eine Einführung in CGA und SeND

"Too old to rock 'n' roll, to young to die?"

Wer spricht?

Carsten Strotmann

Men & Mice, Reykjavik, Island

Forschung, Schulung und Beratung

IPv6 seit 2006

DNS, DNSSEC, DANE, DHCP

Linux, OpenBSD, Solaris/OpenSolaris, MacOS X, Plan 9/Inferno

'Unix für MacOS X Admins', xBSD "pf" Firewall

Forth Programmierung (auf Mikrocontrollern),

CGA und SeND

Cryptographically Generated Addresses (CGA)

Secure Neighbourhood Discovery (SeND)

Alternative Anwendungen von CGA

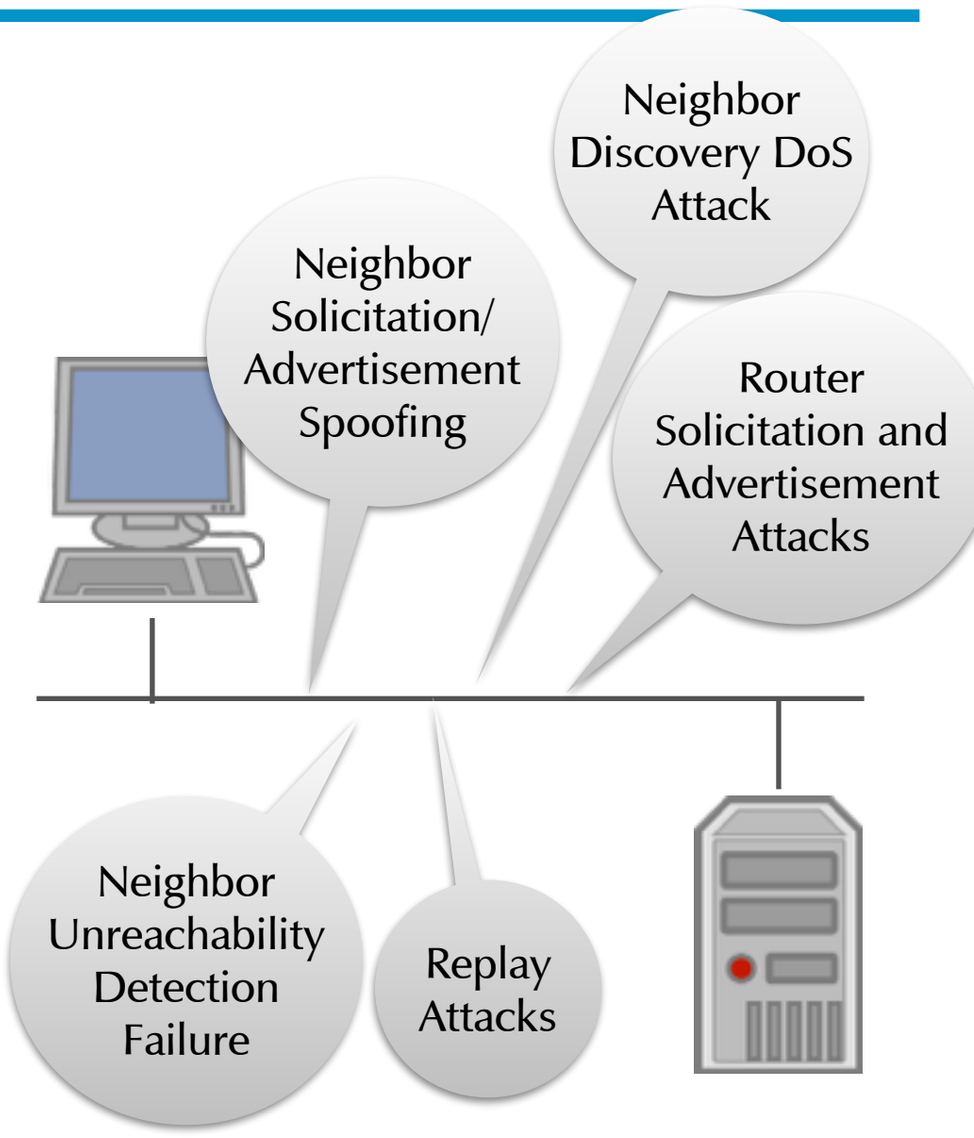
Alternativen zu CGA und SeND

Warum SeND und CGA

Es sind eine Reihe von Angriffen auf das IPv6 Neighbourhood Detection Protokoll (NDP) bekannt

siehe Vorträge von F. Gont, Marc Heuse und Eric Vinke

diese Angriffe sind eine Gefahr; speziell in offenen Drahtlos-Netzwerken



kryptographisch generierte Adressen — CGA

Öffentlicher RSA Schlüssel

Server

Privater RSA Schlüssel

Client

Öffentlicher RSA Schlüssel

Privater RSA Schlüssel

CGA

Öffentlicher RSA Schlüssel

+

```
0100101
1110010
1100110
1101101
1100001
```

Zufallsdaten (Modifier)

+

IPv6 Prefix (2001:db8:100:ae)

Client

SHA1 Hash

`2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c`

CGA

Öffentlicher RSA Schlüssel

+

Server

0100101
1110010
1100110
1101101
1100001

Zufallsdaten (Modifier)

+

IPv6 Prefix (2001:db8:100:ae)

SHA1 Hash

2001:db8:100:ae:34cc:6bba:5947:d208

CGA

2001:db8:100:ae:34cc:6bba:5947:d208

Server

Privater RSA Schlüssel

Client

Anfrage

+

Öffentlicher RSA Schlüssel

+

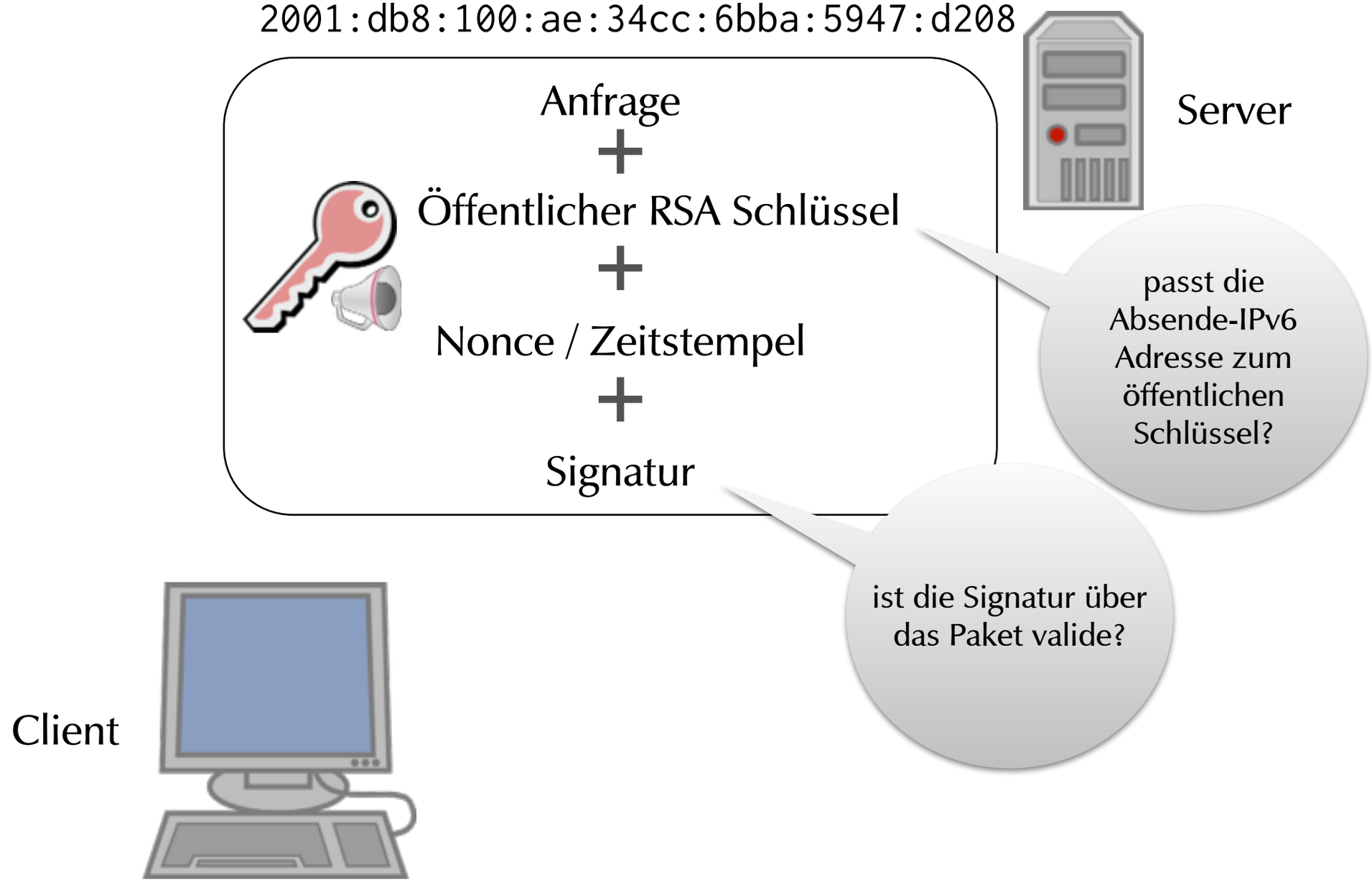
Nonce / Zeitstempel

+

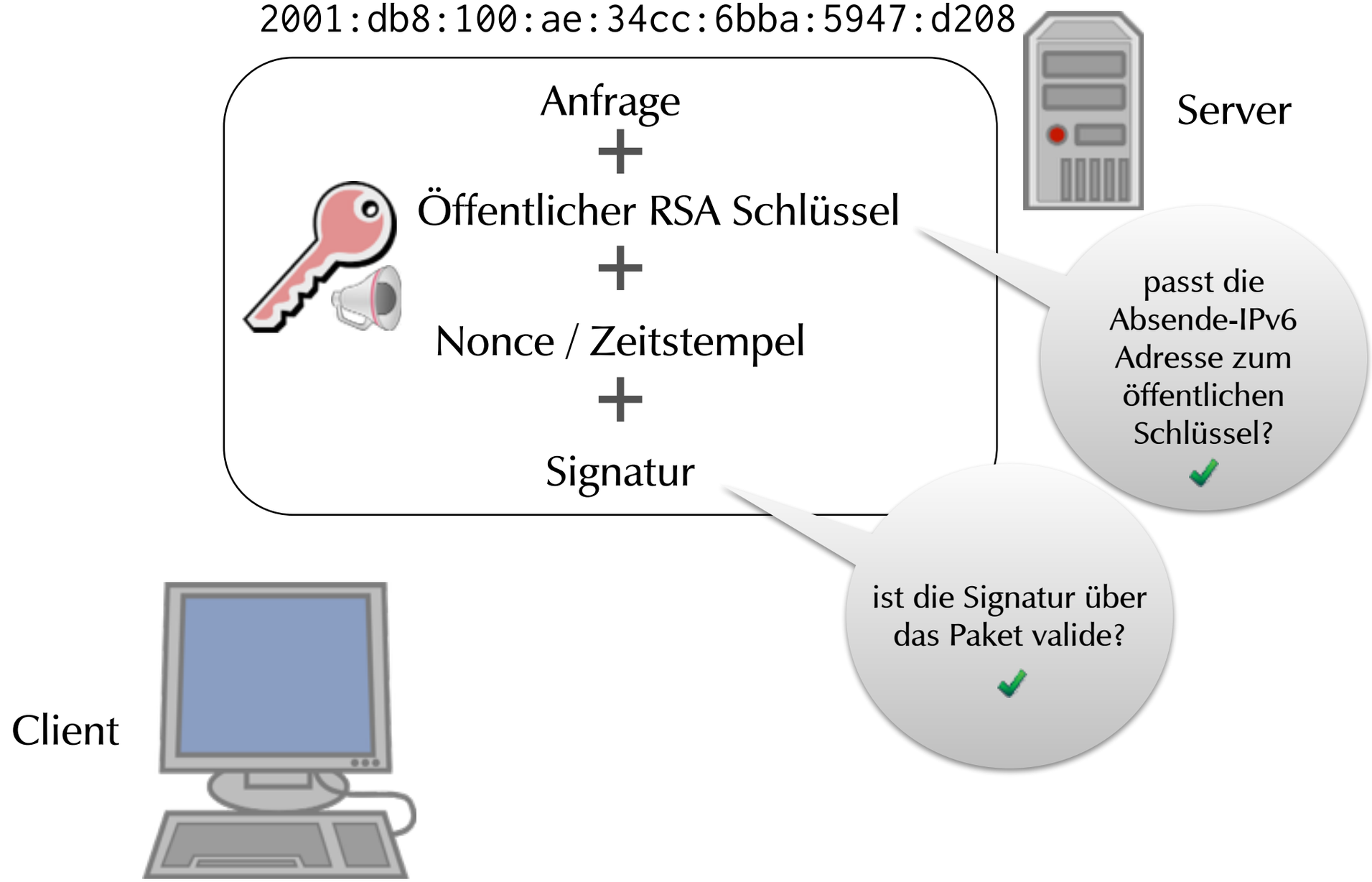
Signatur

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

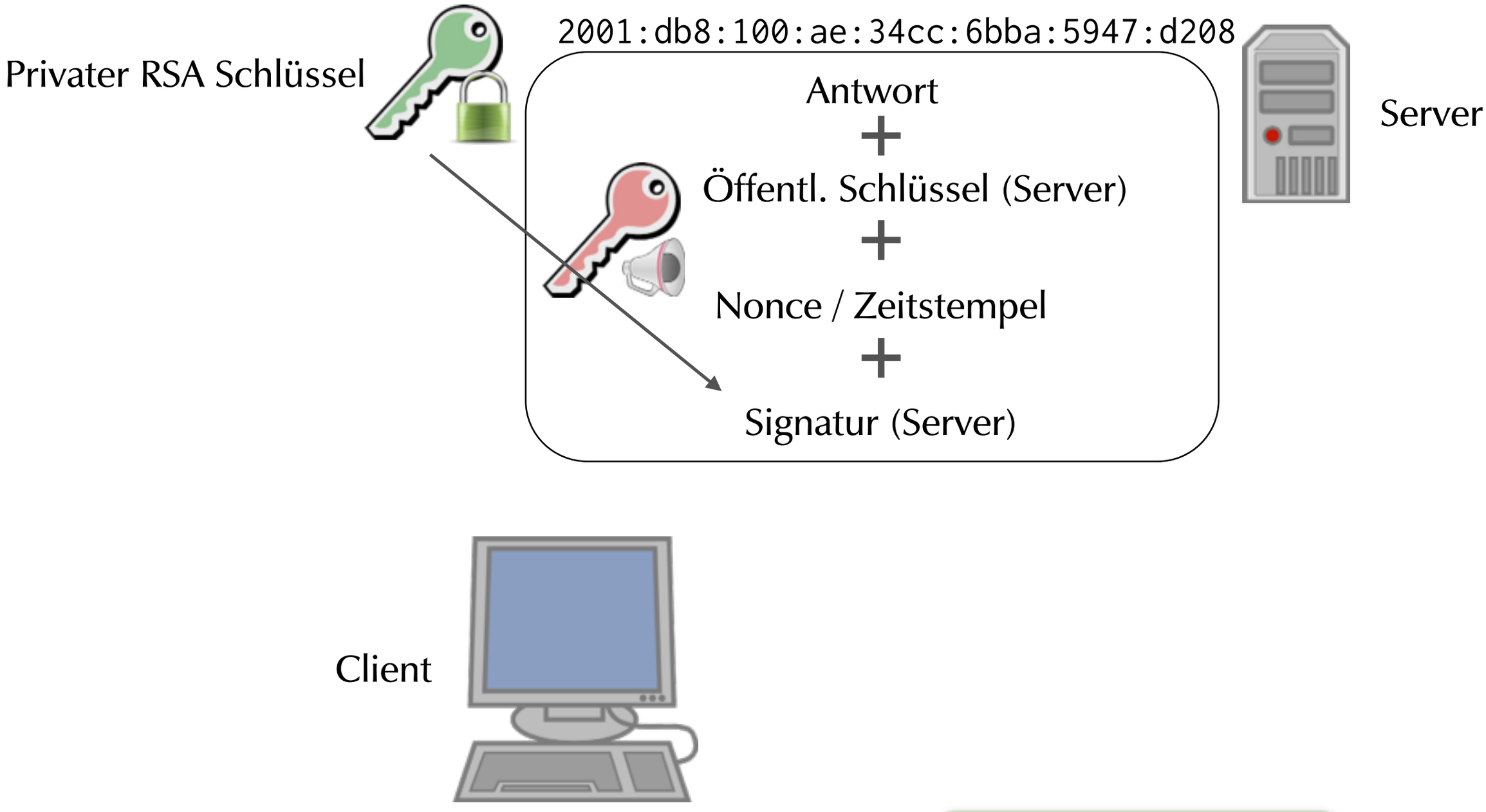
CGA



CGA



CGA



CGA

2001:db8:100:ae:34cc:6bba:5947:d208

Server

passt die Absende-IPv6 Adresse zum öffentlichen Schlüssel?

Antwort

+

Öffentl. Schlüssel (Server)

+

war die NONCE korrekt?

Nonce / Zeitstempel

+

Signatur (Server)

ist die Signatur über das Paket valide?

Client

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

CGA

2001:db8:100:ae:34cc:6bba:5947:d208

Server

passt die Absende-IPv6 Adresse zum öffentlichen Schlüssel?

Antwort

+

Öffentl. Schlüssel (Server)

+

war die NONCE korrekt?

Nonce / Zeitstempel

+

Signatur (Server)

ist die Signatur über das Paket valide?

Client

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

Der CGA Parameterblock

Feld Name	Größe	Beschreibung
Modifier	128 bit	Zufallsdaten
Subnet Prefix	64 bit	IPv6 Subnetz Prefix
Collision Count	8 bit	Anzahl der Adressen-Kollisionen
Public Key	variabel	X509 SubjectPublicKeyInfo (DER kodiert)
Erweiterungs-Felder	variabel	TLV (Type/Length/Value) kodierte Erweiterungen

CGA Adressen erzeugen

CGA Adressen erzeugen

```
wähle einen Wert für "sec" (0-7)
```

CGA Adressen erzeugen

Modifier

```
0100101
1110010
1100110 +
1101101
1100001
```

```
wähle einen Wert für "sec" (0-7)
```

CGA Adressen erzeugen

Modifier 9 x "0"

```
0100101
1110010
1100110 + 000000000 +
1101101
1100001
```

wähle einen Wert für "sec" (0-7)

CGA Adressen erzeugen

Modifier

9 x "0"

```
wähle einen Wert für "sec" (0-7)
```

```
0100101
1110010
1100110
1101101
1100001
```

+ `000000000` +

+

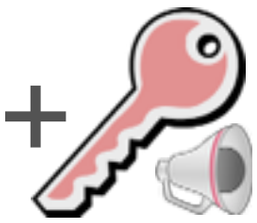
CGA Adressen erzeugen

Modifier

9 x "0"

```
0100101
1110010
1100110
1101101
1100001
```

+ `000000000` +



+ Extension Fields

```
wähle einen Wert für "sec" (0-7)
```

CGA Adressen erzeugen

Modifier

9 x "0"

wähle einen Wert für "sec" (0-7)

CGA Adressen erzeugen

Modifier

9 x "0"

wähle einen Wert für "sec" (0-7)

CGA Adressen erzeugen

Modifier

9 x "0"

```
wähle einen Wert für "sec" (0-7)
```

```
0100101
1110010
1100110
1101101
1100001
```

+ `000000000` +

+ Extension Fields

SHA1

Hash2

112 bits

CGA Adressen erzeugen

Modifier

9 x "0"

wähle einen Wert für "sec" (0-7)

0100101
1110010
1100110
1101101
1100001

+ 000000000 +

+ Extension Fields

SHA1

Hash2

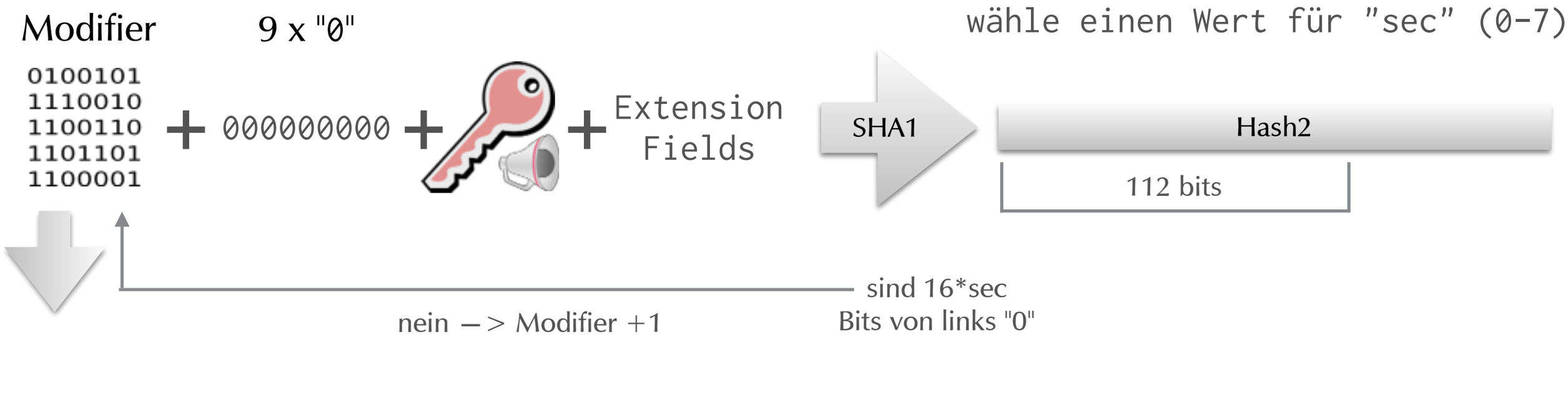
112 bits

sind 16*sec

nein –> Modifier +1

Bits von links "0"

CGA Adressen erzeugen



CGA Adressen erzeugen

Modifier

9 x "0"

wähle einen Wert für "sec" (0-7)

0100101
1110010
1100110
1101101
1100001

+ 000000000 +

+ Extension Fields

SHA1

Hash2

112 bits

sind 16*sec Bits von links "0"

nein —> Modifier +1

Modifier

0100101
1110010
1100110
1101101
1100001

+

CGA Adressen erzeugen

Modifier

0100101
1110010
1100110
1101101
1100001

+ 9 x "0" +  + Extension Fields

SHA1

Hash2

112 bits

wähle einen Wert für "sec" (0-7)

sind 16*sec Bits von links "0"

nein —> Modifier +1

Modifier

0100101
1110010
1100110
1101101
1100001

+ IPv6 Prefix + Collision Count +

CGA Adressen erzeugen

Modifier

9 x "0"

wähle einen Wert für "sec" (0-7)

0100101
1110010
1100110
1101101
1100001

+ 000000000 +

+ Extension Fields

SHA1

Hash2

112 bits

sind 16*sec Bits von links "0"

nein —> Modifier +1

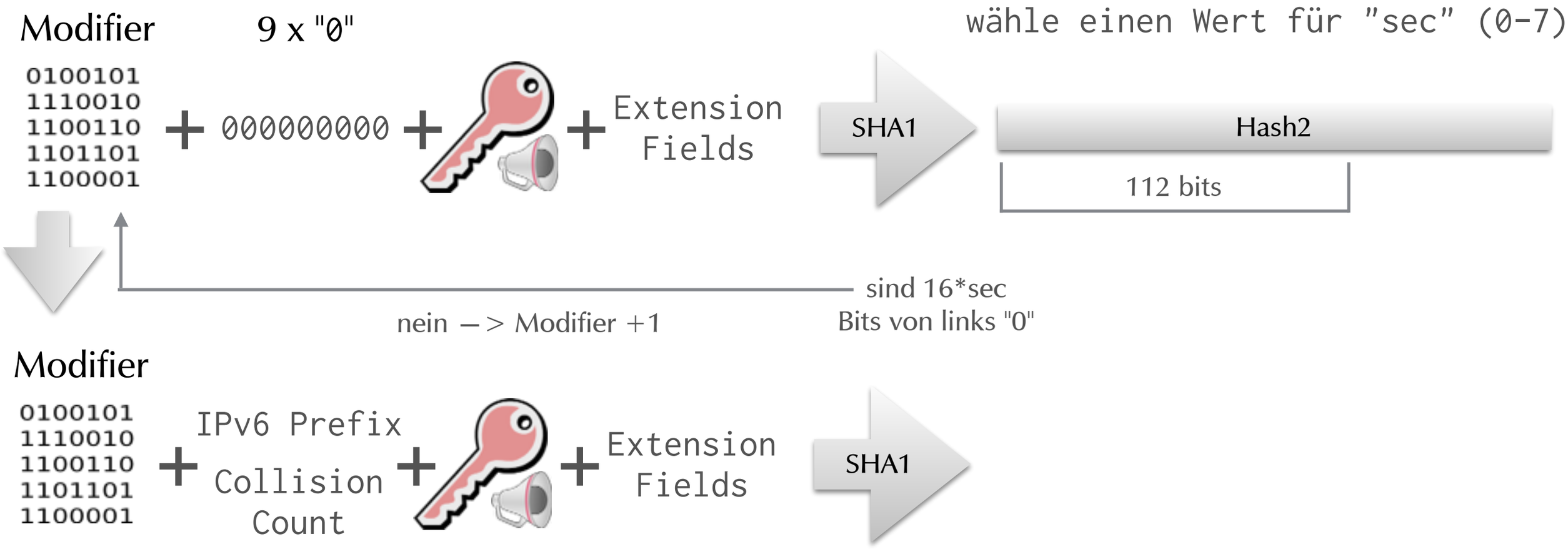
Modifier

0100101
1110010
1100110
1101101
1100001

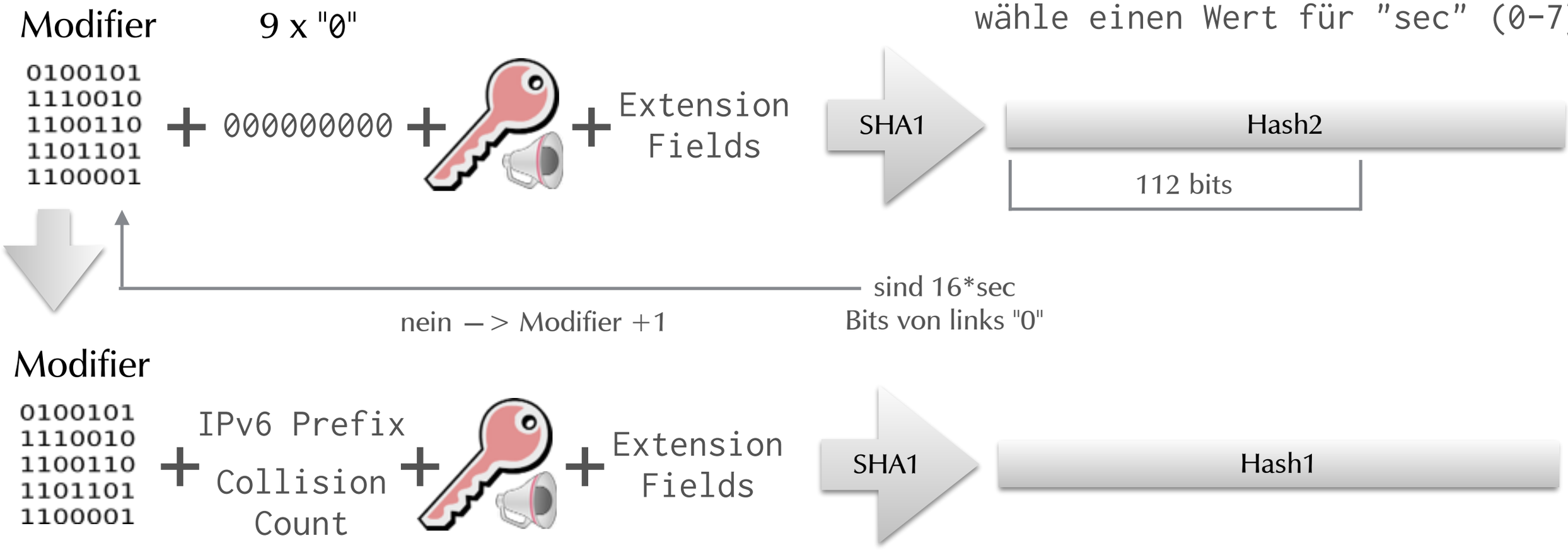
+ IPv6 Prefix + Collision Count +

+

CGA Adressen erzeugen



CGA Adressen erzeugen



CGA Adressen erzeugen

Modifier

0100101
1110010
1100110
1101101
1100001

+ 9 x "0" +

+ Extension Fields

wähle einen Wert für "sec" (0-7)

SHA1

Hash2

112 bits

sind 16*sec Bits von links "0"

nein —> Modifier +1

Modifier

0100101
1110010
1100110
1101101
1100001

+ IPv6 Prefix + Collision Count +

+ Extension Fields

SHA1

Hash1

64 bits

CGA Adressen erzeugen

Modifier

0100101
1110010
1100110
1101101
1100001

9 x "0"

000000000

Extension Fields

wähle einen Wert für "sec" (0-7)

SHA1

Hash2

112 bits

sind 16*sec Bits von links "0"

nein —> Modifier +1

Modifier

0100101
1110010
1100110
1101101
1100001

IPv6 Prefix

Collision Count

Extension Fields

SHA1

Hash1

64 bits

Wert von "sec" in Bits 0-3
"u" und "g" Bits (7,8) auf "0"

CGA Adressen erzeugen

Modifier

0100101
1110010
1100110
1101101
1100001

9 x "0"

000000000

Extension Fields

wähle einen Wert für "sec" (0-7)

SHA1

Hash2

112 bits

sind 16*sec Bits von links "0"

nein —> Modifier +1

Modifier

0100101
1110010
1100110
1101101
1100001

IPv6 Prefix

Collision Count

Extension Fields

SHA1

Hash1

64 bits

Wert von "sec" in Bits 0-3
"u" und "g" Bits (7,8) auf "0"

IPv6 Interface Identifier

CGA Adressen erzeugen

Modifier

0100101
1110010
1100110
1101101
1100001

9 x "0"

+ 000000000 +

+ Extension Fields

SHA1

Hash2

112 bits

wähle einen Wert für "sec" (0-7)

sind 16*sec Bits von links "0"

nein —> Modifier +1

Modifier

0100101
1110010
1100110
1101101
1100001

+ IPv6 Prefix + Collision Count +

+ Extension Fields

SHA1

Hash1

64 bits

Wert von "sec" in Bits 0-3
"u" und "g" Bits (7,8) auf "0"

IPv6 Interface Identifier

Duplicate Address Detection
Collision Count +1
(max 2)

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Router

Router Zertifikat

Client

Trust Anchor
für Router Zertifikat

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Router

Router Zertifikat

Privater RSA Schlüssel

Client

Certification Path Solicitation

+

Öffentlicher RSA Schlüssel

+

Nonce / Zeitstempel

+

Signatur

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Certification Path Solicitation

+

Öffentlicher RSA Schlüssel

+

Nonce / Zeitstempel

+

Signatur

Router

passt die Absende-IPv6 Adresse zum öffentlichen Schlüssel?

ist die Signatur über das Paket valide?

Client

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Certification Path Solicitation

+

Öffentlicher RSA Schlüssel

+

Nonce / Zeitstempel

+

Signatur

Router

passt die Absende-IPv6 Adresse zum öffentlichen Schlüssel?

✓

ist die Signatur über das Paket valide?

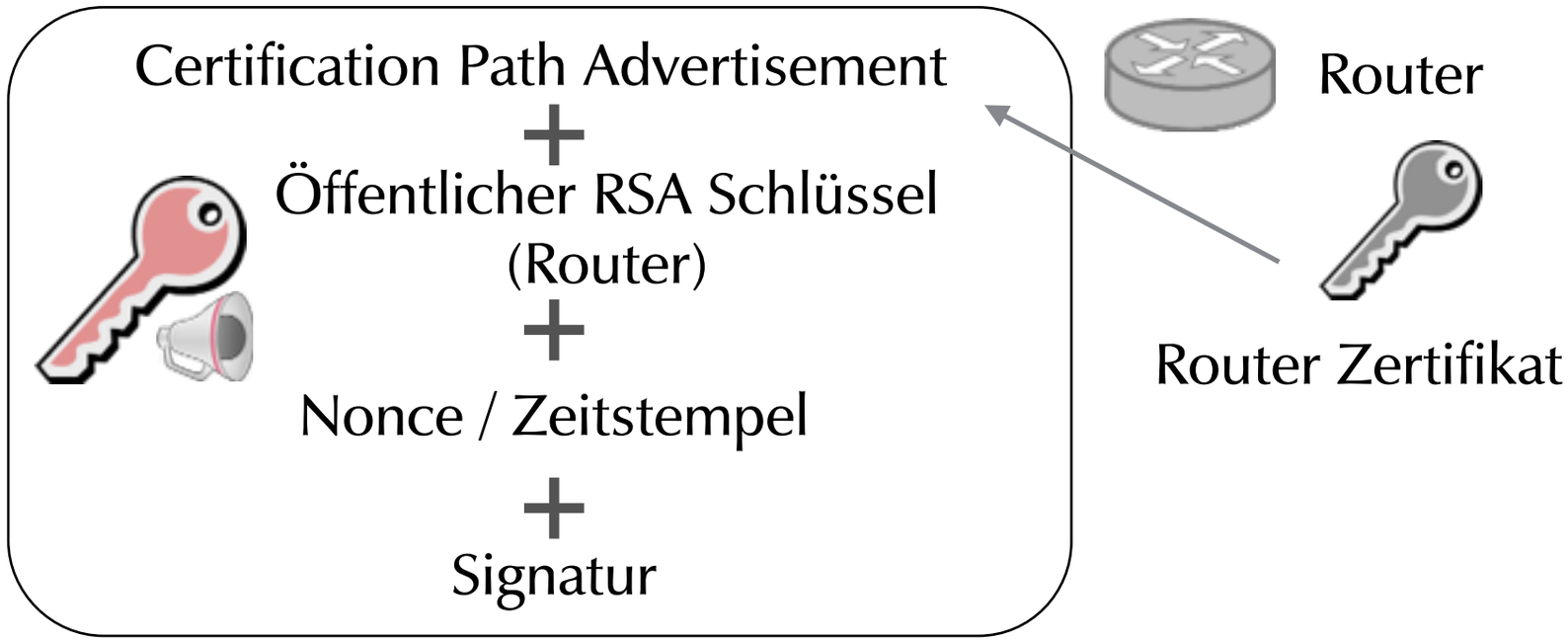
✓

Client

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208



Client

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Router

passt die Absende-IPv6 Adresse zum öffentlichen Schlüssel?

Certification Path Advertisem

+

Öffentlicher RSA Schlüssel (Router)

war die NONCE korrekt?

+

Nonce / Zeitstempel

+

Signatur

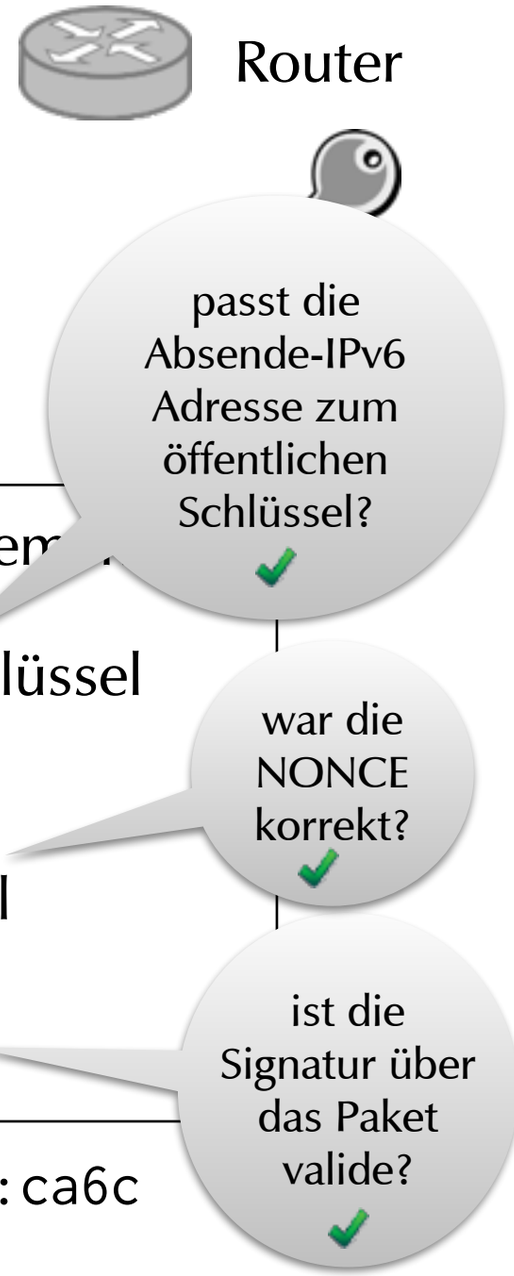
ist die Signatur über das Paket valide?

Client

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208



SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Router

Router Zertifikat

Passt das Router-Zertifikat zum konfiguriertem Trust-Anchor?

Certification Path Advertisement

+

Öffentlicher RSA Schlüssel (Router)

+

Nonce / Zeitstempel

+

Signatur

Client

Trust Anchor für Router Zertifikat

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

SeND Router Authentisierung

2001:db8:100:ae:34cc:6bba:5947:d208

Router

Router Zertifikat

Passt das Router-Zertifikat zum konfiguriertem Trust-Anchor?

Certification Path Advertisement

+

Öffentlicher RSA Schlüssel (Router)

+

Nonce / Zeitstempel

+

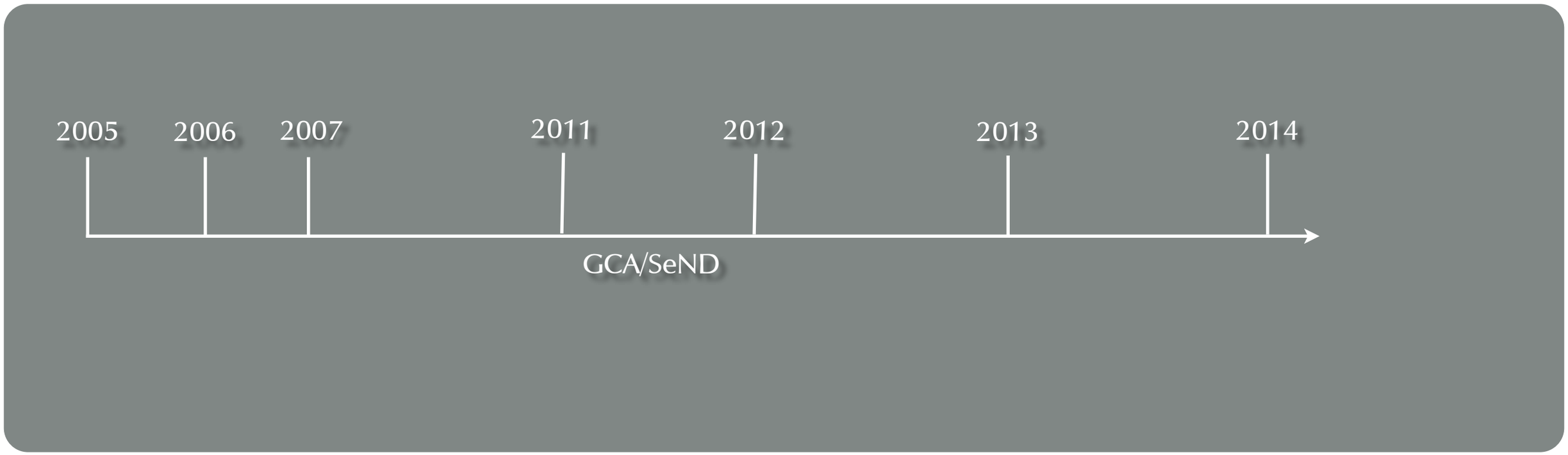
Signatur

Client

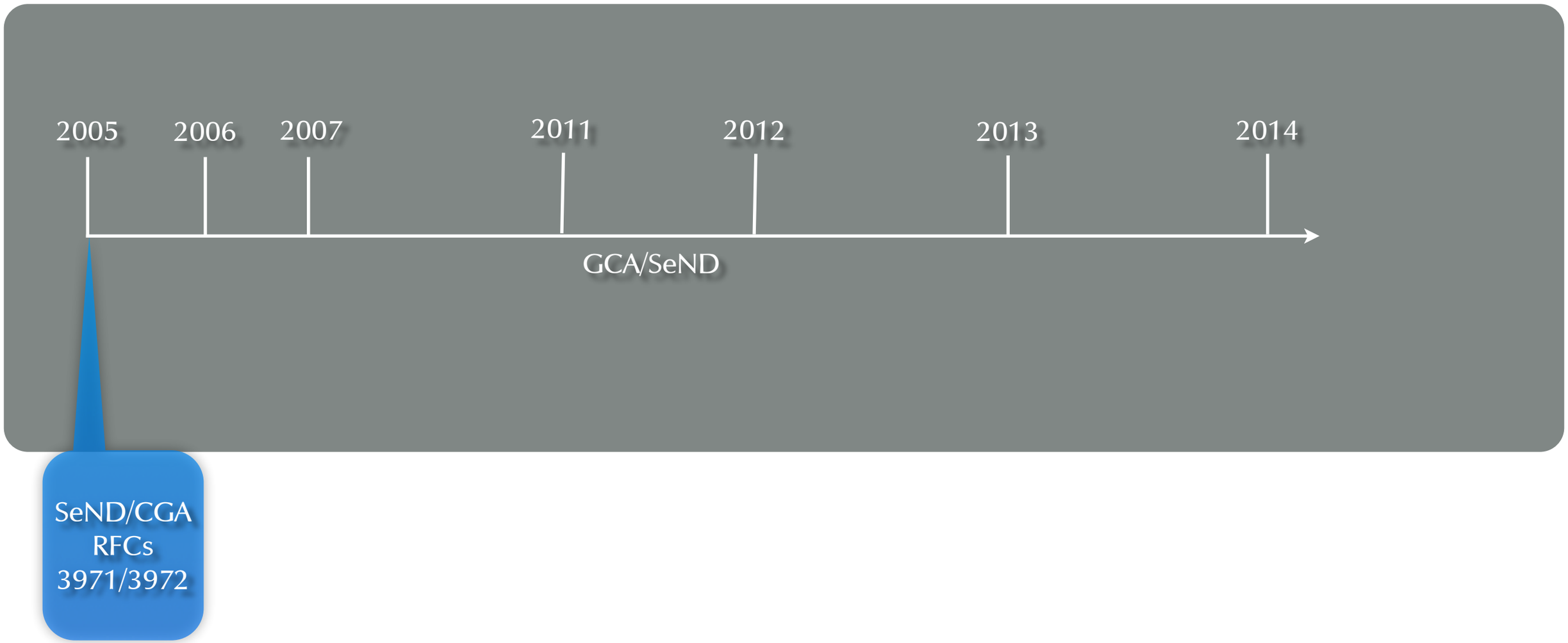
Trust Anchor für Router Zertifikat

2001:db8:100:ae:3c4a:5bf6:ffb4:ca6c

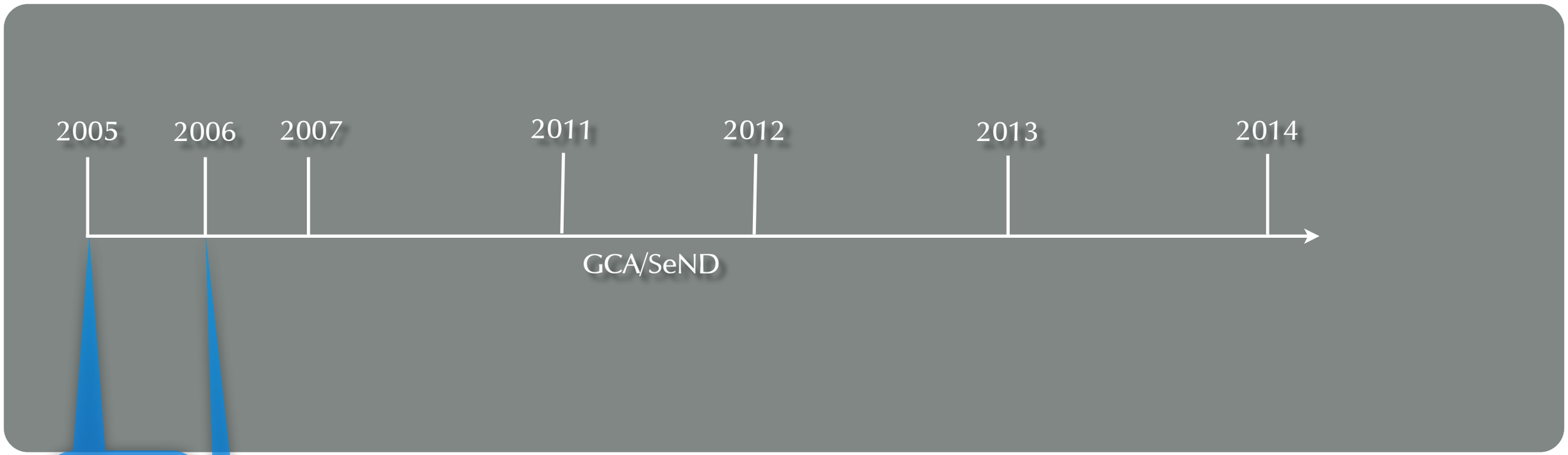
"Too old to rock'n'roll, to young to die?



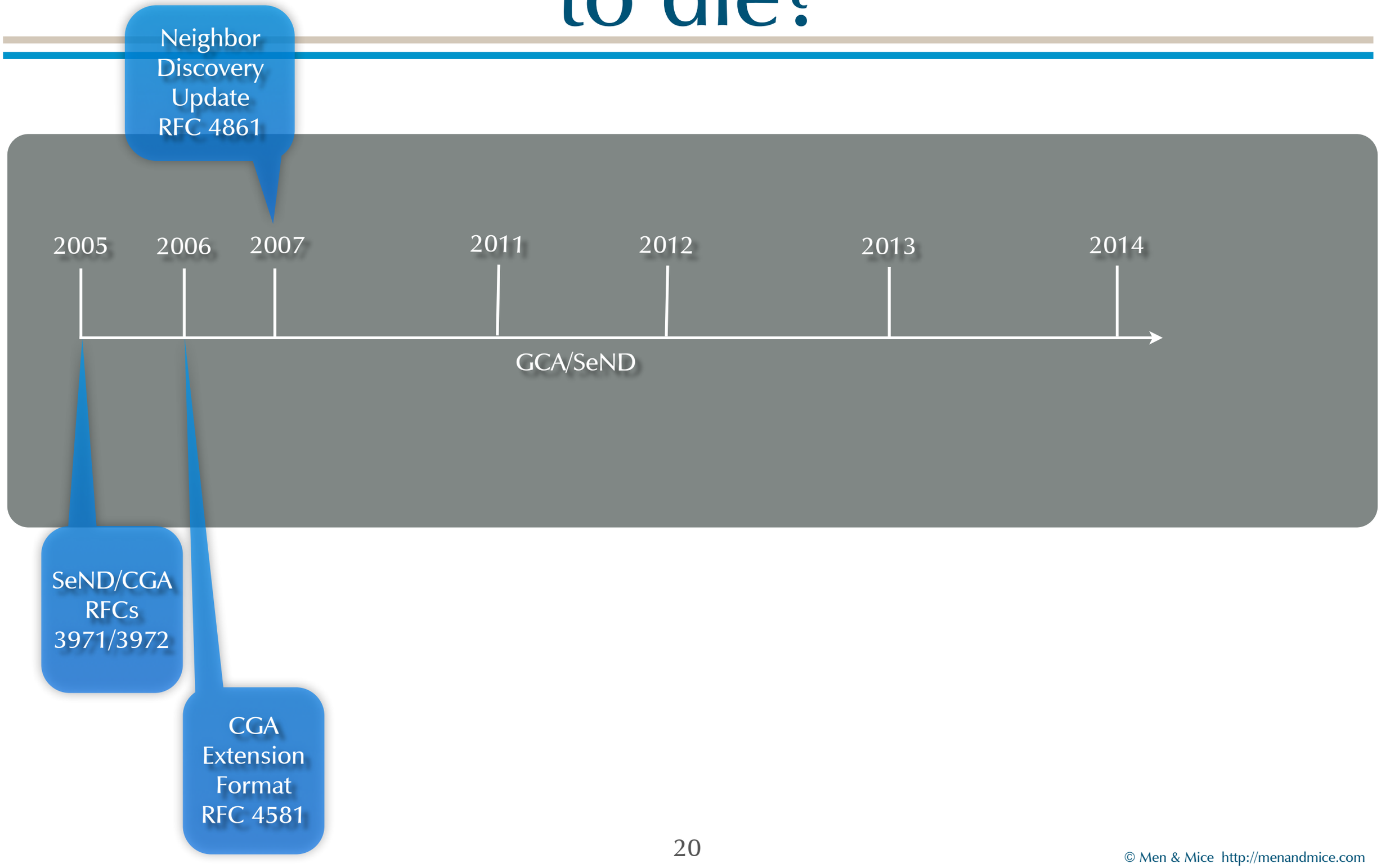
"Too old to rock'n'roll, to young to die?



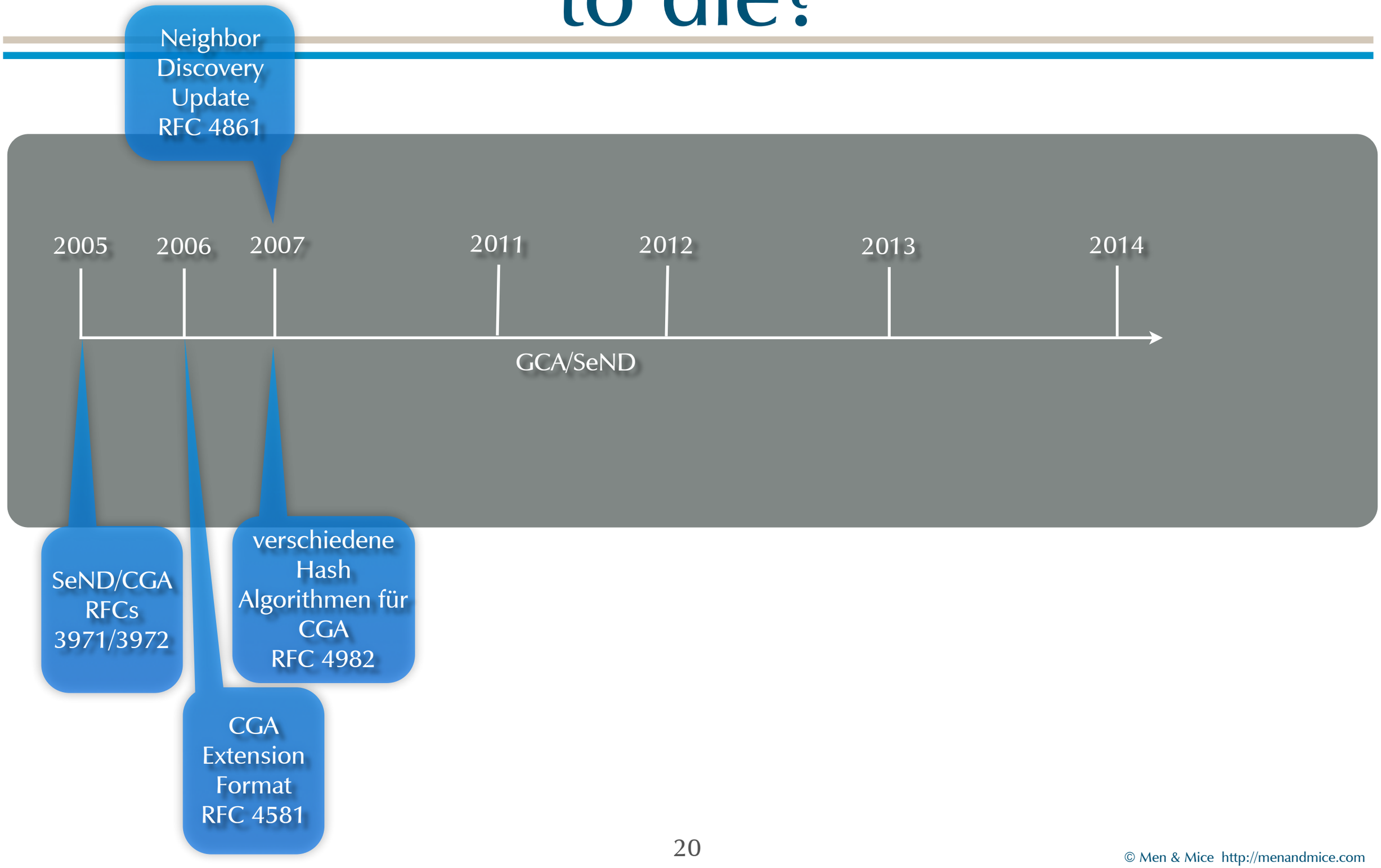
"Too old to rock'n'roll, to young to die?



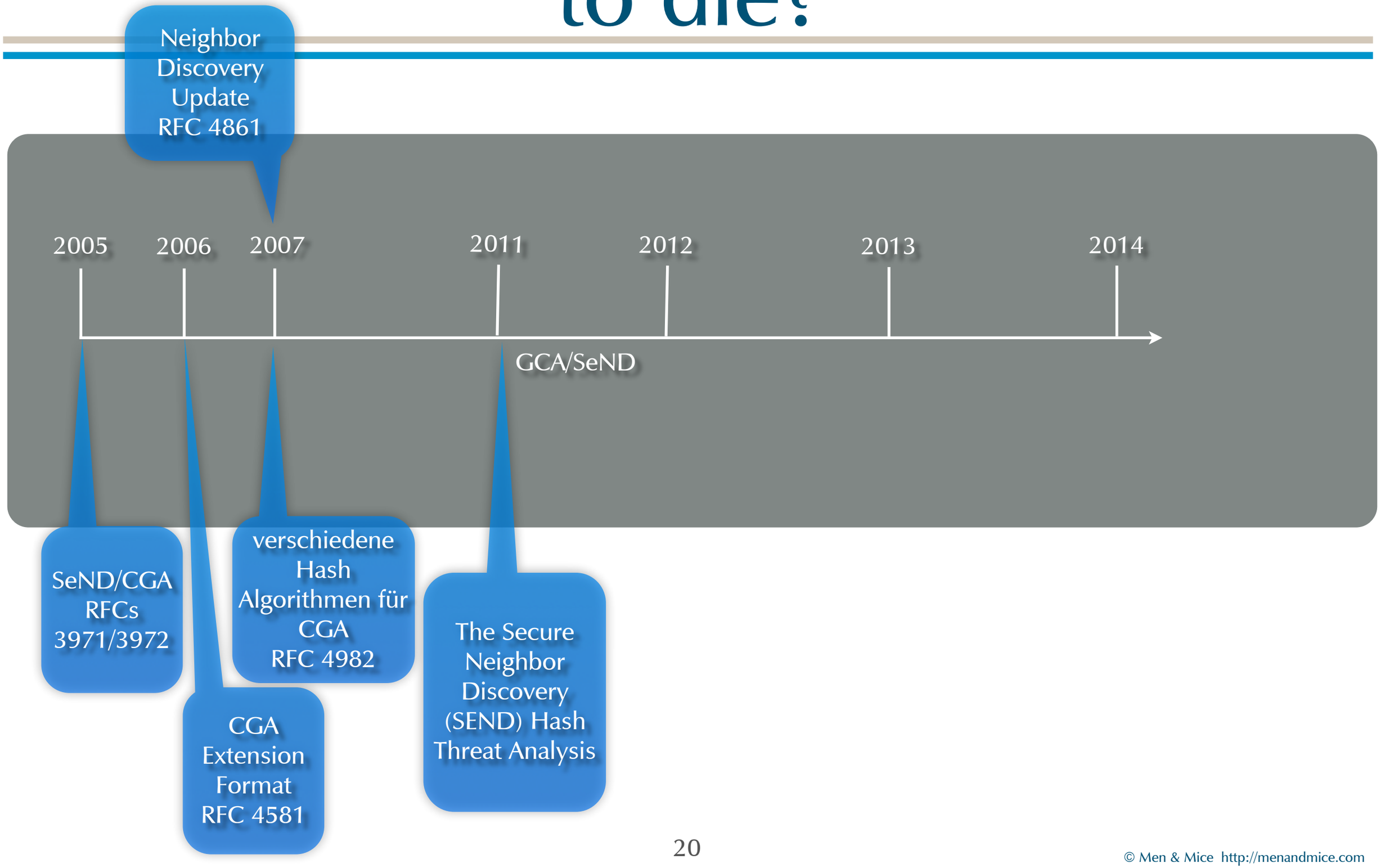
"Too old to rock'n'roll, to young to die?



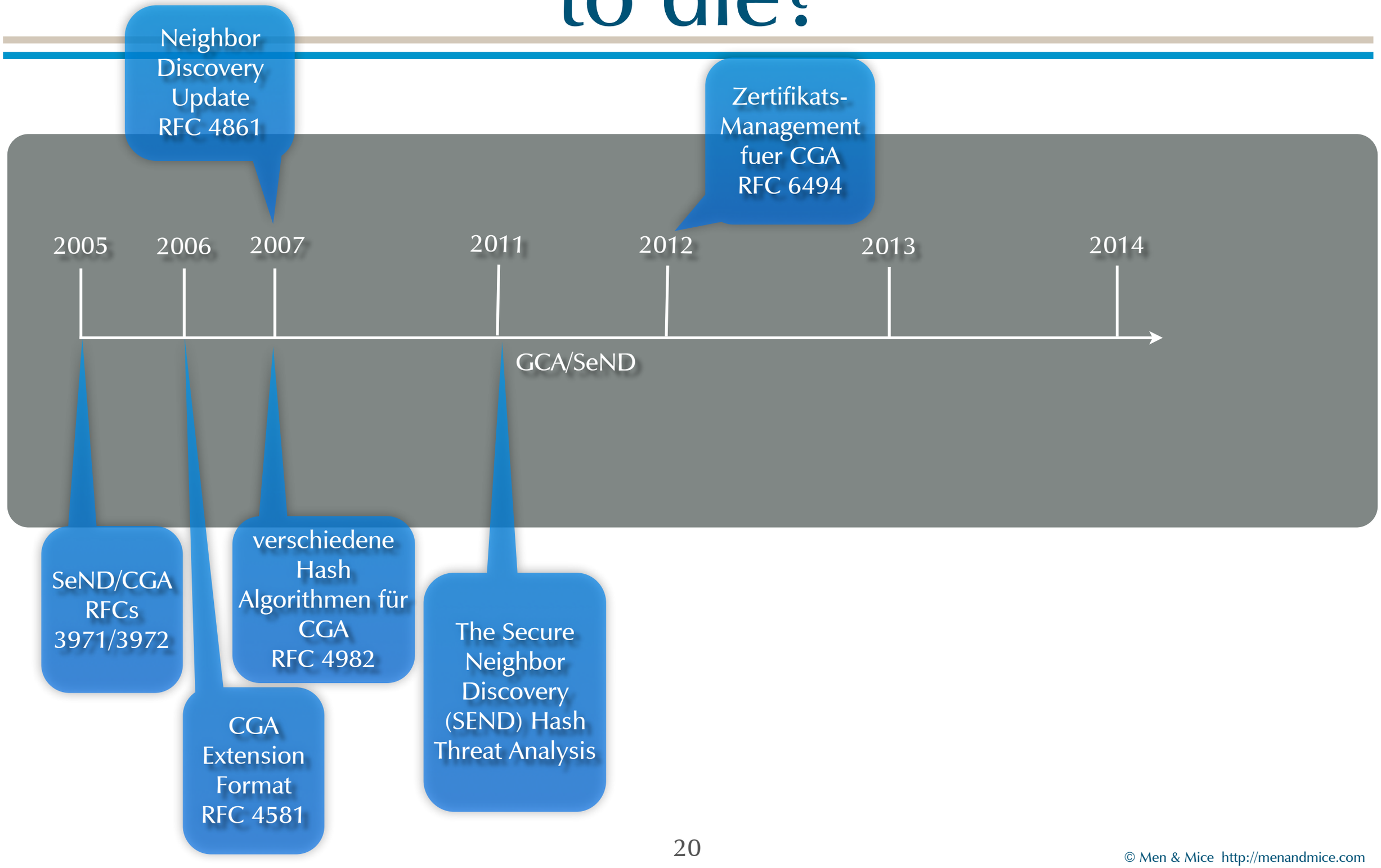
"Too old to rock'n'roll, to young to die?



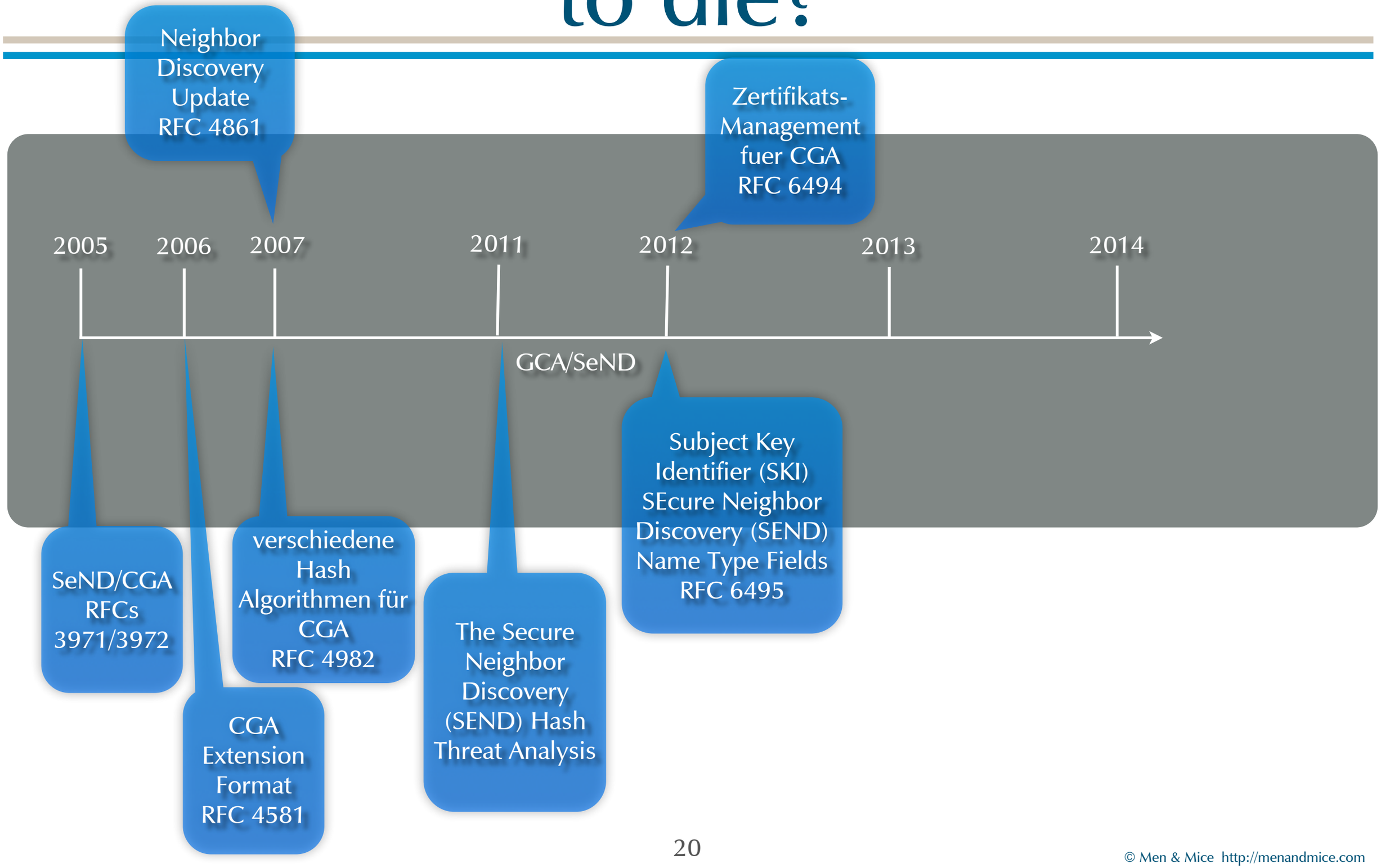
"Too old to rock'n'roll, to young to die?



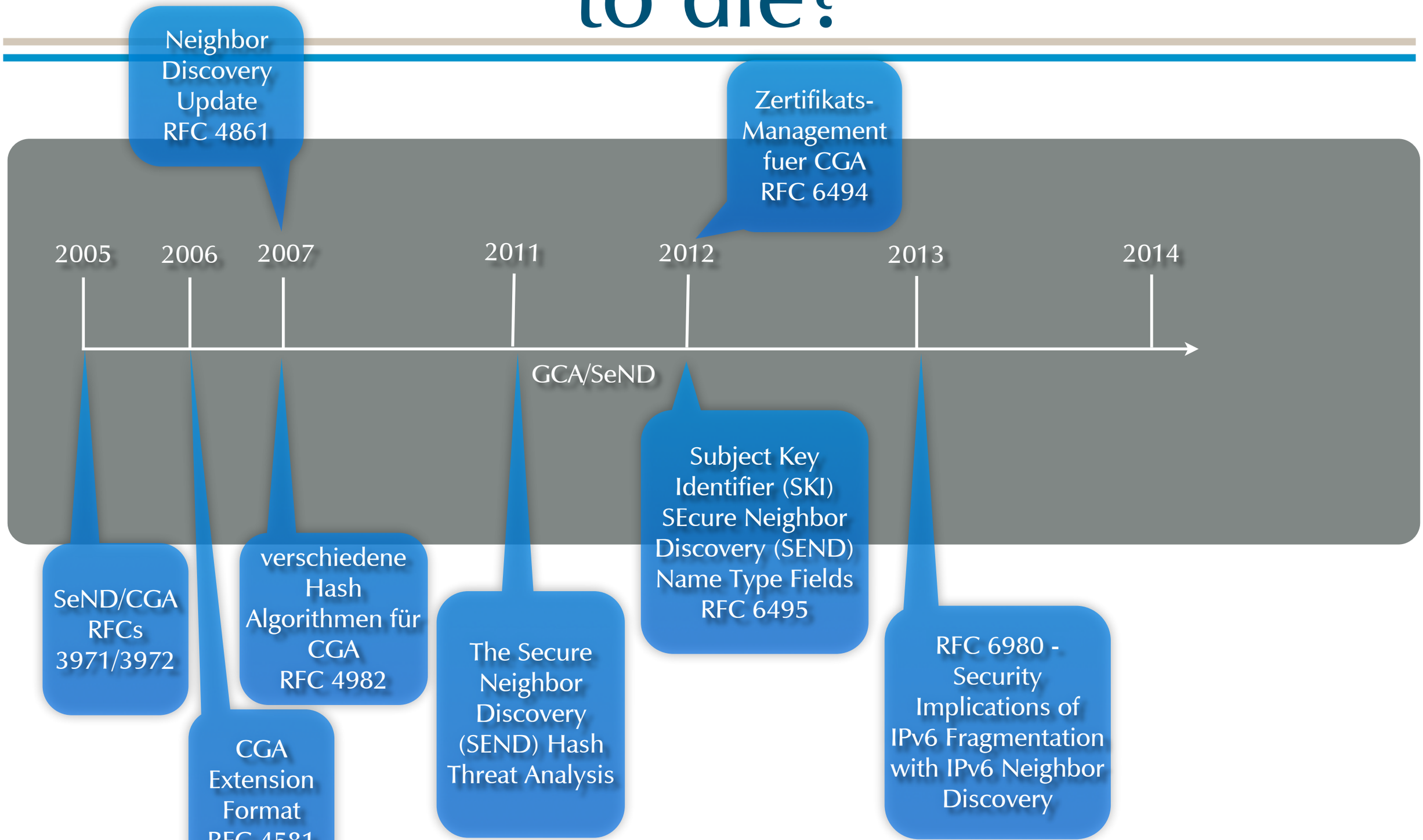
"Too old to rock'n'roll, to young to die?



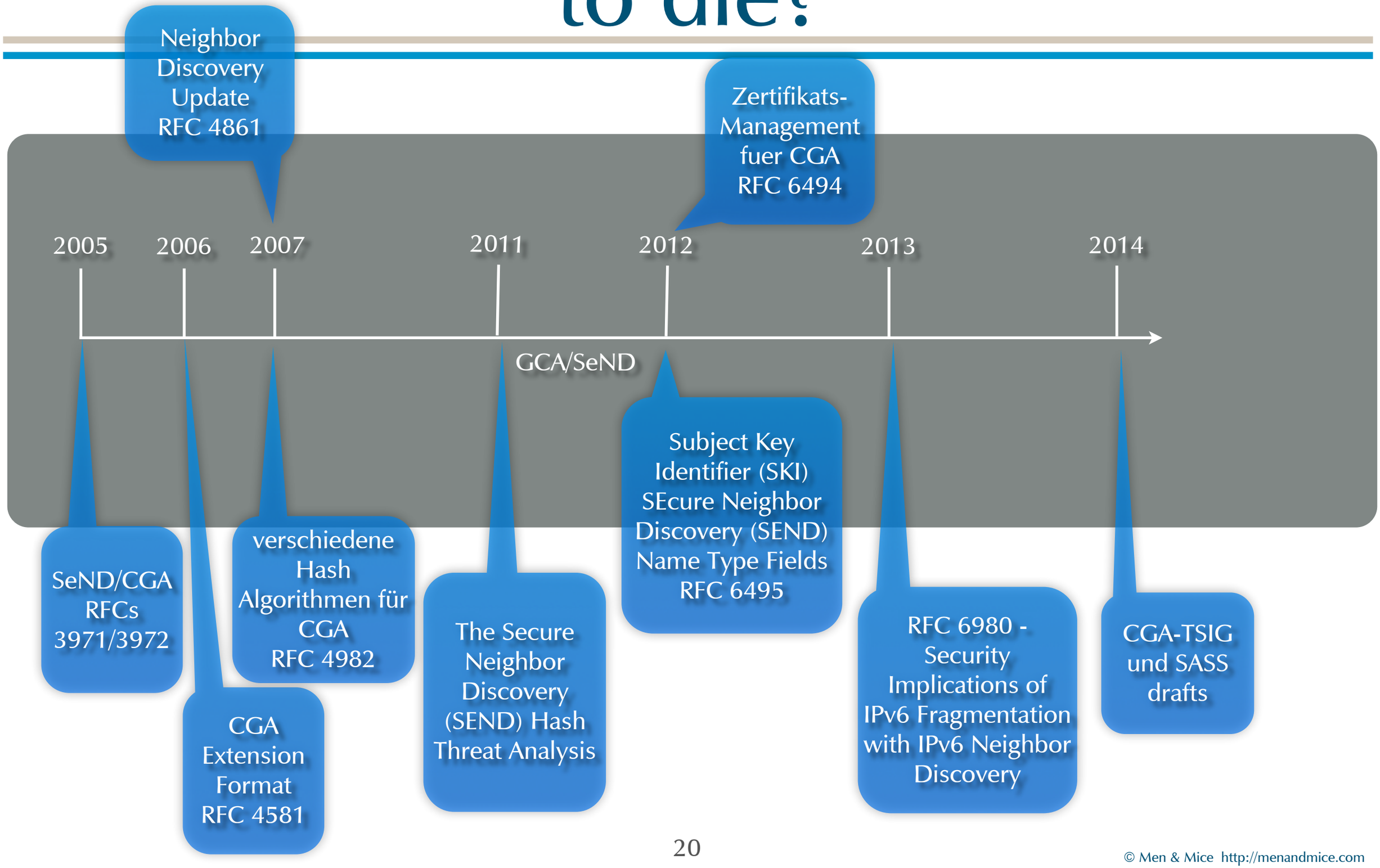
"Too old to rock'n'roll, to young to die?



"Too old to rock'n'roll, to young to die?



"Too old to rock'n'roll, to young to die?



CGA Einsatz in IP Protokollen

RFC 3971/3972 - SeND

RFC 5533 - SHIM6 (Level 3 Multihoming Shim Protocol for IPv6)

RFC 4866 - Enhanced Route Optimization for Mobile IPv6

CGA-TSIG

Internet Draft (Vorschlag) "TSIG using CGA" (April 28, 2014, H. Rafiee et al.)

sichert DNS Kommunikation mit asymmetrischen Schlüsseln

- Betriebssystem zum DNS Cache-Resolver (für DNS Anfragen)
- Betriebssystem zum autoritativen DNS Server (für dynamische DNS-Updates mit SLAAC Adressen)
- zwischen DNS Servern (für DNS Zonentransfer)

kann unabhängig von SeND eingesetzt werden

"Proof-of-Concept" Implementierung auf Basis von "ldns" von Marc Buijsman (mit NLnetLabs)

- "Securing the last mile of DNS with CGA-TSIG - NLnet Labs"

Alternativen zu GCA/SeND

RA-Guard (RFC 6105/7113)

verhindert gefälschte/un-authorisierte Router Advertisements

IEEE 802.X/802.1ae (MACsec)

Layer-2 Authentisierung von Netzwerkgeräten

A Simple Secure Addressing Scheme for IPv6 AutoConfiguration (SSAS)

erweitert SeND um ein neues Verfahren um IPv6 Adressen zu prüfen

Benutzt "Elliptic curve cryptography" (ECC) Schlüssel anstatt RSA (GCA)

Fazit

SeND Implementierungen sind nicht einsatzbereit...

- Windows (WinSEND) - nicht verfügbar
- FreeBSD - Implementierung im Kernel vorhanden, Userland kompiliert in FreeBSD 10 nicht mehr 😞
- Linux - Implementation unvollständig

... das scheint kein wirkliches Problem sein (mangelne Nachfrage)

kryptographisch generierte Adressen (CGA) werden unabhängig von SeND benutzt

Vielen Dank!

Fragen? Kommentare?

Links

Securing IPv6 Neighbor and Router Discovery (http://research.microsoft.com/apps/pubs/default.aspx?id=69145)

Cryptographically Generated Addresses (CGAs): A survey and an analysis of performance for use in mobile environment (http://paper.ijcsns.org/07_book/201102/20110204.pdf)

Easy-SEND: A Didactic Implementation of the Secure Neighbor Discovery Protocol for IPv6 (http://www.iaeng.org/publication/WCECS2009/WCECS2009_pp260-265.pdf)

JunOS - Secure Neighbor Discovery Feature Guide for Security Devices (http://www.juniper.net/techpubs/en_US/junos13.3/information-products/pathway-pages/config-guide-routing/config-guide-routing-secure-neighbor-discovery.pdf)

Cisco - IPv6 Secure Neighbor Discovery (http://www.cisco.com/c/en/us/td/docs/ios/ipv6/configuration/guide/12_4t/ipv6_12_4t_book/ip6-first_hop_security.html)

Native SeND kernel API for *BSD (http://people.freebsd.org/~anchie/SeND_AsiaBSDCon_2010.pdf)

WinSEND (SEcure Neighbor Discovery für Windows) - Software nicht veröffentlicht (http://www.hpi.uni-potsdam.de/meinel/security_tech/ipv6_security/winsend.html)

NIST - Guidelines for the Secure Deployment of IPv6 (http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf)

Towards Provable Secure Neighbor Discovery in Wireless Networks (http://people.kth.se/~papadim/publications/fulltext/provable-secure-neighbor-discovery-wireless.pdf)

Marc Buijsman - Securing the last mile of DNS with CGA-TSIG - NLnet Labs (https://www.nlnetlabs.nl/downloads/publications/report-rp2-buijsman.pdf)

RFCs and Drafts

CGA-EXT Mailing-Liste: http://www.ietf.org/mail-archive/web/cga-ext/current/maillist.html

RFC 3971 - SEcure Neighbor Discovery (https://tools.ietf.org/html/rfc3971)

RFC 3972 - Cryptographically Generated Addresses (https://tools.ietf.org/html/rfc3972)

RFC 4581 - Cryptographically Generated Addresses (CGA) Extension Field Format (https://tools.ietf.org/html/rfc4581)

RFC 4861 - Neighbor Discovery for IP version 6 (IPv6) (https://tools.ietf.org/html/rfc4861)

RFC 4866 - Enhanced Route Optimization for Mobile IPv6 (http://tools.ietf.org/html/rfc4866)

RFC 4982 - Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs) (https://tools.ietf.org/html/rfc4982)

RFC 6273 - The Secure Neighbor Discovery (SEND) Hash Threat Analysis (https://tools.ietf.org/html/rfc6273)

RFC 6494 - Certificate Profile and Certificate Management for SEcure Neighbor Discovery (SEND) (https://tools.ietf.org/html/rfc6494)

RFC 6495 - Subject Key Identifier (SKI) SEcure Neighbor Discovery (SEND) Name Type Fields (https://tools.ietf.org/html/rfc6495)

RFC 6980 - Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery (https://tools.ietf.org/html/rfc6980)

draft-cheneau-csi-ecc-sig-agility-02 (expired) - ECC public key and signature support in Cryptographically Generated Addresses (CGA) and in the Secure Neighbor Discovery (SEND) (https://tools.ietf.org/html/draft-cheneau-csi-ecc-sig-agility-02)

draft-ietf-csi-dhcpv6-cga-ps-09 (expired) - Analysis of Possible DHCPv6 and CGA Interactions (http://tools.ietf.org/html/draft-ietf-csi-dhcpv6-cga-ps-09)

draft-rafiee-6man-ssas (work in progress) - A Simple Secure Addressing Scheme for IPv6 AutoConfiguration (SSAS) (http://www.ietf.org/id/draft-rafiee-6man-ssas-08.txt)

draft-rafiee-intarea-cga-tsig (work in progress) - Secure DNS Authentication using CGA/SSAS Algorithm in IPv6 (https://tools.ietf.org/html/draft-rafiee-intarea-cga-tsig-07)