

Discovery for BRSKI variations

IETF117 ANIMA WG July 2023

Toerless Eckert (Futurewei USA), tte@cs.fau.de

Discovery for

- rfc8995
- draft-ietf-anima-brski-prm (draft-ietf-anima-jws-voucher)
- draft-ietf-anima-brski-ae
- draft-ietf-anima
- draft-ietf-anima-constrained-voucher (constrained BRSKI)
- draft-ietf-anima-constrained-join-proxy
- (draft-ietf-brski-cloud)

Initial problem

- Network has multiple registrars
 - They support different (subsets) of BRSKI variations
- How would a pledge get to the Registrar supporting what the Pledge supports ?
- Some distinction based on transport protocol used:
 - TCP = TLS/HTTPs vs. UDP = CoAP/DTLS
 - But not all aspects of BRSKI variation resolved by what we have.

Registering BRKI variation

Data model for variations

Parameter	Variation	Reference	Type	Notes
mode	rrm	[RFC8995]	Default	Registrar Responder Mode
	prm	[BRSKI-PRM]	Reserved	Pledge Responder Mode
enroll	est	[RFC7030] [RFC8995]	Default	Enrollment over Secure Transport
	cmp	[BRSKI-AE]		Lightweight profile of “Certificate Management Protocol” draft-ietf-lamps-lightweight-cmp-profile
	scep	[ThisRFC]	Reserved	Simple Certificate Enrollment Protocol [RFC8994]
vformat	cms	[RFC8368]	Default	CMS signed JSON Voucher
	jose	[ThisRFC]		JOSE-signed JSON draft-ietf-anima-jws-voucher
....				TBD: not analyzed needs for constrained BRSKI/voucher – eg: stateful/stateless support in Registrar

Semantic of Parameters (Variation-Type)

Mode: Overall mode of interaction with registrar

rrm: Pledge/Proxy is initiator, Registrar is responder

prm: Pledge is ALSO responder. Registrar actually stays responder, but instead of proxy we use an agent that is initiator to both Pledge and Registrar. If Registrar was not responder, it would not need to be discoverable. But PRM requires specific new Service Endpoints, so you can not just use PRM with a Registrar not supporting it. These Endpoints are specified in BRSKI-PPRM.

Enroll: Enrollment protocol . This may have different Endpoints and/or encoding of payloads of Endpoints

Vformat: Voucher Format/Encoding supported. This likely will only impact the parameters of Endpoints, but may also impact what Endpoints need to be supported.

Further Semantics

1. Semantic of “Default”

If service does not indicate any variation of a parameter, it is assumed to support the “Default” of the parameter (e.g.: “rrm”)

As soon as one variation is specified for a service, all supported variations need to be specified (hence “prm”, “rrm” if both are supported)

2. Semantic of “Reserved”

The parameter method is defined, but the details of using it with BRSKI are not specified, as long as the registry experts fear that additional specification is required, the variation can be put into “Reserved” status. As soon as someone wants to use the variation, it requires specification with expert review, and the “Reserved” will be removed.

Registrars and Proxies

TBD Text/explanations

Parameters specified initially are all for selecting an appropriate Registrar by Pledged, Proxies and Agents (BRSKI-PRM)

But they equally apply to BRSKI-Proxies because a Pledge that needs to go through a Proxy to reach the Registrar.

Assume Registrar (1) supports “est”, Registrar (2) supports “cmp”.

Necessary Proxy behavior:

Proxy needs to allocate one Port for “est”, one port for “cmp”. Needs to create separate service announcements for each. When client connects, the proxy will know from the connecting port, which Registrar it needs to proxy the Pledge connection to.

Future Parameters

TBD Text/explanations

Future Parameters may be indicating variations not applicable to Registrar/Proxies, but to other (responder) entities in BRSKI

Example: MASA, Pledge

Not currently required (AFAIK) by any ongoing draft work, but could think of BRSKI-PRM Pledges requiring different details during enrollment that are not necessarily known upfront by PRM Agent.

TBD: Need to specify what exactly needs to be specified at minimum for new parameter, e.g.:

Default variation,

Role Type of BRSKI entities assumed to offer services with this parameter variation.

Encoding a BRSKI variation

into different discovery mechanisms

If we keep the variation strings across all parameters unique (Esko)

We only need to signal all parameter variations but not
... That are not default, but no the parameters (variation-type).

Seems like a good idea: Lets demand that for any future registration!

Example: DNS-SD

“prm”, “rrm” instead of “mode=prm,rrm”

Encoding into GRASP

open issue

Toerless worried about possible too many combinations, when registrar starts to support multiple options

E.g.: registrar that supports prm, rrm with either est, cmp or scep.

A) Could create multiple objectives (even for same socket on registrar) for different variations, we do not need to have strings for all possible variation combinations

B) Could demand that combination of any parameters means any combination is supported:

prm-rrm-est-cmp : MUST support both rrm and prm with either est or cmp

(Note this may not be a relevant example...)

Still need to work through all required “merged-strings”.

If we adopt this optimized encoding, then we should likely add a registrar for explicit brski variation combined strings, to ensure there is no misinterpretation about which strings are valid.

Encoding into DNS-SD

mDNS (Multicast) or unicast DNS

Standard: List of key=value

Should always only have a single instance of each key

Most compact, compliant :?

Each Variation is a type. The only values are 0 or 1 for each variation.

In DNS-SD, “type=1” can be abbreviated as “type”, So ultimately, the DNS-SD TXT RR for all Parameters is just the list of variations supported

“prm” “rrm” “est” “cmp” - Example

Encoding into CoAP discovery

Aligning with constrained BRSKI...

?? Might become popular with non-constrained networks as it may be easier to set up as DNS ???

TBD...

Other issues: Diagnostics (service-instance-name)

Whenever one of multiple service instances (e.g.: Registrar/Proxy) is discovered/selected, it is important to be able to diagnose which one it was.

Any BRSKI Diagnostics may be difficult / post-mortem, from some NVRAM log in pledges, so likely easily overlooked until 2nd gen devices/firmware, after too many customer complaints

In with ACP, the ACP-address is a sufficient Identifier of the Registrar, but IPv6 link-local address of Proxy (in RFC8995 GRASP announcement) is not good diagnostics.

In DNS-SD, there is always a “service-instance” element that identifies the “server”. No equivalent in GRASP

Could use simple format for GRASP objective-value:

[“service-instance-name”, “variation-string”]

Service-instance name could be ACP-address as string if Proxy uses ACP, or DNS-name if not.

Or (of course) Toerless more comprehensive DNS-SD into GRASP encoding ;-)

BRSKI-PRM issue

Actually, generic DULL GRASP issue

In BRSKI-PRM, an agent may want to be able to discover potentially a large number of pledges on a large (but potentially slow) network

Example: Mesh network wide broadcast radio IoT network.

mDNS (RFC6762) specifies correct mechanism. Also other protocols, e.g.: RTP RR (receiver reports via IP Multicast):

Include in query list of known answers to suppress sending those

Provide answer interval, asking responders to pick random response time in that interval

DULL GRASP does not specify any such mechanism. Not well suited to this use-case

Not difficult to fix. But should be easy to specify.

Maybe a piece for Toerless' more general DNS-SD into GRASP proposal...

BRSKI-Cloud

BRSKI cloud only discusses Cloud-Registrar without variations

As soon as BRSKI Cloud is to be used with variation, Pledge has possible problem of selecting correct Owner Registrar

Can not (IMHO) use generic discovery because of trust-model (Pledge should trust only what it got redirected to by Cloud Registrar)

Could add some suggestive text to BRSKI-Cloud or TBD variation document:

Cloud Registrar could support configuration of different Owner-Registrar DNS-Names / URLs, one for each variation – pending on Cloud Registrar being able to know from e.g.: IDevID, which variation(s) Pledge supports

But requires then that different variations are not using different. Ports on registrar, but different IP/IPv6 addresses.

Alternative is to see if redirect to DNS-SD service-instance name is possible in HTTP

Next steps

Would like to quickly (before IETF118) complete text. Maybe in existing draft(s).

Ideally would like to put text into separate draft:

Outsource to a common place, like we also did
for voucher YANG with RFC8366bis

Questions / Suggestions ?