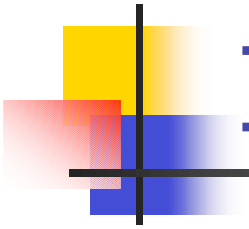
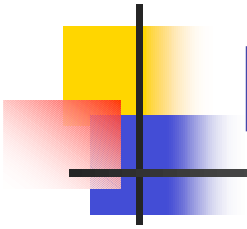
Efficient Authentication and Signing of Multicast Streams over Lossy Channels.



Introduction

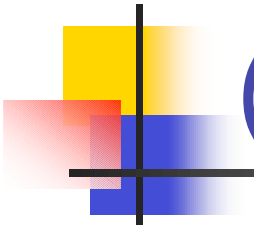
- Need for a separate scheme
- Two Schemes
  - TESLA
    - Sender Authentication
    - Strong loss robustness
    - High Scalability
    - Minimal overhead
  - EMSS
    - Non Repudiation
    - High loss robustness
    - Low overhead



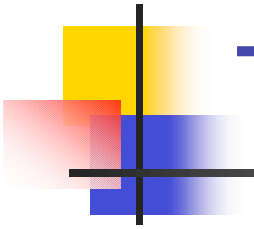
Need for a separate scheme

- Internet
- Need for widespread & trusted streamed media dissemination
  - Attacker may alter stock quotes distributed through IP multicast
- Solution is trivial for 1 sender receiver case
- Multiple receiver – Need to use asymmetric cryptography
  - Digital Signatures: Too inefficient

Need for separate scheme (Contd)

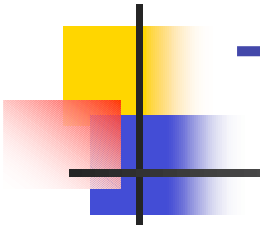


- Needs to scale to millions of users
- Streamed media distribution can have high packet loss



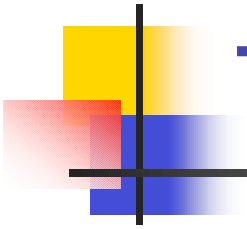
TESLA - Properties

- Low computational overhead
- Low per packet communication overhead
- Arbitrary packet loss tolerated
- Unidirectional data flow
- No Sender side buffering
- High guarantee of authentication
- Freshness of data



TESLA – Basic working

- Timed Efficient Stream Loss – tolerant Authentication
- Based on timed release of keys by the sender
- Sender commits to a random key k and transmits it to the receivers without revealing it
- Sender attaches a MAC to the next packet Pi with k as the MAC key
- In Pi+1 packet sender decommits to the key and receiver uses this key k to verify Pi
- Need a security assurance



TESLA – Scheme I

- Pi-1 = <Di-1, MAC(K'i-1 , Di-1)>
- Di-1 = <Mi-1, F(Ki), Ki-2>
- Mi-1 = Message
- F(Ki) = commitment to Ki
- K'i = F'(Ki)
- Each packet Pi+1 authenticates Pi
- Problems ??

TESLA – Scheme I (Contd)

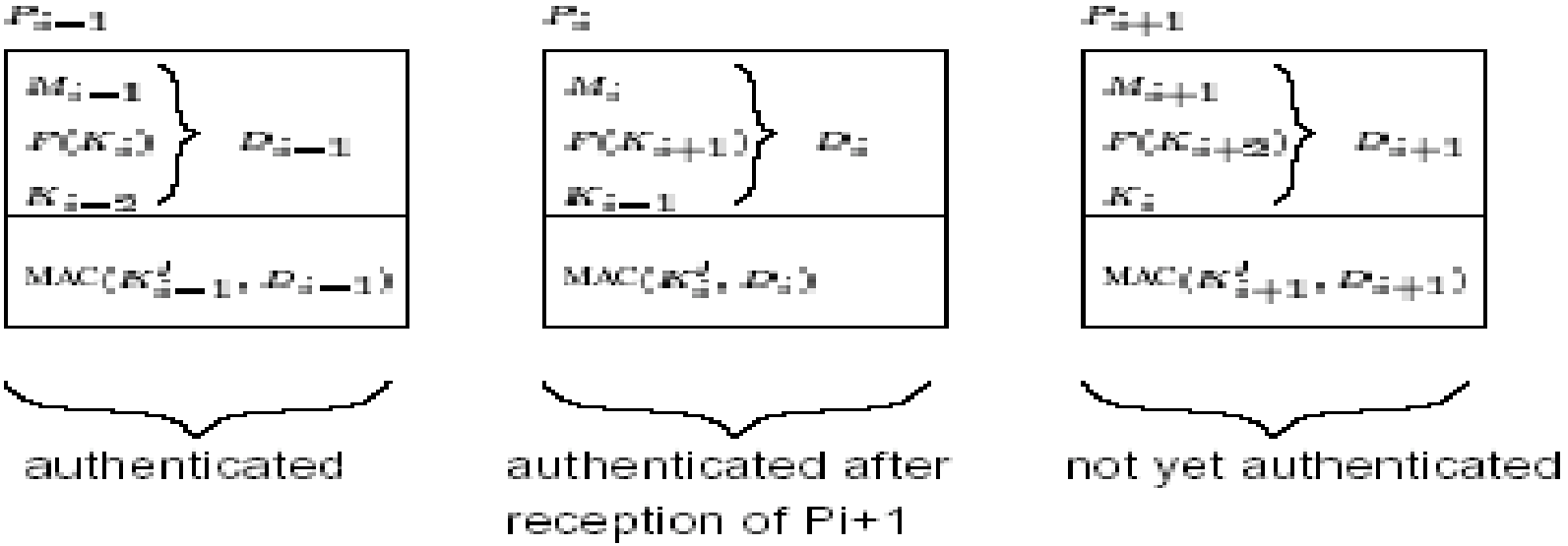
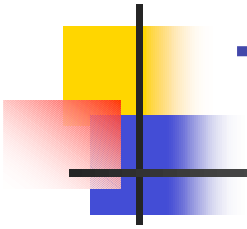
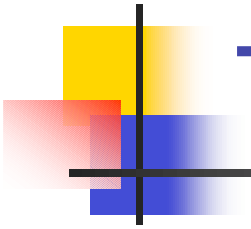


Figure 1. Basic stream authentication scheme. M_i stands for message i, P_i is packet i, K_i denotes the secret key i, F, F' are pseudo-random functions, and $\mathrm{MAC}(K'_i, D_i)$ computes the MAC of packet i using the secret key $K'_i = F'(K_i)$.



TESLA – Scheme I (contd)

- If attacker gets Pi+1 before receiver gets Pi, it can forge Pi
- Security Condition
  - ArrTi + delta(t) < Ti+1
  - Sender's clock is no more than delta(t) secs ahead of that of the recievers
- Packet loss not tolerated



TESLA – Scheme II

- Generate a seq of keys { Ki }
- Fv(x) = Fv-1(F(x))
- Fo(x) = x
- Ko = Fn (Kn)
- Ki = Fn-i(Kn)
- Attacker cannot invert F & compute any Kj given Ki; j>I
- Receiver can compute all Kj from Ki ; j < I
- Kj = Fi-j (Ki) ; K'i = F' (Ki)

TESLA – Scheme II (Contd)

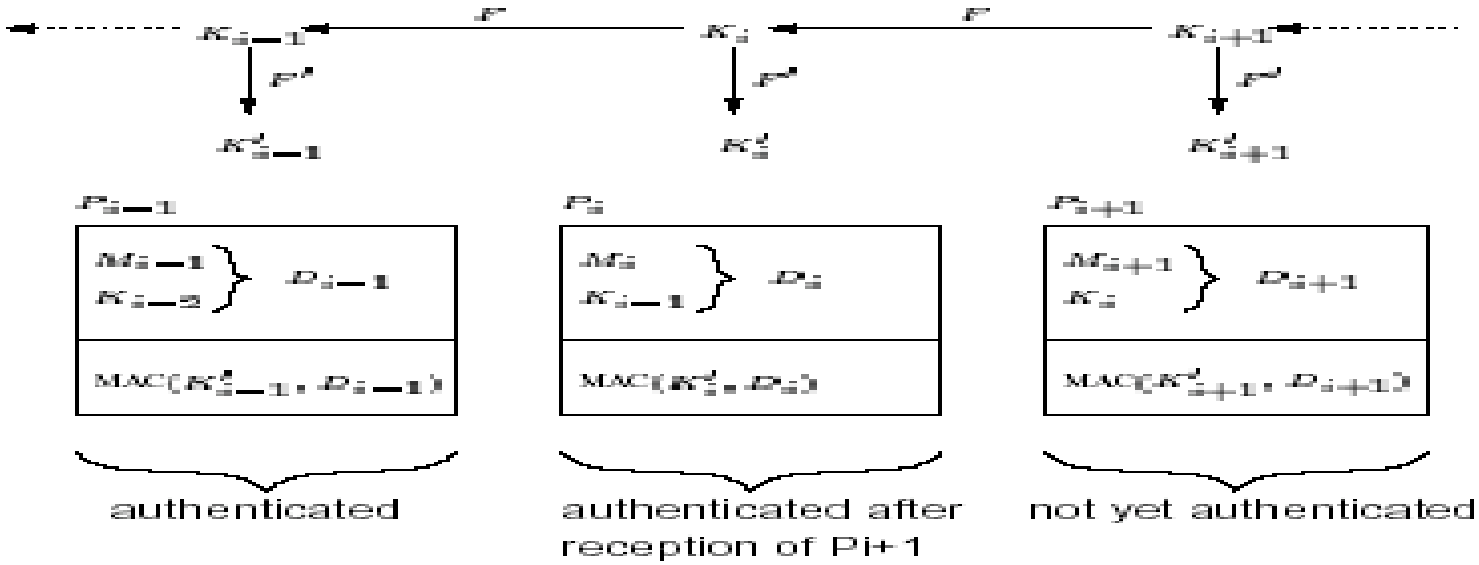
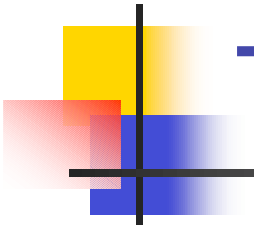
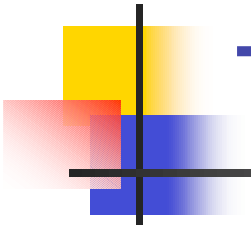


Figure 2. Scheme II. The packet format is the same as in scheme I, except that the commitment $F(K_{i-1})$ is omitted and the keys form a one-way key chain.



TESLA – Scheme III

- Deals with faster packet rates
- Does not require that sender waits for receiver to get Pi before it sends Pi+1
- Disclose Ki in Pi+d instead of Pi+1
- d = (delta(t)max + dnmax)/r
- r = packet rate
- Security Condition:
  - ArrTi + delta(t) < Ti+d
- Very short d = ?
- Very large d = ?



TESLA – Scheme IV

- Deals with Dynamic transmission rates
- Divide time into intervals
- Use the same Ki to compute the MAC of all packets in the same interval i
- All packets in the same interval disclose the key Ki-d
- Achieve key disclosure based on interval basis than on packet index basis

TESLA – Scheme IV (contd)

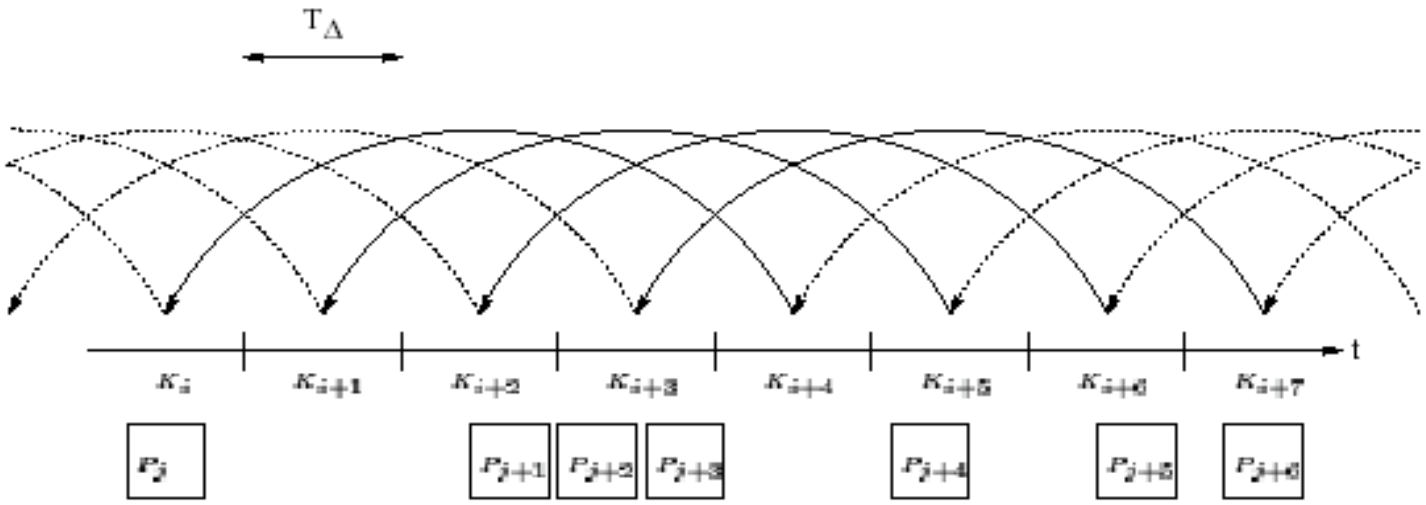
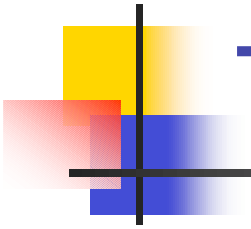
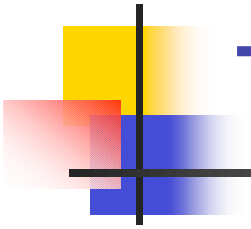


Figure 3. Scheme IV. The MAC key and disclosed key are only dependent on the time interval. The authentication key of P_j is K_i which is disclosed by packets sent during interval $i+4$. In this case, packet P_{j+4} discloses key K_{i+1} which allows the receiver to compute K_i and to authenticate packet P_j. We would like to point out that packets P_{j+2} and P_{j+3} are both authenticated with the same MAC key K'_{i+3}, because they were sent in the same time interval.



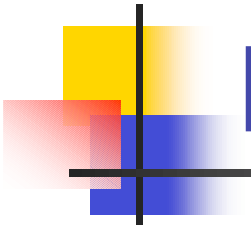
TESLA – Scheme IV (contd)

- i = (t – To)/Tdelta
- Ki’ = F’(Ki) for each packet in interval i
- Pj = < Mj, i, Ki-d, MAC(Ki’, Mj) >
- Security condition:
  - i + d > i’
  - i’ is the interval the sender can be at most
  - i’ = (tj+delta(t)-To)/Tdelta



TESLA – Scheme V

- Allow for a broad spectrum of users
- d is short shall force remote users to drop packets
- d is large shall cause unacceptable delay for fast receivers
- Use multiple authentication chains with different values of d
- Receiver verifies one security condition for each chain Ci, and drops the packet is none is satisfied



EMSS

- Efficient Multichained Streamed Signature
- Useful where
  - Non Repudiation required
  - Time synchronization may be a problem
- Based on signing a small no. of special packets in the stream
- Each packet linked to a signed packet via multiple hash chains

EMSS – Basic Signature Scheme

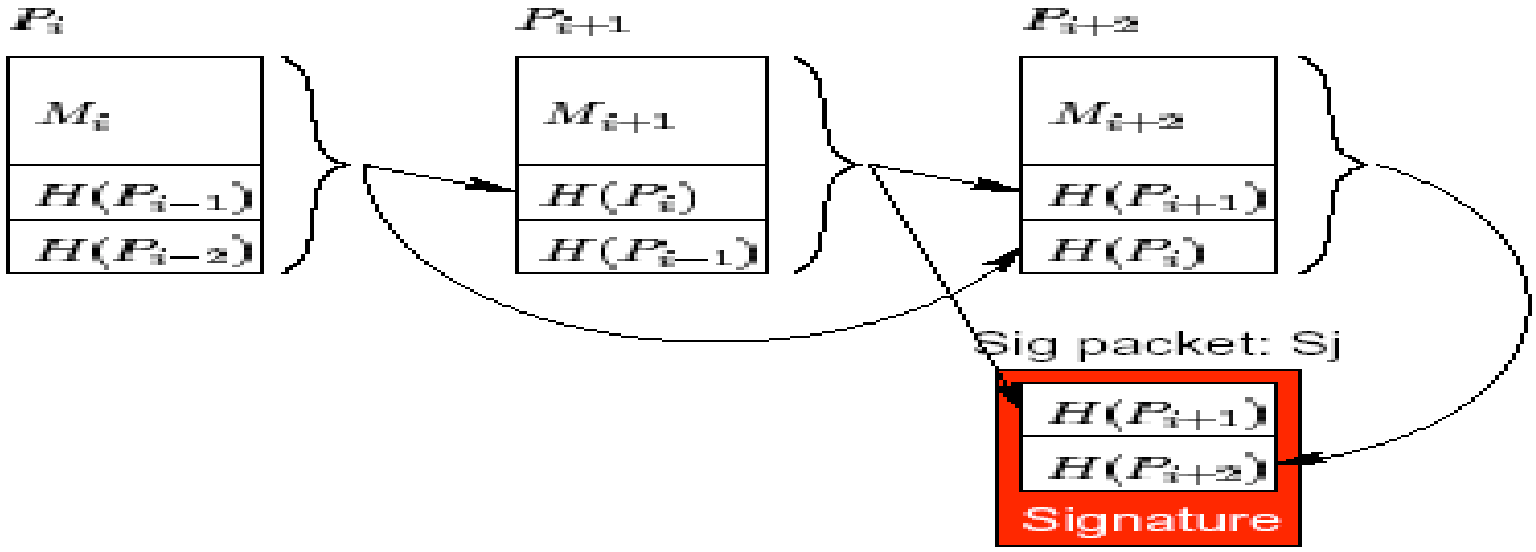
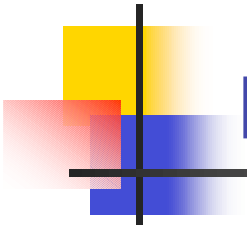
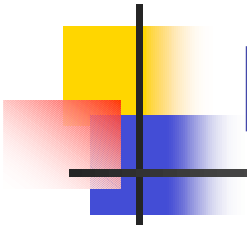


Figure 6. We achieve non-repudiation through periodic signature packets, which contain the hash of several data packets, and the inclusion of the hash of the current packet within future packets. The inclusion of multiple hashes achieves robustness against packet loss.



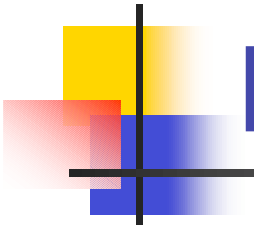
EMSS – Basic Signature Scheme (Contd)

- Sender sends periodic signature packets
- Pi is verifiable if there exists a path from Pi to any signature packet Sj



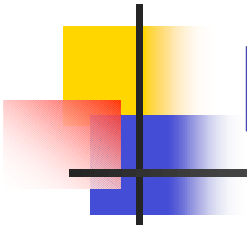
EMSS – Extended Scheme

- Basic scheme has too much redundancy
- Split hash into k chunks, where any k’ chunks are sufficient to allow the receivers to validate the information
  - Rabins Information Dispersal Algorithm
  - Some upper few bits of hash
- Requires any k’ out of k packets to arrive
- More robust



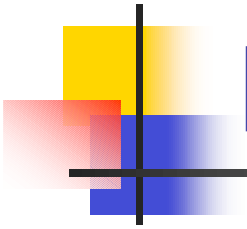
Related Work

- Gennaro and Rohtagi 1997
  - Include in Pi the hash of pi+1
  - Does not tolerate packet loss
- Wong and Lam 1998
  - Use one signature generation & verification over multiple messages
  - Larger space overheard



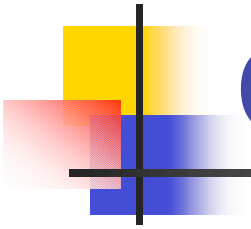
Related Work (Contd)

- Syverson et al. 1997
  - Asymmetric and unlinkable authentication
  - Use a blinded signature token, authenticated every transaction
  - Substantial computational and communication overhead
- Anderson et al. 1998
  - Guy Fawkes protocol
  - Improvise on it, to make it more efficient.



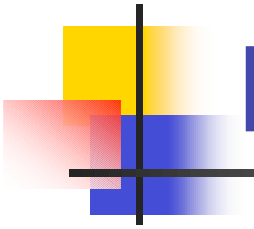
Related Work (Contd)

- Rohatgi 1999
  - Use k time signature scheme
  - Uses a 6 time public key and 300 bytes for each signature
- Cannetti et al. 1999
  - Sender authentication scheme for multicast
  - Use k different keys to authenticate every message with k different MACs
  - More Expensive



Conclusions

- Low computational overhead
- Low communication overhead
- Loss robustness



Future Work

- This paper does not issue DoS attack
- TESLA may suffer from time synchronization issues
- Present Schemes require buffering packets at receiver side