Network Security: Broadcast and Multicast Security, Anonymity

Tuomas Aura, Microsoft Research, UK

Outline

- Broadcast and multicast
- Receiver access control (i.e. data confidentiality)
- Multicast authentication
- DoS protection

- Anonymity and privacy
- High-latency anonymous routing
- Low-latency anonymous routing — Tor

2

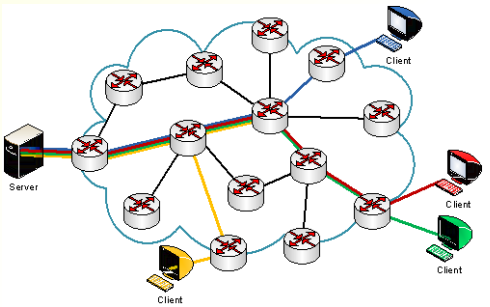
Broadcast and multicast

Broadcast and multicast

- Unicast = send to one receiver
  - Traditional IP routing
  - TCP, HTTP, video and audio streaming
  - Server sends a separate copy to each receiver
- Broadcast = send to everyone
  - Terrestrial radio and television, satellite
  - Link-layer broadcast on Ethernet or WLAN, flood-fill through an overlay network
- Multicast = send to a group of receivers
  - IP multicast, overlay streaming , IPTV
  - Can save bandwidth by routing through a tree

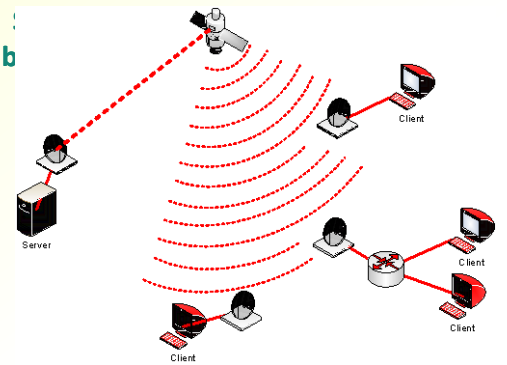
4

IP unicast



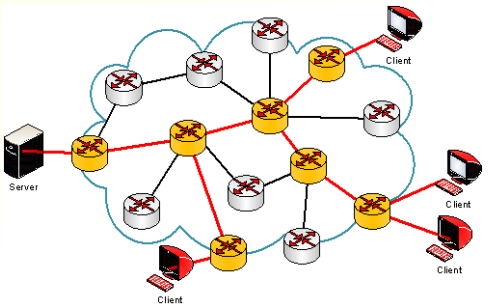
5

b

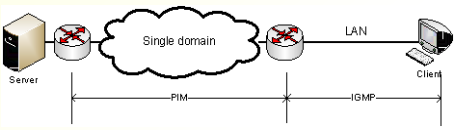


6

IP multicast protocols



IP multicast



- Internet group management protocol (IGMP) in IPv6 and Multicast listener discovery (MLD) in IPv6 between clients and the gateway router
  - Clients use to connect to a local multicast router
- Protocol-independent multicast (PIM) within larger networks
  - PIM sparse mode (PIM-SM) or dense mode (PIM-DM) used closed routing domains
- Inter-domain protocols: Multicast Source Discovery Protocol (MSDP) and MBGP
  - Still experimental
- The multicast routing protocols do not provide security in themselves

Future of multicast and broadcast?

- Multicast tree vs. P2P overlay multicast protocols
- Youtube and unicast

Security goals

- Applications: satellite and cable TV, Internet TV, peer-to-peer content distribution, GPS/Galileo, teleconference
- Access control to multicast and broadcast data
- Data authentication
- DoS protection — access control for senders
- Privacy — confidentiality of subscriber identities (which channel is my neighbor watching?)

Receiver access control

Access control to data

- Goal: allow only authorized access to data
- Encrypt data, distribute keys to authorized recipients (= multicast group)
- Key distribution issues:
  - Revocation speed
  - Amount of communication and computation per joining or leaving node
  - Scalability (teleconference vs. satellite TV broadcast)
  - Possible packet loss when session keys are replaced
  - Sharing keys to unauthorized parties is easier than sharing data

Group key distribution

- Various efficient protocols for distributing keys to a multicast group
- Typical solution: unicast key distribution to individual subscribers
  - Ok for small groups (e.g. teleconference) or slow updates (e.g. IPTV subscription)
- Can piggyback individual key updates on multicast data
  - Does not require separate unicast channel
  - Ok for slow updates (e.g. satellite TV)
- Advanced protocols
  - Typically log(N) communication to revoke one receiver out of N

13

Multicast and broadcast authentication

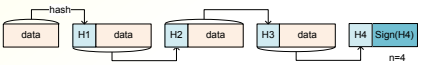
Multicast data authentication

- Security goals:
  - Integrity, data-origin authentication
  - Sometimes non-repudiation
  - Early dropping of spoofed data
- Other constraints:
  - Loss tolerance vs. reliable transmission
  - Real-time requirements
- Small groups could use a shared key and MACs
  - Every member can spoof data
  - Won't work for large or mutually distrusting groups
- Asymmetric crypto seems the right tool
  - One sender and many receivers

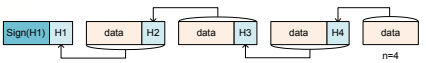
15

Hash chaining

- Forward chaining
  - Amortize the cost of a signature over many data packets
  - Sender can send in real time
  - Receiver should buffer data and consume only after signature received
  - Received vulnerable to DoS from spoofed packets



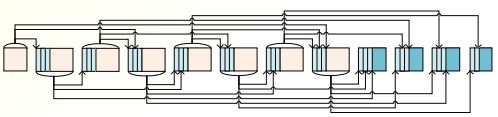
- Backward chaining
  - Received can authenticate and consume data immediately
  - Sender must buffer data before sending and signing



16

Loss tolerant chaining

- Redundant hash chains
- Efficient multi-chained stream signature (EMSS)
  - E.g. 1-3-7 chaining sequence tolerates bursty losses of up to 7 packets:



  - Redundant signatures costly
  - Random chaining sequence shown to be efficient
- Alternative: forward error correction code

17

Guy Fawkes protocol (1)

- Delayed authentication [Ross Anderson 1997]
- Initially, receiver knows Y = hash(X)
- To authenticate message M:
  1. Sender publishes $Z = MAC_X(M)$
  2. Sender reveals M, X
- Z is a commitment that binds the message M and the secret X. Revealing X later authenticates M
- Critical detail:
  - The commitment Z must be received before X is revealed
  - In the Guy Fawkes protocol, Z is published in a news paper = broadcast medium with guaranteed latest delivery time

18

Guy Fawkes protocol (2)

- Out-of-band initialization:
  - Sender selects a random X_0 and computes $Y_0 = \text{hash}(X_0)$
  - Sender publishes Y_0 via an authenticate channel
- Protocol round i=1,2,3,…:
  1. Sender selects a random X_i and computes $Y_i = \text{hash}(X_i)$
  2. Sender publishes in a newspaper $Z_i = \text{MAC}_{X_{i-1}}(M_i, Y_i)$
  3. Sender reveals M_i, $\text{hash}(X_i)$, X_{i-1}
- Z_i is a commitment that binds the message M_i and the secret X_{i-1}. Revealing X_{i-1} later authenticates M_i.
- The next key M_i authenticated together with X.
- Critical:
  - Each Z_i must be received before X_{i-1} revealed

19

Lamport hash chain

- [Leslie Lamport 1981]
- One-time passwords for client-server authentication
- Initialization:
  - Random number X_0
  - Hash chain $X_i = h(X_{i-1})$, i=1…n
  - Server stores X_n
- Client reveals hashes in reverse order: X_{n-1}, X_{n-2},…
- Protects against password sniffing
  - Cannot reuse like a normal password
  - Better than all random passwords X_1, X_2,… because the password database (/etc/password) can be public
- Entity authentication only; not easy to combine with key exchange

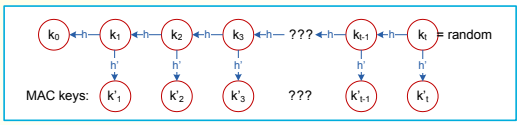
20

TESLA (1)

- Time efficient stream loss-tolerant authentication [Perrig et al. 2000][RFC 4082]
- After initialization, secret-key crypto (cryptographic hash and MACs) only
- Delayed authentication: broadcast sender commits to MAC keys and reveals them after a fixed delay
  - Authentication delay at least one round-trip time (RTT)
  - MAC keys come from a hash chain
- Requires loose clock synchronization
  - Authentication delay must be set to > maximum clock skew
- No buffering of data at sender; buffering for a fixed period at the receiver
- Tolerates packet loss
- Scales to any number of receivers
- No non-repudiation

21

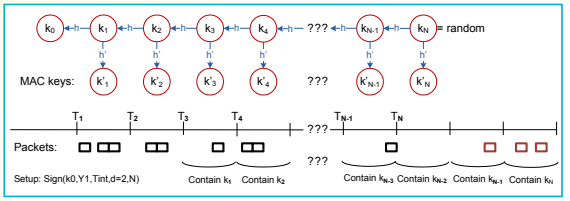
TESLA (2)



- Initialization:
  - Sender commits to the key chain and release schedule by signing: k_0, start time T_1, interval duration T_{int}, disclosure delay d·T_{int}
- Time periods start at T_1, others $T_{i+1}=T_i+T_{int}$
- MAC keys k'_1, k'_2, k'_3,…
  - Used for message authentication in periods starting from T_1, T_2, T_3…
  - k_i revealed d periods later (revealing k_i reveals all k_j, $j \le i$)
  - Sender and receiver must have loosely synchronized clocks

22

TESLA (3)



- Packets received in period i will be authenticated in period i+2 (d=2)
- If a packet that belongs to the period $[T_i, T_{i+1}]$ is received after T_i+T_{int}, it cannot be authenticated
- Ok to have silent periods but dummy packets may be needed to avoid long authentication delays
- Next key chain can be initialized by sending the new k_0 in the last packets of the previous chain (cf. Guy Fawkes)

23

DoS protection

Access control for senders

- Multicast is a mechanism for traffic amplification → can be used for DoS attacks to consume bandwidth
- One-root solution: the root node of the multicast tree authenticates senders and checks for authorization
  - One sender, or the root node relays data from all senders
  - Ok for satellite broadcast but no such root exists for IP multicast in the Internet, for many-to-many communication, or for peer-to-peer content distribution
  - Authentication of data at each router needed to avoid insertion of false data → maybe too expensive
- Reverse path forwarding: each router checks the routing table for the source address and decides whether then packet came from the right direction
  - Prevents some spoofing attacks
  - Needed to prevent routing loops anyway

Non-crypto access control for receivers

- A multicast receiver could subscribe to a large number of multicast streams
  - Packet flood to the location of the receiver
  - Either free, unencrypted streams or streams of encrypted packets it cannot decrypt
- Need some way of limiting subscriptions at the receiver end

26

Exercises

- Combine backward and forward chaining to divide the buffering requirement between sender and receiver
- How could a criminal organization use cryptography to make a series of anonymous but plausible threats? (Hint: Guy Fawkes was a 17th century terrorist)
- If the receiver has no capability for public-key operations, how would you initialize TESLA?

27

Anonymity and privacy

28

Anonymity terminology

- Identity, identifier
- Anonymity — they don't know who you are
- Unlinkability — they cannot link two events or actions (e.g. messages) with each other
- Pseudonymity — intentionally allow linking of some events to each other
  - E.g. sessions, payment and service access
- Authentication — strong verification of identity
- Weak identifier — not usable for strong authentication but may compromise privacy
  - E.g. nickname, IP address, SSID, service usage profile
- Authorization — verification of access rights
  - Does not always imply authentication

29

Anonymity in communications

- Anonymity towards communication peers
  - Sender anonymity — receiver does not know who and where sent the message
  - Receiver anonymity — can send a message to a recipient without knowing who and where they are
- Third-party anonymity — an outside observer cannot know who is talking to whom
  - Unobservability — an outside observer cannot tell whether communication takes place or not
  - Strength depends on the capabilities of the adversary
- Anonymity towards access network
  - Access network does not know who is roaming there
- Relate concept: location privacy

30

Privacy

- Control over personal information
  - Emphasized in Europe
  - Gathering, disclosure and false representation of facts about one's personal life
- Right to be left alone
  - Emphasized in America
  - Avoiding spam, control, discrimination, censorship
- Anonymity is a strong tool for achieving privacy
  - Blending into the crowd

31

Strong anonymity?

- Anonymity and privacy of communications mechanisms are not strong in the same sense as strong encryption or authentication
- Even the strongest mechanisms have serious weaknesses
  - Need to trust many others to be honest
  - Services operated by volunteers and activists
  - Side-channel attacks
- Anonymity tends to degrade over time for persistent communication

32

Weak identifiers

- Lack of strong authentication does not imply anonymity
- Persons or computers can be identified by weak (i.e. implicit) identifiers:
  - Non-unique names, nicknames, usernames, computer names, domain names, addresses
  - Profile of the software and hardware, collected either by passive sniffing or active probing
  - Profile of the network communication and services used
- Weak identifiers are everywhere…

33

Identity protection in key exchange

- Identity protection against passive observers achieved by encrypting the authentication with a Diffie-Hellman key or a secret send with public-key encryption
- Identity protection of one party against active attackers achieved by authenticating the other party first
- Recall these protocols:
  - PGP
  - TLS/SSL
  - IKEv2
  - Kerberos
  - WPA2
- Lower-layer identifiers (MAC and IP address) can still leak identity
- Traffic analysis can still be used to profile the node

34

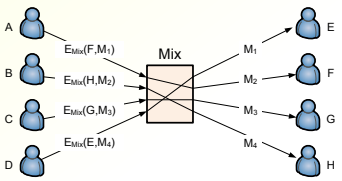
Randomized identifiers

- Replace permanent identifiers with random pseudonyms
- Especially important below the encryption layer
  - Random interface id in IPv6 address [RFC 4941]
  - Random MAC addresses suggested
- Need to consider weak identifiers, too
  - E.g., IPID, TCP sequence number

35

High-latency anonymous routing

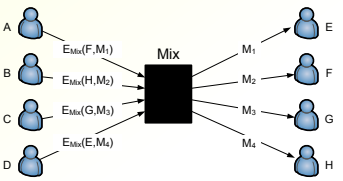
Mix (1)



- Mix is an anonymity service [Chaum 1981]
- Attacker sees both sent and received messages but cannot link them to each other → sender anonymity, third-party anonymity against a global observer
- The mix receives encrypted messages (e.g. email), decrypts them, and forwards to recipients

37

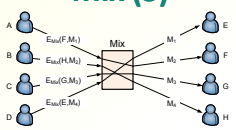
Mix (2)



- Attacker can see the input and output of the mix
- Attacker cannot see how messages are shuffled in the mix
- Anonymity set = all nodes that could have sent (or could be recipients of) a particular message

38

Mix (3)



- Two security requirements:
  - Bitwise unlinkability of input and output messages — cryptographic property, must resist active attacks
  - Resistance to traffic analysis — add delay or inject dummy messages
- Not just basic encryption!
  - Resist adaptive chosen-ciphertext attacks (IND-CCA2 i.e. NM-CCA2)
  - Replay prevention and integrity check needed at the mix
- Examples of bad mix designs:
  - Missing random initialization vector, padding or freshness
  - Malleable encryption, e.g. stream cipher, or no integrity check
  - FIFO order of delivering messages

39

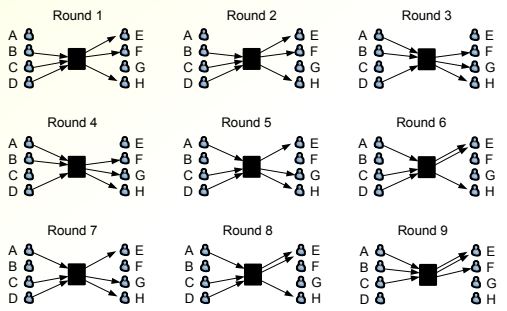
Mixing in practice

- Threshold mix — wait to receive k messages before delivering
  - Anonymity set size k
- Pool mix — mix always buffers k messages, sends one when it receives one
- Both strategies add delay → high latency
- Not all senders and receivers are always active
  - In a closed system, injecting cover traffic can fix this; in the Internet, not
- Real communication (email, TCP packets) does not comprise single, independent messages but common traffic patterns such as connections
  - Attacker can observe beginning and end of connections
  - Attacker can observe requests and response pairs

  → statistical attacks

40

Who sends to whom?



Threshold mix with threshold 3

41

Anonymity metrics

- Size of the anonymity set: k-anonymity
  - Suitable for one round of threshold mixing
- Problems with k-anonymity:
  - Multiple rounds → statistical analysis based on understanding common patterns of communications can reveal who talks to whom, even if k for each individual message is high
  - Pool mix → $k = \infty$
- Entropy: $E = \Sigma_{i=1...n} (p_i \cdot log_2 p_i)$
  - Measures the amount of missing in information in bits: how much does the attacker not know
  - Can measure entropy of the sender, recipient etc.
- Problems with measuring anonymity:
  - Anonymity of individual messages vs. anonymity in a system
  - Depends on the attacker's capabilities and background information
  - Anonymity may degrade over time as attacker collects more statistics

42

Trusting the mix

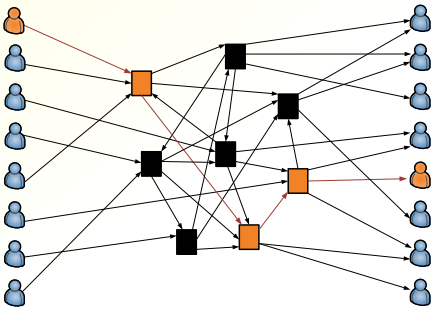
- The mix must be honest
- Example: anonymous remailers for email
  - anon.penet.fi 1993–96

→ Route packets through multiple mixes to avoid single point of failure

- Attacker must compromise all mixes on the route
  - Compromising almost all may reduce the size of the anonymity set

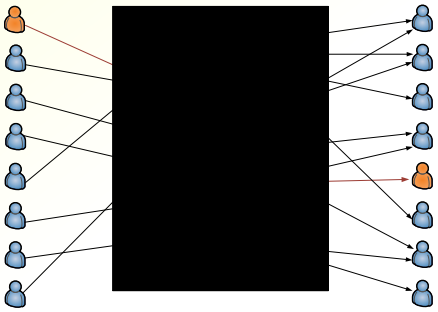
43

Mix network (1)



44

Mix network (2)



- Mix network is just a distributed implementation of mix

45

Mix networks

- Mix cascade — all messages from all senders are routed through the same sequence of mixes
  - Good anonymity, poor load balancing, poor reliability
- Free routing — each message is routed independently via multiple mixes
- Other policies between these two extremes
- Onion encryption:

  Alice → M1: $E_{M1}(M2,E_{M2}(M3,E_{M3}(Bob,M)))$
  M1 → M2: $E_{M2}(M3,E_{M3}(Bob,M))$
  M2 → M3: $E_{M3}(Bob,M)$
  M3 → Bob: M
  - Encryption at every layer must provide bitwise unlinkability
    → detect replays and check integrity
    → for free routing, must keep message length constant
- Re-encryption mix — special crypto that keeps the message length constant with multiple layers of encryption

46

Sybil attack

- Attack against open systems which anyone can join
  - Mixes tend to be run by volunteers
- Attacker creates a large number of seemingly independent nodes, e.g. 50% off all nodes → some routes will go through only attacker's nodes
- Defence: increase the cost of joining the network:
  - Human verification that each mix is operated by a different person or organization
  - The IP address of each mix must be in a new domain
  - Require good reputation of some kind that takes time and effort to establish
  - Select mixes in a route to be at diverse locations
- Sybil attacks are a danger to most P2P systems
  - E.g. reputation systems, content distribution

47

Other attacks

- (n-1) attack
  - Attacker blocks all but one honest sender, floods all mixes with its own messages, and finally allows one honest sender to get though → easy to trace because all other packets are the attacker's
  - Potential solutions: access control and rate limiting for senders, dummy traffic injection, attack detection
- Statistical attacks
  - Attacker may accumulate statistics about the communication over time and reconstruct the sender-receiver pairs based on its knowledge of common traffic patterns

48

Receiver anonymity

- Alice distributes a reply onion:
  $E_{M3}(M2,k2,E_{M2}(M1,k1,E_{M1}(Alice,k3,E_{Alice}(K))))$
- Messages from Bob to Alice:
  Bob → M3: $E_{M3}(M2,k2,E_{M2}(M1,k1,E_{M1}(Alice,k3,E_{Alice}(K)))),$ M
  M3 → M2: $E_{M2}(M1,k1,E_{M1}(Alice,k3,E_{Alice}(K))),$ $E_{k1}(M)$
  M2 → M1: $E_{M1}(Alice,k3,E_{Alice}(K)),$ $E_{k2}(E_{k1}(M))$
  M1 → Alice: $E_{Alice}(K),$ $E_{k3}(E_{k2}(E_{k1}(M)))$
  Alice can be memoryless: ki = h(K, i)

49

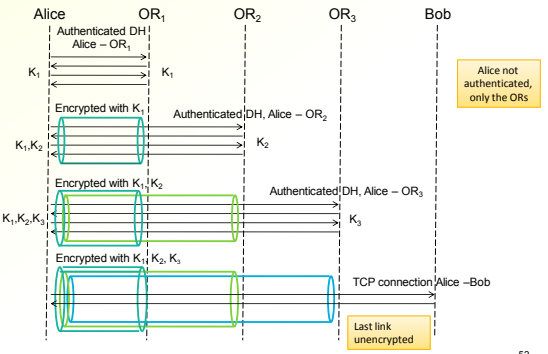
Low-latency anonymous routing

Tor

- "2nd generation onion router"
- Mix networks are ok for email but too slow for interactive use like web browsing
- New compromise between efficiency and anonymity:
  - No mixing at the onion routers
  - All packets in a session, in both directions, go through the same routers
  - Short route, always three onion routers
  - Tunnels based on symmetric cryptography
  - No cover traffic
  - Protects against local observers at any part of the path, but vulnerable to a global attacker
- More realistic attacker model: can control some nodes, can sniff some links, not everything
- SOCKS interface at clients → works for any TCP connection

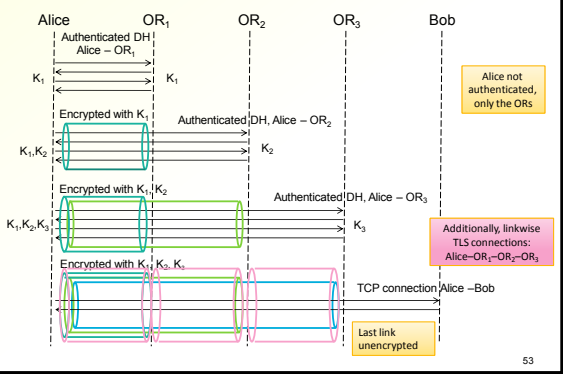
51

Tunnels in Tor [Danezis]



52

Tunnels in Tor [Danezis]



53

Tor limitations (1)

- Identifying packet streams is very easy
  - Passive fingerprinting by packet size, timing
  - Active traffic shaping (stream watermarking)

→ Anonymity compromised if attacker can see or control the first and last link

  - Includes attackers who own the first and last OR → longer routes do not help
  - If c is the fraction of compromised ORs, probability of compromise is c^2
- Why three routers?
  - Out of habit?
  - Attacker in control of 1st or last router cannon immediately go and compromise the other

54

Tor limitations (2)

- Client must know the addresses and public keys of all onion routers
  - If client only knows a small subset of routers, it will always choose all three routers from this subset → implicit identifier
  - E.g. client knows 10 out of 1000 routers = 1%
    → Attacker in control of the last router can narrow down the client identity to $(0.01)^2 = 0.01\%$ of all clients
    → Attacker in control of two last routers can narrow the client identity down to $(0.01)^3 = 0.0001\%$ of all clients
- DNS leaks information to the access network
- Blacklisting of entry or exit nodes

55

Applications of anonymous routing

- Censorship resistance, freedom or speech
- Protection against discrimination, e.g. geographic access control or price differentiation
- Business intelligence, police investigation, political and military intelligence
- Whistle blowing, crime reporting
- Electronic voting
- Crime, forbidden and immoral activities?

56

Exercises

- What is the entropy of a pool mix that has threshold k?
- How does the strength of anonymity protection in a mix network change as a function of the route length?
  - Consider attackers who own 10% or 90% of mixes. Also, longer routes have more failures. How could that affect anonymity?
- Tor does not protect against fingerprinting or watermarking of packet streams. How would anonymity in Tor change if we used 1,2,3,or more routers?
- Install Tor client on your machine and try using it

57