Efficient and Secure Source Authentication for Multicast

Authors: Adrian Perrig, Ran Canetti Dawn Song J. D. Tygar

Presenter: Nikhil Negandhi
CSC774 Network Security

Outline:

- Background
  - Problem
  - Related work
- An Overview of TESLA
- Extended TESLA
- Security Discussion & Robustness to DOS
- Conclusion
- Future Work

Problem: Efficient Source Authentication

- One sender, many receivers
- Receivers not trusted
- Lossy channel (lost packets are not retransmitted)

Related Work:

- Digital Signatures

Advantage: Non-repudiation

Disadvantages: Too expensive, DOS on the receiver

- Simple MAC using shared secret as in UNICAST

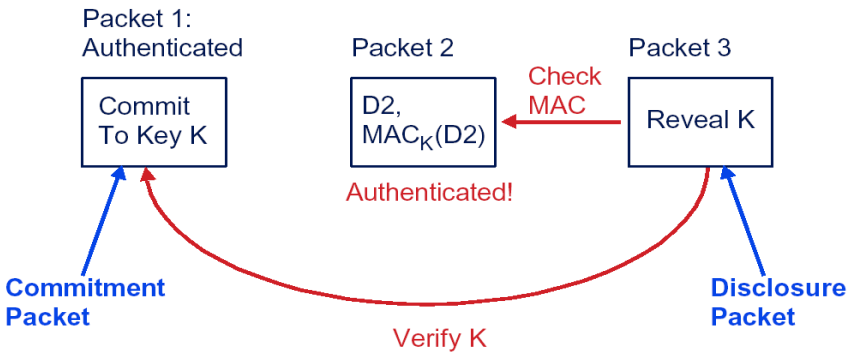
Advantages: Low overhead , MAC computation is fast

Disadvantage: Insecure in multi-party case

An Overview of TESLA

- Provides multicast source authentication
- Efficient:
  - symmetric-key cryptography
  - 1 MAC function computation (1,000,000/s)
  - Low overhead per packet (10-20 bytes)
- Perfect loss robustness
- Scalable: After initial receiver bootstrap, unidirectional data flow
- Drawback: Delayed authentication

TESLA: The basic idea

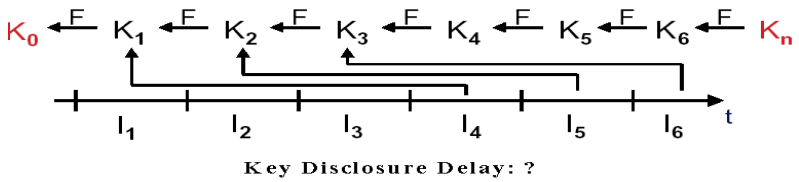


Security Condition

- A packet arrived safely if the receiver is assured that the sender cannot yet be in the time interval in which the corresponding key is disclosed.
- Receiver verifies that packet P arrives before sender discloses KP
- If security condition not satisfied, drop the packet

Sender Setup

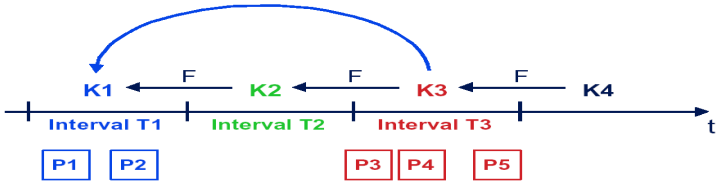
- Interval definition
  - Beginning time of one specific interval
  - Interval duration, disclosure delay
- Key chain
  - Compute using a pseudo-random function F
  - Digitally sign K0, give to receivers at registration.



Key Disclosure Delay: ?

Sending/Receiving Authenticated Packets

- Packet Pj sent in interval Ii is: {Mj/MACki(Mj)/Ki-d}



- The key remains secret for _____ future intervals
- Packets sent in interval Ij can disclose key _____

Drawbacks of TESLA:

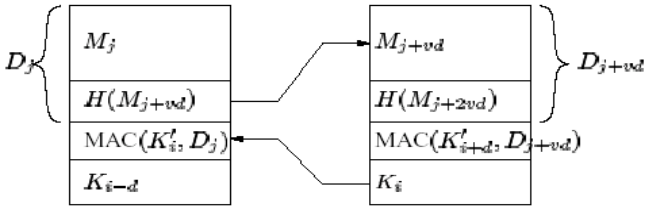
- **Receiver has to buffer packets**
  - DoS attacks on the receiver
- **Overhead for receivers in heterogeneous networks**
  - uses many keys with different disclosure delays
- **Synchronization**
  - costly due to public key operations
- **Not-scalable**
  - sender has synchronized with all the receivers before transmission starts.

Extended TESLA:

- Immediate Authentication
- Concurrent TESLA instances
- Time Synchronization
- Determining Key Disclosure Delay

Immediate Authentication

- Receiver authenticates packets as soon as they arrive
- Sender buffers packets during one disclosure delay
- Sender stores hash value of data of later packet in an earlier packet



If packets are dropped, authentication ?

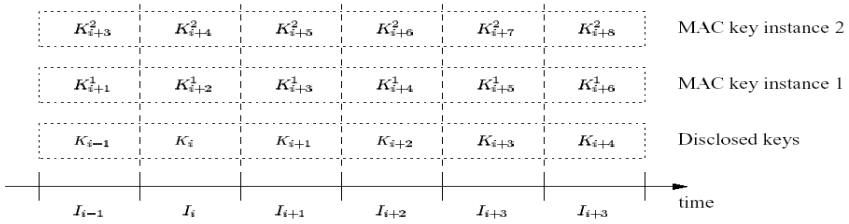
Concurrent TESLA instances

Why require multiple instances of TESLA with different disclosure delays?

- low delay receivers :short key disclosure delay
- high delay receivers: long key disclosure

Solution:

Same key chain but a different key schedule for all instances



K^u_{i+du} = MAC key in Ii→disclosed in I_{i+du}

Concurrent TESLA instances

Advantages

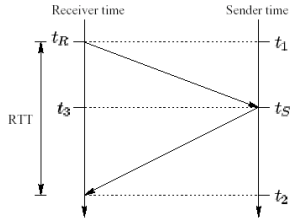
- Generates all different independent keys for each instance
- Sender needs to disclose one key chain independent of number of concurrent instances

Time Synchronization

- Loose time synchronization
- Receiver knows an upper bound of difference between sender's local time and the receiver's local time = (del)

1. Direct:

- Explicit time synchronization with sender
- No extra infrastructure needed
- T $R \rightarrow S$: Nonce
  $S \rightarrow R$: {Sender time t_S, Nonce}$_{K_S^{-1}}$
- Del=ts-tr
- DoS ?



2. Indirect

- Implicit Time Synchronization
- Synchronize with a time reference

Determining Key Disclosure Delay

- Short disclosure delay→ security condition violated→ packets drop
- Long disclosure delay→ long authentication delay
- Security Condition is

$$[(t^R i + del - To)/Tint] - Ij < d$$

d = disclosure delay in time intervals

Tint = interval duration

To = beginning time of 0^{th} time interval

$t^R i$ =receiver's local time when packet Pi arrives

Ij = time at which packet Pi is constructed

del = upper bound of difference between sender's local time and the receiver's local time

[ ] = truncate function

Security Discussion and Robustness to DoS

DoS Attack on the **Sender**

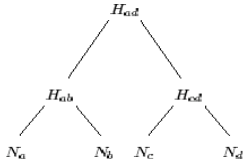
- Not possible with indirect time synchronization
- Possible with direct time synchronization

Solution:

- Sender aggregates multiple requests
- computes and signs a Merkle hash tree that is generated from all the requester's nonces
- Root H_{ad} is included in signed part instead of receiver's nonce
- To verify the signature, each receiver reconstructs hash tree
- Example:

  Node $H_{ab} = H(N_a, N_b)$. $H_{ad} = H(H_{ab}, H_{cd})$.

  A will receive N_b , H_{cd}
- No. of nodes returned = _____



Security Discussion and Robustness to DoS

DoS Attack on the **RECEIVER**

- Replay Attack
  - security condition drops packets if replayed with a long delay
  - add sequence number to each packet
- Flooding with bogus traffic
  - security condition drops a packet after one disclosure delay
  - immediate authentication
- Attacker fools receiver to believe that packet was sent out far in the future
  - in order to verify the disclosed key the receiver would hash until last committed key chain value
  - foiled by checking packet interval <= latest interval sender can be in

    Ij <= [(ti-To)/Tint]

Conclusion

- Basic TESLA provides low computation and communication overhead and perfect lost robustness
- Additional information in a packet used to provide immediate authentication
- Reduced overhead when multiple TESLA instances are concurrently used with different authentication delays
- Derived a tight lower bound on disclosure delay
- Hardened the sender and receiver against various DoS attacks

FUTURE WORK

- Source authentication with non-repudiation
- Perfect time synchronization techniques with less complexity and low overhead