PCH DNSSEC Signer Update

2024-02
DNS OARC 42, Charlotte (NC), US

Tamás Csillag
DNS Services Engineer

Outline

Short intro of PCH's history and what we do.

DNSSEC goals and scope.

Technology and history of the key ceremonies and signer.

Motivations to change.

Choosing knot in part because of their offline-ksk functionality.

Further ideas and developments to improve reliability with verification of the signed zones (nsd).

Packet Clearing House

A Short History of PCH

Originated as an outcome of the 1992 “National Information Infrastructure” transition of Internet governance from the US government to the global private sector.

Responsible for providing operational security and stability for critical Internet infrastructure globally, much like a “fire department” for the Internet.

PCH works primarily in four areas: the core of the DNS, IXPs, regulatory & policy, and cybersecurity coordination.

Packet Clearing House

A Short History of PCH

Operating production anycast since 1994, precursor organizations since 1989

Advocating for the anycasting of the root nameservers since 1996

Began providing ccTLD nameservice in 1997

Started anycasting root nameservers in 2000

Began providing services over IPv6 in 2001

Anycasted the second DNSSEC-signed ccTLD in 2006

Started providing FIPS 140-2 L4 DNSSEC key management in 2011

Deployed first DNSSEC-validating, GDPR-compliant, recursive resolver in 2016

Packet Clearing House

Server clusters at IXPs in 287 cities, 125 countries

FIPS 140-2 Level 4 DNSSEC key management for 28 countries

Physical security facilities in San Jose, Zurich, and Singapore, with a fourth planned in Montevideo

Scope of DNSSEC Signing

Bump in the wire signing, intended to encourage the use of DNSSEC.

Signing 174 zones, including the canonical second-level domains of many ccTLDs.

Two online signing facilities (with ZSKs loaded for signing): Zürich and San Jose.

Two offline key ceremony facilities: San Jose and Singapore with KSKs used during key ceremonies to generate ZSKs and matching RRSIGs. Otherwise kept offline.

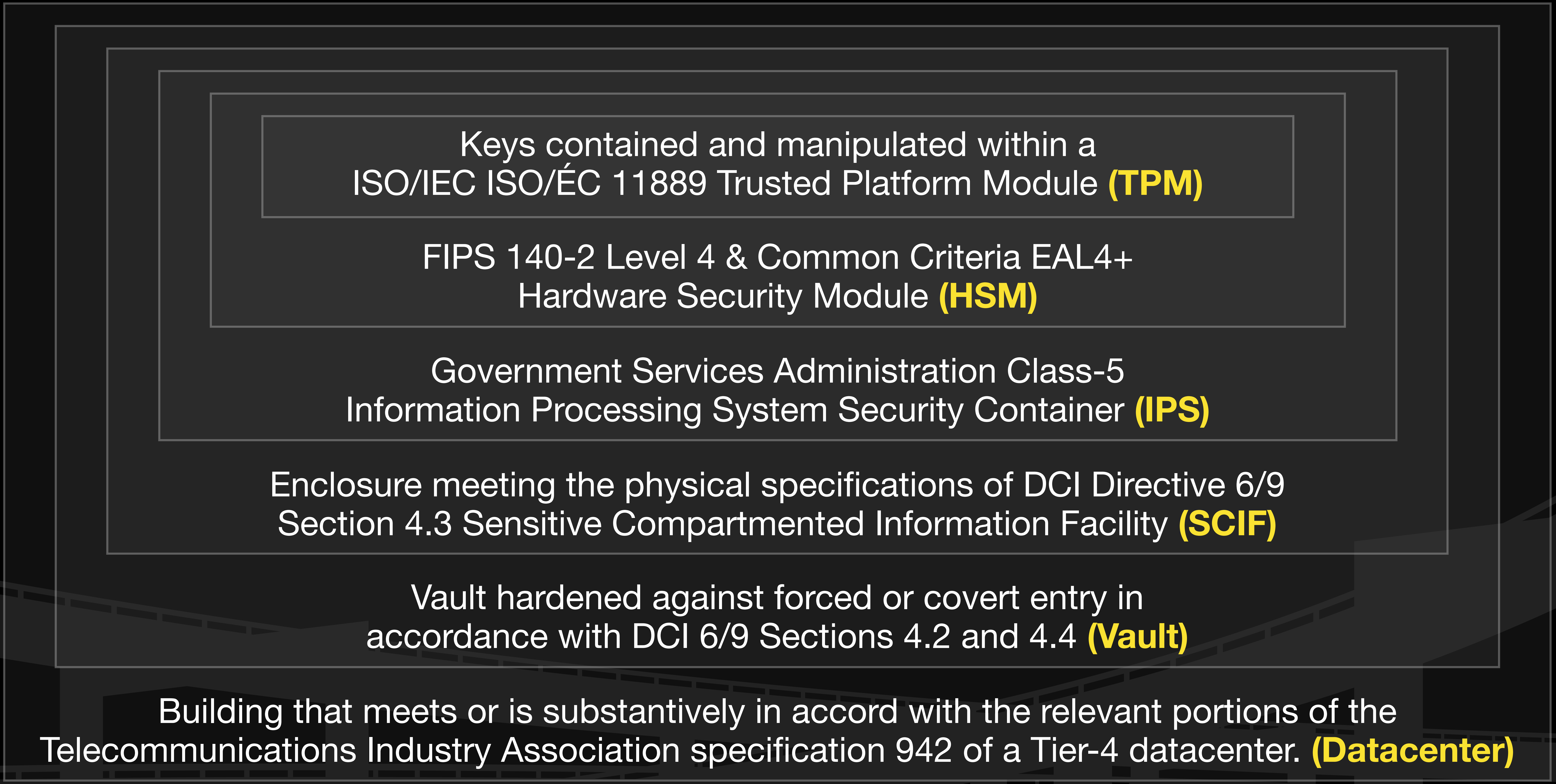
This process is intentionally similar to the ICANN root zone signing process.

FIPS
Level 4 Validated
140-2

Common Criteria
EAL4+ Certified

Packet Clearing House

Physical Security Architecture



Keys contained and manipulated within a
ISO/IEC ISO/ÉC 11889 Trusted Platform Module **(TPM)**

FIPS 140-2 Level 4 & Common Criteria EAL4+
Hardware Security Module **(HSM)**

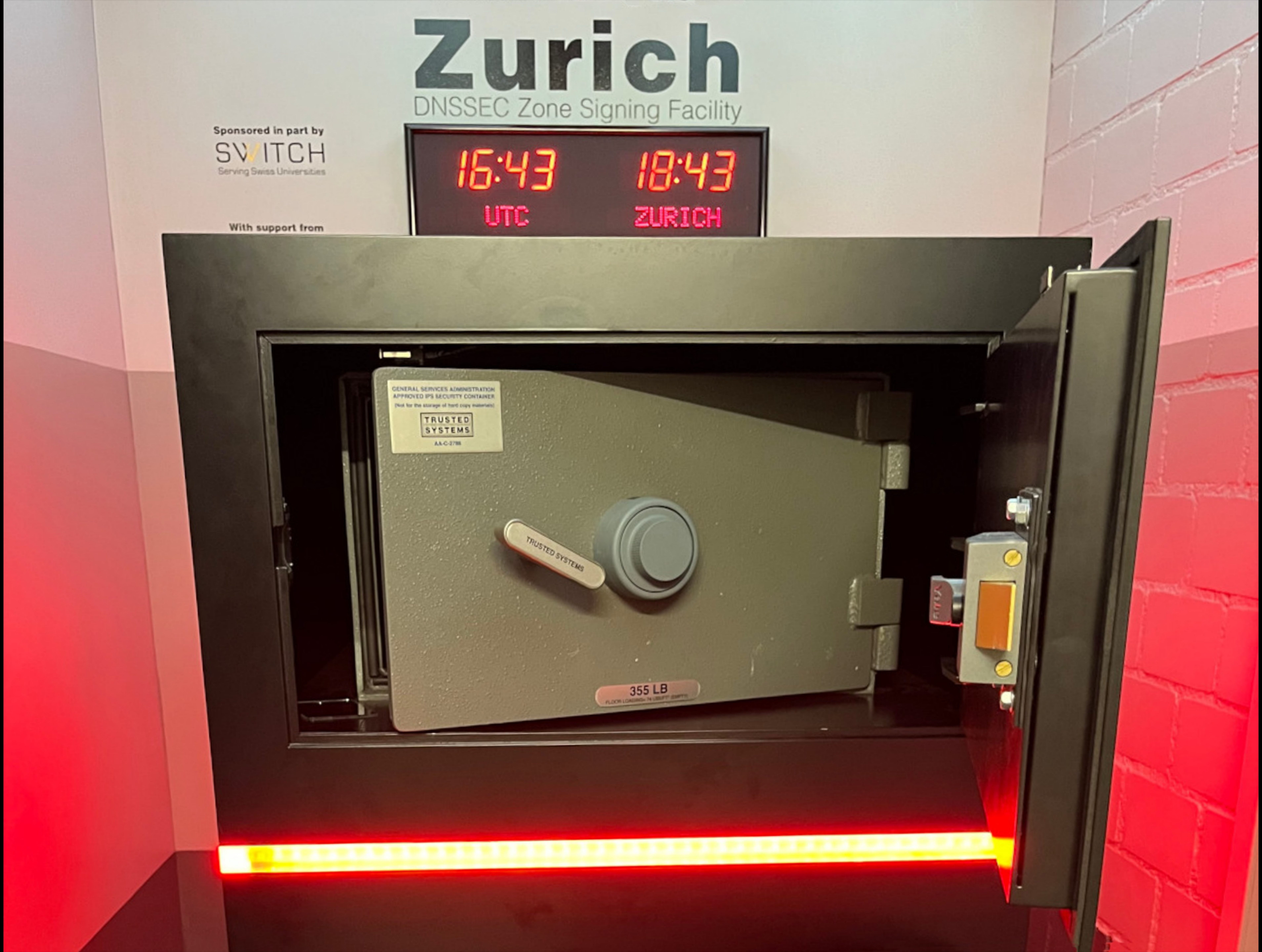
Government Services Administration Class-5
Information Processing System Security Container **(IPS)**

Enclosure meeting the physical specifications of DCI Directive 6/9
Section 4.3 Sensitive Compartmented Information Facility **(SCIF)**

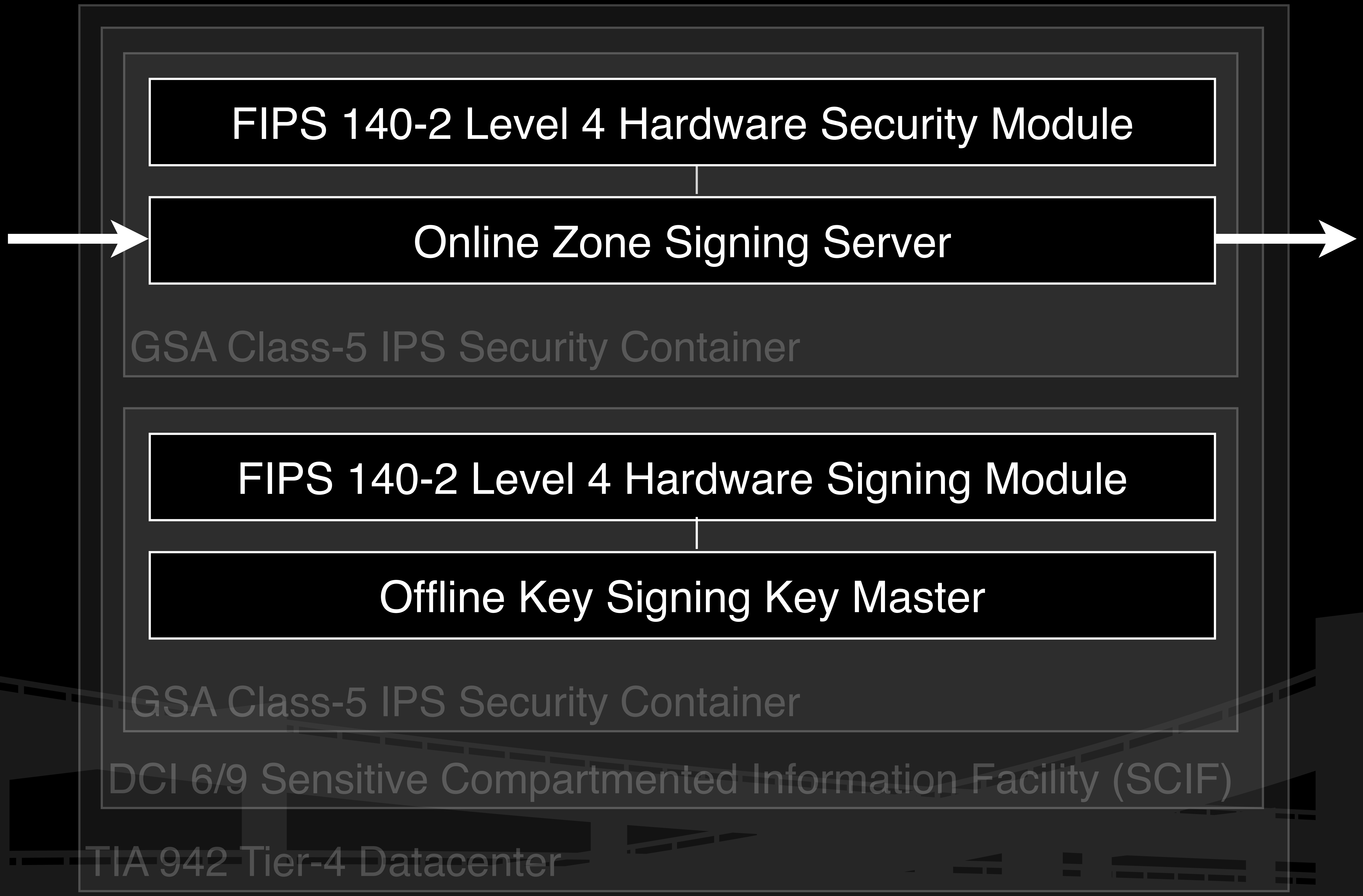
Vault hardened against forced or covert entry in
accordance with DCI 6/9 Sections 4.2 and 4.4 **(Vault)**

Building that meets or is substantively in accord with the relevant portions of the
Telecommunications Industry Association specification 942 of a Tier-4 datacenter. **(Datacenter)**

PCH
Packet Clearing House
Zurich
INOC-DBA


Zurich
DNSSEC Zone Signing Facility
Sponsored in part by
SWITCH
Serving Swiss Universities
With support from
16:43
UTC
18:43
ZURICH
GENERAL SERVICES ADMINISTRATION
APPROVED IPS SECURITY CONTAINER
(Not for the storage of hard copy materials)
TRUSTED SYSTEMS
AA-C-2786
TRUSTED SYSTEMS
355 LB

Packet Clearing House



FIPS 140-2 Level 4 Hardware Security Module

Online Zone Signing Server

GSA Class-5 IPS Security Container

FIPS 140-2 Level 4 Hardware Signing Module

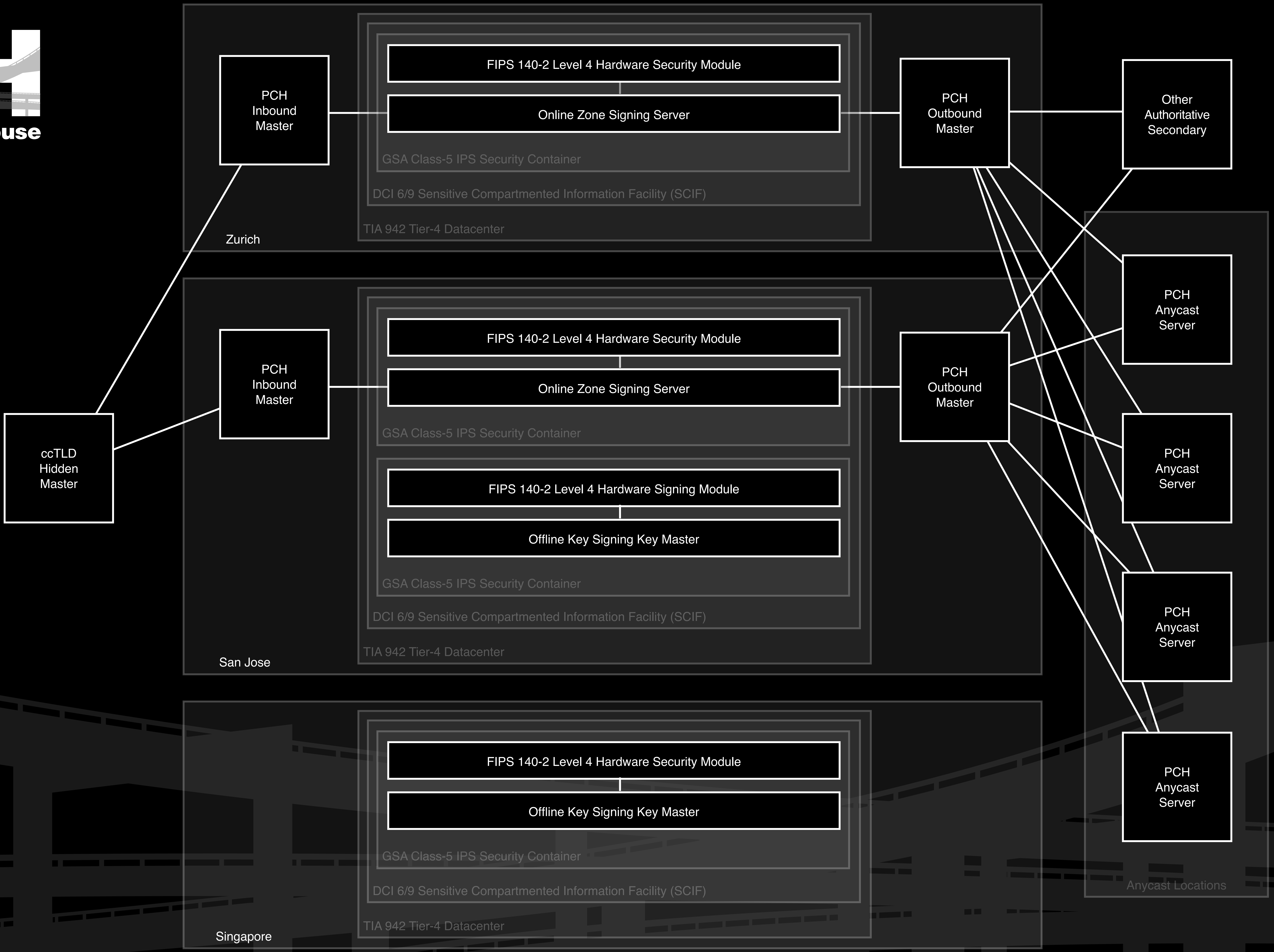
Offline Key Signing Key Master

GSA Class-5 IPS Security Container

DCI 6/9 Sensitive Compartmented Information Facility (SCIF)

TIA 942 Tier-4 Datacenter

PCH
Packet Clearing House



Zurich

TIA 942 Tier-4 Datacenter

DCI 6/9 Sensitive Compartmented Information Facility (SCIF)

GSA Class-5 IPS Security Container

FIPS 140-2 Level 4 Hardware Security Module

Online Zone Signing Server

PCH Inbound Master

PCH Outbound Master

Other Authoritative Secondary

San Jose

TIA 942 Tier-4 Datacenter

DCI 6/9 Sensitive Compartmented Information Facility (SCIF)

GSA Class-5 IPS Security Container

FIPS 140-2 Level 4 Hardware Security Module

Online Zone Signing Server

GSA Class-5 IPS Security Container

FIPS 140-2 Level 4 Hardware Signing Module

Offline Key Signing Key Master

PCH Inbound Master

PCH Outbound Master

ccTLD Hidden Master

Singapore

TIA 942 Tier-4 Datacenter

DCI 6/9 Sensitive Compartmented Information Facility (SCIF)

GSA Class-5 IPS Security Container

FIPS 140-2 Level 4 Hardware Security Module

Offline Key Signing Key Master

Anycast Locations

PCH Anycast Server

PCH Anycast Server

PCH Anycast Server

PCH Anycast Server

Key Ceremonies

One or two per year, either in San Jose or Singapore

Role separation between physical security and software security

Witnesses, notary, other specified roles

KSKs are held exclusively offline

ZSKs and associated RRSIGs generated two years in advance

All key ceremonies are streamed live

Video from three points of view is publicly archived and available online

Output from Key Ceremonies

in encrypted/wrapped format:

- ZSK - files that can be imported into the signer HSMs only
  - they are transferred via offline mechanism to signer (usb stick)
- KSK - files for backup purposes

unencrypted:

- DNSKEY records (the public key records for each KSK + ZSK)
- RRSIG records next to the DNSKEY rrset
- this means that key usage is set in stone during the KC

Output from Key Ceremonies

```
pch.net. 86400 DNSKEY 257 3 8 AwEAAbI5uSB9CnMQP+gSkMjdjuZZHsBr7KE62/il8Pf5Q+1L95Kolz...
; CKA_LABEL K200667 TAG 27692
pch.net. 86400 DNSKEY 256 3 8 AwEAAZHCPVHEvpMrioMKF2WwDGFB98FQzZ2I74o3qbRMauemU8iECQ...
; CKA_LABEL Z220724 TAG 27078
pch.net. 86400 DNSKEY 256 3 8 AwEAAZrHtChsOZZH4uHOh429F7vJLqYAL5W73QliIpQ2toolaGTgj1...
; CKA_LABEL Z219741 TAG 61150
pch.net. 86400 IN RRSIG DNSKEY 8 2 86400 20240217120000 20240131120000 27692 pch.net. I8...
; CKA_LABEL K200667:aep.hsmconfig:1 tag:27692 20230127203914
; pch.net. IN DS 27692 8 1 88BCBCEA9D77CC8D55E0F3EAD07A680B4913A37B
; pch.net. IN DS 27692 8 2 980AB73FF1943DBC1E4235D27FB1C828376AE6AA95F67A255E768D9CD859809A
; USE:Z219741 RSASHA256 1468800 864000 7884000 86400
```

First Generation Signer

Built in 2010 by Rick Lamb, Bob Arasmith, and Robert Martin-Legene

Custom tools built in C, bash and Perl

- `pkcs11-backup` (HSM interactions)

Calling low level BIND 9 tools:

- `dnssec-signzone`
- `dnssec-keyfromlabel`

Signer building blocks

`dnsxnotify`

- receives notifies and invokes the below steps

signing steps

- `donotify` fetches the zone (with `dig`)
- `zsign` works with "keybundles"
- concatenates zone with DNSKEY + RRSIGs
- imports keys into the hsm (if necessary)
- signs zones via an hsm handle with `dnssec-signzone`

Status and Direction

No general errors or failures thus far.

Original software environment had very static goal-oriented construction, rather than implementing all possible features.

Each operation of the key ceremony is planned and scripted in advance, and the sequence of the key ceremony is centered around efficient use of the crypto officers’ time in the vault.

In a mission critical environment no one wanted to deal with the low level operations and development of these tools anymore. We should not reinvent the wheel if there are better tools available that are proven to work.

Status and Direction

DNSSEC best practices have evolved since 2010:

- KASP, CDS, ZONEMD
- NSEC3 recommendations per RFC 9276
- elliptic curve algorithms have become more predominant

We needed more resilience and better support for key rolls (including algorithm rolls) than our original software gave us.

We also wanted incremental signing capability instead of signing the full zone every time.

knot Based Signer

Implements KASP (Key And Signing Policy) to schedule key usage and rotation instead of relying on diverse configuration files

knot includes tools to query and change KASP settings

Has more preexisting safety features

After some format tweaks, knot offline KSK support can make use of the output of our key ceremonies

knot offline-ksk functionality

2 knot environments: one KSKs and one ZSKs.

KSK is needed only to generate the RRSIGs for the DNSKEY rrsets.

The workflow is that each "side" generates its own keys and the ZSK side submits a request to the KSK side which gives the DNSKEYs and RRSIG back (for each period).

Packet Clearing House

New Tools & Integration

Our KC process (almost) produces what the ZSK side expects.
Mostly needed to adapt keybundles (DNSKEY+RRSIGs) format.

1) Process the keybundle
2) Import ZSKs into the HSM (all in advance)
3) Set key usage timings (KASP)
4) Import offline rrsigs into knot (after conversion)

knot - KeySigningRequest

```
;; KeySigningRequest 1.0 1736530108 (2025-01-10T17:28:28Z) ========
pch.net.            3600 DNSKEY 256 3 8 (
                         AwEAAbVvalQtERyfg55JND0qtvVIdSaQezNnGiIY
                         ...
                    ) ; ZSK, RSASHA256 (2048b), id = 16630
pch.net.            3600 DNSKEY 256 3 8 (
                         AwEAAely2j/NiaoLfQAQR4natd6sAtAMb4XYD0iw
                         ...
                    ) ; ZSK, RSASHA256 (2048b), id = 35308
;; KeySigningRequest 1.0 generated at 2024-01-10T17:28:28Z \
 by Knot DNS 3.3.3
```

knot - SignedKeyResponse

```
;; SignedKeyResponse 1.0 1704907708 (2024-01-10T17:28:28Z) =======
pch.net.            3600 DNSKEY 256 3 8 (
                         AwEAAbVvalQtERyfg55JND0qtvVIdSaQezNnGi...
                    ) ; ZSK, RSASHA256 (2048b), id = 16630
pch.net.            3600 DNSKEY 256 3 8 (
                         AwEAAely2j/NiaoLfQAQR4natd6sAtAMb4XYD0...
                    ) ; ZSK, RSASHA256 (2048b), id = 35308
pch.net.            3600 DNSKEY 257 3 8 (
                         AwEAAaHyVCiZyhLIuMtZ1q3Znjz6xDVNjaCuuX...
                    ) ; KSK, RSASHA256 (2048b), id = 46162
pch.net.            3600 RRSIG DNSKEY 8 1 3600 20240124172828 (
                         20240110155828 46162 hu. B/rZjC89pbu8C...
                    )
;; SignedKeyResponse 1.0 generated at 2024-01-10T17:28:51Z \
by Knot DNS 3.3.3
```

keybundles (reminder)

```
pch.net. 86400 DNSKEY 257 3 8 AwEAAbI5uSB9CnMQP+gSkMjdjuZZHsBr7KE62/il8Pf5Q+lL95Kolz...
; CKA_LABEL K200667 TAG 27692
pch.net. 86400 DNSKEY 256 3 8 AwEAAZHCPVHEvpMrioMKF2WwDGFB98FQzZ2I74o3qbRMauemU8iECQ...
; CKA_LABEL Z220724 TAG 27078
pch.net. 86400 DNSKEY 256 3 8 AwEAAZrHtChsOZZH4uHOh429F7vJLqYAL5W73QliIpQ2toolaGTgj1...
; CKA_LABEL Z219741 TAG 61150
pch.net. 86400 IN RRSIG DNSKEY 8 2 86400 20240217120000 20240131120000 27692 pch.net. I8...
; CKA_LABEL K200667:aep.hsmconfig:1 tag:27692 20230127203914
; pch.net. IN DS 27692 8 1 88BCBCEA9D77CC8D55E0F3EAD07A680B4913A37B
; pch.net. IN DS 27692 8 2 980AB73FF1943DBC1E4235D27FB1C828376AE6AA95F67A255E768D9CD859809A
; USE:Z219741 RSASHA256 1468800 864000 7884000 86400
```

More on knot

Practical and uptodate html based documentation.

Also complete manpages.

The developers are nice and helpful when dealing with requests for fixes or improvements.

bind9 future?

It seems that bind9 is also gaining this functionality:

https://gitlab.isc.org/isc-projects/bind9/-/issues/1128
Implement offline key support.
"All the things that need to be fixed before 9.20" reading on gitlab.

This is good because we want to maintain software diversity.

Further Enhancements
nsd Zone Verification (v4.6.0)

When NSD loads a new version of a zone, it can now run a validity-verification script against it before deciding whether to bring the new version live, replacing the old version.

Further Enhancements
nsd Zone Verification (v4.6.0)

Our script uses Perl with Net::DNS to check:

Validate DNSSEC chain from parent
to guard against both operational errors or software bugs.

If DS record present

- Checks if DNSKEY rrset is signed by the key defined by DS
- Checks if NS+SOA are properly signed by any of the keys from the DNSKEY rrset

Prevent bad data propagation if any of the checks fail

Sample Output
nsd verification

```
nsd[459]: verify: started verifier for zone pch.net (pid 163)
nsd[459]: + /usr/local/bin/check-dnskey-rrsigs-new.pl pch.net
127.0.0.1 5347
nsd[459]: dnskey_referenced_from_ds_found_in_dnskey_rrset
nsd[459]: dnskey rrset with valid signature
nsd[459]:  61150 RSASHA256 (2048) ZSK
nsd[459]:  27692 RSASHA256 (2048) KSK
nsd[459]: soa signature valid
nsd[459]: ns signature valid
nsd[459]: verify: verifier for zone pch.net (pid 163) exited with 0
nsd[458]: zone pch.net serial 2024010600 is updated to 2024010800
```

Further Enhancements
nsd Zone Verification

Related to a recent accident

Check the next time an RRSIG expires

Notify operators in case expiry is sooner than threshold.

```
while(<>) {
  my @F = split(' ');
  next unless m/IN\sRRSIG/;
  next unless $smallest_rrsig_exp > $F[8]
  $smallest_rrsig_exp = $F[8];
} # then calculate how far that day is...
```

Sample Output
nsd checking rrsig expiry

```
nsd[955518]:
[  pch.net] soonest RRSIG expiry is in 13 days
```

Future Work

The Keyper HSM will no longer be supported after March 2027, so we are now evaluating other FIPS 140-2 Level 4 hardware signing modules.

Trustworthy Elliptic Curve key options are needed.

Other capabilities by nsd verifier?

Offering Multi Signer for interested parties.

Thanks, and Questions?

https://pch.net

Tamás Csillag
DNS Services Engineer
Packet Clearing House
tom@pch.net