IETF Hackathon Secure Channel for CDNI delegation

IETF 103

3-4 November, 2018

Bangkok

Secure Channel for CDNI delegation: IETF 103 Hackathon

- What we aim to build is:
  - An open standards based keyless-SSL technique that:
    - Protects security credentials by way of a secure channel between any edge server and the key server
    - Decouples operations associated to these credentials into specific cryptographic services

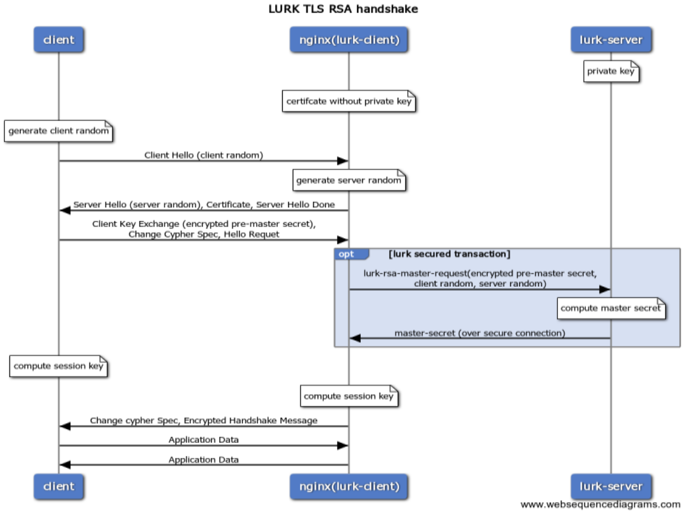
- Relevant Internet Drafts:
  - https://datatracker.ietf.org/doc/draft-mglt-lurk-lurk/
  - https://datatracker.ietf.org/doc/draft-mglt-lurk-tls12/
  - https://tools.ietf.org/pdf/draft-mglt-lurk-tls13-00.pdf

- Where this effort aims to help?
  - Delegation of streaming video sessions
    - without sharing of private keys between a delegating entity and the delegate
    - Protects CDNs own IP around keyless SSL but offers use of standards-based Interconnection when delegating across CDNs

What we are doing? And, yet to do… IETF 103 Hackathon

Current:

- Working on the handshake between LURK client (edge server) and the key server



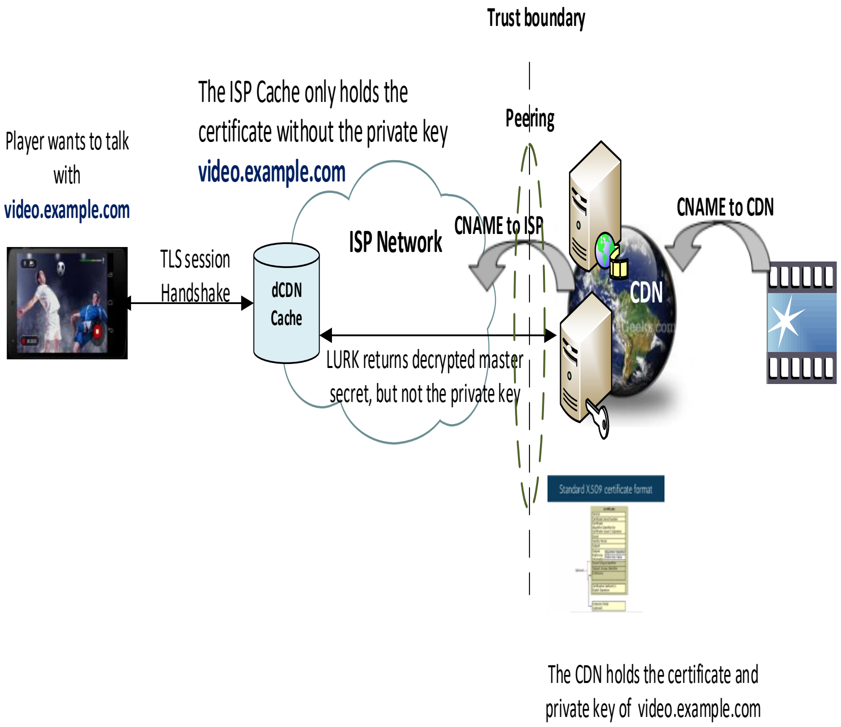
LURK TLS RSA handshake
client
nginx(lurk-client)
lurk-server
private key
certifcate without private key
generate client random
Client Hello (client random)
generate server random
Server Hello (server random), Certificate, Server Hello Done
Client Key Exchange (encrypted pre-master secret), Change Cypher Spec, Hello Requet
opt
[lurk secured transaction]
lurk-rsa-master-request(encrypted pre-master secret, client random, server random)
compute master secret
master-secret (over secure connection)
compute session key
compute session key
Change cypher Spec, Encrypted Handshake Message
Application Data
Application Data
client
nginx(lurk-client)
lurk-server
www.websequencediagrams.com

To Do:

- Specify the PRF hash function as a parameter
- Defines the LURK capabilities exchange so the LURK client knows what the LURK server is able to provide
- Introduce Multithreading to LURK Server
- Define additional transport (TCP, HTTPS, UDP/DTLS, TCP/TLS)
- A Go implementation (besides cLURK and pyLURK
- Complete implementation in TLS 1.2

Team Members
IETF 103 Hackathon

- Daniel Migault (Daniel.Migault@ericsson.com)
- Sanjay Mishra (sanjay.Mishra@verizon.com)
- Ori Finkelman (orif@qwilt.com)
- Dmitry Kravkov (dmitryk@qwilt.com)
- Frederic Fieau (Frederic.fieau@orange.com)
- Emile Stephane (emile.stephan@orange.com)
- Jesús Alberto Polo (ietf@jesusalberto.me)



Trust boundary

Peering

Player wants to talk with video.example.com

The ISP Cache only holds the certificate without the private key video.example.com

ISP Network

CNAME to ISP

CNAME to CDN

TLS session Handshake

dCDN Cache

CDN

LURK returns decrypted master secret, but not the private key

Standard X509 certificate format

The CDN holds the certificate and private key of video.example.com