DNSSEC + x509

Leveraging DNSSEC for DV certificates

Danny Groenewegen[1], Pieter Lange[1]

[1]System and Network Engineering
Universiteit van Amsterdam

Michiel Leenaars[2]
[2]NLnet Labs
NLnet Foundation

2^{nd} of February 2011
RP1 presentations

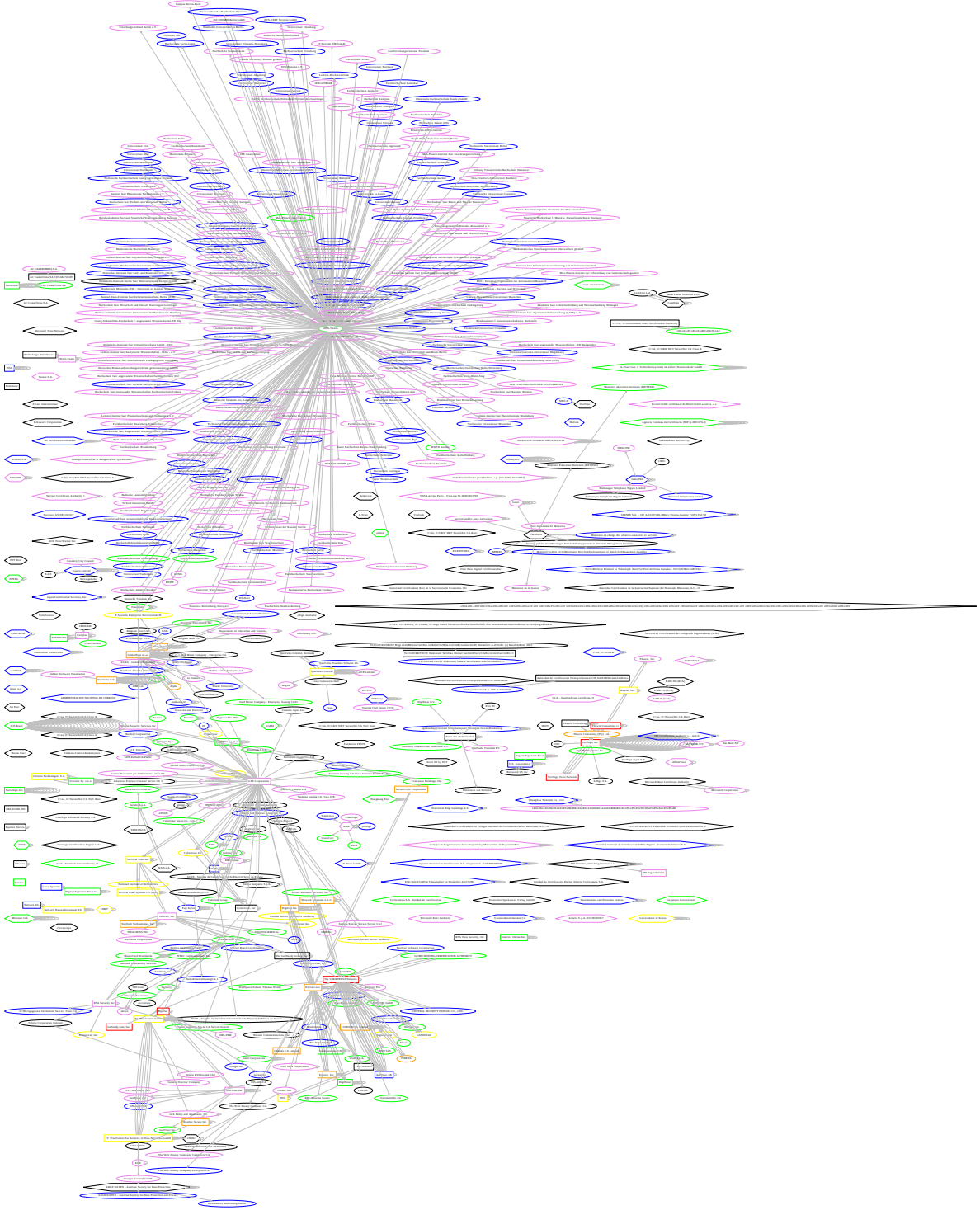
Outline

Outline

Certificate Authorities

- We don't know who we've delegated our trust decisions to.
- Only one has to misbehave. We have over 600 SPOFs.[1]

[1] https://www.eff.org/observatory

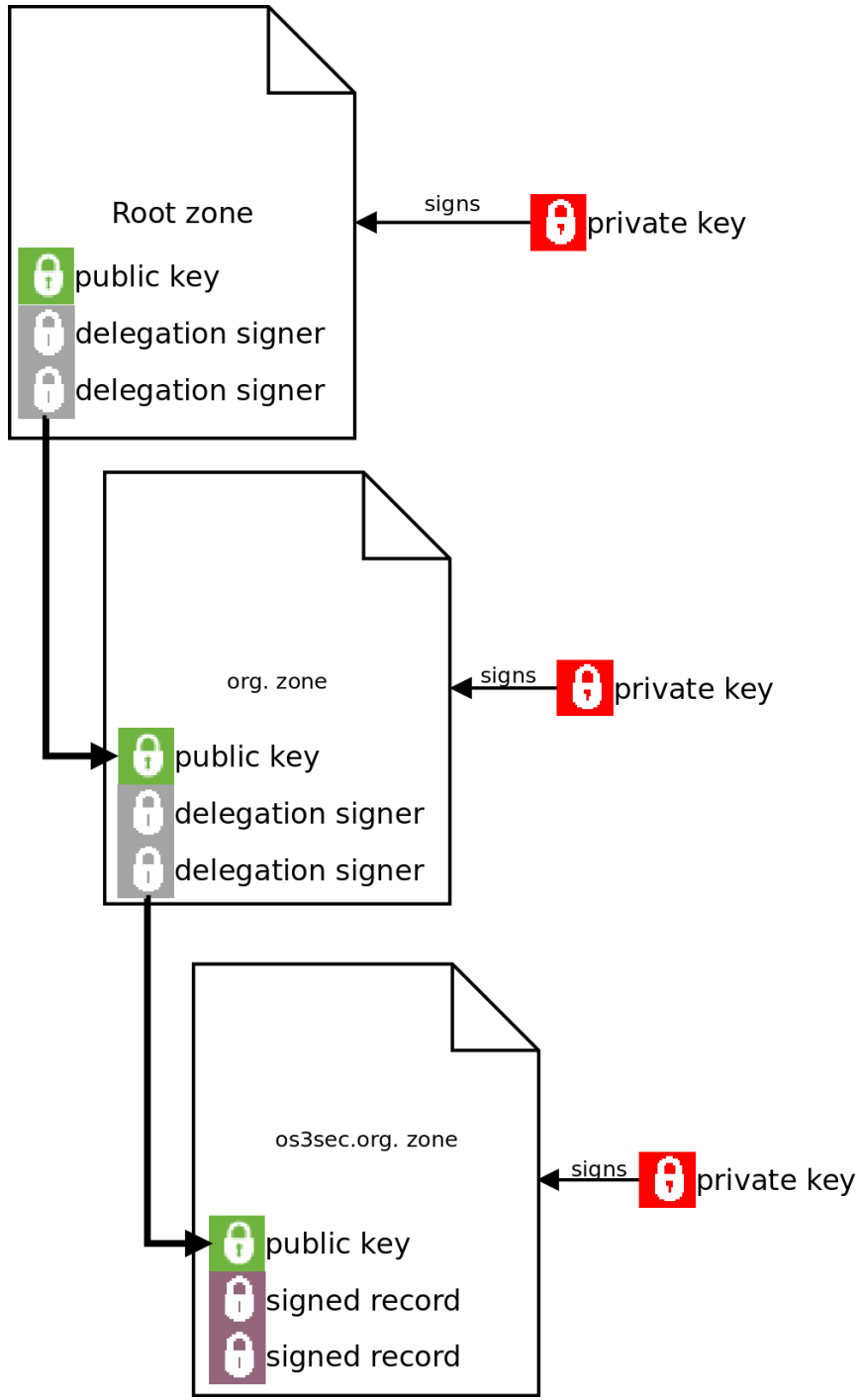


DNSSEC Provides Infrastructure

for trust.

Puts zone owner in control

- End to end integrity
  - announce our public keys to the world


Root zone
public key
delegation signer
delegation signer
signs
private key
org. zone
public key
delegation signer
delegation signer
signs
private key
os3sec.org. zone
public key
signed record
signed record
signs
private key

DNSSEC Provides Infrastructure

for trust.

Puts zone owner in control

- End to end integrity
  - announce our public keys to the world
- We're not the first to come up with this idea...

DNSSEC Provides Infrastructure

for trust.

Puts zone owner in control

- End to end integrity
  - announce our public keys to the world
- We're not the first to come up with this idea...
  - but we have an implementation!

Outline

How should we do this?

Not as easy as it looks

- *Where* to place the key?

How should we do this?

Not as easy as it looks

- *Where* to place the key?
  - in the label

How should we do this?

Not as easy as it looks

- *Where* to place the key?
  - in the label
  - create new label from hash of certificate

How should we do this?

Not as easy as it looks

- *Where* to place the key?
  - in the label
  - create new label from hash of certificate
- *What* is the key? *What* format to use?

Work In Progress

- IETF (dane) $\rightarrow$ TLSA
- Dan Kaminsky

Dan Kaminsky

"Domain Key Infrastructure"

- Blogs count as documentation nowadays
- Iterative development
- Took some shortcuts. . .[2]

[2] not necessarily bad.

Dan Kaminsky

The format: where

- Chose to use `TXT` records

Dan Kaminsky

The format: where

- Chose to use `TXT` records
- Combination of "*where*":
  - in the label:

```
www          IN TXT "v=key1 ha=sha1 h=e242...fba1"
```

Dan Kaminsky

The format: where

- Chose to use `TXT` records
- Combination of "*where*":
  - in the label:

    ```
    www          IN TXT "v=key1 ha=sha1 h=e242...fba1"
    ```
  - label+hashlabel:

    ```
    www                            IN TXT "v=key1 lh=1"
    _keyhash-e242...fba1.www IN TXT "anything"
    ```

Dan Kaminsky

The format: what

- Hashes
  - Entire certificate (`TXT "... hr=cert"`)
  - Only the public key (`TXT "... hr=pubkey"`)

Dan Kaminsky

The format: what

- Hashes
  - Entire certificate (`TXT "... hr=cert"`)
  - Only the public key (`TXT "... hr=pubkey"`)
- Entire key:
  - `www IN TXT "v=key1 pka=rsa e=65537 m=ANknyBHye+RFyUa2Y3WDsXd+F0...KtT"`

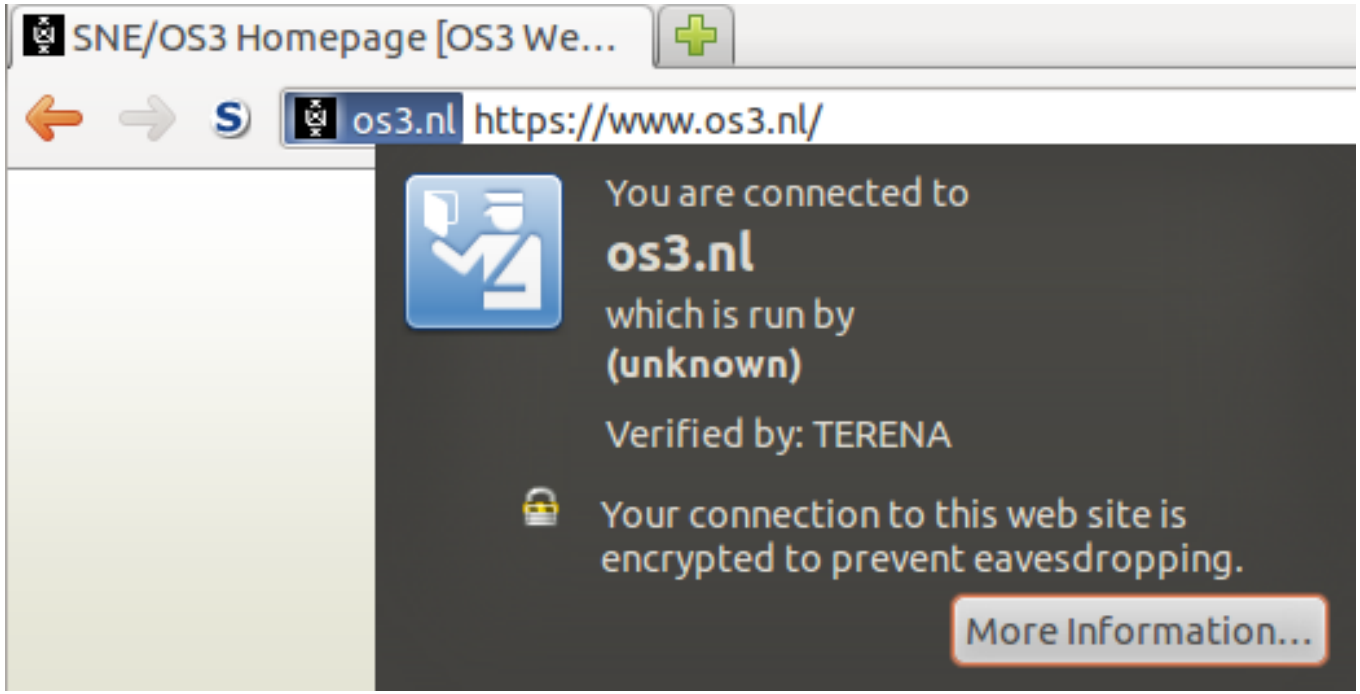
Dan Kaminsky

The format: what

- Hashes
  - Entire certificate (`TXT "... hr=cert"`)
  - Only the public key (`TXT "... hr=pubkey"`)
- Entire key:
  - `www IN TXT "v=key1 pka=rsa e=65537 m=ANknyBHye+RFyUa2Y3WDsXd+F0...KtT"`
  - Saves round trip in TLS/IPSEC handshake

Outline

What it does


SNE/OS3 Homepage [OS3 We...
os3.nl
https://www.os3.nl/
You are connected to
os3.nl
which is run by
(unknown)
Verified by: TERENA
Your connection to this web site is encrypted to prevent eavesdropping.
More Information...

What it does


SNE/OS3 Homepage [OS3 Webs...
os3.nl
https://www.os3.nl/
DNSSEC
Domainname is secured by DNSSEC and the certificate is validated by DNSSEC and CA.
You are connected to
os3.nl
which is run by
(unknown)
Verified by: TERENA
Your connection to this web site is encrypted to prevent eavesdropping.
More Information...

What it does


```
;; QUESTION SECTION:
;www.os3.nl. IN TXT

;; ANSWER SECTION:
www.os3.nl. 82721 IN CNAME info4u.os3.nl.
info4u.os3.nl. 86302 IN TXT "v=key1 ha=sha1 h=4a2662313f6e5d7b706e3a21742177281a2938f1"
```

Functionality

1. Integration with `libunbound`

Functionality

1. Integration with `libunbound`
   - (We're better than dnssec-validator.cz!)

Functionality

1. Integration with `libunbound`
   - (We're better than dnssec-validator.cz!)
2. TXT record
   - Strict Transport Security

Functionality

1. Integration with `libunbound`
   - (We're better than dnssec-validator.cz!)
2. TXT record
   - Strict Transport Security
3. TLSA record
   - No STS, but we have a button

Functionality

1. Integration with `libunbound`
   - (We're better than dnssec-validator.cz!)
2. TXT record
   - Strict Transport Security
3. TLSA record
   - No STS, but we have a button
4. Supports FF4 on Linux, Mac OSX and Windows

What it doesn't do

It's a proof of concept after all

1. Anything but SHA1 hashes
2. TXT:
   - `LH=1`
   - `HR=[cert|pubkey]`
   - STS doesn't work for self-signed certificates
3. TLSA:
   - CA in TLSA validation
4. Proper caching
5. Getting records' TTL

Outline

DEMO

Summary

- Very easy to integrate `libunbound`
- Very hard to inform users
- Certificate authorities need to find a new business (EV)

Summary

- Very easy to integrate `libunbound`
- Very hard to inform users
- Certificate authorities need to find a new business (EV)

- Outlook
  - Cheap and reliable PKI is coming.

Thanks NLnet!

Questions?

- Add-on – `https://os3sec.org`
- TXT spec – `http://dankaminsky.com/`
- Dane WG – `http://tools.ietf.org/wg/dane/`
- NLnet – `http://nlnet.nl/dnssec`
- OS3 – `https://www.os3.nl`

IETF workgroup

The format

- *What?*
  - For now, TLS. Both hashes and entire certificates.
  - End entity and certificate authorities.

IETF workgroup

The format

- *What?*
  - For now, TLS. Both hashes and entire certificates.
  - End entity and certificate authorities.
- Current format:
  - Certificate type (1=hash of EE, 2=full EE cert,...)

IETF workgroup

The format

- *What?*
  - For now, TLS. Both hashes and entire certificates.
  - End entity and certificate authorities.
- Current format:
  - Certificate type (1=hash of EE, 2=full EE cert,...)
  - Hash type (0=none, 1=sha1, 2=sha256, 3=sha384,...)

IETF workgroup

The format

- *What?*
  - For now, TLS. Both hashes and entire certificates.
  - End entity and certificate authorities.
- Current format:
  - Certificate type (1=hash of EE, 2=full EE cert,...)
  - Hash type (0=none, 1=sha1, 2=sha256, 3=sha384,...)
  - Certificate for association
  - `www IN TLSA ( 1 1 e242...fba1)`

How should we do this?

Aren't we forgetting something?

Policies...

- The default is still *insecure*

How should we do this?

Aren't we forgetting something?

Policies...

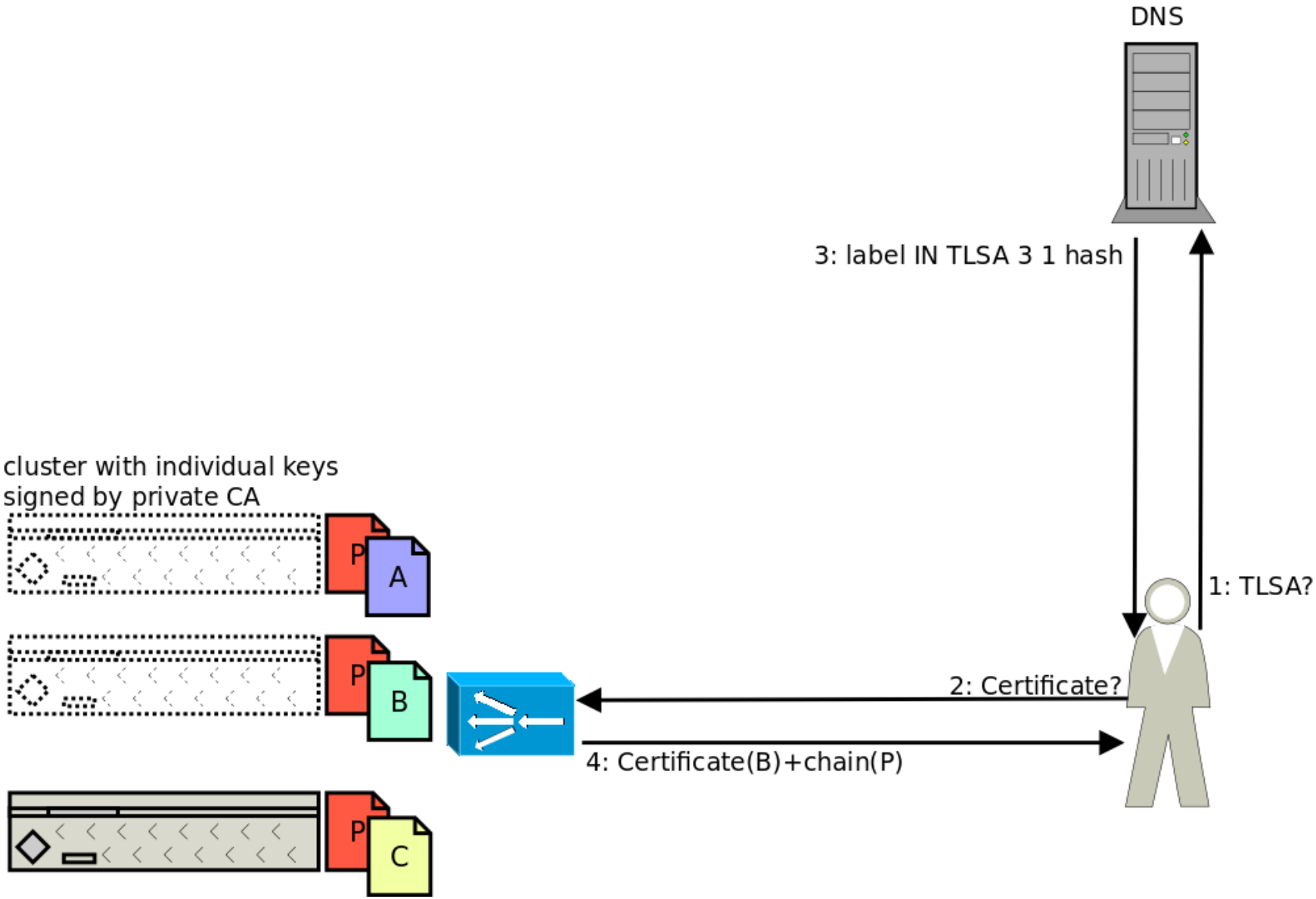
- The default is still *insecure*
- Now is the time to fix that.
  - HASTLSA discussion
  - Lots of bickering...

How should we do this?

Aren't we forgetting something?

Policies...

- The default is still *insecure*
- Now is the time to fix that.
  - HASTLSA discussion
  - Lots of bickering...
- TXT has this: STS, SN...


DNS
3: label IN TLSA 3 1 hash
1: TLSA?
cluster with individual keys
signed by private CA
P
A
P
B
P
C
2: Certificate?
4: Certificate(B)+chain(P)

os3sec.org zone

```
scsigned IN A 145.100.105.212
IN TXT "v=key1 ha=sha1 h=5F8B024DEE05CF820517A7C471BF3D234I
selfsigned IN A 145.100.105.211
IN TXT "v=key1 ha=sha1 h=570651DA8D1D42C34937A0FDF4E29F93FI
sts=1" broken IN A 145.100.105.211
IN TXT "v=key1 ha=sha1 h=THISISBROKENTHISISBROKENTHISISBROK

signedtlsa IN A 145.100.105.214
IN TYPE65534 # 22 ( 0101052D9B22DDB83DF87FB458CFF5BFB676E03
)
wikileaks IN A 145.100.105.211
IN TXT "v=key1 ha=sha1 h=570651DA8D1D42C34937A0FDF4E29F93FI
sts=1"
```