Authentication in Peer-to-peer Systems

Vivek Pathak
Outline

- Introduction
  - Public key authentication
    - Existing models
- Motivation for Peer-to-peer authentication
  - Other solutions
- Byzantine fault tolerant authentication
  - Security model
  - Outline of correctness and performance
- Future work
Public Key Encryption

- Public-private key pair
- Bootstrap shared secret encryption
- Validation of digital signature
Authentication of Public Keys

- Mapping identities to public keys
  - Trusted third parties (TTP)
    - Certificate authority (CA)
  - Web of trust
    - PGP

```
$ssh eden.rutgers.edu
The authenticity of host 'eden.rutgers.edu (128.6.68.10)' can't be established.
Are you sure you want to continue connecting (yes/no)? [no]
```
Authentication through CA

- Provide public key certificate
- Use secure channel for bootstrapping
Authentication through CA
Authentication through CA

- Represent centralized aggregation of trust
  - Long lived CA keys
  - Single point of failure

- Public key revocation
  - Scalability with number of certified keys
Web of Trust

- Informal human authentication
  - PGP key rings
  - Levels of trust
Web of Trust

- Peers take on role of CA

- Decentralized trust
  - No single point of failure
  - Key authentication depends on human connections

- How to apply to autonomous systems
  - Sophisticated users
Outline

- Introduction
  - Public key authentication
    - Existing models
- Motivation for Peer-to-peer authentication
  - Other solutions
- Byzantine fault tolerant authentication
  - Security model
  - Outline of correctness and performance
- Future work
Characteristics of Peer-to-peer Systems

- Heterogeneous peers
  - Lack of trusted third parties
  - Hierarchical Certificate Authorities

- Large scale peer-to-peer systems
  - Need decentralized solution
  - Administrative burden on CA
  - Scalability of key revocation
Characteristics of Peer-to-peer Systems

- Autonomous operation
  - Unsophisticated users
  - Sensors and devices
  - Web of trust depends on constant human feedback

- Short lived public keys
  - Peers may be attacked and recover
  - Public key certificates require secure channel

- Malicious peers
Other Solutions

- Threshold encryption systems
  - Share the secret among a set of parties
  - Defend against a few compromised parties
  - Secure initialization phase

- Crypto based network IDs
  - Choose ID as function of public key
  - Depends on the routing infrastructure
Outline

- **Introduction**
  - Public key authentication
    - Existing models

- **Motivation for Peer-to-peer authentication**
  - Other solutions

- **Byzantine fault tolerant authentication**
  - Security model
  - Outline of correctness and performance

- **Future work**
System Model

- Mutually authenticating peers
  - Associate network end-point to public key

- Asynchronous network
  - No partitioning
  - Eventual delivery after retransmissions

- Disjoint message transmission paths
  - Man-in-the-middle attack on Ø fraction of peers
Attack Model

- Malicious peers
  - Honest majority
  - At most $t$ of the $n$ peers are faulty or malicious peers
  where $t = \frac{1-6\theta}{3} n$

- Passive adversaries

- Active adversaries
  - Relax network-is-the-adversary model
    - Unlimited spoofing
    - Limited power to prevent message delivery
Authentication Model

- Challenge-response protocol
  - No active attacks

- Man in the middle attack
  - Limited number of attacks

- Proof of possession of $K_a$
  \[
  \{b,a,\text{Challenge},K_a(r)\}_b, \{a,b,\text{Response},r\}_a
  \]
Authentication Model

- Distributed Authentication
  - Challenge response from multiple peers
  - Gather proofs of possession
- Lack of consensus on authenticity
  - Malicious peers
  - Man-in-the-middle attack
Authentication Correctness

- Validity of proofs of possession
  - \( \{e, a, \text{Challenge}, K_a(r)\}_e \), \( \{a, e, \text{Response}, r\}_a \)

- All messages are signed
  - Required for proving malicious behavior
  - Recent proofs stored by the peers

<table>
<thead>
<tr>
<th>From peers</th>
<th>( P_B )</th>
<th>( P_C )</th>
<th>( P_D )</th>
<th>( P_E )</th>
<th>( P_F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>From A</td>
<td>( P_B )</td>
<td>( P_C )</td>
<td>( P_D )</td>
<td>( P_E )</td>
<td>( P_F )</td>
</tr>
</tbody>
</table>
Byzantine Agreement Overview

- Publicize lack of consensus
  - Authenticating peer sends proofs of possession to peers

- Each peer tries to authenticate A
  - Sends its proof-of-possession vector to every peer
  - Byzantine agreement on authenticity of $K_A$

- Majority decision at every peer
  - Identify malicious peers
  - Complete authentication

<table>
<thead>
<tr>
<th>From B</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>From C</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>From D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>From E</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>From F</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Byzantine Agreement Correctness Overview

- Consider proofs received at a peer $P$

Set of Peers of $P$
$t$ malicious peers

$\Phi_n$ on compromised path to $A$
$\Phi_n$ on compromised path to $P$
Byzantine Agreement Correctness
Overview

- $t + 2\theta n$ may not arrive
  - $P$ receives at least $n-t-2\theta n$ proofs

- $t + 2\theta n$ may be faulty
  - $P$ receives at least $n-2t-4\theta n$ correct agreeing proofs
  - $P$ decides correctly by majority if $n-2t-4\theta n > t + 2\theta n$

- Agreement is correct if $t < \frac{1-6\theta}{3} n$
Trust Groups

- Execute Authentication on smaller Trust groups
  - Quadratic messaging cost
  - Peer interest

- Trusted group
  - Authenticated public keys
  - Not (overtly) malicious

- Probationary group

- Un-trusted group
  - Known to be malicious
Growth of Trust Groups

- Governed by communication patterns
- Discovery of new peers
  - Authentication of discovered peers
  - Addition to trusted set
- Discovery of un-trusted peers
Evolution of Trust Groups

- Covertly malicious peers
  - May wait until honest majority is violated
  - Lead to incorrect authentication

- Periodic pruning of trusted group
  - Unresponsive peers
  - Remove older trusted peers from trust group
    - Reduce messaging cost
    - Randomize trusted group membership
  - Group migration event

- Probability of violating honest majority
Bootstrapping Trust Group

- Authentication needs an honest trust group
  - Initialize a Bootstrapping trust group
  - Needed for cold start
  - Authenticate each bootstrapping peer

- Size of bootstrapping trust group
  - Recover from trusting a malicious peer
    \[ n > \frac{3}{1-\delta} \]
Public Key Infection

- Optimistic trust
  - Lazy authentication
  - Reduced messaging cost

- Cache of undelivered messages
  - Use peers for epidemic propagation of messages
  - Anti-entropy sessions eventually deliver messages
  - Infect peers with new undelivered messages
Public Key Infection

- Use logical and vector timestamps
  - Determine messages to exchange for anti-entropy
  - Detect message delivery

- Double exponential drop in number of uninected peers with time

- Number of cached messages is in $O(n \log n)$
Simulation

- Implemented Byzantine Fault Tolerant Authentication as a C++ library

- Simulation program
  - Make library calls and keeps counters
  - Study effects of
    - Group size
    - Malicious peers
Effects of Group Size

- Constant Cost for trusted peers
- Probationary peers process $O(n^2)$ messages
- Trust graph does not affect the cost
  - Randomized trusted sets from Bi-directional trust
Effects of Malicious Peers

- Rapid increase of messaging cost
  - With group size
  - With proportion of malicious peers
- Byzantine agreement has quadratic messaging cost
Conclusion

- Autonomous authentication without trusted third party
  - Incremental approach to security
  - Suited for low value peer-to-peer systems

- Tolerate malicious peers
  - Suited for applications spanning multiple administrative domains

- Scalable to large peer-to-peer systems

- Eliminate total trust and single point of failure

- Made feasible by using stronger network assumptions
  - Network adversary is not all powerful
Outline

- Introduction
  - Public key authentication
    - Existing models

- Motivation for Peer-to-peer authentication
  - Other solutions

- Byzantine fault tolerant authentication
  - Security model
  - Outline of correctness and performance

- Future work
Future Work

- Enhancement of the model for Ad-hoc networks
  - Lack the network IDs assumed
  - Apply to vehicular computing
    - Does the key belong to the car stalled on GWB?

- Applications
  - Provide key authentication capability to Open-SSH
  - Other peer-to-peer systems
Future Work

- Enhancements to Byzantine fault tolerant authentication mechanism
  - Address denial of service
  - Avoid expensive public key cryptography

- Study hybrid trust models
  - Hierarchical, peer-to-peer, web of trust
Authentication Protocol

[Diagram showing a protocol with nodes A, B, C, D, and E.]

- Encrypted Nonce
- Recover and Sign Nonce
- Unauthenticated public key at peer A
- Challenge response pairs sent by peers
- Public key of A authenticated to B
- Challenge response pairs sent by A
- B identifies the malicious peer D
- If no consensus on authenticity
- Set of peers with honest majority

Authentication in Peer-to-peer Systems