IPv6 CGAs: Balancing between Security, Privacy and Usability

Ahmad Alsadeh

Birzeit university

Outline

- IPv6 Configuration
- IPv6 StateLess Address Auto-Configuration
  - Extended Unique ID (EUI-64)
  - Privacy Extension RFC 4941
  - Cryptographically Generated Addresses (CGA)
- Problem statement
- Our Proposed Approach (Modified CGA)
  - CGA Modifications
  - Implementation
  - Limitations and Deployment Considerations
- SEND Implementations
- Conclusion

IPv6 Configuration

IPv6 Address (128 bits)

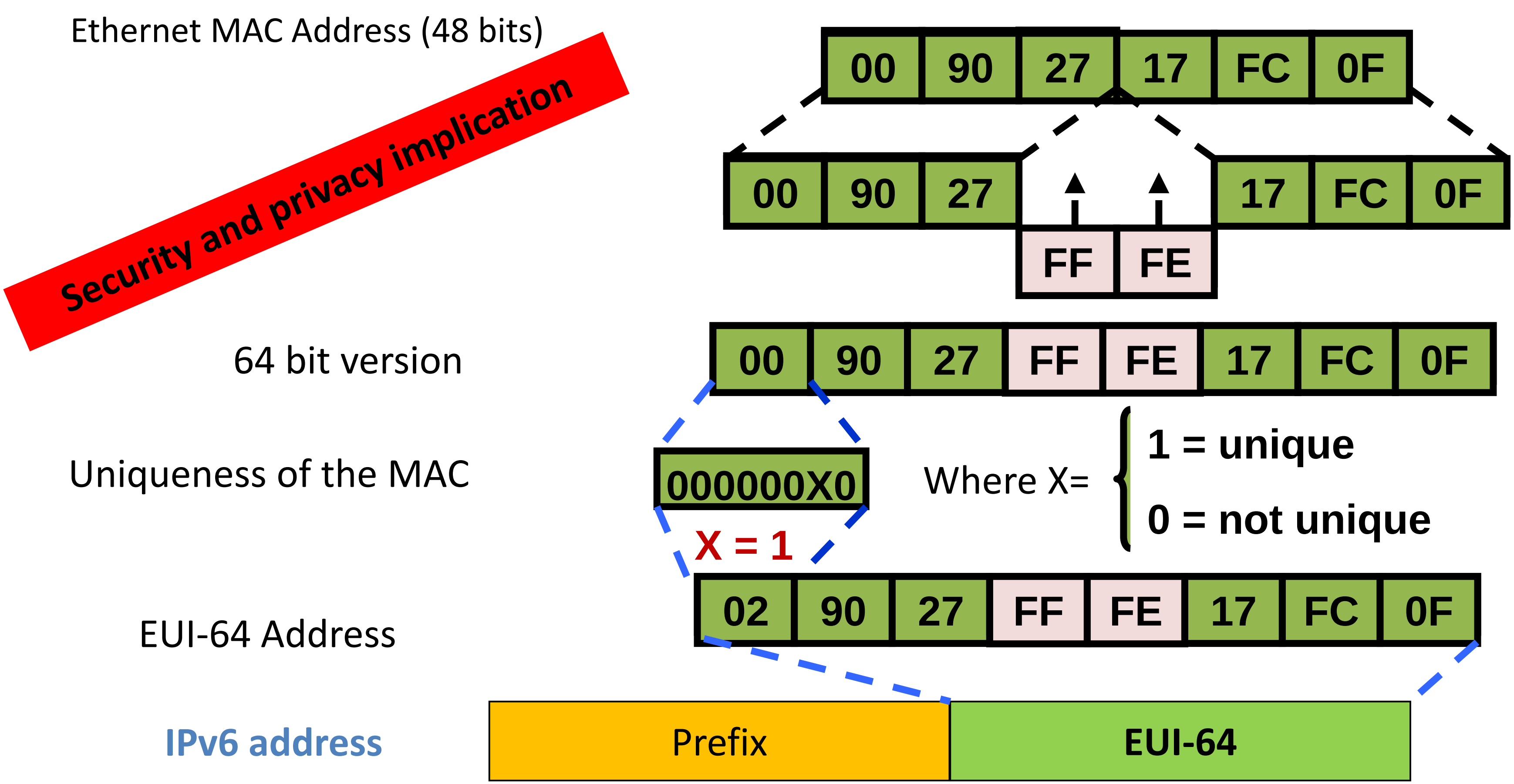
64 bits	64 bits
Subnet Prefix	Interface ID

Network ID can be configured
- Manual
- Stateful
- Stateless: prefex can be
  - Link-Local prefix (FE80::/64)
  - Global prefix (2001:DB8:123:/64)

Interface ID can be configured
- Manually
- Stateful (DHCPv6)
- Stateless
  - Auto-configuration Based on the MAC address (EUI-64-based interface ID)
  - Privacy Extension (Pseudo Random ID)
  - Cryptographically Generated Addresses (CGA)

Our foucus on IPv6 StateLess Address Auto-Configuration (SLAAC)

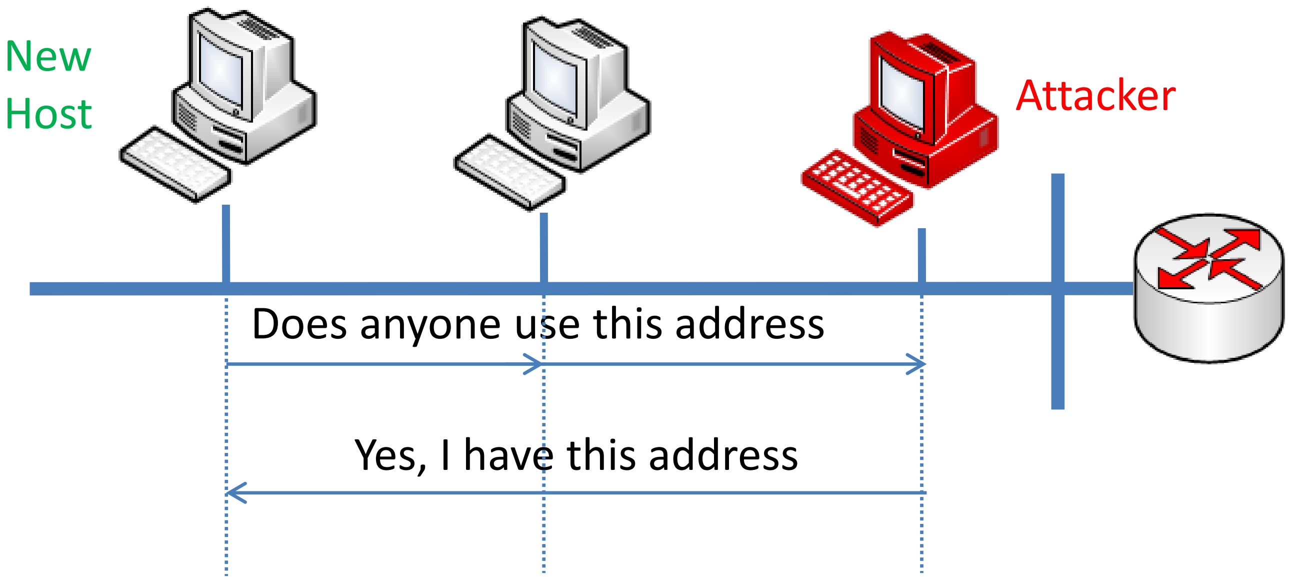
1. Extended Unique ID (EUI-64)



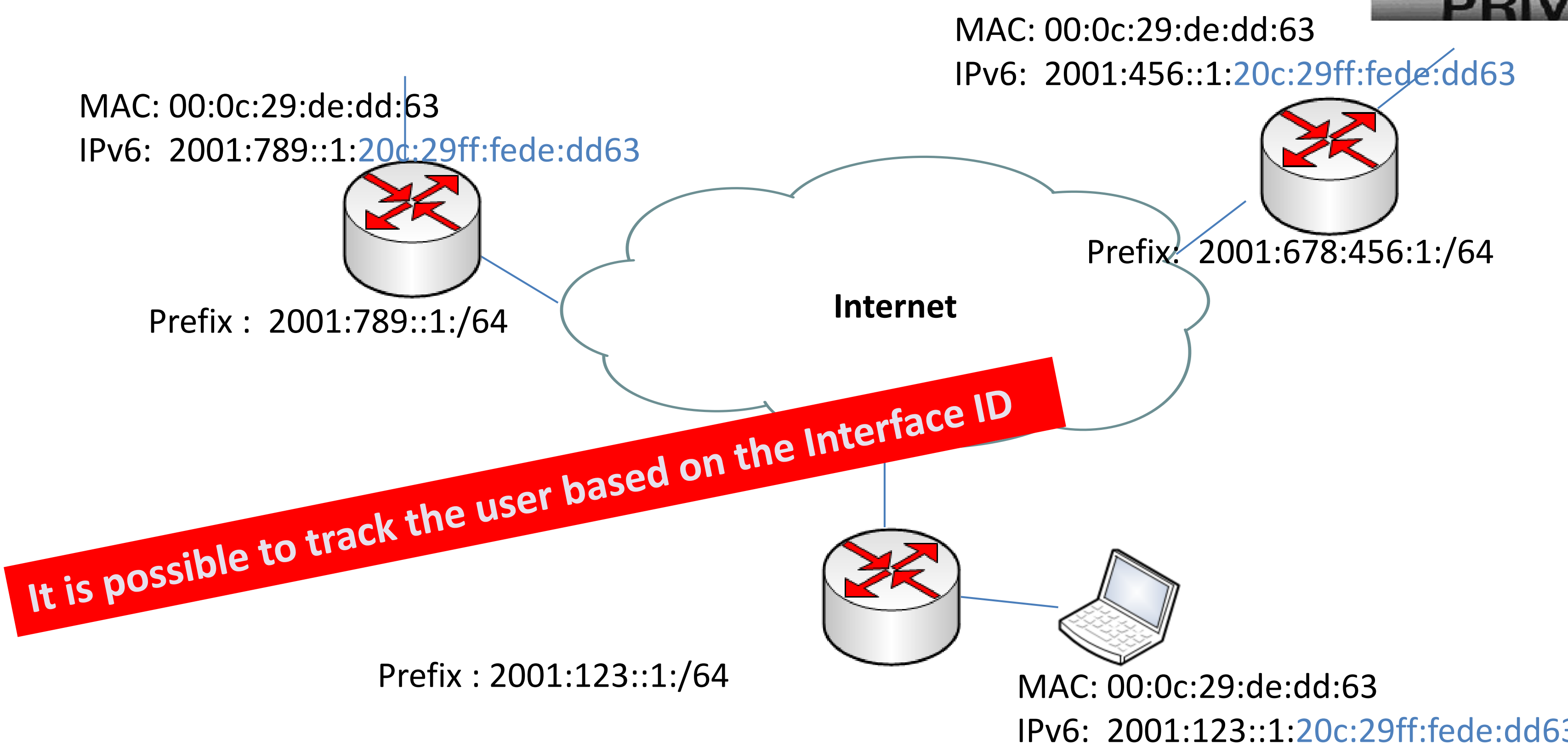
Reference: http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucm/srnd/ipv6/ipv6srnd/basics.pdf

EUI-64: Security Implication

– Duplicate Address Detection (DAD) DoS attack
  - THC-IPv6 Attack Suite http://www.thc.org/thc-ipv6/
    – *dos-new-ip6*

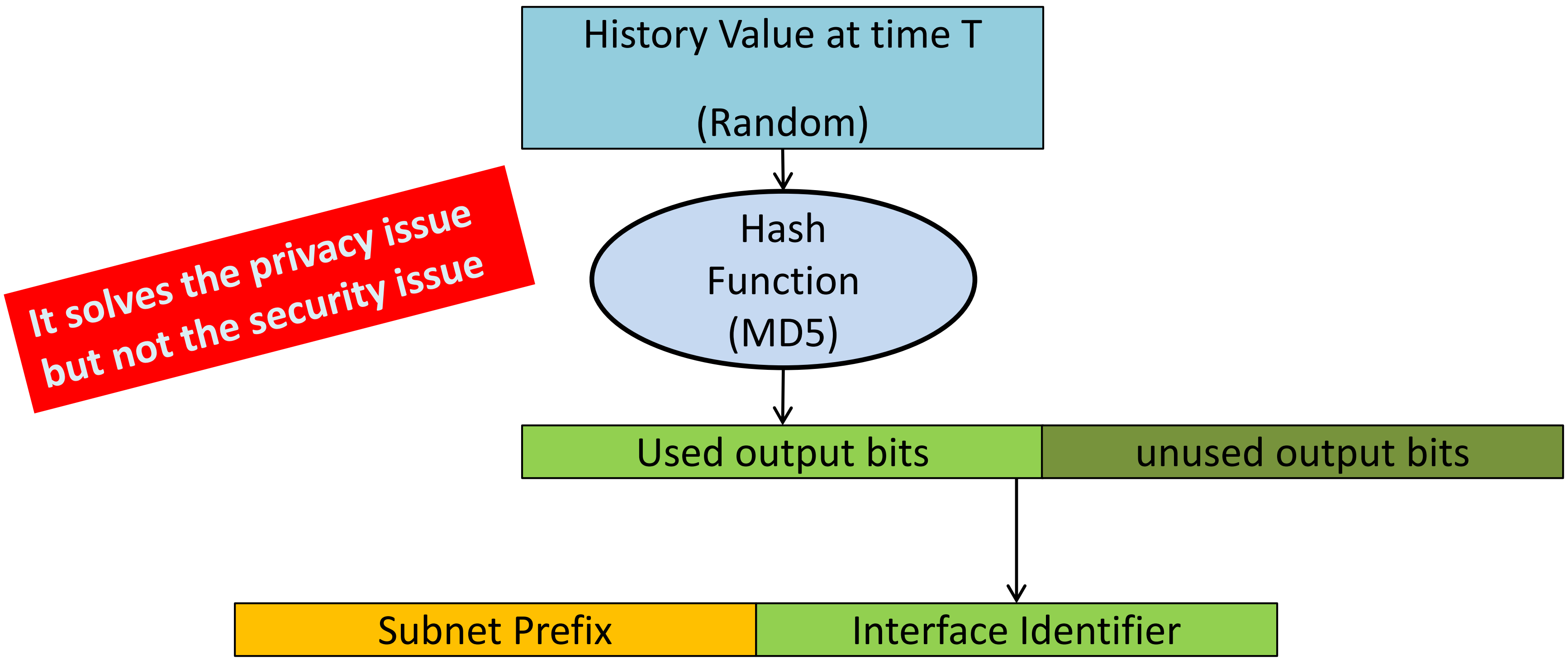


EUI-64: Privacy Implication



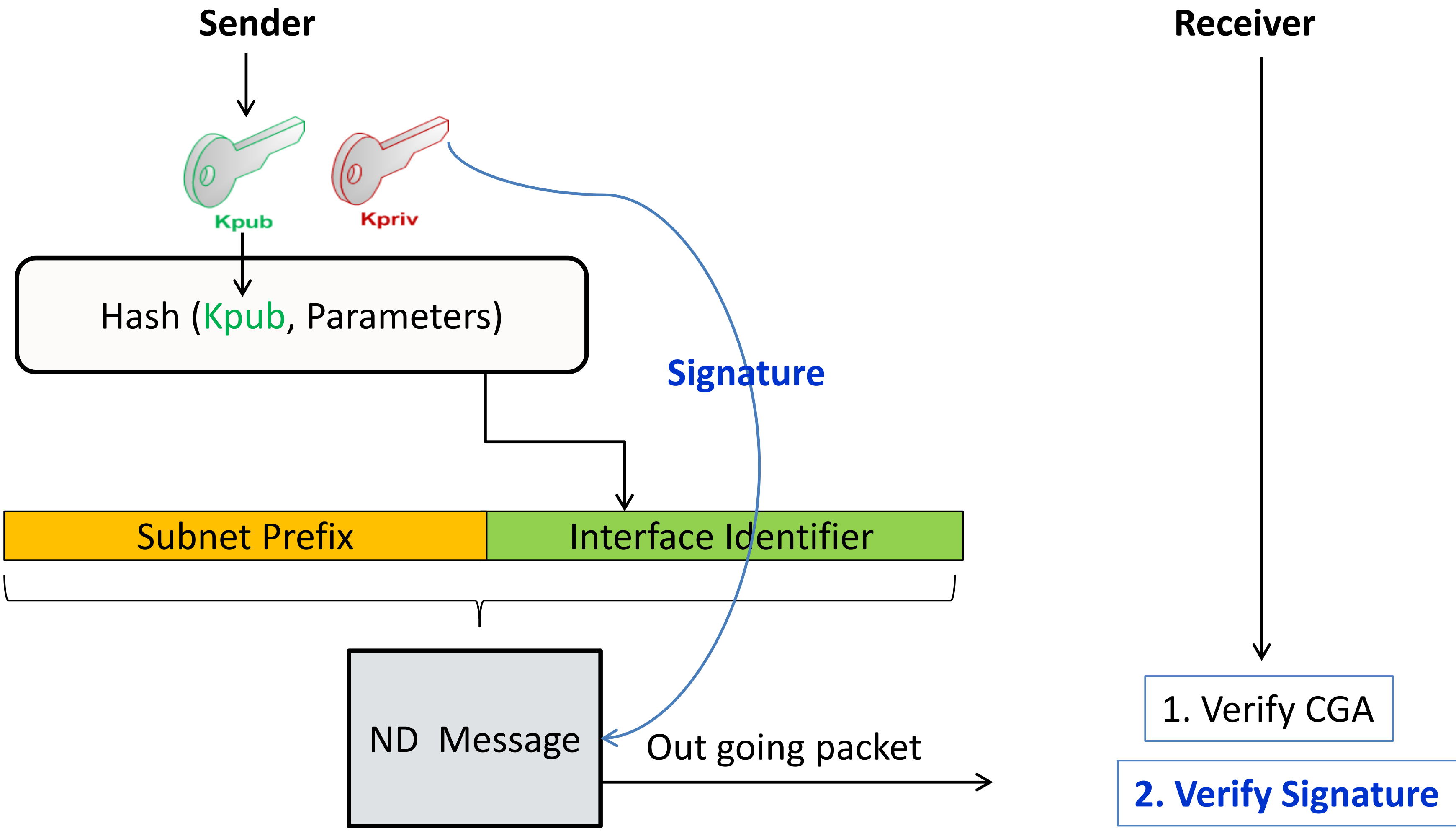
MAC addresses are usually the least of a user's security concern - most people happily accept browser cookies without thinking

2. Privacy Extension - RFC 4941



Reference: J. Ullrich and E. Weippl, "**Privacy is not an option: Attacking the IPv6 privacy extension**," in Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID), 2015, pp. 448-468.

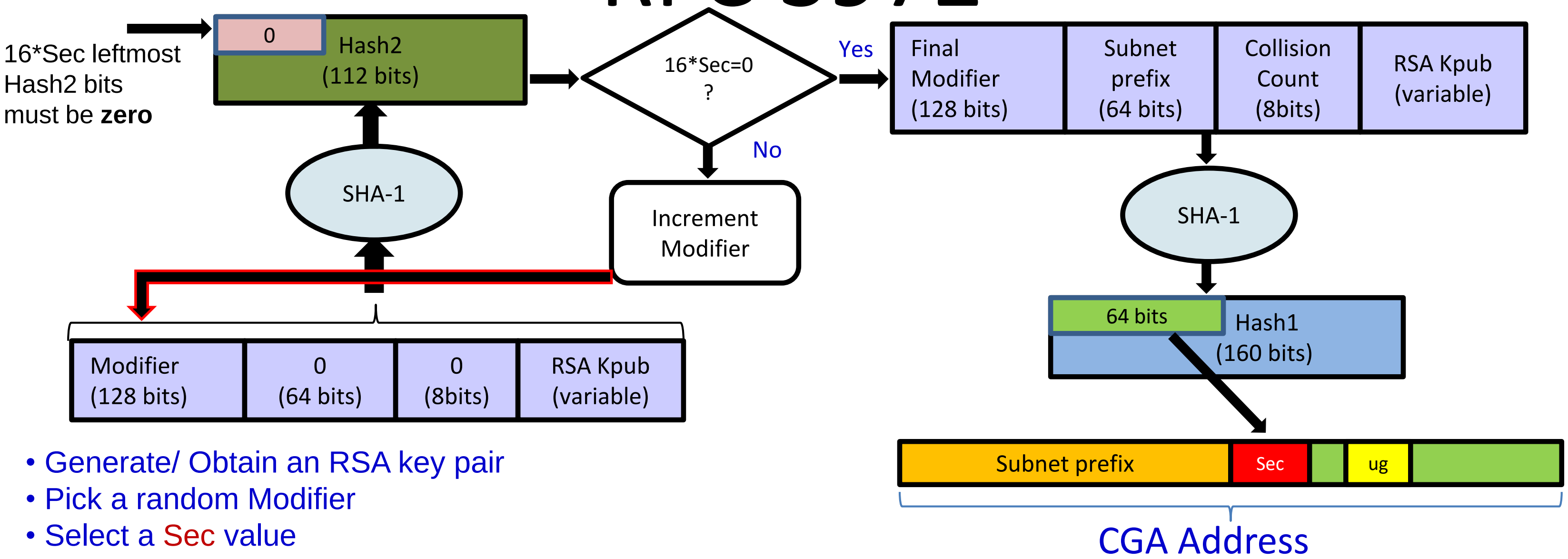
3. Cryptographically Generated Addresses (CGA): Basic idea



CGA Solves the Security and Privacy

- Security
  - CGA bound the address with corresponding public key. Therefore, no address spoofing – prevent the spoofing attack
- Privacy
  - The Interface ID a hash value (random) -- protect the tracking possibility
- But at what cost the security and privacy have been achieved?
  - Let us see CGA in more details

CGA: Generation algorithm RFC 3972

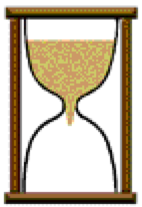


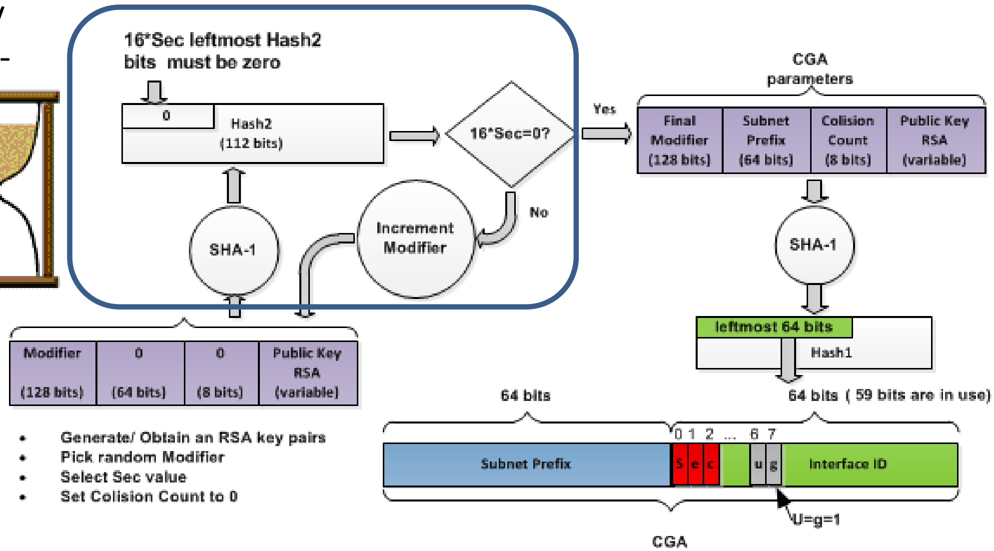
- Generate/ Obtain an RSA key pair
- Pick a random Modifier
- Select a Sec value
- Set Collision Count to 0

1. Set CGA initial values
2. Concatenate (modifier, 0, 0, Kpub)
3. Execute SHA-1 algorithm
4. Compare the 16xSec = 0 ?
5. Concatenate (CGA parameters)
6. Execute SHA-1 algorithm
7. Form an interface ID
8. Concatenate (Prefix, Interface ID)
9. Check the uniqueness of IPv6 address

CGA – Computation Cost Concerns

Increase artificially the cost of a brute-force attack

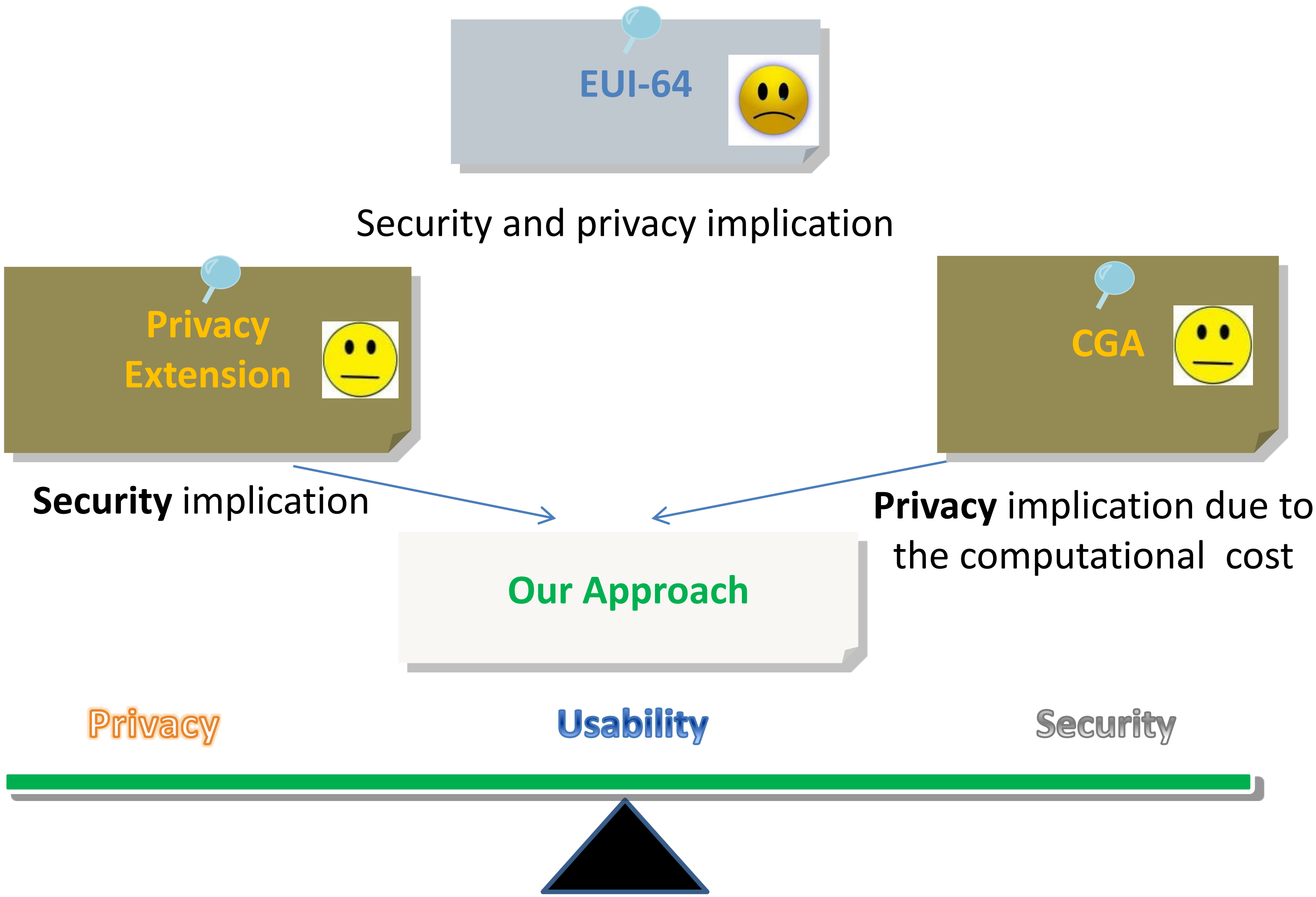




CPU 2.6 GHz	
Sec	**Time**
1	~ 1 Sec
2	~ 3 hours
3	~ 12 years

- Sec (0 to 7), unsigned 3-bit integer, is scale factor
  - The address generator needs on average O(2^{16xSec})
  - high Sec value may cause unacceptable delay
- It is likely that once a host generates an acceptable CGA, it will continue to use this address→ hosts using CGAs still being susceptible to privacy related attacks.

Problem statement



Our proposed approach (Modified CGA)

- Two **main modifications** to CGA
  1. Setting a CGA Address lifetime
  2. Reducing the granularity of CGA security levels

and

  3. Automatic key pair generation

1. Setting a Lifetime for Temporary CGA

- A CGA address has an associated lifetime that indicates how long the address is bound to an interface
- Once the lifetime expires, the CGA address is deprecated
  - The deprecated address should not be used for new connections
- A new temporary CGA address should be generated:
  - When a host joins a new subnet
  - Before the lifetime for the in-use CGA address has expired
  - When the subnet prefix lifetime has expired
  - When the user needs to override the default value

2. Reducing the Granularity of CGA Security Levels

– The granularity factor 16 is relatively large

- Sec value 0 or 1 can be used in practice

Sec	Granularity		
	16	**8**	**4**
1	427 ms	121 ms	117 ms
2	5923857 ms	425 ms	128 ms
3	*	88217 ms	135 ms

– We choose the granularity factor 8 for the following reasons:

- It is unnecessary to select a high Sec when using a short lifetime
- computation costs of CGA is usually much more important for mobile devices which have limited resources (e.g., CPU, battery, …)
- The multiplication factor of 8 increases the maximum length of the *Hash Extension* up to 56 bits which is sufficient (59-115 bits total hash length)

3. Automatic Key Pair Generation

– Setting the keys automatically is better for the following reasons:
  - Protects the user's privacy
  - The keys are not vulnerable to theft
  - Easier for end user
  - The key generation is small portion of the total CGA generation time

Secure neighbor discovery (SEND)

- SEND has three ingredients
  1. CGA-based signatures
     - Prevents NA spoofing
     - Prevents address squatting in DAD
     - Zero-configuration security!
  2. Certificate-based authorization of routers
     - Certificate authorizes router for a an address prefix
     - Extension to X.509 to certify IPv6 address allocation [RFC 3779]
     - Requires hosts to know the root key; currently no global CA hierarchy
  3. Freshness:
     - Timestamp in unsolicited advertisement and redirect
     - Nonce in NS and RS, copied to NA and RA

Modified-CGA Implementation

- We modified the CGA part of our SEND implementation (WinSEND) to include the proposed modifications
  - lifetime, granularity, and the automatic key generation
- The user can override the default parameters
  - Sec value
  - Granularly : 8*sec
  - Max IP validation: 24 hours
  - Key generation

SEND Implementations

- WinSEND
- NDprotector, Telecom SudParis
- Cisco IOS 12.4(24)T and newer
- Easy-SEND
- Docomo USL SEND fork
- ipv6-send-cga, Huawei and Beijing University of Posts and Telecommunications
- Native SeND kernel API
- TrustRouter
- USL SEND (discontinued), NTT DoCoMo

Limitations and Deployment Considerations

- Changing the CGA granularity to 8 requires updating the CGA RFC
- The other modifications do not affect the CGA algorithm and the way of communicating
- There are some implications and deployment considerations for the use of changeable addresses
  - May cause unexpected difficulties with some applications
  - May have performance implication that might impact user experience
  - Protecting the users‘ privacy may conflict with the administrative needs
  - Deleting the deprecated addresses requires awareness of the upper layers applications

Conclusion

- CGA can be used to prove the ownership of an IPv6 address, but it might be susceptible to privacy related attacks
- the privacy extensions protect the users' privacy but are of no value to related address spoofing attacks
- We integrate the privacy extensions into CGA to resolve both privacy and security issues for IPv6 addresses in a practical way

Thank you