

PQ Authentication @ Netflix

Ian McDiarmid-Sterling
IAB PQWS

This Talk

- Authentication for member devices; going PQ
- A multifunctional tool: crypto-agility
- What's missing ?

A Simplified Cryptographic Protocol™



The member device authenticates Netflix servers

Netflix servers authenticate the member device

- Fraud/piracy prevention
- Device partner rev. sharing
- Free trial enforcement
- Etc

A Simplified Cryptographic Protocol™



The member device authenticates Netflix servers

Netflix servers authenticate the member device

- Fraud/piracy prevention
- Device partner rev. sharing
- Free trial enforcement
- Etc

Not TLS
Shaped

A Simplified Cryptographic Protocol™



**Netflix Backend
Servers**

The member device authenticates Netflix servers

Netflix servers authenticate the member device

- Fraud/piracy prevention
- Device partner rev. sharing
- Free trial enforcement
- Etc

**Not TLS
Shaped**

Deep Dive: The Member Device Auths. Netflix Servers

Step 1: verify the expected code is running / the hardware measurement are correct with Secure Boot^{*}

Step 2: run our Simplified Cryptographic Protocol™

Deep dive: Protocol Goals

Establish:

- **Authenticity**: Is my TV talking to Netflix?
- **Confidentiality**: Can anyone else see what I'm watching?
- **Integrity**: Is this the show I asked for?

Deep dive: Protocol Goals

Establish:

- **Authenticity**: Is my TV talking to Netflix?
- **Confidentiality**: Can anyone else see what I'm watching?
- **Integrity**: Is this the show I asked for?

Deep Dive: Authenticity



Step #1: Send some
device-specific data



**Netflix Backend
Servers**

Step #2: Respond with a
signed bundle of assets

Step #3: Use
hardware public key
to **verify** signature

The Protocol Must Update to PQ-Auth

Member devices must be able to securely stream Netflix for their **expected lifespan**

Upgrade to PQ-Auth; Overview

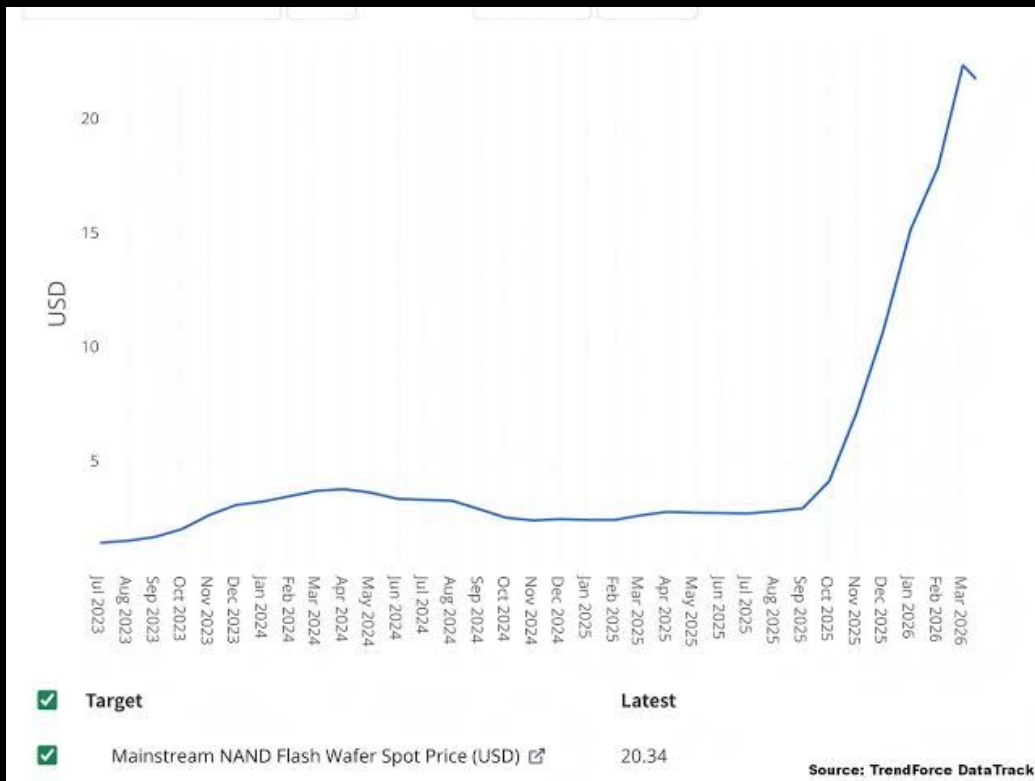
What do we need to do ?

- Provision PQ-Auth keys to member devices
- Support PQ-Auth in the protocol
- Support PQ-Auth in the Netflix backend

Deep Dive: Key Provisioning for Member Devices

PQ keys are **larger** - collaborate with ecosystem partners to address concerns around memory (DRAM) and storage (NAND Flash, eMMC) pressure

Deep Dive: Why Key Size Matters



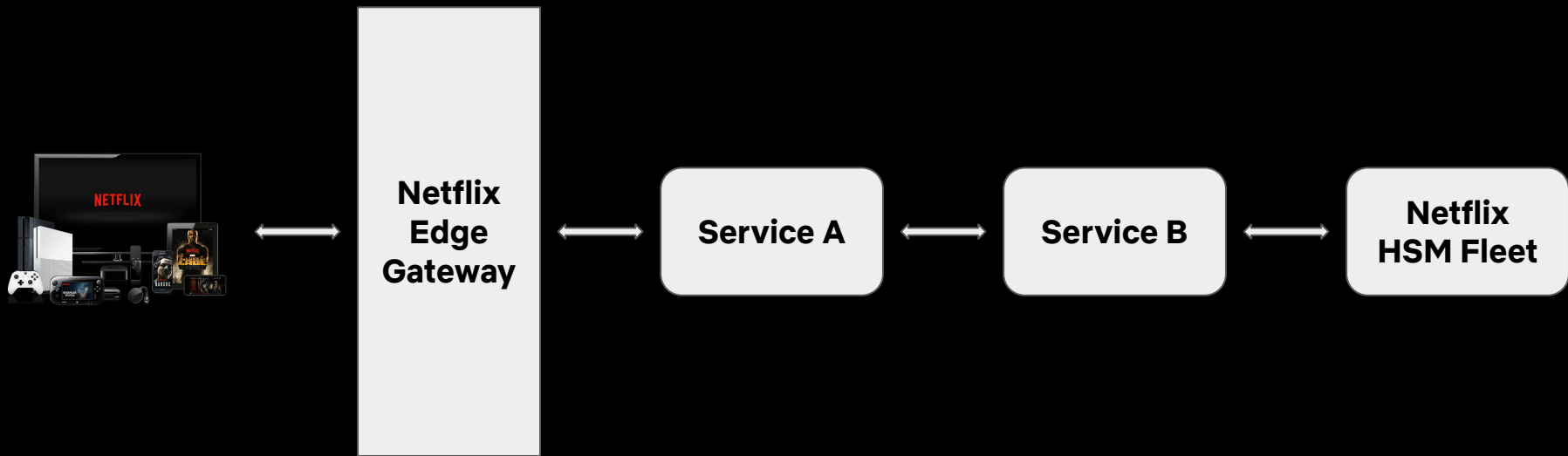
Deep Dive: Protocol Changes

Where do we need signature-based authentication vs. authentication based on key-derivation w/ AEAD encryption or HMAC schemes

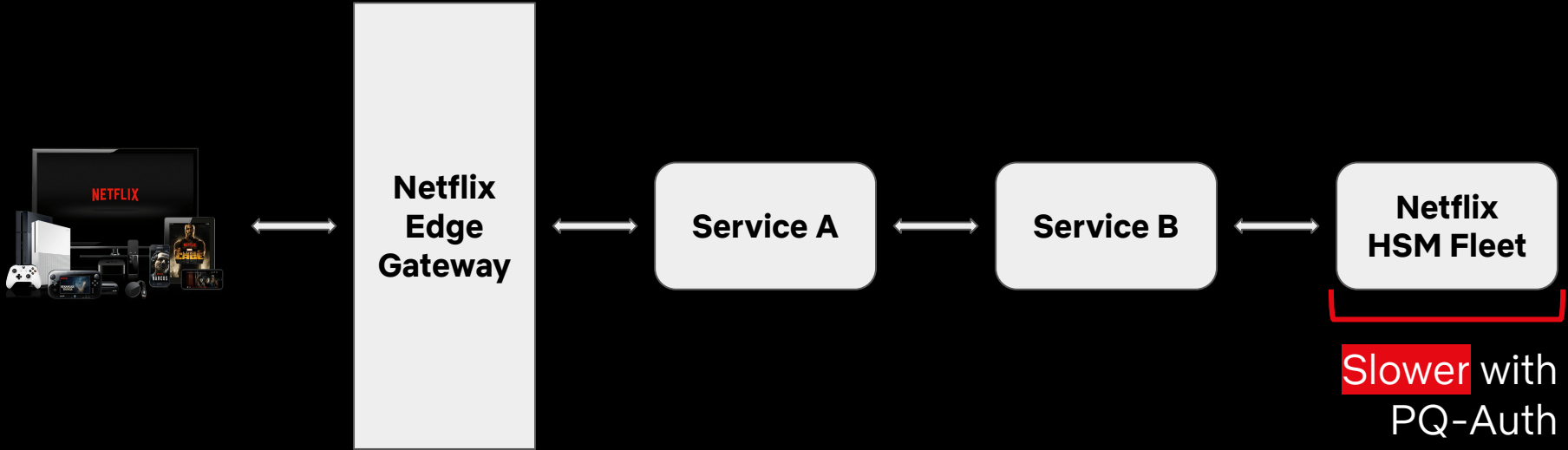
Deep Dive: Netflix Backend



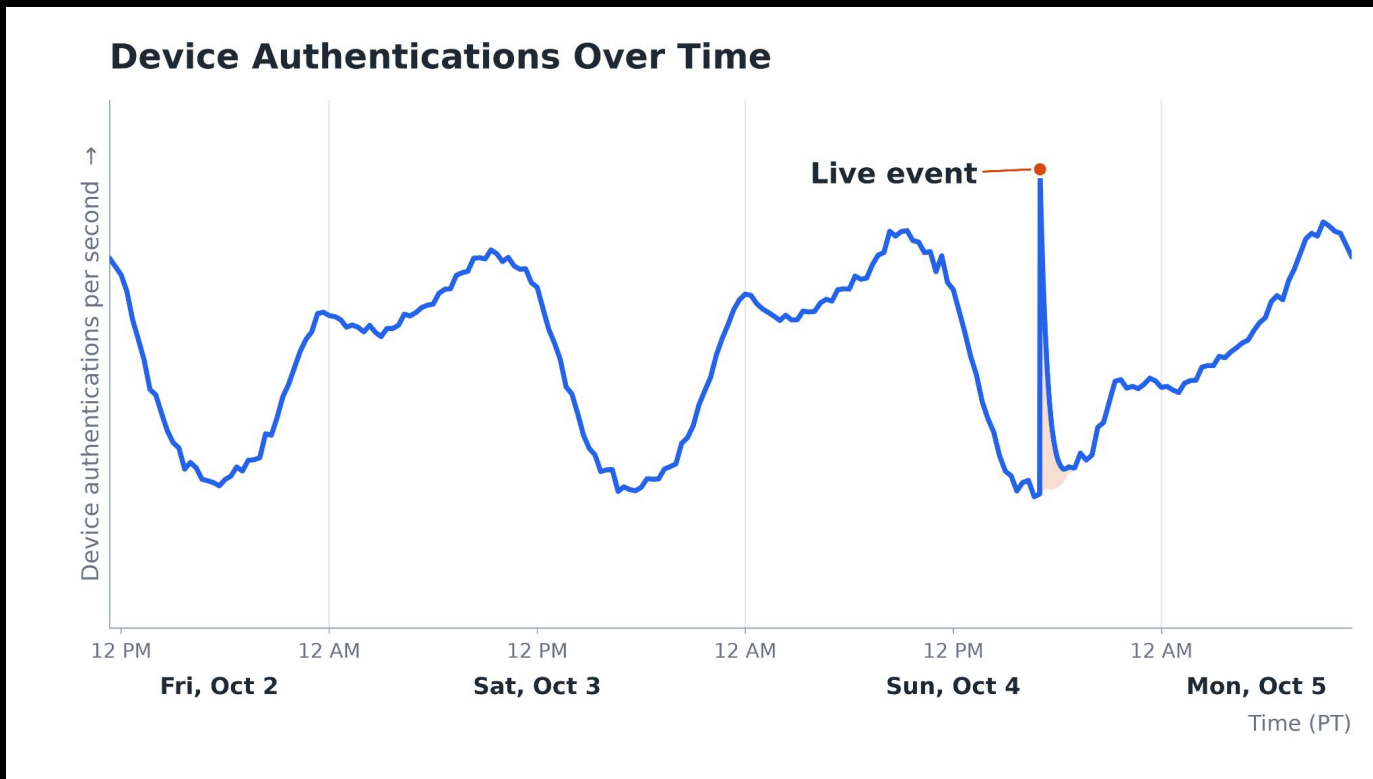
Deep Dive: Netflix Backend



Deep Dive: Netflix Backend



Deep Dive: Netflix Backend



Deep Dive: Netflix Backend

Experimentally validate the impact to the ecosystem of slower authentication operations (signing)

Improved capacity management reduces overall business cost without impacting the member experience

What about everything else?



Crypto Agility

Crypto Agility @ Netflix

- Inventory of current cryptographic usage
- Centralized implementations and management
- Shift-left techniques to change developer behavior
- TLS automation + shift to private trust*

Crypto-agility Case Study: JWT signing

Crypto-agility Case Study: JWT signing

Today

- Teams manage their own JWT signing with common libs
- Teams manage keys, select algos etc for their integrations
- Teams manage JWKS cache lifetimes, etc

Future

- Centrally managed JWT signing
- Provided key management, rotation, algo selection
- Teams manage JWKS cache lifetimes, etc

Deep Dive: JWT signing

Inventory flags JWT signing as a common ecosystem pattern lacking centralized support

Deep Dive: JWT signing

We build a centralized implementation to handle this use case and provide sensible defaults + opaque advanced functionality

Deep Dive: JWT signing

AI systems provide shift-left guidance directly to developers as they open pull requests

What's Missing ?

Ecosystem adoption of PQ in JOSE (+standardized test vectors ?)

Alignment between standards bodies; differing regulatory requirements

This slide intentionally left blank

Deep Dive: Secure Boot

