How to get a trustworthy DNS Privacy enabling recursive resolver

an analysis of authentication mechanisms for DNS Privacy enabling recursive resolvers

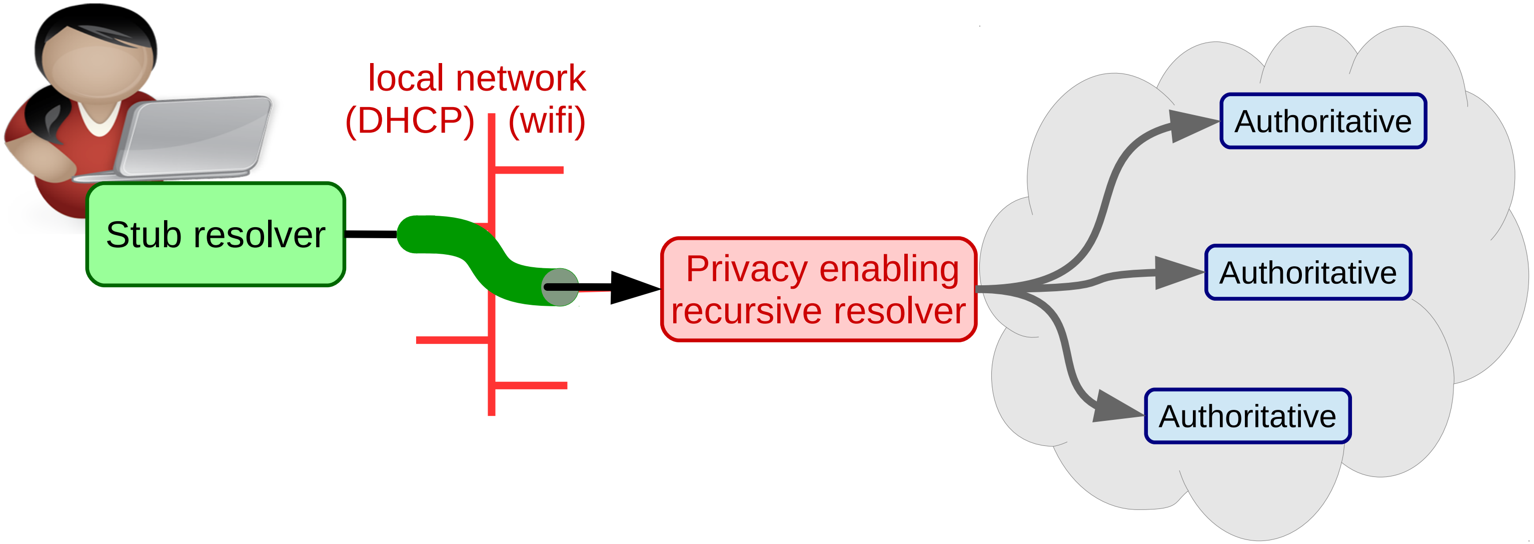
Willem Toorop
NLnet Labs
(presenter)

Melinda Shore
Fastly

Benno Overeinder
NLnet Labs

DNS over TLS
What are the actors, and what are their relationships?

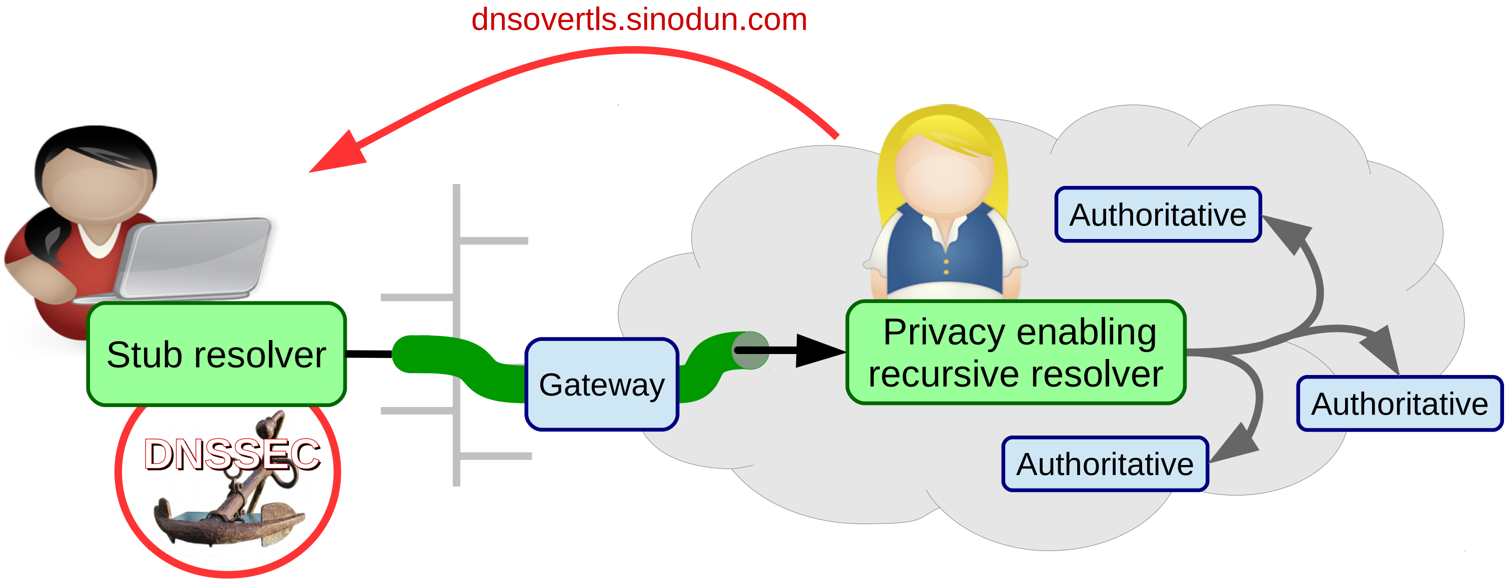
- Current Spec (RFC7858) focuses on securing stub to recursive traffic



- TLS from the system stub client to a privacy enabling recursive resolver can withstand he power and capabilities of a passive pervasive monitor (i.e. an eavesdropper)
- The user entrusts her queries with the *Privacy enabling recursive resolver*
- How did the stub resolver learn the recursive resolver? (traditionally via *DHCP*)

DNS over TLS
What are the actors, and what are their relationships?

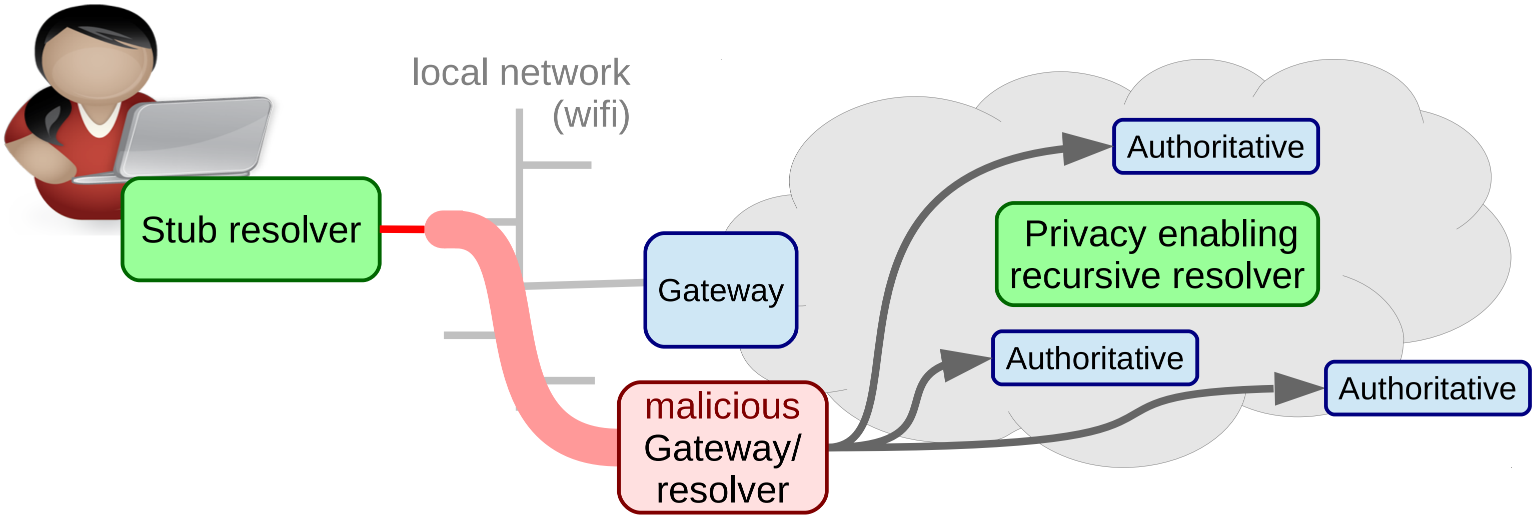
- Current Spec (RFC7858) focuses on securing stub to recursive traffic



- User trusts the channel (Verbally? Website?) over which the connection end-point (IP-address? Name?) was communicated *(what is most reliable to get right, name or IP?)*
- How to get the IP-address for a name securely, and privately *(what is acceptable to leak?)*
- Trust the DNSSEC root trust-anchor + provisioning channel + TLD of the name ?

Authentication

- TLS from stub to resolver cat withstand the power and capabilities of an eavesdropper, it does not withstand an attacker that plugs itself into the path



- Trust in *the network* can be replaced with authentication
- In RFC7858 and draft-ietf-dtls-and-tls-profiles authenticated TLS is called ***Strict***.
- **Oppertunistic** is the *best you can get* modus operandi

Analysis of authentication mechanisms

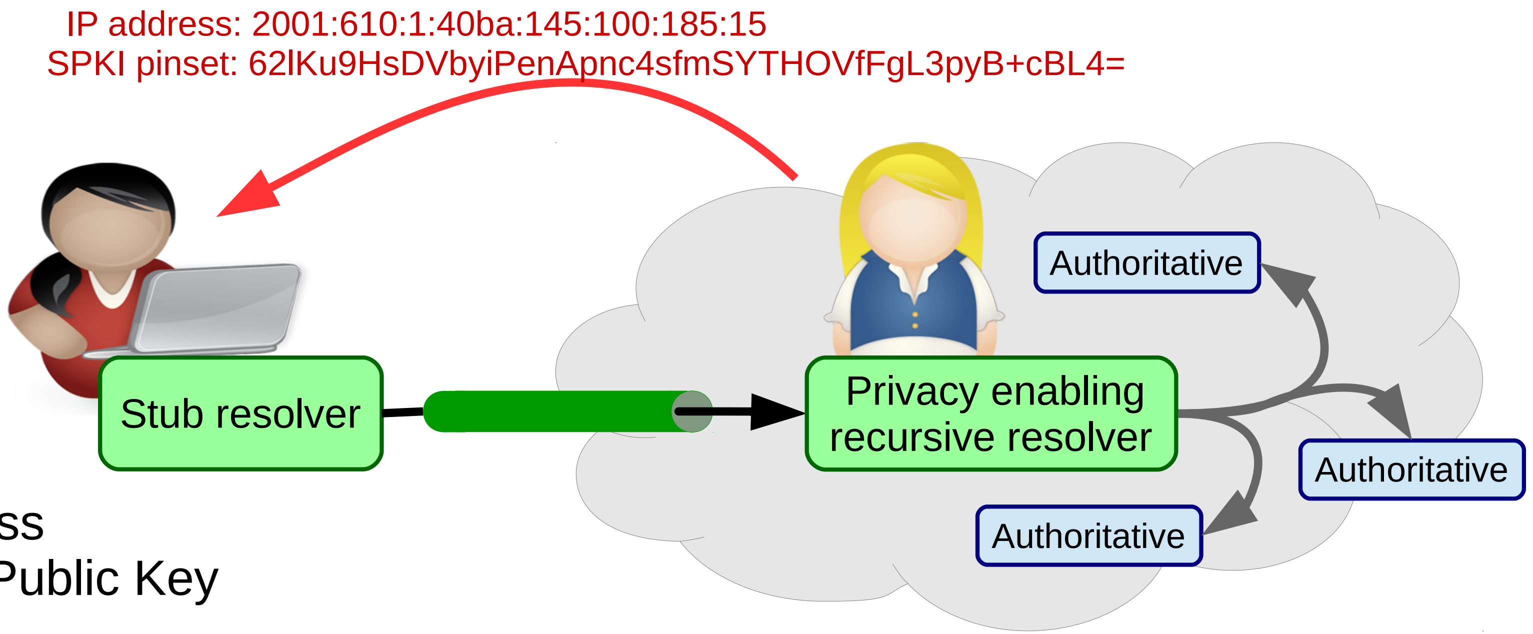
- *Analyzed mechanisms: (from draft-ietf-dprive-dtls-and-tls-profiles)*
  - SubjectPublicKeyInfo pinning … SPKI
  - Traditional Public Key Infrastructure for X.509 Certificates
    - Statically configured Authentication Domain Name and IP address … PKIX ADN + IP
    - Statically configured Authentication Domain Name + dynamically obtained IP … PKIX ADN only
  - DNS Based Authentication of Named Entities … DANE
  - TLS DNSSEC Authentication Chain Extension
- There are key trade-offs between
  - Usability & provision flexibility (important for adoption and correct usage)
  - meta queries leaking information in these mechanisms
  - Requirements on additional dependencies (fewer deps, less can break; i.e. Robustness)
    - Availability of unhampered DNSSEC and DNSSEC capable stub resolver
    - Third parties (Trust anchor/CA store) that do the authentication

Analysis of authentication mechanisms

	1	2	3	4	5
SPKI					
PKIX ADN + IP					
PKIX ADN only					
DANE					
Chain Extension					

- *We did an analysis on the basis of these considerations:*
  1) Ease of configuration … Least possible config to identify the trusted recursive resolver
  2) Key management … Can it handle updated, rolled or withdrawn keys
  3) Information leakage … Leaks info about the *trusted* recursive resolver, via DNS or SNI
  4) DNSSEC dependency … Needs DNSSEC availability and capability for bootstrapping
  5) Trust requirements … Dependencies and maintainability on Trust Anchor and/or CA store

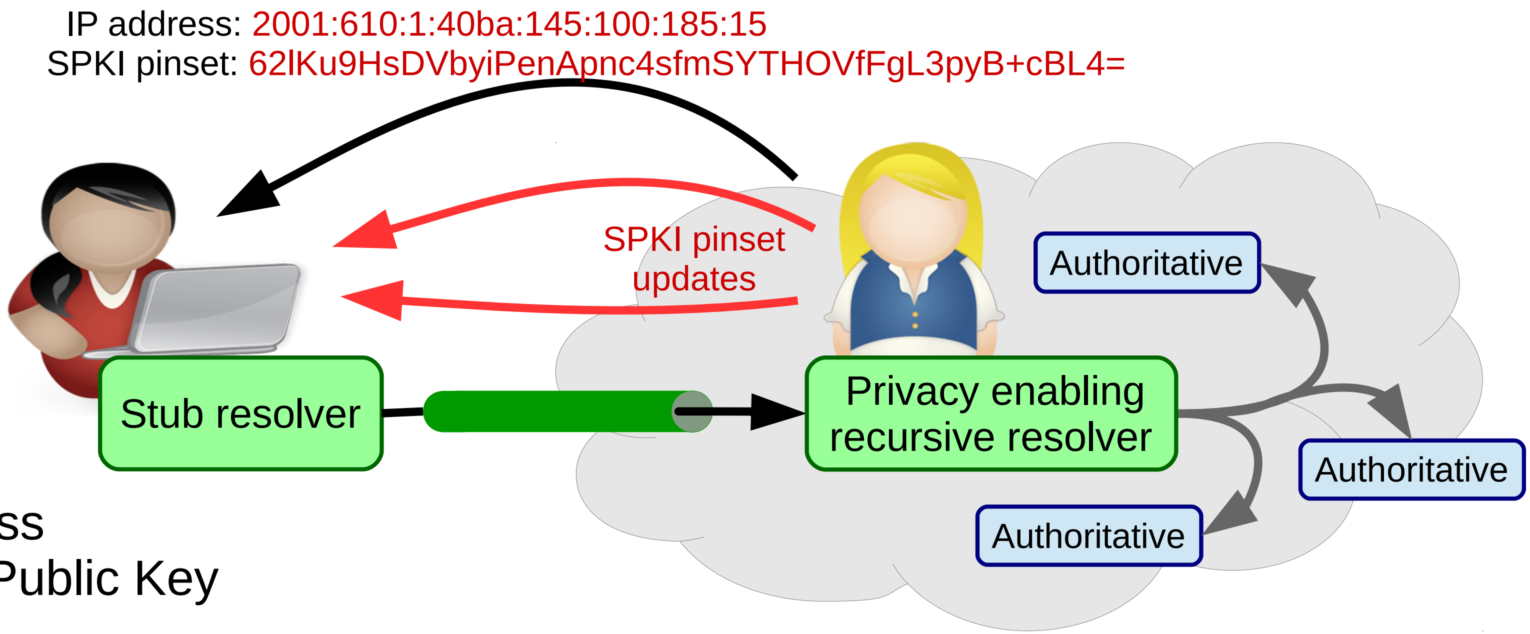
SubjectPublicKeyInfo (SPKI) pinning



? IP address
? hash of Public Key

+ direct and simple
+ nothing is leaked
+ no additional network activity

SubjectPublicKeyInfo (SPKI) pinning



? IP address
? hash of Public Key

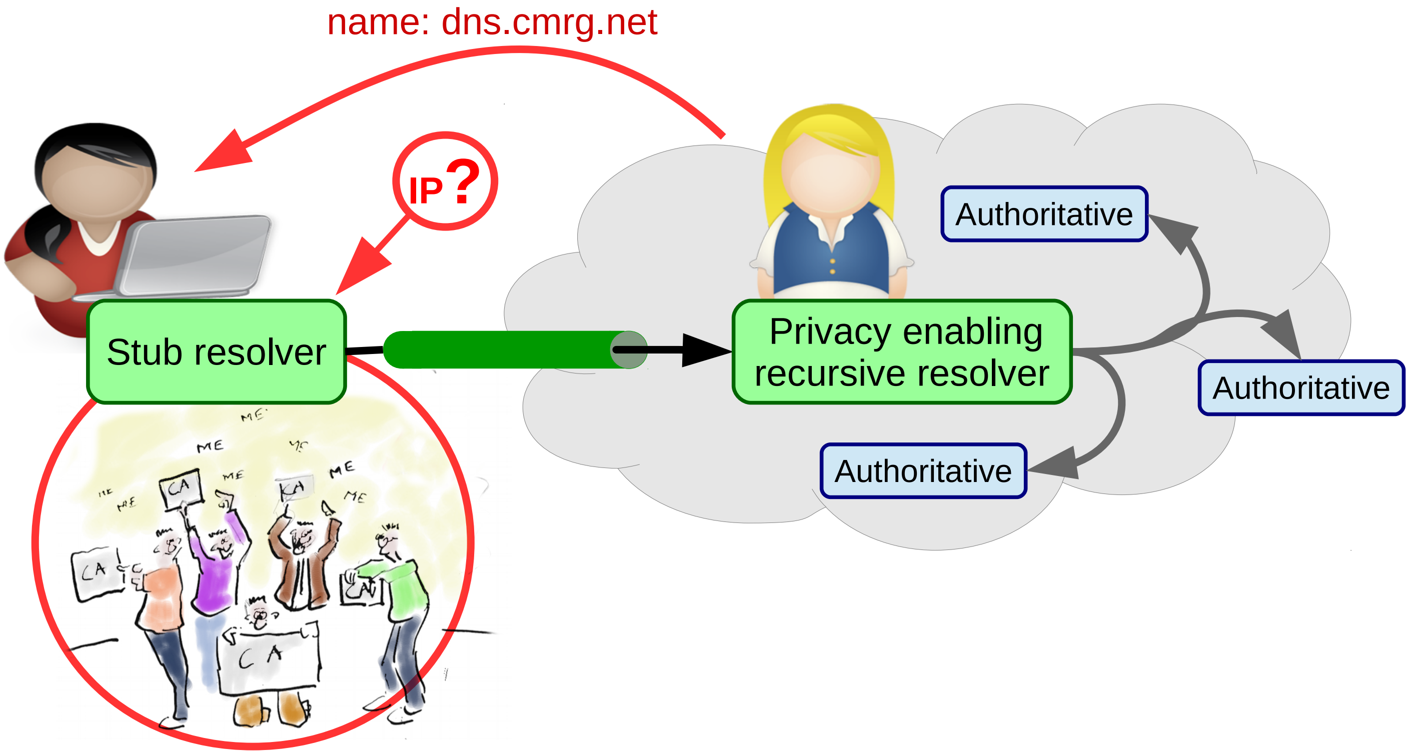
+ direct and simple
+ nothing is leaked
+ no additional network activity

- IP-address and pinset are easy to get wrong
- Lacks provisioning
- Lacks compromised and updated keys signaling

Tip! Backup pinsets

Traditional Public Key Infrastructure for X.509 Certificates (PKIX)

? name
? IP address
- static, DHCP or DNS



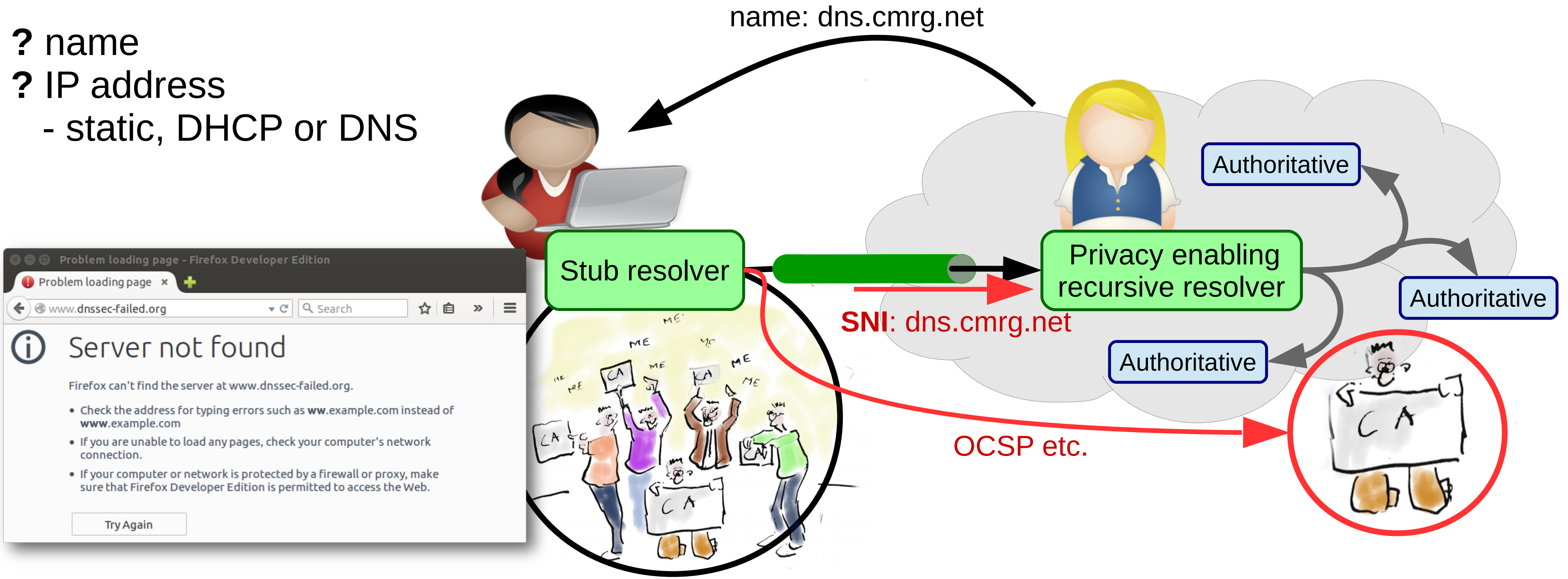
+ traditional, well-known OS managed

+ keys can be rolled

- All CA’s in the store can vouch for any name

Traditional Public Key Infrastructure for X.509 Certificates (PKIX)

? name
? IP address
  - static, DHCP or DNS

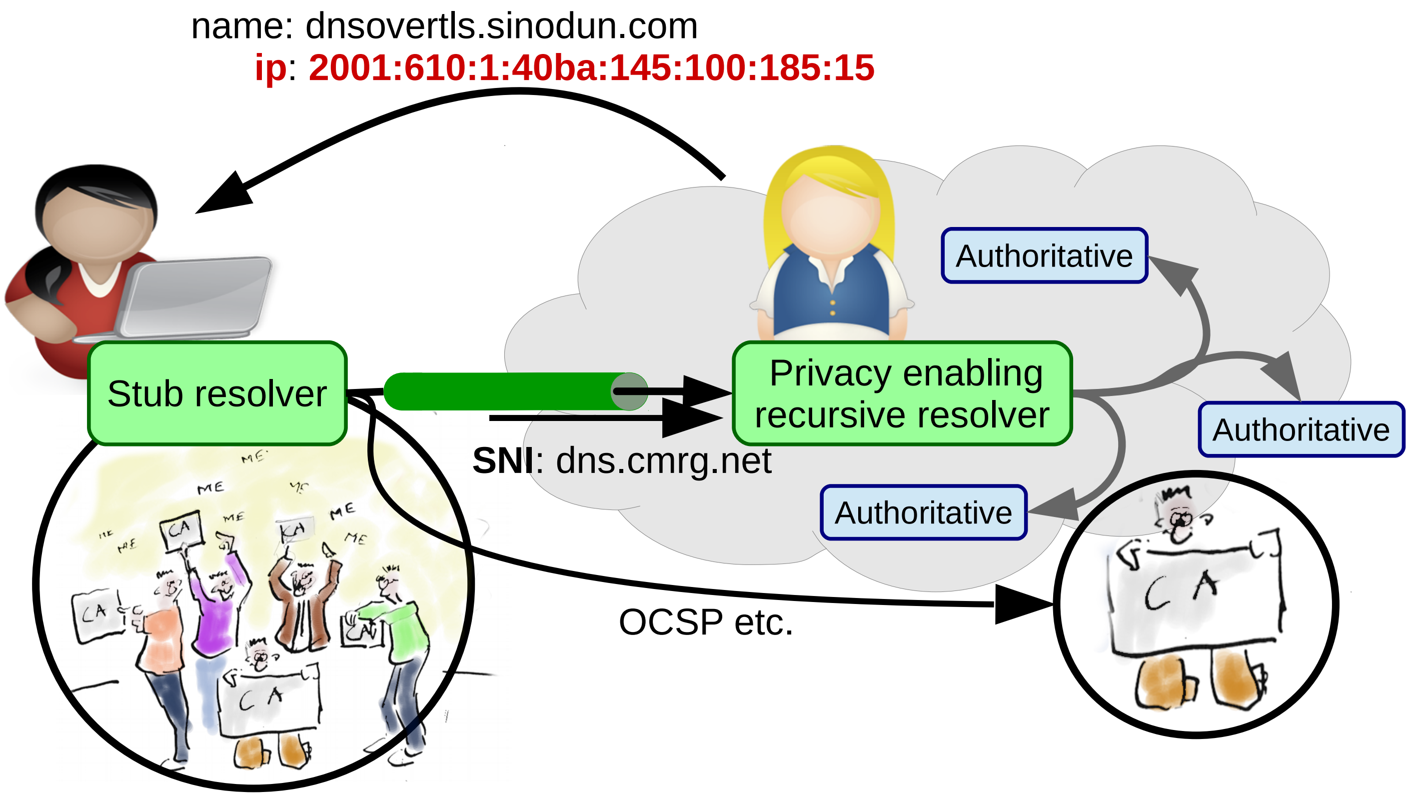


- All CA's in the store can vouch for any name
- no signaling of unknown CA *(reason for opportunistic encryption with SMTPS)*
- network access + DNS is already needed for OCSP etc.

PKIX - statically configured IP address

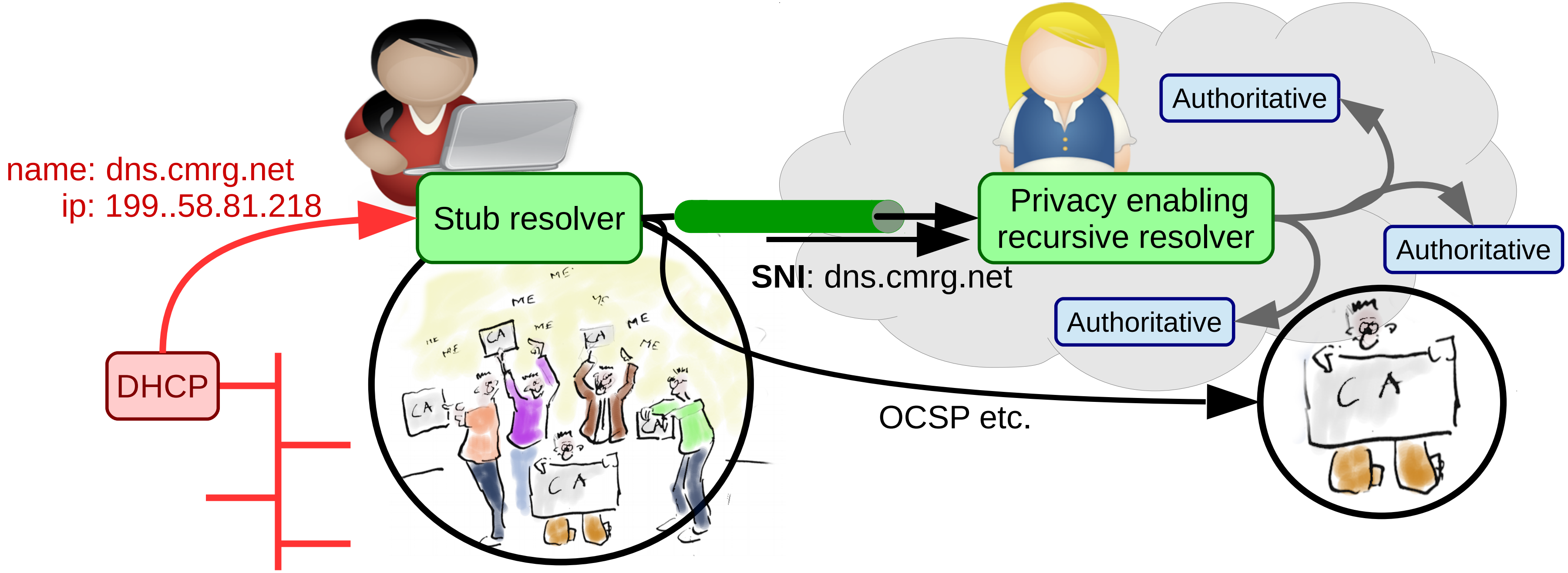
? name
? static IP address

- IP easy to get wrong
- no IP change signalling



PKIX – Both name and IP address came from DHCP

+ Dynamically configured Authentication Domain Name

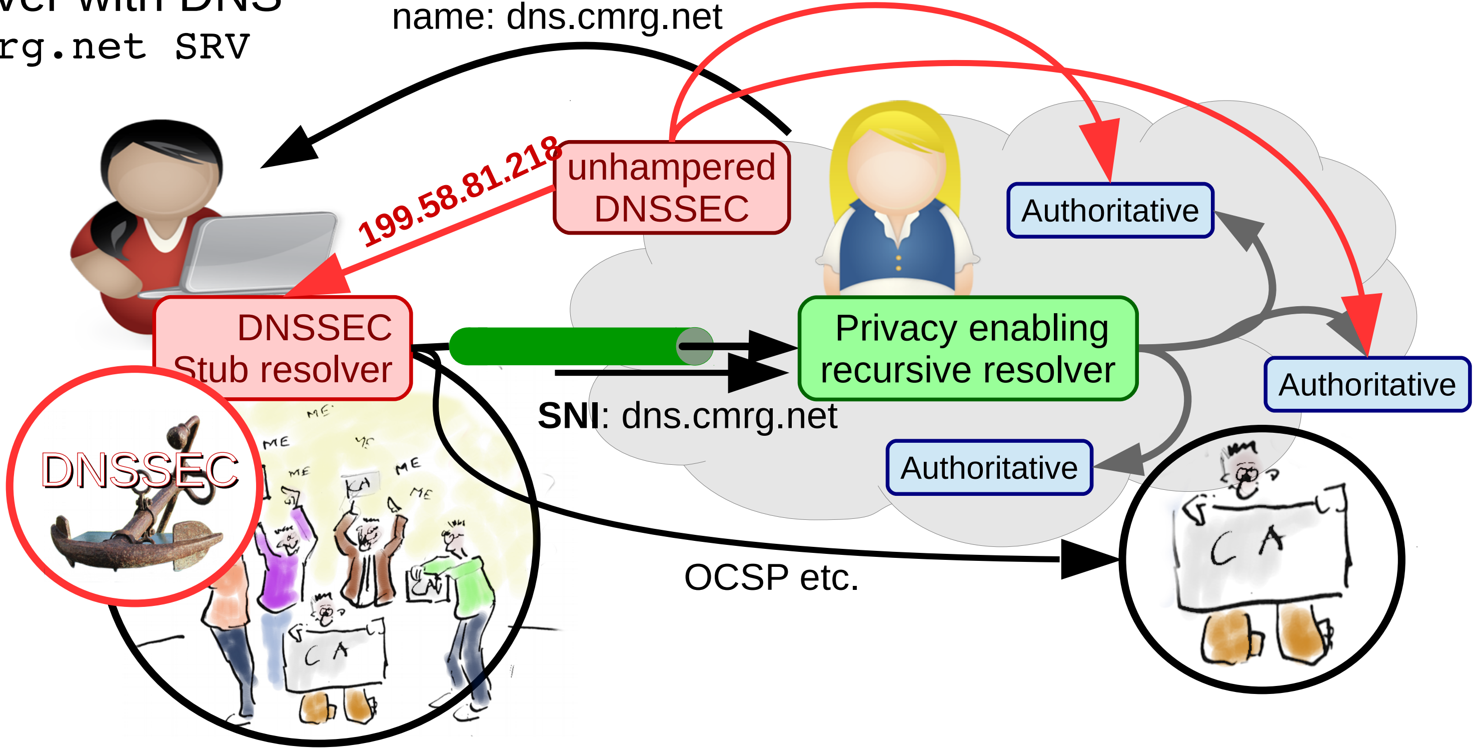


- Needs secure DHCP (does not exist) + extension to convey the ADN
- Shifts problem to bootstrapping secure DHCP *(how is that statically configured?)*

PKIX – statically configured name, IP address from DNS

Lookup the privacy resolver with DNS
`_domain-s._tcp.dns.cmrg.net SRV`

+ IP change provisioning



`draft-ietf-dprive-dtls-and-tls-profiles` requires DNSSEC for lookup

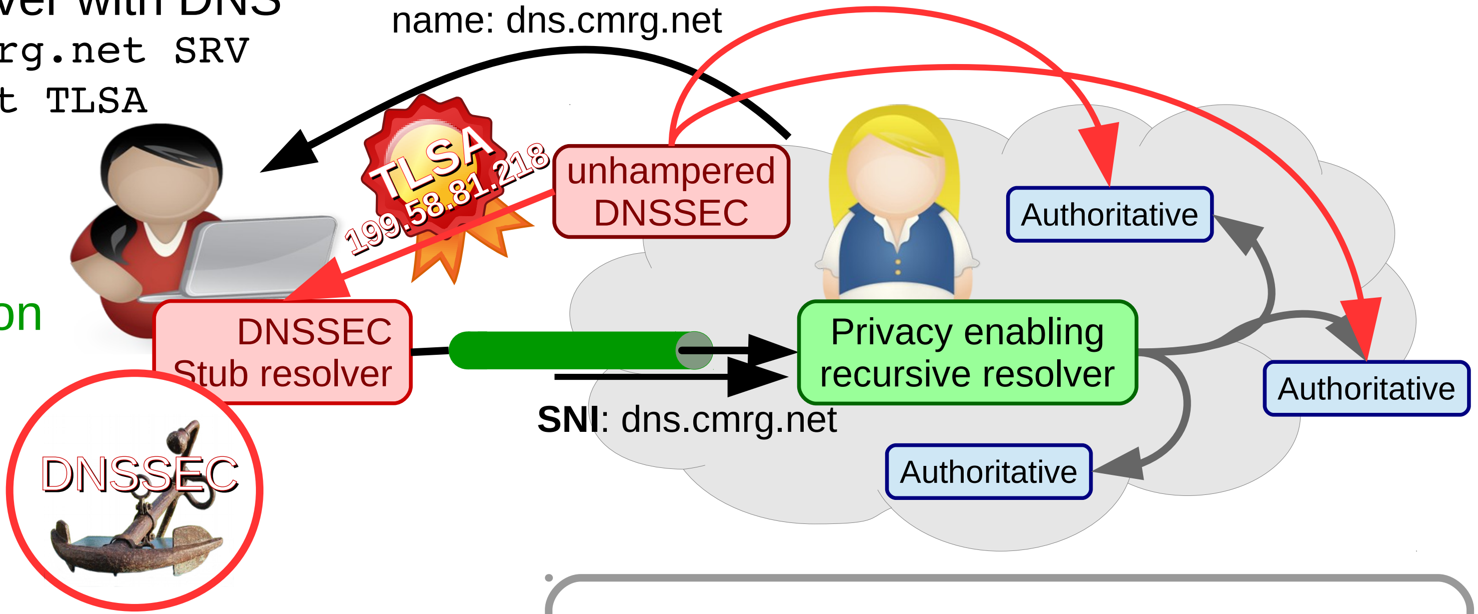
- Needs unhampered DNSSEC
- Additional trust in DNSSEC trust anchor
- DNSSEC capable stub resolver needed
+ In protocol trust anchor rollover (RFC5011)

DNS Based Authentication of Named Entities (DANE)

Lookup the privacy resolver with DNS

```
_domain-s._tcp.dns.cmrg.net SRV
_853._tcp.dns.cmrg.net TLSA
```

+ IP change provisioning
+ No more dependency on CA infrastructure



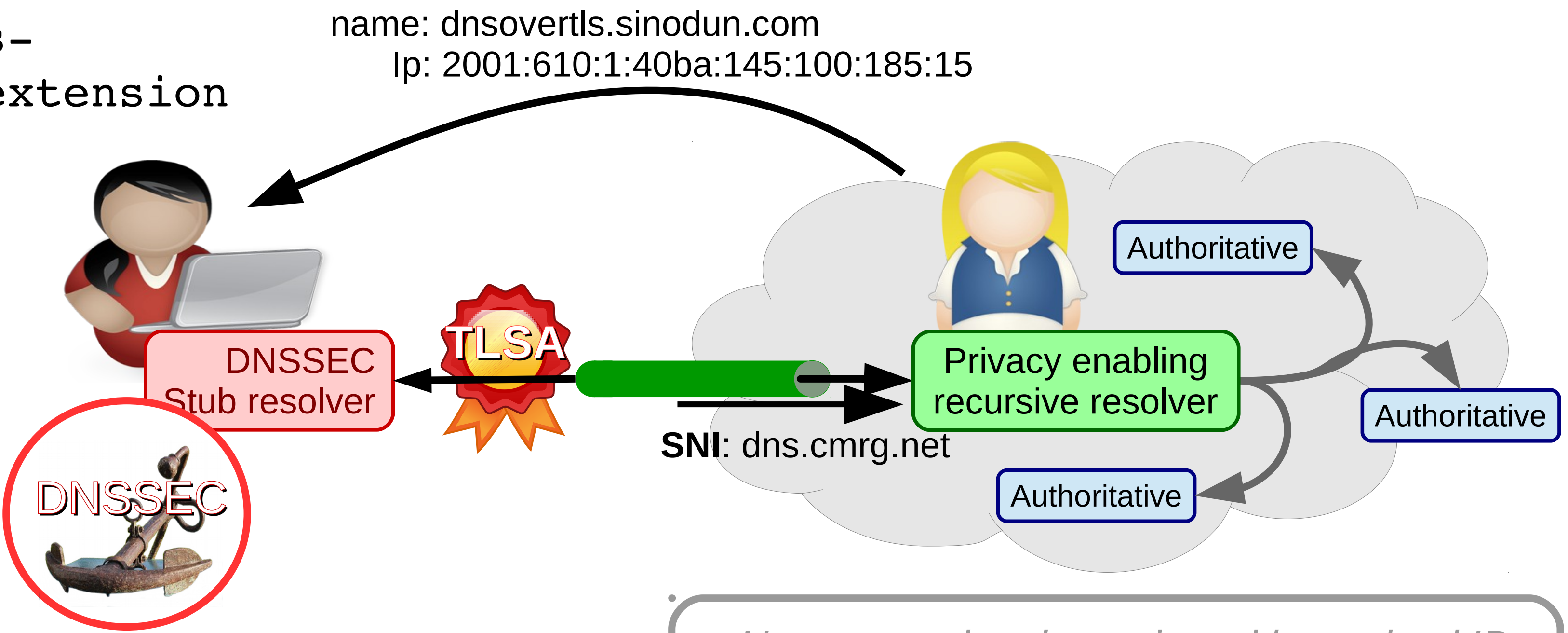
Not concerning the option with provided IP, because that has no additional benefits

- Needs unhampered DNSSEC
- Additional trust in DNSSEC trust anchor

- DNSSEC capable stub resolver needed
+ In protocol trust anchor rollover (RFC5011)

TLS DNSSEC Authentication Chain Extension

`draft-ietf-tls-dnssec-chain-extension`



Not concerning the option with resolved IP, because that has no additional benefits compared to the pure DANE option

+ Smallest setup latency *(same as SPKI)*
– No IP change provisioning
+ No more dependency on CA infrastructure
+ No need for unhampered DNSSEC
– Additional trust in DNSSEC trust anchor

– DNSSEC capable stub resolver needed
+ In protocol trust anchor rollover (RFC5011)

Comparison of the different considerations per mechanism

	Ease of configuration
SPKI	**--**
PKIX ADN + IP	**-**
PKIX ADN only	**++**
DANE	**++**
Chain extension	**-**

++) PKIX ADN only, DANE — need only the name

-) PKIX ADN + IP, Chain ext. — need name + IP
(IPv6 addresses are hard to communicate)

--) SPKI — needs IP + pinset
(Base64 pinset is impossible to communicate)

Comparison of the different considerations per mechanism

	Ease of configuration	Key management
SPKI	--	--
PKIX ADN + IP	-	-
PKIX ADN only	++	-
DANE	++	+
Chain extension	-	+

+) DANE, Chain extension — DNSSEC has single trust anchor
in protocol key management (RFC5011)
bootstrap problem when of for long period?

-) PKIX ADN's — Traditional, well known, managed by OS, but
weakest link problem
lack of unknown CA signaling

--) SPKI — Complete manual provisioning with long Base64 string

Comparison of the different considerations per mechanism

	Ease of configuration	Key management	Information leakage
SPKI	--	--	++
PKIX ADN + IP	-	-	-
PKIX ADN only	++	-	--
DANE	++	+	--
Chain extension	-	+	+

++) SPKI — No non-TLS communications, no SNI

+) Chain extension — No non-TLS communications, leaks name by SNI

-) PKIX ADN + IP — No non-TLS communications, leaks name by SNI, leaks CRL checking

--) PKIX ADN only, DANE — DNS communication before TLS setup, leaks SNI
PKIX also leaks CRL

Comparison of the different considerations per mechanism

	Ease of configuration	Key management	Information leakage	DNSSEC dependency
SPKI	--	--	++	++
PKIX ADN + IP	-	-	-	++
PKIX ADN only	++	-	--	--
DANE	++	+	--	--
Chain extension	-	+	+	+

++) SPKI, PKIX ADN + IP — No DNSSEC dependency

+) Chain extension — Not affected by DNSSEC hampering middle boxes
Requires DNSSEC capable stub resolver

--) PKIX ADN only, DANE — Requires unhampered DNSSEC availability
Requires DNSSEC capable stub resolver

Comparison of the different considerations per mechanism

	Ease of configuration	Key management	Information leakage	DNSSEC dependency	Trust requirements
SPKI	--	--	++	++	++
PKIX ADN + IP	-	-	-	++	-
PKIX ADN only	++	-	--	--	--
DANE	++	+	--	--	+
Chain extension	-	+	+	+	+

++) SPKI — trust the outbound communication channel connection endpoint details

+) DANE, Chain extension — Additional trust on DNSSEC trust anchor + TLD

-) PKIX ADN + IP — Additional trust on all CA’s in the trust store

--) PKIX ADN only — Additional trust on DNSSEC trust anchor + TLD
Additional trust on all CA’s in the trust store

Comparison of the different considerations per mechanism

	Ease of configuration	Key management	Information leakage	DNSSEC dependency	Trust requirements
SPKI	--	--	++	++	++
PKIX ADN + IP	-	-	-	++	--
PKIX ADN only	++	-	--	--	--
DANE	++	+	--	--	-
Chain extension	-	+	+	+	+

How would you weigh the considerations?