A Signature Agility solution for CGA & SEND

draft-cheneau-csi-cga-pk-agility-00
draft-cheneau-csi-send-sig-agility-00
draft-cheneau-csi-ecc-sig-agility-00

76th IETF

CGA & SEND maintenance WG

Tony Cheneau
email: tony.cheneau@it-sudparis.eu
Michaela Vanderveen
Maryline Laurent
Sean Shen
Overview

- Advantages of ECC/ECDSA
- Recommendations for hosts and routers
- Changes since the last version
  - New options
  - Negotiation process
- Authorization Delegation Discovery
Advantages of ECC/ECDSA

- Faster Public Key and signature generation
- Shorter Public Key size:
  - A 1024 bits long, DER encoded, RSA Public Key's size is 160 octets
  - Equivalent, DER encoded, ECC P-256 curve's size is ~88 octets
- Shorter Signature size:
  - PKCS#1 v1.5 signature when using 1024 bits long RSA Public Key is 128 octets
  - ECDSA signature is ~71 octets long
Recommendations for hosts

- Ability to generate CGA based on Multiple Keys (as a transitional mechanism, not mandatory)
- Ability to verify additional signature types besides the one used to generate a signature (not mandatory)
- Ability to communicate with RFC 3971 nodes (not mandatory)
Recommendations for routers

- Routers are likely to verify more Signature Algorithm types (even when they don't use M-CGA)
- When using one or many M-CGA, router can add multiple “Signature” options (for each Public Key available) to ND messages
- Routers should have their certificate(s) signed with the same algorithm they use to generate their (M-)CGA
- A single router may have multiple CGA of different types. It implies that the router will likely have multiple Certification Path (one for each addresses)
Changes since last version

- draft-cheneau-cga-pk-agility-01 → draft-cheneau-csi-cga-pk-agility-00
  - text clarification
- draft-cheneau-send-sig-agility-01 → draft-cheneau-csi-send-sig-agility-00
  - removes ECC/ECDSA related stuff
  - removes « resend » ('r') flag
  - removes « router as a notary » functionality
  - updated options
- draft-cheneau-csi-ecc-sig-agility-00
  - provides a basic skeleton for other signature algorithms
Multiple Key CGA (1 of 2)

CGA Parameters Data Structure's Public Key extension field:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
|          Extension Type          |    Extension Length     |
+-----------------------------------------------+
|                                                                                      |
~                       Public Key                              ~                     
|                                                                                      |
+-----------------------------------------------+
```
Multiple Key CGA generation process:

```
+-------------------------+   +-------------------------+   +-------------------------+
|                         |   |                         |   |                         |
|  Modifier                |   |  Subnet Prefix          |   |  Extension Fields       |
|                         |   | (variable length)      |   | (optional, variable length) |
|                         |   |                         |   |                         |
|  Col Count              |   |                         |   |                         |
|                         |   |                         |   |                         |
|  Public Key 1           |   |  Public Key 2          |   |                         |
| ~                       |   | (variable length)      |   |                         |
|                         |   | Extension              |   |                         |
|  Public Key 2           |   | Public Key N          |   |                         |
| ~                       |   | (variable length)      |   |                         |
|  Public Key N           |   | ...                    |   |                         |
```

Multiple Key CGA (2 of 2)
Supported Signature Algorithms Option (SSA)
Signature Algorithm field (detailed)

- Signature generation bit:
  - Value 0: can only verify this signature algorithm
  - Value 1: can sign and verify with this signature algorithm

- Signature Type Identifier subfield:
  - Value 0: RSA/SHA-1
  - Value 1: RSA/SHA-256
  - Value 9: ECDSA (P-256)/SHA-256
  - ...

(ECC values follows registry IKEv2 authentication method)
Universal Signature Option (USO)

Backward compatible with the RSA Signature Option when Key Pos=0 and Sig Id=0 (i.e. when using a single Public Key and RSA/SHA-1 signature algorithm)
Negotiation phase with legacy RFC 3971 nodes

Node A (RSA, ECC)

NS {SSA: s/v RSA, s/v ECC; signed with ECC}

...timeout... (fallback)

...process the NA...

Node B (RFC 3971)

NS {SSA: s/v RSA, s/v ECC; signed with RSA}

NA {signed with RSA}

...ignore SSA opt... ...ignore specificity of USO...
Negotiation phase with compatible nodes

Node A (RSA, ECC)

Node B (RSA)

Unidirectional IPv6 traffic

NS {SSA: s/v RSA, s/v ECC; signed with ECC}

NA {SSA: s/v RSA; signed with RSA}

Unidirectional IPv6 traffic

NS {SSA: s/v RSA; signed with RSA}

NA {SSA: s/v RSA, s/v ECC; signed with RSA}

Bidirectional IPv6 traffic

NA {SSA: s/v RSA, s/v ECC; signed with RSA}

...process the NA...
...record the L2 address...

...can't check ECC signature, does not record L2 address...
...however SSA indicates RSA...

...missing L2 address...

11/9/2009
Negotiation phase with incompatible nodes

Depending on local policies, node A and B could register unsecured L2 addresses
Authorization Delegation Discovery

• Is this out of the scope of our work?
• Should we state requirements such as:
  ◦ « the Certification Path SHOULD only contain certificates containing verifiable content for the requesting node based on its Supported Signature Option »
  ◦ « if the router as a CGA based on algorithm XXX, then all the Certification Path certificates must use the same XXX algorithm »
Thanks for listening

draft-cheneau-csi-cga-pk-agility-00
draft-cheneau-csi-send-sig-agility-00
draft-cheneau-csi-ecc-sig-agility-00

Questions ? Thoughts ? Volunteering for reviewing ?