Security and Privacy in Wireless Networks

Mohammad Hossein Manshaei

manshaei@gmail.com

Chapter 4: (secowinet.epfl.ch)

Address stealing, Sybil attack, node replication attack

NAMING AND ADDRESSING

- **Naming and addressing are fundamental for networking**

  notably, routing protocols need addresses to route packets
  services need names in order to be identifiable, discoverable, and useable

Chapter Outline

4.1 The future of naming and addressing in the Internet

4.2 Attacks against naming and addressing

4.3 Protection techniques

Future of Naming in the Internet

Principle 1:

Names **should bind protocols only to the relevant aspects** of the underlying structure; binding protocols to irrelevant details unnecessarily limits flexibility and functionality.

New Identification Layers

- User-level descriptors (ULDs)
- Service identifiers (SIDs)
- Endpoint identifiers (EIDs)

Example: Web Browsing

1. A user types a ULD (in this case a search query) in a search engine running on the client.
2. The search engine returns an SID.
3. The application then resolves that SID, thus receiving one or more EIDs that identify the end-hosts that run the service.
4. The client will then establish one or more connections (e.g., TCP) with the service EIDs.
5. The transport layer then resolves the EID to the current set of IP addresses to which the EID is attached

Future of Naming in the Internet

Principle 2:

Names, if they are to be persistent, **should not impose arbitrary restrictions** on the elements to which they refer.

Future of Naming in the Internet

Principle 3:

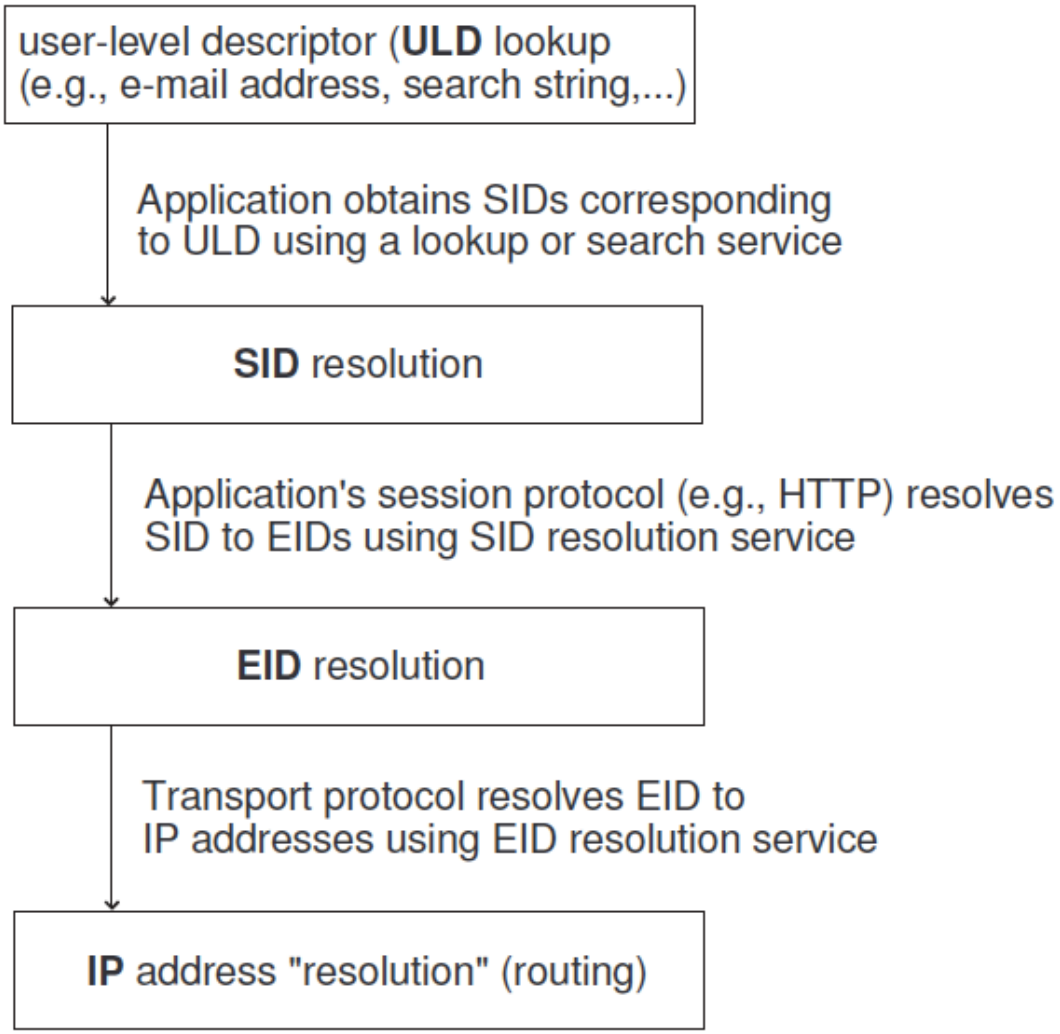
A network entity should be able to direct **resolutions** of its name not only to its own location, but also **to the locations or names of chosen delegates**.

Future of Naming in the Internet

Principle 4:

Destinations, as specified by sources and also by the resolution of SIDs and EIDs, should be generalizable to **sequences of destinations**

The Naming Layers in a Possible Future Organization of the Internet



Resistance to attacks of the described architecture

- Viruses and spam
- Phishing attacks
- Denial-of-Service

Chapter Outline

4.1 The future of naming and addressing in the Internet

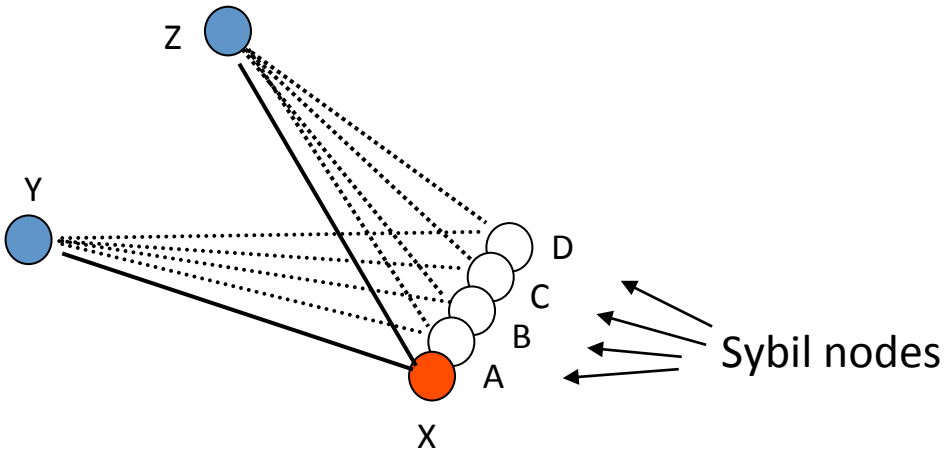
4.2 Attacks against naming and addressing

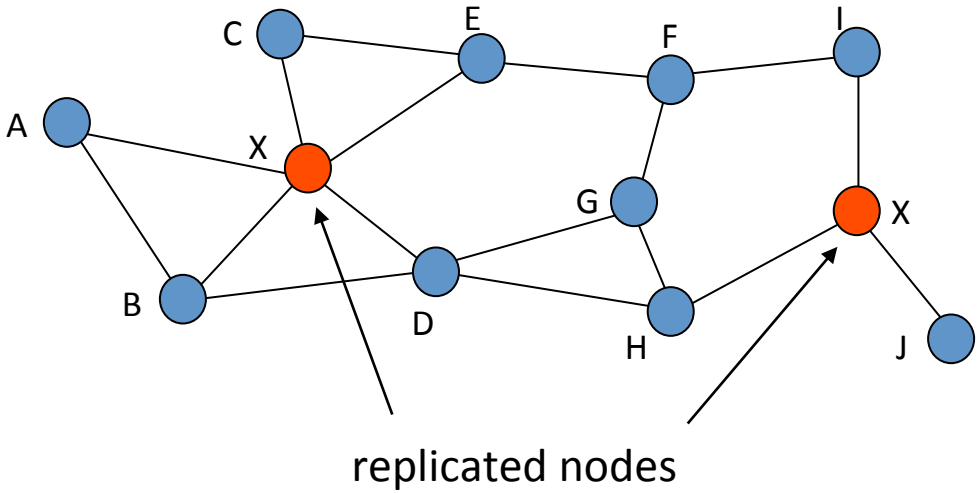
4.3 Protection techniques

Attacks Against Naming and Addressing

- **Address stealing**
  - adversary starts using an address already assigned to and used by a legitimate node
- **Sybil attack**
  - a single adversarial node uses several invented addresses
  - makes legitimate nodes believe that there are many other nodes around
- **Node replication attack**
  - dual of the Sybil attack
  - the adversary introduces replicas of a single compromised node using the same address at different locations of the network

Illustration of the Sybil and Node Replication Attacks





Chapter Outline

4.1 The future of naming and addressing in the Internet

4.2 Attacks against naming and addressing

4.3 Protection techniques

Centralized Protection Technique

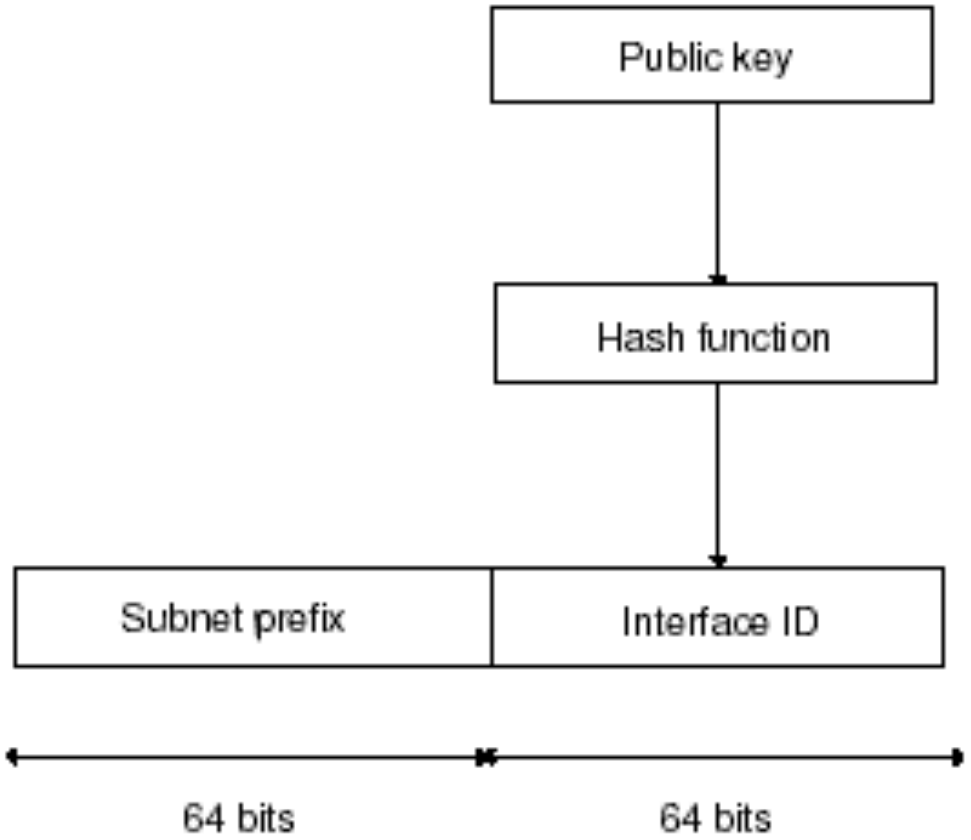
- Network operator manually distribute the identity along with a symmetric key to the subscriber (e.g., GSM)
- Internet Key Exchange (IKE) offers a centralized solution in the Internet
  - require a global key management infrastructure

➔ Hence, we need a distributed scheme

Cryptographically Generated Addresses (CGA)

- Aims at preventing address stealing
- General idea:
  - generate node address from a public key
  - corresponding private key is known only by the legitimate node
  - prove ownership of the address by proving knowledge of the private key

Example: IPv6



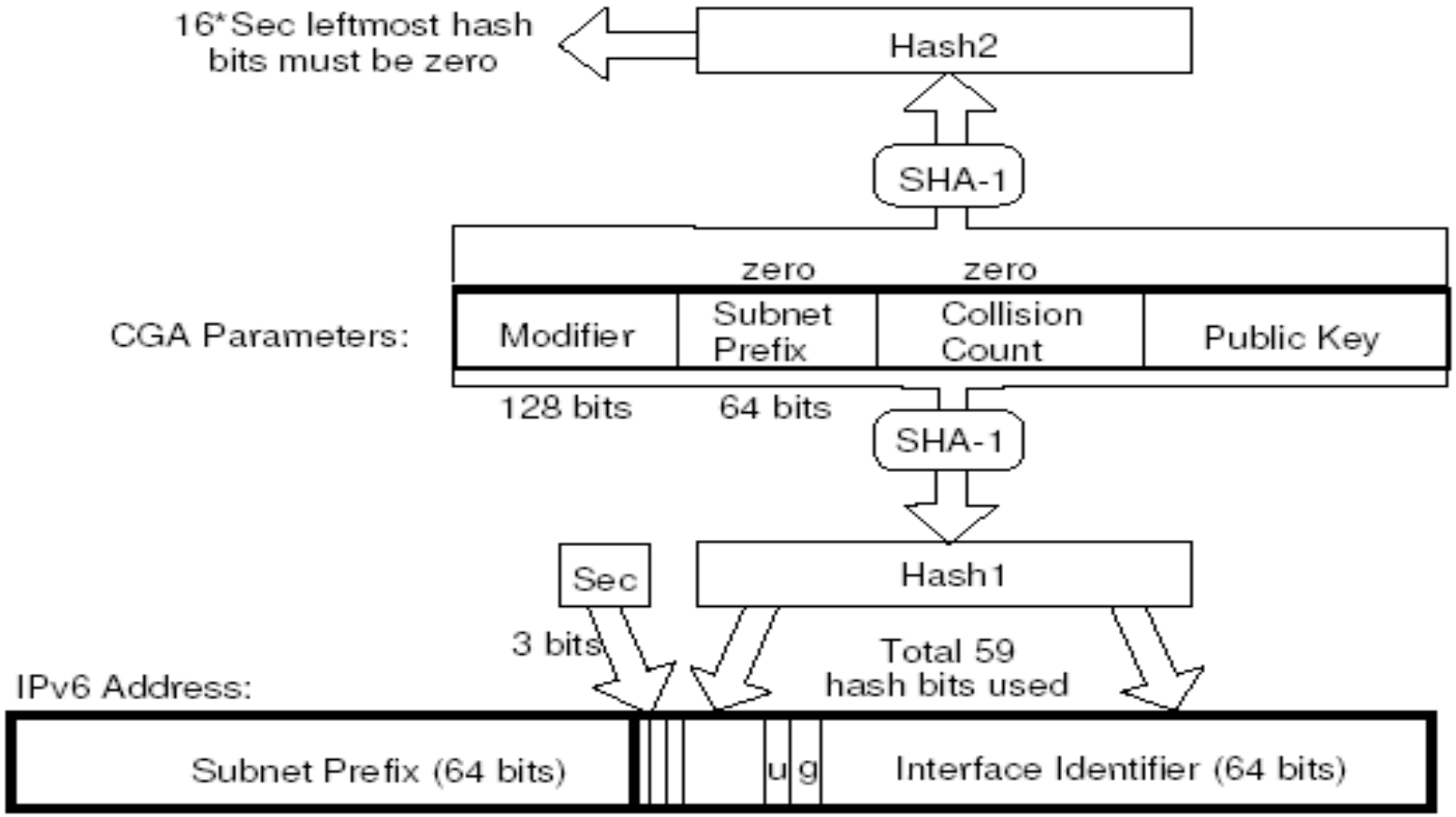
A Potential Problem with CGA

- Often only **a limited number of bits** of the address can be chosen arbitrarily (64 in our example)

- This number may be **too small to guarantee second pre-image resistance**
  - an adversary could pre-compute a large database of interface identifiers from public keys generated by himself, and use this database to find matches to victims' addresses

- A solution can be the technique called ***hash extension***
  - increase the cost of address generation, and hence the cost of brute-force attacks, while keep constant the cost of address usage and verification

Hash extension



Protocol for CGA Generation

1. Set the modifier field to a random 128-bit value.
2. Hash the concatenation of the modifier, 64+8 zero bits, and the encoded public key. The leftmost 112 bits of the result are Hash2.
3. Compare the 16*Sec leftmost bits of Hash2 with zero. If they are all zero (or if Sec=0), continue with Step (4). Otherwise, increment the modifier and go back to Step (2).
4. Set the collision count value to zero.
5. Hash the concatenation of the modifier, subnet prefix, collision count and encoded public key. The leftmost 64 bits of the result are Hash1.
6. Form an interface identifier by setting the two reserved bits in Hash1 both to 1 and the three leftmost bits to the value Sec.
7. Concatenate the subnet prefix and interface identifier to form a 128-bit IPv6 address.
8. If an address collision with another node within the same subnet is detected, increment the collision count and go back to step (5). However, after three collisions, stop and report the error.

Protocol for CGA Verification

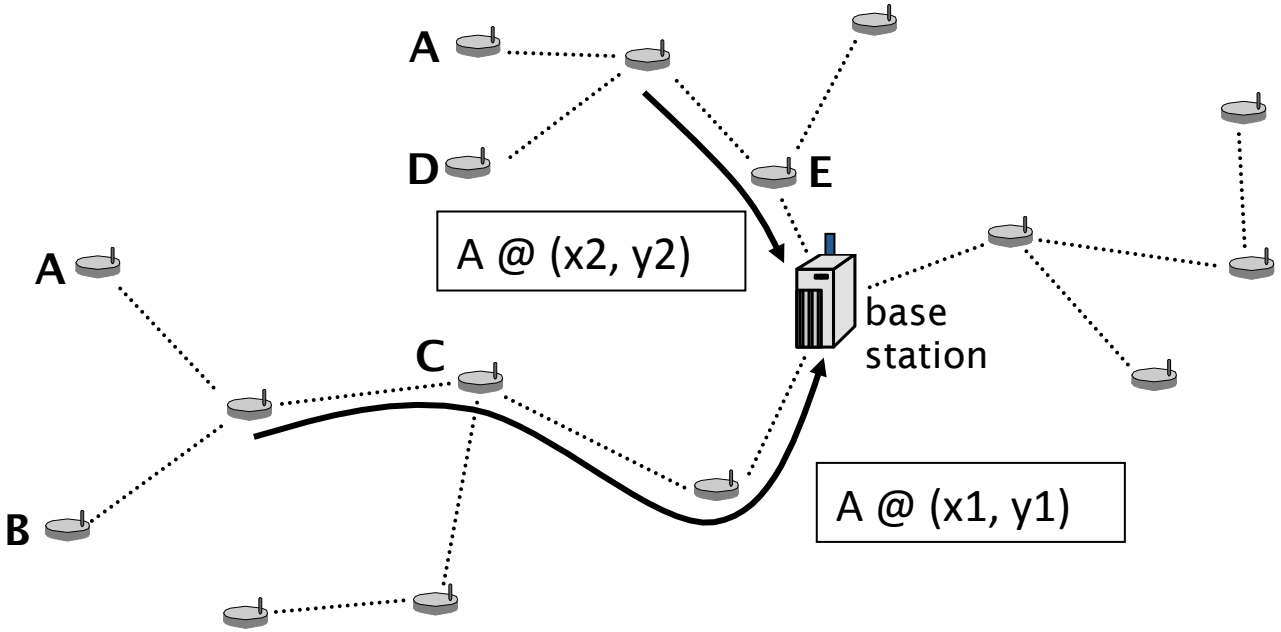
1. Check that the collision count value is 0, 1 or 2, and that the subnet prefix value is equal to the subnet prefix (i.e. leftmost 64 bits) of the address. The CGA verification fails if either check fails.
2. Hash the concatenation of the modifier, subnet prefix, collision count and the public key. The 64 leftmost bits of the result are Hash1.
3. Compare Hash1 with the interface identifier (i.e. the rightmost 64 bits) of the address. Differences in the two reserved bits and in the three leftmost bits are ignored. If the 64-bit values differ (other than in the five ignored bits), the CGA verification fails.
4. Read the security parameter Sec from the three leftmost bits of the interface identifier of the address.
5. Hash the concatenation of the modifier, 64+8 zero bits and the public key. The leftmost 112 bits of the result are Hash2.
6. Compare the 16*Sec leftmost bits of Hash2 with zero. If any one of these is nonzero, CGA verification fails. Otherwise, the verification succeeds.

Thwarting the Sybil Attack

- Note that CGAs do not prevent the Sybil attack
  - an adversary can still generate addresses for herself
- A solution based on a central and trusted authority
  - the central authority vouches for the one-to-one mapping between an address and a device
  - e.g., a server can respond to requests concerning the legitimacy of a given address
- Other solutions take advantage of some physical aspects
  - e.g., identify the same device based on radio fingerprinting

Thwarting the Node Replication Attack (1/2)

- **A centralized solution**
  - each node reports its neighbors' claimed locations to a central authority (e.g., the base station in sensor networks)
  - the central authority detects if the same address appears at two different locations
  - assumes location awareness of the nodes



Thwarting the Node Replication Attack (2/2)

- **A decentralized variant**
  - neighbors' claimed location is forwarded to *witnesses*
  - witnesses are randomly selected nodes of the network
  - if a witness detects the same address appearing at two different locations then it broadcasts this information and the replicated nodes are revoked

Analysis of the Decentralized Variant

- Total number of nodes is n
- Average number of neighbors is d
- Each neighbor of A forwards A's location claim with probability p to g randomly selected witnesses
- Average number of witnesses receiving A's location claim is p*d*g
- If there are L replicas of A, then for the probability of detection:

$$P_{det} > 1 - \exp(- L(L-1)(pdg)^2 / 2n)$$

- Numerical example:

  n = 10000, d = 20, g = 100, p = 0.5

  L = 2 → P_{det} ~ 0.63

  L = 3 → P_{det} ~ 0.95

Summary

- There are various attacks against naming and addressing
  - address stealing
  - Sybil attack
  - node replication attack
- Decentralization and lack of a central authority renders the defense against these attacks difficult
- Proposed solutions (CGA, node replication detection using witnesses) provide only probabilistic guarantees
  - parameters should be chosen carefully