



The Sybil Attack

John R. Douceur
Microsoft Research

Presented for Cs294-4 by
Benjamin Poon

1



Outline

- Background
- Motivation
- Model
- Lemmas
- Conclusion

Cs294-4

Benjamin Poon
bpoon@uclink.berkeley.edu

2



Outline

- **Background**
- **Motivation**
- Model
- Lemmas
- Conclusion

Cs294-4

Benjamin Poon
bpoon@uclink.berkeley.edu

3



Background

- P2P systems use multiple, independent entities to mitigate possible damage by other hostile entities
 - Replication
 - Computations
 - Storage
 - Fragmentation
 - Protects against privacy violations
- Sybil Attack
 - Attacker can assume multiple identities
 - Aims to control substantial fraction of system

Cs294-4

Benjamin Poon
bpoon@uclink.berkeley.edu

4

Motivation (1/2)

- Must protect against Sybil Attack
 - Using replication or fragmentation requires ability to determine if entities are really different
- Paper claims Sybil attacks always possible except under extreme, unrealistic assumptions
 - Need logically centralized authority

Motivation (2/2)

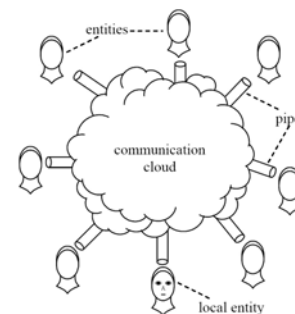
- Centralized authority possibilities
 - VeriSign
 - Explicit certification
 - CFS (cooperative storage)
 - Identities assigned using hash of IP address
 - Problem?
 - SFS (network file system)
 - Identities assigned using host identifier + DNS name
 - EMBASSY (multiparty trust system)
 - Identities assigned using hardware-embedded cryptographic keys
 - Issues?

Outline

- Background
- Motivation
- **Model**
- Lemmas
- Conclusion

Model – Overview

- Generic distributed computing environment



Entities E

- Correct C union Faulty F = E
- Send messages

Pipe

- Messages → cloud

Messages

- Uninterrupted, finite-length bit string

Cloud

- Broadcast
- Bounded time
- Guaranteed
- Unordered

Model – Definitions

- Identity
 - Abstract representation that persists across multiple communication events
 - Entities perceive other entities through *identities*
- Present
 - Each entity *presents* an identity to other entities in the system
- Local entity L
 - A specific entity that results are stated with respect to
- Accept
 - If an entity *e* successfully presents an identity *i* for itself to L, L *accepts* *i*

Model – Characteristics

- General
 - Leaves internals of cloud unspecified
 - Includes any topology/geometry
- Friendly
 - Limits obstructive power of corrupt entities
 - DoS attacks not possible
 - Strengthens negative results
- Q: Any drawbacks of the model?

Model – Resources

- Entities can perform operations with complexity polynomial in n
 - Allows public-key cryptography
 - Entities can establish point-to-point communication

Model – Main Idea

- Each entity *e* attempts to present one legitimate identity
- Each faulty entity *f* additionally attempts to present one or more counterfeit identities
- Goal: system should accept all legitimate identities, zero counterfeit

Outline

- Background
- Motivation
- Model
- Lemmas
- Conclusion

Lemmas

- Four lemmas
 - Collectively show impracticality of establishing distinct identities in large-scale distributed system
 - Proofs trivial (refer to paper)
- In absence of trusted authority, entities accept identities only when identity is:
 1. Validated by entity itself (direct validation)
 - Lemmas 1 and 2
 2. Vouched for by other already validated identities (indirect validation)
 - Lemmas 3 and 4

Lemma 1: Resources

- Let
 - \min = minimally capable entity
 - R_x = resources of x
 - $\rho = R_f / R_{\min}$
- f can present $\text{floor}(\rho)$ distinct identities to L

Lemma 1: Resources

- Gives lower bound on damage achievable by f
- Achieve upper bound by exploiting limitations in resources
 - Communication: L broadcasts request for identities and accept replies that come within given time interval
 - Storage: L challenges identities to store large amount of unique, uncompressible data
 - Computation: L challenges identities to solve unique computational puzzle

Lemma 1: Resources

- Computational puzzle example
 - Generate large random value y
 - Challenge identity to find (in limited time) pair of values x and z such that least significant n bits of $\text{hash}(x \mid y \mid z) = 0$

Given y , find x, z s.t. $\text{LSB}_n(\text{hash}(x \mid y \mid z)) = 0$

- Time to solve proportional to 2^{n-1}
- Time to verify constant

Lemma 2: Concurrency

- If L accepts entities not validated simultaneously
 - f presents a distinct identity to L using R_f
 - R_f is freed and f repeats process
 - Single f can present many counterfeit identities to L

Lemma 2: Concurrency

- Works for temporal resources (computation and communication), not storage
 - L can indefinitely extend challenge duration: periodically demand different data excerpts
 - Challenge consumes R , so real work limited

Lemma 3: Resources

- If L accepts any identity vouched for by q accepted identities, a group F can present many counterfeit identities to L if either:
 - Size of group $F \geq q$
 - $R_F \geq$ resources taken by $q + |F|$ minimally capable entities

Lemma 4: Concurrency

- If correct entities C do not coordinate time intervals to accept identities, and if L accepts any identity vouched for by q accepted identities
 - Minimally capable f can present $\text{floor}(|C| / q)$ counterfeit identities to L

Lemma 4: Concurrency

- Shows need for multiple entities in C to issue concurrent challenges
- May or may not be possible depending on resource
 - Communication: possible because of broadcast cloud
 - Storage: information theory says “probably not”
 - Computation: possible by combining puzzles

Lemma 4: Concurrency

- Simultaneous computational puzzle example
 - Same puzzle, but has m of them to solve
- Given m puzzles y_1, y_2, \dots, y_m , find w s.t.
 $\text{LSB}_n(\text{hash}(0 | y_1 | y_2 | \dots | y_m | w)) = 0$
- Solution to each puzzle y_k is
 $x_k = 0 | y_1 | y_2 | \dots | y_{k-1}$ and
 $z_k = y_{k-1} | \dots | y_m | w$
- If validating entity challenges m identities all made by one f, then f can use this method
 - Validating entity can check if this happens by

$$x_1 | y_1 | z_1 = x_2 | y_2 | z_2$$

Outline

- Background
- Motivation
- Model
- Lemmas
- **Conclusion**



Conclusion

- Without centralized authority, Sybil attacks always possible except when:
 - All entities have nearly identical resources
 - All presented identities are validated simultaneously
 - When accepting identities not directly validated, required number of vouchers exceeds number of system-wide failures
- Not justifiable as assumptions
- Not practically realizable as requirements



Pros/Cons

- Pros
 - General model for distributed computing environments with well thought-out reasons behind design
 - Good reminder to keep Sybil attacks in mind when designing large-scale distributed systems
- Cons
 - Leaves the fact that faulty nodes can solve multiple puzzles by combining them