CGA as alternative security credentials with IKEv2: implementation and analysis

SAR-SSI 2012

Orange Labs

Jean-Michel Combes (France Telecom - Orange)

Aurélien Wailly (France Telecom - Orange)

Maryline Laurent (Telecom Sud Paris)

outline

- IPsec/IKEv2
- Authentication methods for IKEv2
- Cryptographically Generated Addresses (CGA)
- CGA as alternative method?
- Integration of CGA into IKEv2
- IKEv2 with CGA implementation
- Conclusion and future works

IPsec/IKEv2 (1/5)

- IPsec [RFC4301]
  - IP(v4/v6) security
  - *Authentication Header* (**AH**) for authentication
  - *Encapsulating Security Payload* (**ESP**) for authentication/encryption
  - 2 modes
    - Transport
    - Tunnel (e.g., "VPN" is ESP/Tunnel)

IPsec/IKEv2 (2/5)

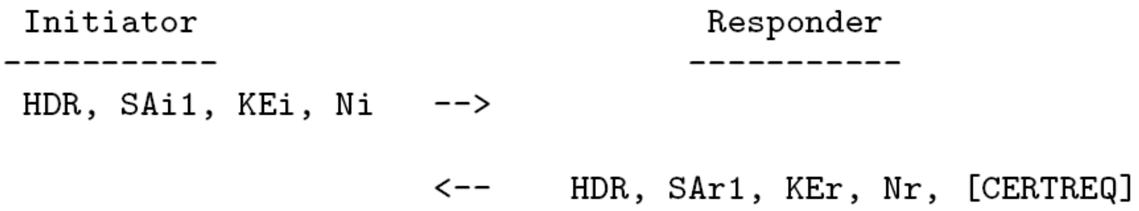
- 3 databases
  - Security Policy Database (**SPD**)
    - Allow/Discard/IPsec policy for a specific IP flow
  - Security Association Database (**SAD**)
    - Configuration (e.g., algorithm, key, etc.) of an IPsec connection, *IPsec Secure Association*, for a rule from the SPD
  - Peer Authorization Database (**PAD**)
    - Configuration of the security material used by an IPsec peer (i.e., ID, authentication method, security credentials)

IPsec/IKEv2 (3/5)

- Internet Key Exchange version 2 (IKEv2) [RFC5996]
  - To configure SAD dynamically
  - Use SPD and PAD
  - 4 types of exchange
    - IKE_SA_INIT
      - To set up IKE Secure Association
    - IKE_AUTH
      - To authenticate IPsec peers and set up initial IPsec Secure Association
    - CREATE_CHILD_SA
      - To create additional IPsec Secure Association
    - INFORMATIONAL
      - To inform about errors, etc.

IPsec/IKEv2 (4/5)

- IKE_SA_INIT
  - Diffie-Hellman key exchange (KEi, KEr)
  - IKEv2 Security Association (SA) negotiation (SAi1, SAr1)

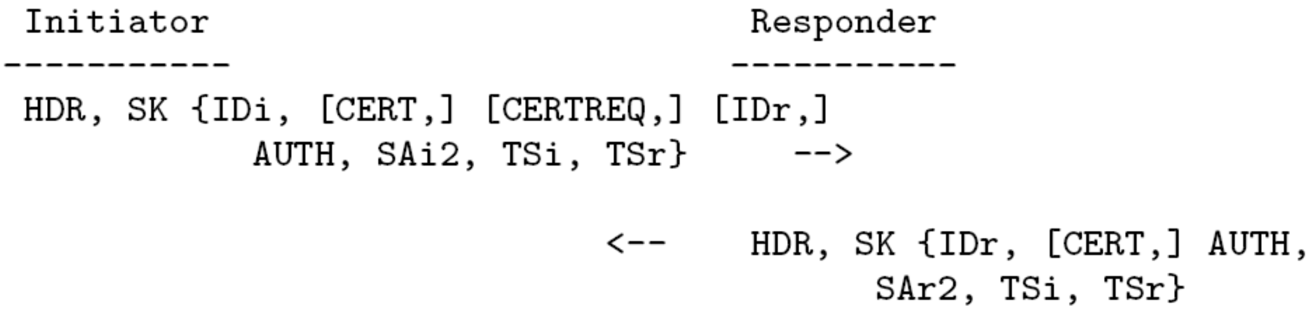


```
 Initiator                         Responder
-----------                       -----------
 HDR, SAi1, KEi, Ni   -->

                      <--    HDR, SAr1, KEr, Nr, [CERTREQ]
```

IPsec/IKEv2 (5/5)

- IKE_AUTH
  - Peers identification (IDi, IDr)
  - Peers' security material exchange (CERTREQ, CERT)
  - Peers authentication (AUTH)
  - IPsec SA negotiation (SAi2, SAr2, TSi, TSr)



```
 Initiator                          Responder
-----------                        -----------
 HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,]
            AUTH, SAi2, TSi, TSr}     -->

                               <--    HDR, SK {IDr, [CERT,] AUTH,
                                            SAr2, TSi, TSr}
```

Authentication methods for IKEv2 (1/2)

- Most common
  - pre-shared keys
    - complex provision
    - not scalable
  - X.509 certificates
    - require a *Public Key Infrastructure* (PKI)
      - associated costs
      - introduction of potential vulnerabilities
  - *Extensible Authentication Protocol* (EAP)
    - not mandatory

Authentication methods for IKEv2 (2/2)

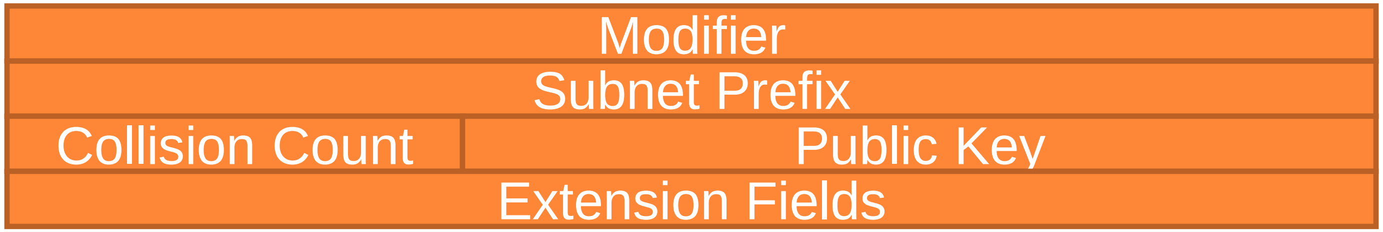
- Others (less known)
  - IPSEC_KEY RR [RFC4025]
    - Public key in the DNS
    - DNSSEC must be deployed
  - Better Than Nothing Security (BTNS) [RFC5386]
    - Assumption: no malicious node doing a MitM attack during IKE_SA_INIT exchange
    - So … no authentication needed.

Cryptographically Generated Addresses (1/3)

- Cryptographically Generated Addresses (CGA) [RFC3972]
  - IPv6 addresses resulting from the hash of parameters
  - Used with Secure Neighbor Discovery (SEND) [RFC3971]
    - Neighbor Discovery "equivalent" to ARP for IPv6
    - SEND, security for Neighbor Discovery

Cryptographically Generated Addresses (2/3)

- Generation
  - IPv6 address
    - Subnet Prefix (64 bits) || Interface ID (64 bits)
  - Public/private key pair
    - Algorithm: RSA
  - CGA Parameters



Modifier

Subnet Prefix

Collision Count | Public Key

Extension Fields

  - Interface ID = First64(Hash(CGA Parameters))
    - Algorithm: SHA-1

Cryptographically Generated Addresses (3/3)

- Verification
  - Step 1: regeneration of the CGA, based on received CGA Parameters
  - Step 2: validity of data signed with the CGA private key associated to the public one

CGA as alternative method? (1/3)

- Based on an academic paper [CMLN04] and an IETF draft [LMK07]
- Advantages
  - Equivalent security level to X.509 certificate
  - No need of a PKI
  - Self-generated by the owner
  - All the needed material to check a CGA sent directly to the receiver

CGA as alternative method? (2/3)

- Limitations
  - Identity
    - CGA, hard to remember for a human
    - Need to be associated to a Fully Qualified Domain Name (FQDN) stored in Domain Name Server (DNS)
  - "Hard-coded" cryptographic algorithms
    - SHA-1 mandatory
    - RSA (minimum key length is 384 bits)
  - No revocation

CGA as alternative method? (3/3)

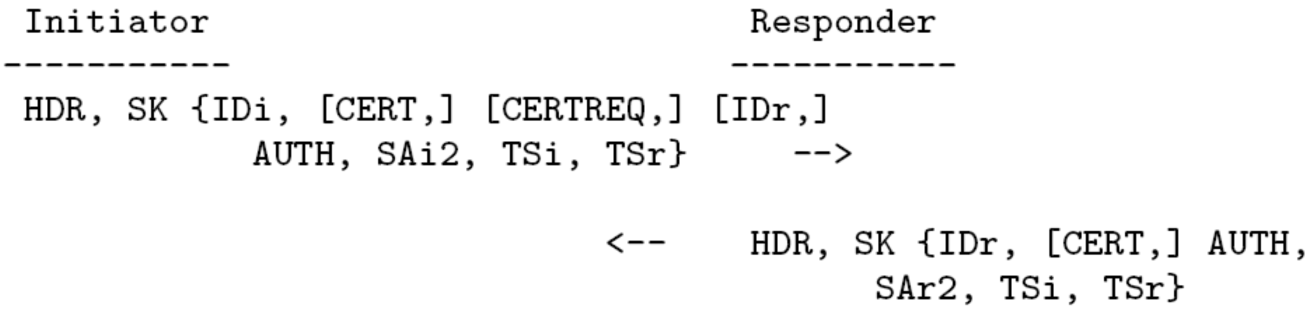
- To mitigate/solve the limitations
  - Identity: DNS use
    - To keep same security level
      - DNSSEC: FQDN <-> CGA
      - TSIG, SIG(0): for the CGA registration
  - "Hard-coded" cryptographic algorithms
    - SHA-1
      - Replaced by SHA-3 in CGA IETF RFC
    - RSA
      - Allow ECC use
  - No revocation
    - Potential solution based on Time To Live (TTL) field in DNS ressource records???

Integration of CGA into IKEv2 (1/4)

- IPsec
  - Peer Authorization Database (PAD)
    - Peer identity (ID_IPV6_ADDR) associated with CGA authentication method

Integration of CGA into IKEv2 (2/4)

- IKEv2
  - IDi, IDr
    - ID_IPV6_ADDR == CGA
  - CERT
    - New type: 222
    - Includes CGA parameters (self-signed certificate format)
  - CERTREQ
    - New type: 222
  - AUTH
    - Signature using the CGA's private key



```
 Initiator                              Responder
-----------                            -----------
 HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,]
             AUTH, SAi2, TSi, TSr}     -->

                                  <--    HDR, SK {IDr, [CERT,] AUTH,
                                               SAr2, TSi, TSr}
```

Integration of CGA into IKEv2 (3/4)

- AUTH validity
  - CGA ownership checking
    - Step 1: regeneration of the CGA, based on received CGA Parameters
    - Step 2: validity of data signed with the CGA private key associated to the public one

Integration of CGA into IKEv2 (4/4)

- Comparisons with other existing solutions
  - IETF draft [LMK07]
    - opportunistic encryption
    - no details about CGA use triggering
    - no details about CGA information exchanges
  - Microsoft
    - for IKEv2 (Windows 7 and Windows Server 2008 R2)
    - for IKEv1 only (other Windows OS)
    - Design choices

IKEv2 with CGA implementation (1/3)

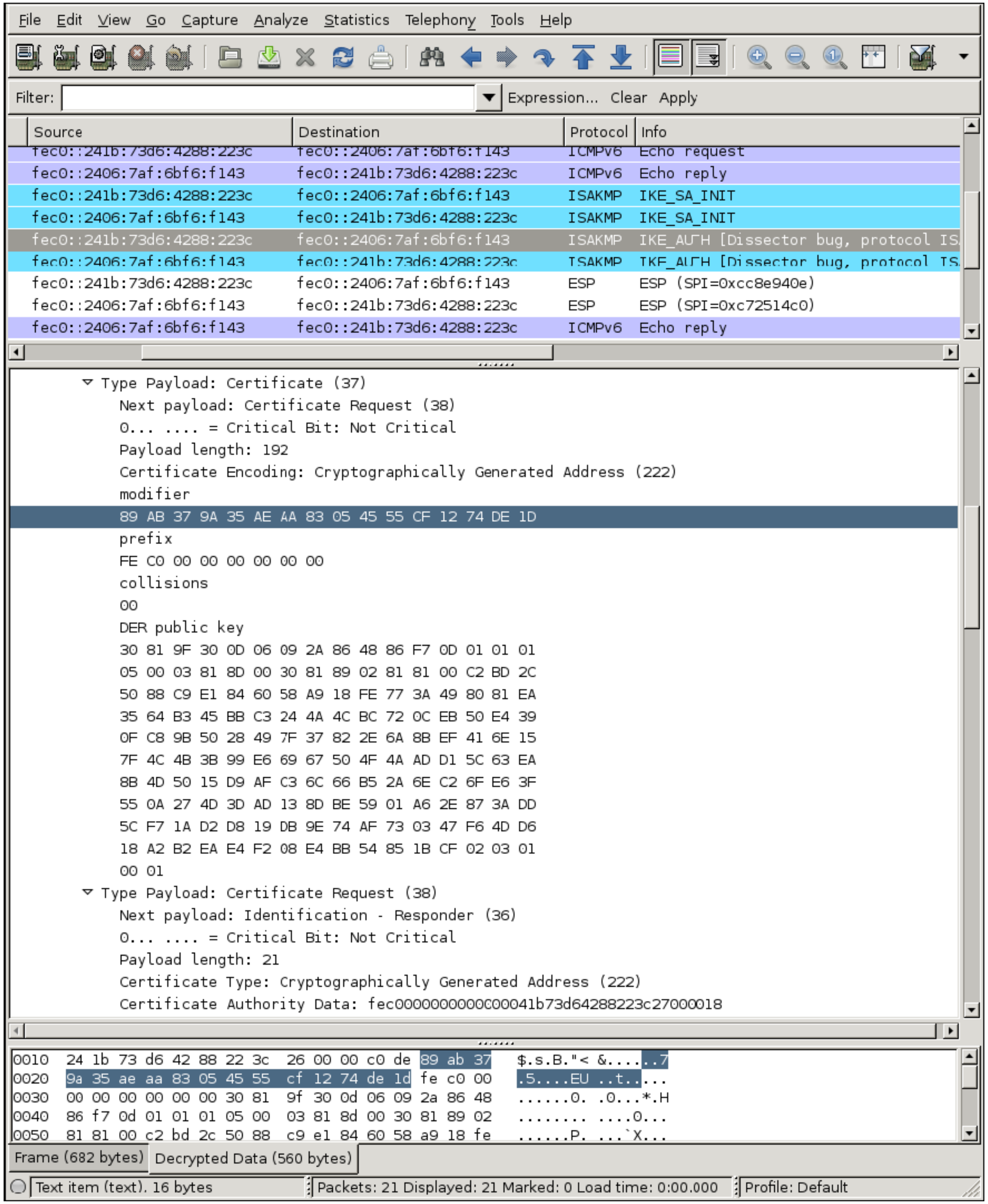
- Based on
  - StrongSwan
    - Linux IPsec/IKEv2 implementation
  - Docomo USA Labs
    - FreeBSD/Linux SEND/CGA implementation
- Debian

IKEv2 with CGA implementation (2/3)

- StrongSwan modifications
  - IPsec configuration file parser
  - IKEv2 payloads(ID, CERTREQ, CERT)
    - CERT: new plugin for StrongSwan
  - IKEv2 AUTH
  - IKEv2 State Machine (AUTH checking)
    - CGA ownership checking

IKEv2 with CGA implementation (3/3)

- Wireshark
  - Plugin to check the IKEv2+CGA exchanges


File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
Source                       Destination                  Protocol  Info
fec0::241b:73d6:4288:223c    fec0::2406:7af:6bf6:f143     ICMPv6    Echo request
fec0::2406:7af:6bf6:f143     fec0::241b:73d6:4288:223c    ICMPv6    Echo reply
fec0::241b:73d6:4288:223c    fec0::2406:7af:6bf6:f143     ISAKMP    IKE_SA_INIT
fec0::2406:7af:6bf6:f143     fec0::241b:73d6:4288:223c    ISAKMP    IKE_SA_INIT
fec0::241b:73d6:4288:223c    fec0::2406:7af:6bf6:f143     ISAKMP    IKE_AUTH [Dissector bug, protocol IS
fec0::2406:7af:6bf6:f143     fec0::241b:73d6:4288:223c    ISAKMP    IKE_AUTH [Dissector bug, protocol IS
fec0::241b:73d6:4288:223c    fec0::2406:7af:6bf6:f143     ESP       ESP (SPI=0xcc8e940e)
fec0::2406:7af:6bf6:f143     fec0::241b:73d6:4288:223c    ESP       ESP (SPI=0xc72514c0)
fec0::2406:7af:6bf6:f143     fec0::241b:73d6:4288:223c    ICMPv6    Echo reply
```

```
    ▽ Type Payload: Certificate (37)
        Next payload: Certificate Request (38)
        0... .... = Critical Bit: Not Critical
        Payload length: 192
        Certificate Encoding: Cryptographically Generated Address (222)
        modifier
        89 AB 37 9A 35 AE AA 83 05 45 55 CF 12 74 DE 1D
        prefix
        FE C0 00 00 00 00 00 00
        collisions
        00
        DER public key
        30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
        05 00 03 81 8D 00 30 81 89 02 81 81 00 C2 BD 2C
        50 88 C9 E1 84 60 58 A9 18 FE 77 3A 49 80 81 EA
        35 64 B3 45 BB C3 24 4A 4C BC 72 0C EB 50 E4 39
        0F C8 9B 50 28 49 7F 37 82 2E 6A 8B EF 41 6E 15
        7F 4C 4B 3B 99 E6 69 67 50 4F 4A AD D1 5C 63 EA
        8B 4D 50 15 D9 AF C3 6C 66 B5 2A 6E C2 6F E6 3F
        55 0A 27 4D 3D AD 13 8D BE 59 01 A6 2E 87 3A DD
        5C F7 1A D2 D8 19 DB 9E 74 AF 73 03 47 F6 4D D6
        18 A2 B2 EA E4 F2 08 E4 BB 54 85 1B CF 02 03 01
        00 01
    ▽ Type Payload: Certificate Request (38)
        Next payload: Identification - Responder (36)
        0... .... = Critical Bit: Not Critical
        Payload length: 21
        Certificate Type: Cryptographically Generated Address (222)
        Certificate Authority Data: fec0000000000c00041b73d64288223c27000018
```

```
0010  24 1b 73 d6 42 88 22 3c  26 00 00 c0 de 89 ab 37   $.s.B."< &.....7
0020  9a 35 ae aa 83 05 45 55  cf 12 74 de 1d fe c0 00   .5....EU ..t....
0030  00 00 00 00 00 00 30 81  9f 30 0d 06 09 2a 86 48   ......0. .0...*.H
0040  86 f7 0d 01 01 01 05 00  03 81 8d 00 30 81 89 02   ........ ....0...
0050  81 81 00 c2 bd 2c 50 88  c9 e1 84 60 58 a9 18 fe   ......P. ...`X...
```

Text item (text), 16 bytes | Packets: 21 Displayed: 21 Marked: 0 Load time: 0:00.000 | Profile: Default

Conclusion and future works

- IKEv2+CGA works
  - Implementation (PoC)
- CGA RFC needs modifications
  - SHA-3 and ECC integrations
- IKEv2+CGA with DNSSEC
  - Needs of more works on (i.e., a PoC)
- CGA revocation
  - Still an open issue …
- Performances

Questions?

Thanks!