Putting ACP domain information in LDevID

Benjamin Kaduk

20 November 2019

Table of Contents

Success condition for today

I'll try to describe my understanding of what the pieces of the domain information do, and how the different aspects of it fit into the PKIX/X.509 structure. (Note: **my** understanding; probably wrong, and don't be shy about jumping up to the mic!)
I'd like to end up with something that meets the ANIMA requirements without diverging from the expected PKIX usage patterns, or diverging only in the smallest sense needed.

Domain Information

Pulling the ABNF from -21:

```
domain-information = local-part "@" acp-domain-name
local-part = key [ "." local-info ]
key = "rfcSELF"
local-info = [ acp-address ] [ "+" rsub extensions ]
acp-address = 32HEXDIG | 0 ; HEXDIG as of RFC5234 section B.1
rsub = [ <subdomain> ] ; <subdomain> as of RFC1034, section 3.5
acp-domain-name = ; <domain> ; as of RFC 1034, section 3.5
extensions = *( "+" extension )
extension = ; future standard definition.
            ; Must fit RFC5322 simple dot-atom format.

routing-subdomain = [ rsub " ." ] acp-domain-name
```

Quick Tangent

An ABNF nit:

```
domain-information = local-part "@" acp-domain-name
local-part = key [ "." local-info ]
key = "rfcSELF"
local-info = [ acp-address ] [ "+" rsub extensions ]
acp-address = 32HEXDIG | 0 ; HEXDIG as of RFC5234 section B.1
rsub = [ <subdomain> ] ; <subdomain> as of RFC1034, section 3.5
acp-domain-name = ; <domain> ; as of RFC 1034, section 3.5
extensions = *( "+" extension )

extension = dot-atom; future standard definition.

            ; Must fit RFC5322 simple dot-atom format.

routing-subdomain = [ rsub " ." ] acp-domain-name
```

Quick Tangent

An ABNF nit:

```
domain-information = local-part "@" acp-domain-name
local-part = key [ "." local-info ]
key = "rfcSELF"
local-info = [ acp-address ] [ "+" rsub extensions ]
acp-address = 32HEXDIG | 0 ; HEXDIG as of RFC5234 section B.1
rsub = [ <subdomain> ] ; <subdomain> as of RFC1034, section 3.5
acp-domain-name = ; <domain> ; as of RFC 1034, section 3.5
extensions = *( "+" extension )

extension = dot-atom; future standard definition.

            ; Must fit RFC5322 simple dot-atom format.

routing-subdomain = [ rsub " ." ] acp-domain-name
```

Note that dot-atom allows for "+", which is our extension marker...

local-part.key

Drilling down...

```
domain-information = local-part "@" acp-domain-name
```

```
local-part = key [ "." local-info ]
key = "rfcSELF"
```

This “`rfcSELF`” acts as a type marker.

local-info.acp-address

```
local-part = key [ "." local-info ]
local-info = [ acp-address ] [ "+" rsub extensions ]
acp-address = 32HEXDIG | 0 ; HEXDIG as of RFC5234 section B.1
```

This looks like an IP address — tells the private-key-holder its own address, and is used to certify to other domain members that this node has been assigned this IP address.
We have some cases where there is no IP address or an empty one (these currently have different semantics).

local-info.rsub

```
domain-information = local-part "@" acp-domain-name
local-info = [ acp-address ] [ "+" rsub extensions ]
rsub = [ <subdomain> ] ; <subdomain> as of RFC1034, section 3.5
routing-subdomain = [ rsub " ." ] acp-domain-name
```

`rsub` is a subdomain under the full ACP domain to which this node belongs. The routing-subdomain value is used as an input to the hierarchical addressing mode (its hash is 40 bits of address prefix)...but we can't assume that at any point other than address-assignment time; it's a "heuristic". Being in the routing subdomain implies some level of locality of routing (though RPL isn't sensitive to the locality anyway and doesn't use the /48 hierarchy), but ACP nodes still set up RPL adjacencies to other routing subdomains (when such nodes are adjacent). ~~`rsub` might be related to Intent at some point.~~ The ACP is mostly indifferent to how network topology relates to "rsub" topology.
What uses `rsub` other than the registrar/etc. assigning addresses?

local-info.extensions

```
domain-information = local-part "@" acp-domain-name
local-part = key [ "." local-info ]
extensions = *( "+" extension )
extension = dot-atom ; future standard definition.
            ; Must fit RFC5322 simple dot-atom format.
```

Are any potential extensions known? Arbitrary extension points seem rather incompatible with being a name.

acp-domain-name

```
domain-information = local-part "@" acp-domain-name
acp-domain-name = ; <domain> ; as of RFC 1034, section 3.5
```

This is pretty key. The ACP domain name is used to determine domain membership, which in turn is a key part of authorization policy.
That said, it seems to only be used for determining "is the remote peer using the same value as me?"

Information content of rfc822Name/domain information

Tabulating (omitting “syntactic sugar”/framing):

- rfcSELF: type marker: “this is an ACP domain-information address”
- acp-address: assigned IP address, or ¡all-zeros¿ (address assigned externally), or ¡absent¿ (ACP connect)
- rsub: used as input to (hash function for) IP prefix allocation. No other impact on ACP node behavior?
- extensions: No current use planned? Very hard to justify as a “name”
- acp-domain-name: Name for the ACP to which a node belongs. Used for authorizing peers as being in the same domain.

Classifying needed information into an X.509 vocabulary

- type marker: dedicated X.509 extension, can be “critical”, so this cert is discarded by any peer that doesn’t know about ACP. Doesn’t have to be critical, to allow usage for other purposes, which we have some use cases for
- IP address: a node’s identity, and an attribute of the peer that can be extracted from the connection and authenticated by the certificate
- routing subdomain: ???
- extensions: X.509 has built-in extensibility
- domain name: not always needed to identify the thing being certified (e.g., for rfc822Name the domain part is needed in order to scope the local-part). Probably can be an attribute of the extension used as a type marker.