CGA

Cryptographically Generated Addresses RFC3972

- Hash generated address
  - Hash defined data
    - Result -> local part
  - Include data with NS/NA
    - Other nodes can verify calculation
- Defined data
  - Could be anything
    - ... network prefix
  - If data constant
    - hash stable
    - same result every time
  - If data variable
    - result will vary
- Of itself not very useful
  - Anyone can copy data

CGA (2)

- What data to hash?
  - Network prefix
    - Means local part cannot be copied
      - from one prefix to another
  - Some other stuff...
  - Public Key
    - Half of public/private pair
- All this data included with NA/NS packet
  - Allows receiver to verify hash
  - Also tells receiver public key of sender
- Public key
  - Allows sending secret data to owner
    - Not useful here
  - Allows verification of signature

CGA (3)

- NS/NA message can be signed
  - Generate hash of all message contents
    - If anything altered, has changes
  - Encrypt hash value
    - Using private key
  - Include result in packet
- Receiver
  - Calculates hash of same data from packet
  - Decrypts signature
    - using public key from packet
  - compares hash values
    - calculated vs decrypted
  - If equal
    - packet verified

CGA (4)

- From this we know
  - Data originated at owner of address
    - Only node that generated address
  - Public key
    - used to generate address
    - used to decrypt signature
      - The same
  - Private key required to sign message
    - Only generator of address knows it
  - Data not modified
    - Hash verifies that
  - No replay
    - Timestamp added for that
  - NA is reply to NS
    - Nonce sent and returned

CGA (5)

- Note:
  - Anyone can generate a CGA
    - No element of authorisation to conect
  - Generated CGA known to belong to node
    - No information about node
      - except that it owns the CGA
- Knowing the CGA we communicate with
  - deciding it is the correct one
    - Problems for a higher layer

RA security

- More difficult problem
  - Need to demonstrate authorisation
  - This is an authorised router
  - Router is authorised to
    - forward packets
    - announce network prefixes
      - particular prefixes
- Authorisation
  - Usually based upon certificates
    - Statement from higher authority
      - Node does own certificate
      - Its contents are correct
- How do we know certificate is valid?
  - How do we know who is appropriate higher authority?

RA Security (2)

- Trust chains
  - New ICMPv6 messages
    - Request
    - Reply
      - Certificate Trust Chain
  - Each certificate in chain
    - authorised by the next
      - authorised to advertise certain prefixes
      - to be a router
    - authorisation is a subset
      - of that authorised in higher certificate
- Trust ends
  - high level certificate
  - allows advertising any prefix
    - Allocated to address registry
    - authorised by IANA
      - (or similar)

RA Security (3)

- Trusting top level certificate
  - It claims to be valid?
  - How do we know?
- Configured certificate
  - Nodes configured with high level certificate
  - Allows verification based upon local knowledge
- Configured vs Autoconfiguration
  - Aim is no configuration
    - No specific config of node
    - Certificate is common to everyone
      - Installed by manufacturer