PEACE THROUGH SUPERIOR PUZZLING: AN ASYMMETRIC SYBIL DEFENSE

Diksha Gupta, Jared Saia and Maxwell Young
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 agency, Sybil attacks are always possible except in a few special cases, and 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 signatory scheme. However, recent work [37] has shown that, in the absence of trusted third parties, even an entity (or unrealistic assumptions of security (or unrealistic assumptions of security by an attacker), a Sybil attack on the system can be performed. The following sections explore the initial
• In an open system, an entity may claim **multiple identities** in order to gain **unfair share** of the network’s resources.

• Specially challenging in the absence of a central certifying authority.
Real-World Sybil Attacks in BitTorrent Mainline DHT

Liang Wang
Dept. of Computer Science
University of Helsinki
Helsinki, Finland

Jussi Kangasharju
Helsinki Institute for Information Technology
University of Helsinki
Helsinki, Finland

Abstract—Distributed hash tables (DHT) are a key building block for modern P2P content-distribution systems, for example in implementing the distributed tracker of BitTorrent Mainline DHT. DHTs, due to their fully distributed nature, are known to be vulnerable to certain kinds of attacks and different kinds of defenses have been proposed against these attacks. In this paper, we consider two kinds of attacks on a DHT, one already known and one new kind of an attack, and show how they can be targeted against Mainline DHT. We complement this by an extensive measurement study using honeypots which shows that attacks have been going on for a long time in the network, but still happening. We present numbers showing the number of sybils in the Mainline DHT network is currently around 300,000. We analyze the attacks and propose simple fixes.

- We present two possible routing table attacks on the system: horizontal and vertical attack.
- We analyze the damages to network attacks and their potential targets.
- Through extensive analysis of these,
PageRank is vulnerable to Sybil Attacks

One of the first things that struck me while learning about the PageRank algorithm is the following problem: What happens if you just make a lot of fake pages to boost your ranking? It seemed clear that this would be an effective strategy, and indeed it is [1].

Consider a Sybil attack. One sort of case you might be familiar with is the case of dummy nodes in a network in order to boost a reputation in Reddit. Sybil attacks are an open component to Facebook and are advertised impossible to beat without the

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.
Renewed interest due to the success of cryptocurrencies such as Bitcoin, Ethereum.

Computational Puzzles are moderately hard mathematical problem, solving which requires measurable computational effort but can be verified efficiently.
Renewed interest due to the success of cryptocurrencies such as Bitcoin, Ethereum.

Computational Puzzles are moderately hard mathematical problem, solving which requires measurable computational effort but can be verified efficiently.

Popular systems require solutions to be generated continuously in order to provide security guarantees.
Bitcoin Energy Consumption w.r.t several countries

Energy Consumption by Country Chart

TWh per Year

Qatar
Hungary
New Zealand
Bitcoin
Peru
Hong Kong
Iraq

BitcoinEnergyConsumption.com
PoW is expensive!
OUR MODEL
THE SYSTEM

Good IDs
- Follow the protocol.

Bad IDs
- Can deviate arbitrarily from the protocol, under the control of an adversary.
• **Random Oracle Assumption**: We have a function, h, such that \( h(x) \) is chosen uniformly at random on \((0, 1]\) the first time bit string \(x\) is input to \(h\).

• **Computational Cost**: Number of times \(h\) is called.
Diffuse protocol:

- Every time a good ID sends a message, it is delivered to all other good IDs.
- Communication time is negligible compared to computation time.
- Has at most a small constant fraction of the computational power of the system.

- Knows our algorithms, not our random bits.
OUR RESULTS
**Theorem:** Let $T$ be the rate of adversarial spending and $J$ be the rate of join of good IDs. Then:

- There is always an honest majority.
- The rate of algorithmic spending is $O(J + \sqrt{JT})$. 
- We consider **purge-based algorithms**: Periodically all IDs must solve puzzles.

- Duration between periodic puzzling is called an **epoch**.
**Lower Bound:** For any purge-based algorithm, there is an adversarial strategy ensuring the following for any epoch: The algorithmic spending rate is $\Omega(J + \sqrt{JT})$, where the algorithmic spending rate $T$, and $J$ are taken over the epoch.
NAIVE APPROACH
INITIALLY...

Reduce the fraction of bad IDs to 1/3 using [1].

NEW ID JOINS

Here is my 1-difficult puzzle solution.
ONLY JOINS

1/3 fraction of new IDs join.
PURGE

Solve a puzzle or be ejected.
Reduce the fraction of bad IDs to $\alpha < 1/3$. 

ITERATION ENDS
NAIVE APPROACH

Total computational cost to algorithm: $O(T + J)$
Can we do better?
Fix an iteration $i$. Suppose

- $T$: adversarial rate of spending
- $J$: actual join rate
- $J_g$: join rate of good IDs
- $\chi$: average entrance cost
- $\ell$: length of iteration
Assume $T = \mathcal{X} J$

Total entrance cost to good IDs: $\theta(\mathcal{X} J_g \ell)$

Total purge cost to good IDs:

$|G| \leq |S| = \theta(J \ell)$

Spending rate of good IDs:

$\theta(J + \mathcal{X} J_g)$
Assume $T = XJ$

Spending rate of good IDs: $\theta(J + XJ_g)$

To balance, we get:

$$X = \frac{J}{J_g}$$
Suppose computational power with the adversary is $\alpha$. Then:

$$J_g = \frac{\alpha |S|}{L}$$
CANDIDATE JOIN RATE ESTIMATE

\[ J_g = \frac{\alpha |S|}{L} \]

where

- **L** - Duration over which at least \(2\alpha\) - fraction of current set of IDs joined
- **S** - Set of IDs at the end of iteration
Reduce the fraction of bad IDs to $\alpha$. 

\[ t_0 \quad t_k \quad t'_k \]
ENTRANCE COST FUNCTION

\[ e(k) = \frac{J}{J_g} \]

where

- \( e(k) \) - entrance cost to the \( k^{th} \) ID joining in the current iteration
- \( J \) - join rate in the current iteration
- \( J_g \) - estimated join rate of good IDs
$e(k) = \frac{J}{J_g}$

Here is my $e(3)$-difficult puzzle solution.
Solve a puzzle or be ejected.
EMPIRICAL STUDY
EMPIRICAL SETUP

- Compare the performance of our algorithm - GMCom, against 2 computational puzzles based algorithms:
  ‣ CCom
  ‣ Sybil Control

- Over 4 different p2p network churn data sets:
  ‣ Bitcoin
  ‣ Skype Supernodes
  ‣ BitTorrent - Debian
  ‣ BitTorrent - FlatOut
EMPIRICAL RESULTS
EMPIRICAL RESULTS
FUTURE WORK
DHT

- Design Sybil resistant DHT using computational puzzles.

- Challenges:
  - Maintaining correctness properties in the presence of adaptive adversarial churn
  - Preventing eclipse attack
Thank You!

Questions?