Cryptographically Generated IPv6 Addresses (CGA)

- Basic idea:
  
  Interface Id = hash (Public Key)

  The public key is used to authenticate messages sent from the CGA address.

  ➡ Proof of address ownership without security infrastructure.

- Prior work:
  draft-roe-mobileip-updateauth, draft-montenegro-sucv, draft-nikander-ipng-pbk-addresses, draft-moskowitz-hip

- Covered by IPR
Problems

- 64 bit limit for hash length
  - eventual failure because of Moore’s law
  - pre-computation attacks ($2^{64}$ memory)
- Detailed formats and algorithms missing
- Drafts incompatible with each other and with standard authentication protocols
- Fully specified address formats and address-generation and verification algorithms

- The 64-bit limit effectively removed:
  - security parameter (Sec)
  - cost of generating an address multiplied by $2^{12 \times \text{Sec}}$
  - cost of attacks increased from $\sim 2^{62}$ to $2^{59+12 \times \text{Sec}}$
  - cost of authentication remains constant

- CGA address indicated by $g=1, u=1$
  (not essential but allows mixing of authenticated and unauthenticated nodes)
CGA Address Format

Hash1 = h (Public Key, Modifier, Routing Prefix, Collision Count)

- Routing Prefix: 64 bits
- Interface Id: 59 hash bits
- Security Parameter (Sec): 3 bits

\( \text{Hash1} = h (\text{Public Key}, \text{Modifier}, \text{Routing Prefix, Collision Count}) \)
CGA Address Format

Hash\(_1\) = \(h\) (Public Key, Modifier, Routing Prefix, Collision Count)

\[64\text{ bits}\]

Hash\(_2\) = \(h\) (Public Key, Modifier)

**New requirement:** Modifier must be chosen so that Hash\(_2\) begins with \(12 \times \text{Sec}\) zero bits.
Two CGA Parameter Formats

1. Certificate format:
   - Public key and parameters stored in a self-signed X.509 certificate — Easy to use in certificate-based authentication protocols
   - New certificate extension contains the parameters: Modifier, Routing Prefix, Collision Count

2. Optimized (short) format:
   - Concatenation of the public key and parameters
   - Public key + 29 bytes
   - Verifier needs: signed message (e.g. NA), source IP address, and parameters in either format