Authenticating streamed data in the presence of random packet loss

Philippe Golle
Nagendra Modadugu
(Stanford University)

Network and Distributed System Security Symposium (2001)

Internet Radio Station

= 1 2 3 4 5 6
Signing Streams

- **Goal:** authenticity, non-repudiation
  - Digital Signature

- **3 requirements for signing streams**
  - ‘On the fly’ authentication
  - Low overhead (computation and communication)
  - Robustness (resist packet loss)

Sign each packet

- Sign each packet (RSA, DSA,...)
- **Properties**
  - Immediate authentication
  - Robust: packets individually verifiable
  - **Computational load too high**
- **Optimization**
  - Maximum: 100 signatures / second
Amortization: hash function

- Collision-resistant hash function $h$:
  Given $h(x)$, hard to find $y$ such that $h(x) = h(y)$
- Hash 100 times faster than digital signature

Hash chain (Gennaro, Rohatgi)

- Sender processes the stream backwards
  - Append the hash of $P_{i+1}$ to $P_i$
  - Sign only the first packet
- Properties
  - Immediate authentication
  - Extremely efficient:
    - 1 hash computation / packet
  - Vulnerable to packet loss
Packet groups (Wong & Lam)

- **Sender:**

  (Simple Example)

- **Packet 3 is sent as:**

- **Robust against packet-loss**

- **Trade-off**
  - More packets per group: buffering, communication overhead
  - Fewer packets per group: computational overhead

Recap

- **Started with a hash chain**
  - Immediate authentication
  - Low computation and communication overhead
  - **Vulnerable to packet loss**
  - Offline streams only

- **Improvement: Wong and Lam**
  - Immediate authentication
  - **Higher computation and communication overhead** (trade-off)
  - Resists packet loss
  - Some buffering on sender side, none on receiver side
The Scheme

- Existing solutions
  - Resistant to packet loss
  - Trade-off between Computation/Communication cost
- Communication overhead *matters*.
- We propose a solution which is
  - Resistant to *average* loss
  - New trade-off: computational cost and authentication speed

The scheme (contd)

- Model of packet loss
  - Bursts (UDP)
  - Goal: maximize length of single worst-case burst
  - Resists multiple bursts

- Authentication
  - complete
Hash chain with redundancy

- Divide the stream into sequences of fixed length (say 50 or 100 packets)
- The last packet in each sequence is signed (and is presumed never lost)
- Property: the signature on the last packet ‘covers’ the hash of every packet in the sequence

No Packet Buffering On Sender-Side

- Chain of strength $a$: the hash of packet $P_i$ is appended to two other packets: $P_{i+1}$ and $P_{i+a}$
- Only the last packet is signed.
Characteristics of a Chain

- **Sender:**
  - Buffers 1 packet
  - Stores a hashes

- **Receiver**
  - Buffers 2 hashes
  - Can authenticate at the end of the sequence

- **Resistance to loss**
  - Maximum burst length = a-1

---

With Packet Buffering on the Sender Side

Example: augmented chain of strength 3

- If the sender can buffer a single packet:
Generalization

- Sender buffers:
  - p packets
  - h hashes

Insert new packets in-between

Stage 1

- Very simple to implement
- Optimally resistant to loss
- But: the maximum number of hashes appended to a packet grows linearly with p
Stage 2

Recursive embedding

Characteristics

**Sender**
- Buffers $p$ packets
- Hash buffer of size $h = a + p$

**Receiver**
- Buffers $(p+3)/2$ hashes

**Resistance to loss:**
- Maximum burst length $= p(a-1)$
Conclusion

- Efficient stream authentication scheme.
- Strength: resistance to random loss (bursts)
- New trade-off: between computational complexity and time to authentication