Threshold Cryptography lab test

Kevin Gong/ BII

Email: dbgong@biigroup.cn

2020-03-29 / Beijing / Yeti DNS Workshop

Bii
天地互连

Outline

- Purpose
- Topology
- Procedures
- Conclusion

Background

- RFC 8483 Yeti DNS testbed
  - Multi-signer: model 1/Three DMs: Unique KSK, multiple ZSKs, PINZ
  - Large DNSKEY RRSET size: more ZSKs
  - Zone transfer issue: AXFR/IXFR
    - Mixed RRSIGs from different DM
  - Operational issue
    - More Dms add possibility of failure points

Purpose

- Study the technology for Fault-tolerant Distribution Master Architecture
- Threshold Cryptography
  - Reduce DNSKEY RRSET
  - More DM operators
- Available Threshold Cryptography library test

天地互连

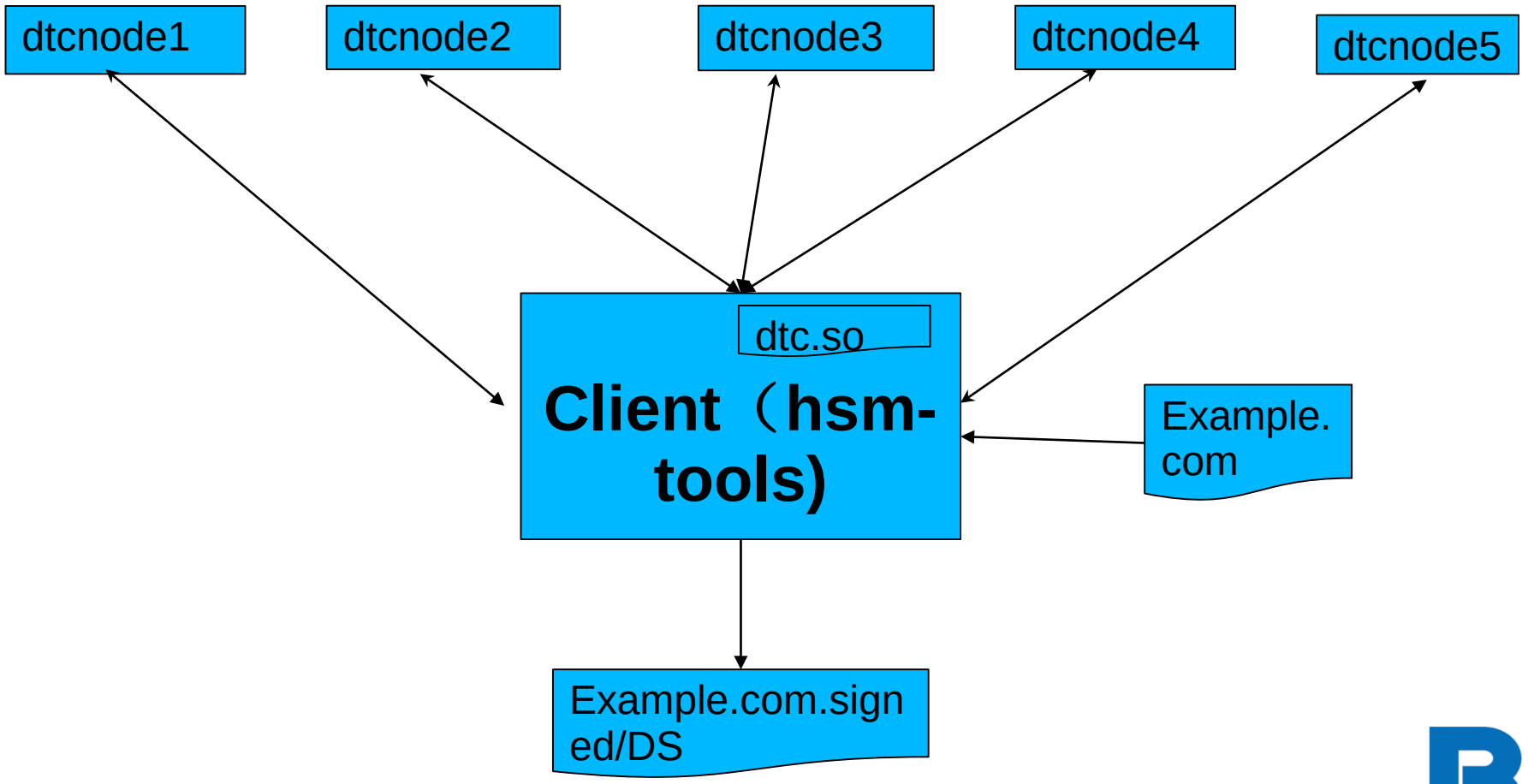
Threshold cryptography

- Practical Threshold Signatures
  - TCHSM
    - NIC Chile Research Labs
    - TCRSA
    - TCECDSA 2019
- Distributed Threshold cryptography: a novel way to distribute keys and signing for DNSSEC （Hugo/Tokyo 2019 Yeti DNS workshop)

TCHSM components

- TCRSA/TCECDSA
- DTC
  - PKCS11-Compatible Distributed Threshold Cryptography Library Signer
- DTCnode
  - ZMQ Node for PKCS11-Compatible DTC Library
- HSM-tools/OpenDNSSEC
  - a set of tools for zone digest and DNSSEC zone signer

Topology


dtcnode1
dtcnode2
dtcnode3
dtcnode4
dtcnode5
dtc.so
Client（hsm-tools)
Example.com
Example.com.signed/DS

Procedures

1. Compile the Library: dtc.so
   - Git,tar,wget,libzmq3-dev libczmq-dev,gcc
   - sqlite3 (used in HSM data storage)
   - Go (1.13.4 or higher)
   - sudo apt install libzmq3-dev libczmq-dev build-essential sqlite3 pkg-config git tar wget
   - git clone https://github.com/niclabs/dtc
   - cd dtc &&./build.sh
2. DTCnode
   - git clone https://github.com/niclabs/dtcnode.git
   - cd dtcnode && go build
   - Quick Node Deployment(docker)

天地互连

Procedures

1. DTCnode(continue)

- Quick Node Deployment(docker)

`./docker/test.sh`

This deploys five nodes and creates configuration files using a threshold value of 3.

The deployed nodes are listening on the following local ports:

- 9871
- 9872
- 9873
- 9874
- 9875

Then, you should copy the generated `config.yaml` file in `/etc/dtc/config.yaml` . on your testing machine. The database file is going to be by default in `/tmp/dtc.sqlite3` , so it is not recommended for persistent testing.

When you want to stop the nodes, you can simply stop the `./docker/test.sh` script.

天地互连

Procedures

1. Compile the HSM-tools
   - git,gcc,Go
   - sudo apt install build-essential pkg-config git
   - git clone https://github.com/niclabs/hsm-tools --branch v1.1.0
   - cd hsm-tools && go build
2. How to sign a zone
   - ./hsm-tools sign pkcs11 -p ./dtc.so -f ./example.com -3 -z example.com -o example.com.signed -c
3. How to verify a zone
   - ./hsm-tools verify -f ./example.com.signed

Procedures

4. How to delete PKCS11 keys

- ./hsm-tools reset-pkcs11-keys -p ./dtc.so

Issues

- HSM-tools
  - Zone file do not support ‘@’
    - Feature or bug?

Conclusion

- TCHSM works well so far
- Future work
  - More dtcnode
  - Simulate latency
  - Performance test
  - Key rollover?
  - use it in Yeti2

Questions?

Thank you