Peer-to-peer Sender Authentication for Email

Vivek Pathak and Liviu Iftode
Rutgers University
Email Trustworthiness

- Sender can be spoofed
Need for Sender Authentication

- Importance depends on sender
Update on Spam Filters

- Circumvention of content based spam classification
- False positives
End-to-end Issues

- Can the mail server decide importance for the receiver?
Characteristics of Email

- Social networks of collaborating users
  - Limited trust infrastructure

- Usability expectation
  - Automatic authentication

- Asynchronous
  - Delayed authentication is better than none
Outline

- Byzantine fault tolerant public key authentication
  - Basis of sender authentication for email

- Application to Email
  - Thunderbird sender authentication plugin

- Usability
  - Micro-benchmark
  - Simulation on University and Industry mail trace
Public-key Authentication 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 $t = \frac{1-6\delta}{3} n$

- Passive adversaries

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

- Challenge-response protocol
  - No active attacks

- Man in the middle attack
  - Limited number of attacks

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

- Distributed Authentication
  - Challenge response from multiple peers
  - Gather proofs of possession

- Lack of consensus on authenticity
  - Malicious peers
  - Man-in-the-middle attack

- Detect and correct through Byzantine agreement on authenticity of $K_A$
Scalability of Authentication

- Authentication cost and group size
  - Scale to large peer-to-peer network
    - Operate on local trusted group
  - Tolerate bad group selection
    - Periodic recycling of group members
    - Eventual authentication

- Operate through epidemic algorithm
  - Eliminate direct connectivity requirement
  - Improve messaging cost
Outline

- Byzantine fault tolerant public key authentication
  - Basis of sender authentication for email
- Application to Email
  - Thunderbird sender authentication plugin
- Usability
  - Micro-benchmark
  - Simulation on University and Industry mail trace
Sender Authentication Design

Backward Compatibility
- SMTP ignores user defined fields
- Operate as an overlay on SMTP
Overlay Limits

- Size limit 32 Kb on SMTP header
- High compression for XML format protocol messages
  - 300 message limit

![Graph showing protocol message size and compressed size](image)
About 20% emails are to new peers
Trusted Group Size

- Authentication messages per email
  - System limitation 300

- Peers to authenticate per email
  - Mailbox observation 1/5

- Quota of 1500 messages per peer
  - Protocol messaging cost analysis
  - Trusted group size limit 75
Sender Authentication Plugin

- Thunderbird mail client
  - XPCOM layer
    - Implements Public-key authentication

- Javascript layer
  - Transfer protocol messages to and from SMTP extension fields
Sender Authentication Plugin

Thunderbird Mail Client User Interface

Events

Scripted Extension Access

Authentication Interface

Shared Object

nsISupports

Authentication Adapter XPCOM

Byzantine Fault Tolerant Authentication Library

Authentication Data
Bootstrapping Trusted Group

- University mail trace shows Receiving bias
Bootstrapping Trusted Group

- Required for automatic operation

- Select trusted group
  - Two-way
  - Outgoing

- Selected 53 peers with 10 or more trusted peers using Two-way rule
Implementation Status

- Email application
  - Automatic sender authentication
  - Overlay authentication protocol on SMTP

- Available as Thunderbird extension module
  - Tested on 32bit and 64bit Linux
  - http://discolab.rutgers.edu/sam
Implementation

Screenshot
Outline

- Byzantine fault tolerant public key authentication
  - Basis of sender authentication for email

- Application to Email
  - Thunderbird sender authentication plugin

- Usability
  - Micro-benchmark
  - Simulation on University and Industry mail trace
Micro-benchmarks

- Record the processing time overhead
  - Average over multiple messages

- Operational parameters
  - Public key length
  - Trusted group size
Overhead with Trusted Group Size

- Increasing on Sender
  - Serialization and compression of larger messages
Overhead with Key Length

- Increasing on receiver
  - Digital signature verification
  - Responding to challenges
Micro-benchmark Summary

- Sending path overhead of 250ms
- Receiving path overhead of 500ms
  - Can be done asynchronously
- Acceptable level of overhead
Simulation Study

- Process the entire email trace on a single machine
  - Anonymous log records from mail server
  - Exact times have been removed

- University trace of 92 days and 1.19M messages

- Industry trace of 56 days and 2.5M messages
Overhead on Email Size

- Recover the designed 10KB overhead
Disk Space Usage

- Epidemic algorithm overhead
  - Trusted group size is 100
  - Overhead about 10MB per peer
Partial completion on 92 day trace
- About 40% of peers authenticated
Completion of Authentication Industry Trace

- Reduced progress
  - Trace collected upstream of spam filter
  - Effectiveness of Authentication is near 40%
Trace Analysis Study

- Achieve 40% completion on about 3 months of email traffic
  - Using two way bootstrapping group
  - Effectiveness depends on bootstrapping group selection
- Modest cache overhead
- Message overhead is respected as designed
Conclusion

- Implemented and evaluated automatic sender authentication for email

Future work

- Data collection from deployment
- Improve bootstrapping group selection
- Address authenticity vs. importance
Q&A
Peer-to-peer Sender Authentication for Email
Extra Slides
Extra Slides Outline

- Authentication protocol details
  - Distributed Authentication
  - Byzantine Agreement
  - Trust Groups
  - Public Key Infection

- Simulation results
  - Group size
  - Malicious peers
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
Consider proofs received at a peer P

Set of Peers of P

$\Phi_n$ on compromised path to A

$t$ malicious peers

$\Phi_n$ 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 untrusted 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 uninfected peers with time
- Number of cached messages is in $O(n \log n)$
Extra Slides Outline

- Authentication protocol details
  - Distributed Authentication
  - Byzantine Agreement
  - Trust Groups
  - Public Key Infection

- Simulation results
  - Group size
  - Malicious peers
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