Achieving Reduced-Delay and Multi-Grade Authentication in Multicast

Qing Li
Wade Trappe
Talk Overview

- Multicast Source Authentication
- Delayed Key Disclosure Approach
  - Key Example: The TESLA protocol
- Staggered TESLA
  - Scheme overview
  - Results
- Conclusion
Multicast Source Authentication
Source Authentication

- Ensures data is from trusted source
- Ensures data is not modified en route
Unicast Source Authentication

• Digital Signatures
  – Examples: RSA
    • Private Signing Function
    • Public Verification Function
    • Large per packet computation: ~0.5 seconds

• Message Authentication Code (MAC)
  – Uses hash functions with shared key K, “MAC(M,K)=h(M||K)”
  – Signed document {M,MAC(M,K)}
  – Computationally efficient: ~1msec
Multicast Source Authentication

• Can we use Digital Signatures?
  – Yes, but too much computation needed.

• Can we use traditional MAC?
  – No!
  – Why?
Alice shares the same key with the multicast group.

**Problem**: Is the packet really from Alice?
Delayed Key Disclosure Scheme
Delayed Key Disclosure Scheme

- **Delayed Key Disclosure**: (e.g. TESLA)
  - All Packets Authenticated with $K_1$ have arrived to all group members

- **Weakness**:
  - Use of buffers allows for a simple denial of service (DoS) attack
  - Since there is no way to check packets until key is disclosed, buffer will overflow

- **Goal**: To protect against DoS attacks.
Understanding Delayed Key Disclosure

Now Sender Releases Key Seed For Interval i

Releases Packet With TESLA Authentication

No More Packets From Time Interval i allowed!

Gets Packet Sent in Time Interval i
What if we release key seed earlier?

- Now Sender Releases Key Seed For Interval i
- Releases Packet With TESLA Authentication
- Both Real and Forged Packets are Authenticated!!!

Adversary Receives Key Seed Packet

Adversary Forges Packets for Interval i

Gets Forged Packets for Interval i

No More Packets From Time Interval i allowed!

Gets Packet Sent in Time Interval i

i
i+1
i+2
i+3
i+4
i+5
i+6

WINLAB
Definition of Trust in Delayed Key Disclosure

- **Assumptions**
  - Adversary has 0 Forge time
  - Adversary has 0-delay link to receiver
  - Original TESLA disclosure delay is \( d \)

- **Security Condition**
  - Packets sent at interval \( i \) will be discarded if received after \( i + d \)

- **Key released at time \( i + t \), \((t<d)\):**
  - Adversaries within delay radius \( d-t \) can forge packets
  - Adversaries outside radius \( d-t \) will cause a violation of the TESLA security condition

- **Trust**
  \[
  1 - \frac{\text{Area of Forge Capable}}{\text{Area of Whole Network}} = 1 - \frac{\pi(d-t)^2}{\pi d^2}
  \]
Staggered TESLA: Multicast/Broadcast Authentication
Staggered TESLA: Sender Setup

- In TESLA, the sender attaches 1 MAC computed by $K_i$
- Staggered TESLA: The sender attaches $d$ MACs computed by $K_i, \ldots, K_{i-d+1}$
- Using multiple MACs simultaneously allows us to effectively “release” keys earlier than in TESLA
Staggered TESLA: Authentication at Receiver

- Receivers have a chained buffer
  - As keys arrive, MACs are verified
  - If matches, it puts the packet into the next layer. If not, the packet is dropped.
  - As the packets move to lower buffer layers, the trustworthiness of the packets increases
Forge-Capable Area

- Forge Capable Area
  - $K_{i-d+1}$ Release Time: $i+1$ Radius: $d-1$
  - $K_{i-d+2}$ Release Time: $i+2$ Radius: $d-2$
  - ...
  - $K_{i-1}$ Release Time: $i+d-1$ Radius: 1
  - $K_i$ Release Time: $i+d$ Radius: 0

- Forge capable area shrinks as time moves on

- The position of an adversary governs the amount of “damage” he may cause to the authentication.

- Adversaries closer to the source can forge more.

- We remove the potential for forging faster!
Results and Discussions

- Single sender, receiver, adversary case
- Interval size: 200ms
- Delayed key disclosure: 4
- Source sends out packets at 25 Packets/Second
- Different number of MACs are attached
- Adversary sends out packets at different rates

Table 1: Number of Packets in Buffer

<table>
<thead>
<tr>
<th>A’s Rate</th>
<th>0</th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC1</td>
<td>18.0302</td>
<td>36.0668</td>
<td>54.4166</td>
<td>87.5416</td>
<td>155.5825</td>
<td>365.3545</td>
</tr>
<tr>
<td>MAC2</td>
<td>18.0302</td>
<td>30.5037</td>
<td>42.3275</td>
<td>67.3730</td>
<td>117.5745</td>
<td>265.6503</td>
</tr>
<tr>
<td>MAC3</td>
<td>18.0302</td>
<td>26.6845</td>
<td>33.2116</td>
<td>49.8919</td>
<td>78.0683</td>
<td>167.1517</td>
</tr>
<tr>
<td>MAC4</td>
<td>18.0302</td>
<td>25.4828</td>
<td>30.6354</td>
<td>45.0250</td>
<td>72.4581</td>
<td>146.6373</td>
</tr>
</tbody>
</table>

Table 2: Average Time to purge a Forged Packet

<table>
<thead>
<tr>
<th>A’s Rate</th>
<th>0</th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC1</td>
<td>N/A</td>
<td>719.1367</td>
<td>710.9694</td>
<td>707.1318</td>
<td>703.9731</td>
<td>702.2391</td>
</tr>
<tr>
<td>MAC2</td>
<td>N/A</td>
<td>521.1570</td>
<td>517.2279</td>
<td>508.5111</td>
<td>504.1845</td>
<td>502.6923</td>
</tr>
<tr>
<td>MAC3</td>
<td>N/A</td>
<td>318.8805</td>
<td>313.3443</td>
<td>307.4677</td>
<td>303.7401</td>
<td>301.5933</td>
</tr>
<tr>
<td>MAC4</td>
<td>N/A</td>
<td>279.5491</td>
<td>271.3238</td>
<td>269.4597</td>
<td>267.5881</td>
<td>260.9229</td>
</tr>
</tbody>
</table>

Conclusion: Staggered TESLA makes better use of buffer resources!
Staggered TESLA Summary

- Allows for multi-grade authentication
- The receivers can partially authenticate packets during each time interval
- Packets can be released earlier as partially authenticated
- Advantages:
  - False packets are removed from the buffer quicker than conventional TESLA
  - An adversary must attempt a DoS attack at a faster data rate than conventional TESLA