Security Considerations for Peer-to-Peer Distributed Hash Tables

Emil Sit and Robert Morris

Laboratory for Computer Science, MIT

presented by Markku Kekkonen

Introduction

- what problems do large peer-to-peer systems based on hash lookup systems have?
- focus on attacks that prevent data retrieval
- presented by examples, proposing defenses

Background

- Lookup protocols have common basic components:
  - a key and node identifier space
  - rules for associating keys to particular nodes and for updating routing tables
  - per-node routing tables
- Storage layer must maintain some invariants

  => responsible for storing, replicating etc. the data

Attacks and defenses

- Different kinds of attacks:
  - Routing attacks
    - Incorrect lookup routing
    - Incorrect routing updates
    - partition
  - Storage and retrieval attacks
  - Miscellaneous attacks
    - Inconsistent behaviour
    - Overload of targeted nodes
    - Rapid joins and leaves (churn)

Routing attacks and defenses (1/2)

- In a DHT routing is critical => define verifiable system invariants and verify them

- Attack: Incorrect lookup routing
  - Malicious node could forward lookups to a non-existent or incorrect node
- *Defense: Allow the querier to observe the lookup process and assign keys to nodes in a verifiable way (eg. cryptographic hash of its IP address and port)*

Routing attacks and defenses (2/2)

- A: Incorrect routing updates
  - Malicious node misleads innocent nodes and causes them to send misdirected queries
- *D: Reachability must be confirmed, server selection in routing may be abused*
- A: Partition
  - A new node may inadvertantly join a parallel all-malicious network
- *D: Cross-check routing tables using **random** queries*

Storage and retrieval attacks and defenses

- A: Denying the existence of data or refusing to give data
- *D: Avoid single points of responsibility and consult multiple replicas*

Miscellaneous attacks and defenses (1/2)

- A: Inconsistent behaviour
  - A node acts maliciously only to part of the network
- *D: use of public keys*
- A: Overload of targeted nodes (DoS)
- *D: node identifiers must be distributed randomly and replicas kept in physically disparate locations*

Miscellaneous attacks and defenses (2/2)

- A: Rapid joins and leaves (churn)
  - unnecessary rebalancing => excess traffic
- *D: Balancing between replication and overloading*

- A: Unsolicited messages
- *D: random nonce, digital signatures*

Conclusions

- Design principles derived from defenses for discussed attacks:
  - Define verifiable system invariants (and verify them!)
  - Allow the querier to observe lookup process
  - Assign keys to nodes in a verifiable way
  - Server selection in routing may be abused
  - Cross-check routing tables using random queries
  - Avoid single points of responsibility!