attack [30] on the system.

It is tempting to envision a system in which established identities vouch for other identities, so that an entity can accept new identities by trusting the collective assurance of multiple (presumably independent) signatories, analogous to the PGP key of trust [37] for human entities. However, we show that, in the absence of a trusted authority (or unrealistic assumptions about the initial ownership of the identities), this approach is not secure.
What is a Sybil Attack?

- In open systems, participants are virtual identities (IDs).
What is a Sybil Attack?

- In open systems, participants are virtual identities (IDs).
- Adversarial entity can generate multiple IDs - Sybil IDs.
What is a Sybil Attack?

- In open systems, participants are virtual identities (IDs).
- Adversarial entity can generate multiple IDs - Sybil IDs.
- Obtains unfair share of the network’s resources.
ELECTRUM BOTNET STEALS $4.6 MILLION IN BITCOIN, CRYPTOCURRENCIES

Specifically, attackers took advantage of the fact that anyone could operate on the network as a public Electrum peer. Attackers then launched what's called a Sybil attack that introduces compromised nodes into the network. The result of such an attack was that hundreds of thousands of computers have been compromised through the false security update and other means shown at the start of this article.
Existing Defenses

• Continuously solve puzzles to limit number of Sybil IDs
Existing Defenses

- Continuously solve puzzles to limit number of Sybil IDs
- Proof-of-Work (PoW) commonly used for cryptocurrencies
Existing Defenses

- Continuously solve puzzles to limit number of Sybil IDs
- Proof-of-Work (PoW) commonly used for cryptocurrencies
Existing Defenses

- Continuously solve puzzles to limit number of Sybil IDs
- Proof-of-Work (PoW) commonly used for cryptocurrencies

WHAT IF I TOLD YOU

NOTHING IS FREE
Resource Burning

- Verifiable consumption of resources. E.g.;
  - Proof of Work
  - Proof of Space & Space-Time
  - Proof of Useful Work
  - Radio Resource Testing
Resource Burning

• Verifiable consumption of resources. E.g.;
  • Proof of Work
  • Proof of Space & Space-Time
  • Proof of Useful Work
  • Radio Resource Testing

Proof-of-Stake is Not!
Our Result

**Theorem:** Let $T$ be adversarial spend rate and $J_G$ be good join rate. Then, ERGO limits Sybil IDs with an algorithm spend rate:

$$O(J_G + \sqrt{J_G T})$$
Our Result

**Theorem:** Let $T$ be adversarial spend rate and $J_G$ be good join rate. Then, ERGO limits Sybil IDs with an algorithm spend rate:

$$O(J_G + \sqrt{J_G T})$$

Parameters attached to Churn
Still allows for exponential change in system size.
Our Model
Our Model

• System consists of $n$ good IDs and Sybil IDs

- Good IDs
  - Follow the protocol.

- Sybil IDs
  - Can deviate arbitrarily from the protocol, under the control of an adversary.
Our Model

- System consists of \( n \) good IDs and Sybil IDs.
- Adversary has \( \kappa \) fraction of resources.
Our Model

- System consists of $n$ good IDs and Sybil IDs.
- Adversary has $\kappa$ fraction of resources.
- Churn - IDs can join and depart.
Our Model

- System consists of $n$ good IDs and Sybil IDs.
- Adversary has $\kappa$ fraction of resources.
- Churn - IDs can join and depart.
- Adversary can’t target specific good IDs.
Our Model

- System consists of $n$ good IDs and Sybil IDs.
- Adversary has $\kappa$ fraction of resources.
- Churn - IDs can join and depart.
- Adversary can’t target specific good IDs.
- System size is always “sufficiently large”.
Problem Statement

- **Good** IDs always know a set $S$ that contains
Problem Statement

- **Good** IDs always know a set $S$ that contains
  - All good IDs
  - At most $c\kappa < 1/2$ fraction of bad IDs
What if I told you we have an oracle for $J_G$?
Initially

Reduce the fraction of bad IDs to 1/3.
New IDs Join

New IDs solve Entrance Puzzles of difficulty $\max(\frac{J}{J_G}, 1)$. 
New IDs Join

Join + Departures $\geq |S_i|/3$
Purge

Solve a puzzle or be evicted.
Iteration Ends

Reduce the fraction of bad IDs to $\kappa$. 
Intuition for Entrance Cost

Fix an iteration:

• T - Resource Budget of the adversary
• J - Join rate of IDs
Intuition for Entrance Cost

Fix an iteration:

• T - Resource Budget of the adversary
• J - Join rate of IDs
• Assume: T = XJ
Intuition for Entrance Cost

Fix an iteration:

• $T$ - Resource Budget of the adversary
• $J$ - Join rate of IDs
• Assume: $T = XJ$
  • Good spend rate for entrance: $XJ_G$
Intuition for Entrance Cost

Fix an iteration:

• T - Resource Budget of the adversary
• J - Join rate of IDs
• Assume: \( T = XJ \)
  • Good spend rate for entrance: \( XJ_G \)
  • Good spend rate for purges: J
Intuition for Entrance Cost

Fix an iteration:

• T - Resource Budget of the adversary
• J - Join rate of IDs
• Assume: $T = XJ$
  • Good spend rate for entrance: $XJ_G$
  • Good spend rate for purges: J
  • Setting $J = XJ_G$
Intuition for Entrance Cost

Fix an iteration:

- **T** - Resource Budget of the adversary
- **J** - Join rate of IDs

Assume: $T = XJ$

- Good spend rate for entrance: $XJ_G$
- Good spend rate for purges: $J$
- We get: $X = J/J_G$
Issue:
Don’t know $J_G$ in advance
Challenge: Estimating $J_G$

- Cannot differentiate between Good from Bad IDs
Challenge: Estimating $J_G$

- Cannot differentiate between Good from Bad IDs
- Developed an algorithm to estimate join rate of good IDs assuming:
  - $\alpha, \beta$ smoothness parameters for churn
  - Always a majority of Good IDs
Churn Model
Definition: Epoch

Duration over which set of good IDs ($G_t$) changes by:

$$|G_t \oplus G_{t'}| \geq \frac{3}{4} |G_t|$$

For some $t$ and $t' > t$. 
$\alpha, \beta$-Smoothness

$\rho_j$ : join rate of good IDs in epoch $j$
\( \rho_j : \) join rate of good IDs in epoch \( j \)

- Join rate of good IDs changes by at most \( \alpha \) between successive epochs.

\[
\frac{\rho_{j-1}}{\alpha} \leq \rho_j \leq \alpha \rho_{j-1}
\]
$\alpha, \beta$-Smoothness

$\rho_j$ : join rate of good IDs in epoch $j$

- Let $\ell$ : length of some duration in epoch $j$

  $n_a$: number of good IDs that join

  $n_d$: number go good IDs that depart
\( \alpha, \beta \)-Smoothness

\( \rho_j \): join rate of good IDs in epoch \( j \)

- Let \( \ell \): length of some duration in epoch \( j \)
  
  \( n_a \): number of good IDs that join
  
  \( n_d \): number of good IDs that depart

\[
\left\lfloor \frac{\ell \rho_j}{\beta} \right\rfloor \leq n_a \leq \lceil \beta \ell \rho_j \rceil \quad \text{and} \quad n_d \leq \lceil \beta \ell \rho_j \rceil
\]
Estimating $J_G$
Interval Begins

$S_t$
IDs Join and Depart

Wait until $|S_t \oplus S'_t| \geq \frac{5}{8} |S'_t|$.
Set $\tilde{J}_G = \frac{|S_r|}{t - t'}$. 

Interval Ends
Intuition for $\tilde{J}_G = \frac{|S_{t'}|}{(t' - t)}$

$S_t$ $\cdots$ $S_{t'}$

Bad ID < 1/2
Intuition for $\tilde{J}_G = |S_{t'}|/(t' - t)$

Since $|S_t \oplus S_{t'}| \geq \frac{5}{8}|S_{t'}|$, at least $1/8$ fraction of good join events.
Our Result

**Theorem:** Fix an interval. Let $J_G$ be true join rate of good IDs rate, and $\tilde{J}_G$ be the estimate of good join rate. Then:

$$\frac{J_G}{78} \leq \tilde{J}_G \leq 14\beta^2 J_G$$
Empirical Result
Empirical Results

• Four data sets: Bitcoin, Ethereum, Gnutella and BitTorrent
Empirical Results

- Four data sets: Bitcoin, Ethereum, Gnutella and BitTorrent
- For a given:
  - Adversarial spending rate, T:
  - Population Invariant: Max. Fraction of Sybil IDs in the system at any time.
Empirical Results

• Four data sets: Bitcoin, Ethereum, Gnutella and BitTorrent

• For a given:
  • Adversarial spending rate, $T$:
  • Population Invariant: Max. Fraction of Sybil IDs in the system at any time.

• Measured:
  • Approximation Factor: \( \frac{\tilde{J}_G}{J_G} \)
Empirical Result

![Graph showing approximation factor vs. population invariant for different values of T: T = 1/100, T = 1, and T = 100.]
SO WHAT?
Our Result

**Theorem:** Let $T$ be adversarial spend rate and $J_G$ be good join rate. Then, ERGO limits Sybil IDs with an algorithm spend rate:

$$O(J_G + \sqrt{J_G T})$$
Empirical Results

- Four data sets: Bitcoin, Ethereum, Gnutella and BitTorrent
- ERGO against existing techniques:
  - CCom - ERGO with entrance cost of 1.
  - SybilControl: Puzzle every 5 seconds.
  - REMP: Puzzle every $x$ seconds, where $x$ is based on upper bound of adversary power
Empirical Results

Whoa!
Future Work

• Lower bounds for Resource Burning
  • Existing LB for algorithms with periodic purges
  • Need a Generic LB
Questions!
Not RB: Proof of Stake

• Used in: Algorand, Ouroboros, Ethereum
• Proof of Stake is a measurement
• ID’s stake must be known
Committee

Logarithmic size

Use state-machine replication to get committee to act in concert

After every purge, old committee elects a new committee from set of current IDs, using Byzantine-resilient coin-flipping