Redesigning PKI

To Solve Revocation, Expiration, & Rotation Problems

neustar

Brian Knopf @DoYouQA

WHO AM I

- Sr Director of Security Research & IoT Architect @Neustar
- @DoYouQA
- 20+ years in IT, QA, Dev & Security
- Home is IoT research lab with 130 devices
- Previously
  - CEO at BRK Security
  - Principal Security Advisor at Wink
  - Director of Application Security at Belkin & Linksys
  - Principal Test Architect, Office of the CTO at Rapid7
  - Director of QA at MySpace

AGENDA

1. Introduction
2. What happened?
3. 140 threat models and the breakthrough
4. Key Terms
5. Components of TDI
6. PKI vs TDI
7. The Workflow
8. The Demo
9. The Code

Introduction


WHAT IS TDI?

What Happened

WHAT HAPPENED

IoT worm can hack Philips Hue lightbulbs, spread across cities

Hacker Claims To Push Malicious Firmware Update to 3.2 Million Home Routers

Enterprises are racing to connect their products to the Internet. But without plugging the IoT vulnerabilities, they are risking their reputation, bottom line and customer data.

CIRCUIT BREAKER \ TECH \ CYBERSECURITY

Hacked webcams that helped shut down the internet last week are being recalled

Oops! 185,000-plus Wi-Fi cameras on the web with insecure admin panels

Just unplug them now before someone writes a botnet, okay?

Someone DDoSed A University Server By Hacking Its Vending Machines

The University's Internet connection was blocked using infected IoT devices including vending machines and bulbs.

US & WORLD \ TECH \ CYBERSECURITY

The CIA is hacking Samsung Smart TVs, according to WikiLeaks docs

by Russell Brandom | @russellbrandom | Mar 7, 2017, 10:14am EST

Survey: Cyber Attacks Against Smart City Services May Pose Public Safety Threat

Eighty-eight percent of state and local government IT professionals are concerned about cyber attacks targeting critical city infrastructure

AT&T

*…security risks are rising with the exponential growth of connected devices. The company alone has noted a **3,198% increase** in attackers prowling for vulnerabilities in IoT devices over the past three years.*

The Register®
Biting the hand that feeds IT

DATA CENTER SOFTWARE SECURITY TRANSFORMATION DEVOPS BUSINESS PERSONAL TECH

Security

12

Comodo admits 2 more resellers pwned in SSL cert hack

How d

By John L

Comodo
were hit

No forge
attack, b
compron

Comodo
Europe c
Windows
hours aft

CA Security Council

News About Us Get Educated Tools Blog Cont

Leading Certificate Authorities and Microsoft Introduce New Standards to Protect Consumers Online

- **Stronger protection for private keys:** The best practice will be to use a FIPS 140-2 Level 2 HSM or equivalent. Studies show that code signing attacks are split evenly between issuing to bad publishers and issuing to good publishers that unknowingly allow their keys to be compromised. That enables an attacker to sign malware stating it was published by a legitimate company. Therefore, companies must either store keys in hardware they keep on premise hardware, or in a new secure cloud-based code signing cloud-based service.
- **Certificate revocation:** Most likely, a revocation will be requested by a malware researcher or an application software supplier like Microsoft, if they discover users of their software may be installing suspect code or malware. After a CA receives request, it must either revoke the certificate within two days, or alert the requestor that it has launched an investigation.
- **Improved code signatures time-stamping:** CAs must now provide a time-stamping authority (TSA) and specifies the requirements for the TSA and the time-stamping certificates. Application software suppliers are encouraged to allow code signatures to stay valid for the length of the period of the time-stamp certificate. The standard allows for 135-month time-stamping certificates.

Microsoft will require CAs that issue code signing certificates for Windows platforms must adhere to these guidelines beginning on February 1, 2017.

theguardian

nion sports soccer tech arts lifestyle fashion business travel environment

DigiNotar SSL certificate hack amounts to cyberwar, says expert

Dutch government revokes certificates used for all its secure online transactions, while CIA, Google, Microsoft and others affected by hack called 'worse than Stuxnet'

International Contact Abonneren RSS Vacatures Sitemap Help

Mozilla Security Blog

MAR 23 2015

Revoking Trust in one CNNIC Intermediate Certificate

Kathleen Wilson

96 responses

Mozilla was recently notified that an intermediate certificate, which chains up to a root included in Mozilla's root store, was loaded into a firewall device that performed SSL man-in-the-middle (MITM) traffic management. It was then used, during the process of inspecting

engadget

IOActive®

Gear Gaming Entertainment Tomorrow Video Reviews Find a Product Buyer's Guide

NEWS Overview

Latest in Bricked

Wink smart home hubs knocked out by security certificate (update)

PRESS RELEASES

IOActive Lights Up Vulnerabilities for Over Half a Million Belkin WeMo Users

Popular home automation devices are wide open to attackers

Seattle, US — February 18, 2014 — IOActive, Inc., the leading global provider of specialist information security services, announced today that it has uncovered multiple vulnerabilities in Belkin WeMo Home Automation devices that could affect over half a million[1] users. Belkin's WeMo uses Wi-F control home electronics anywhere in the world direc

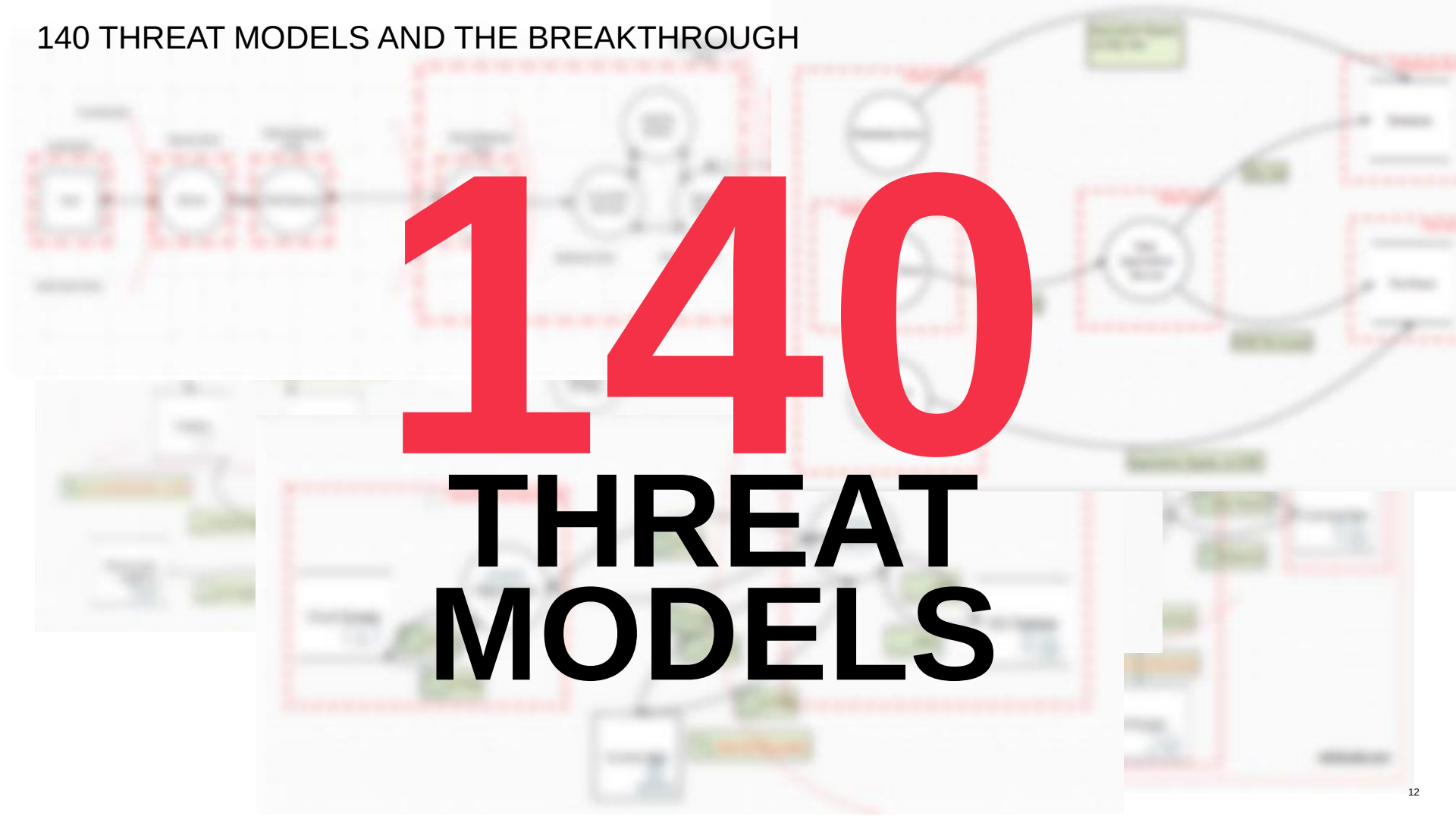
Mike Davis, IOActive's principal research scientist, u in the WeMo product set that gives attackers the abi

- Remotely control WeMo Home Automation attac
- Perform malicious firmware updates
- Remotely monitor the devices (in some cases)

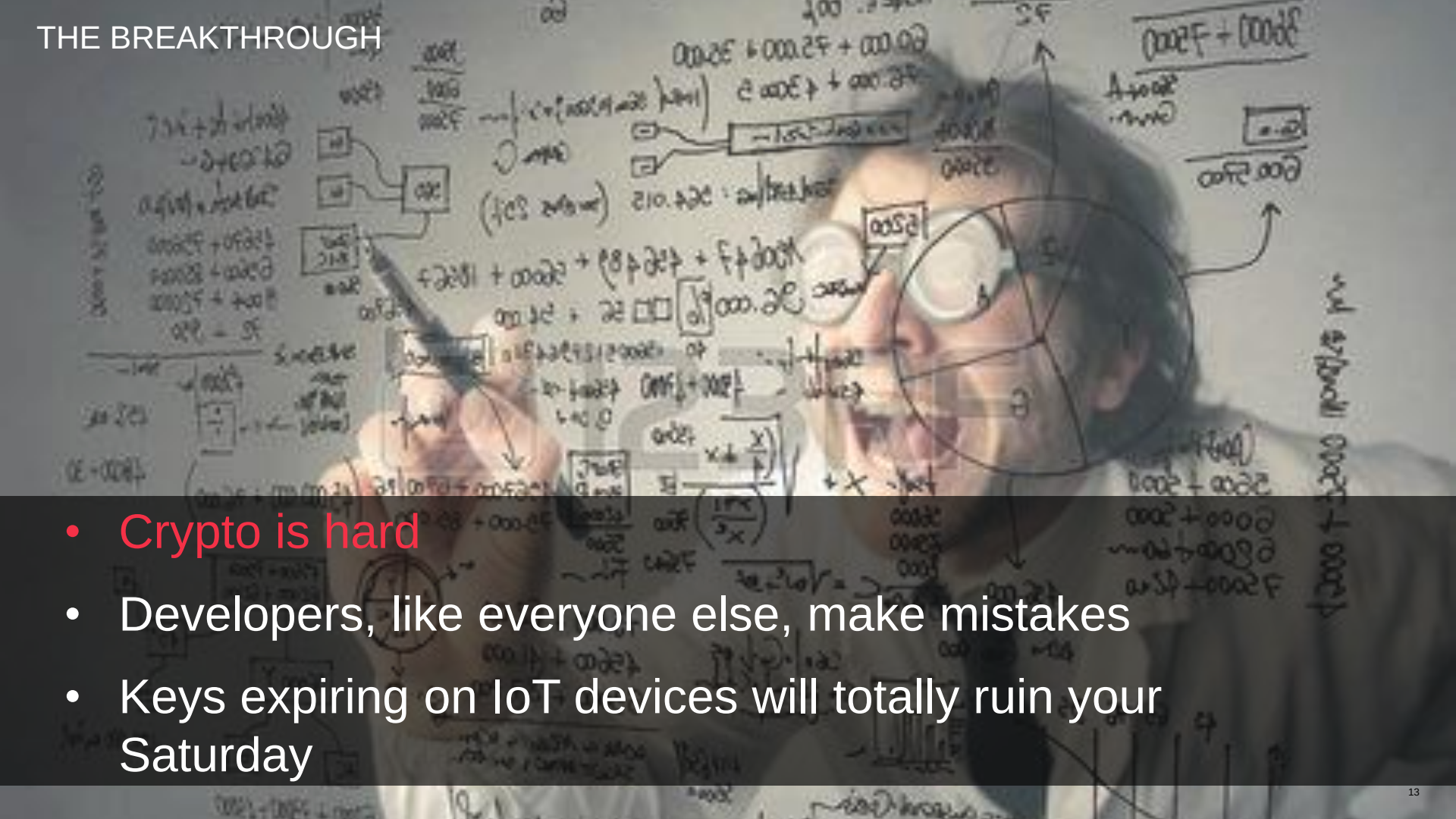
The Vulnerabilities

The Belkin WeMo firmware images that are used to update the devices are signed with public key encryption to protect against unauthorised modifications. However, the signing key and password are leaked on the firmware that is already installed on the devices. This allows attackers to use the same signing key and password to sign their own malicious firmware and bypass security checks during the firmware update process.

140 Threat Models and the Breakthrough


140
THREAT
MODELS

THE BREAKTHROUGH



- Crypto is hard
- Developers, like everyone else, make mistakes
- Keys expiring on IoT devices will totally ruin your Saturday

REQUIREMENTS FOR PKI REPLACEMENT

- NOC or SOC should be in control, not users/site managers
- Do NOT rely on the router & firewall as your security model
- Trust nothing unless proven otherwise… constantly
- Servers should not share keys for signing
- Revocation should be instant
- Key rotation should be easy & fast
- Keys should never expire unexpectedly
- Plan for complete failure

Key Terms

FLEET

A fleet defines the scope of the deployment. Typically, a Fleet is comprised of devices with a certain SKU and those devices' supporting services.

The public key of the fleet is recognized by all elements of the system and represents a Fleet's base authority and identity.

FLEET SERVER

The private fleet key is stored in an HSM and paired with a server(s) associated with that Fleet of devices. This server acts as a verification point of messages and can sign any messages directed to devices on behalf of the fleet.

CO-SIGNING SERVICE

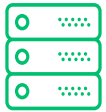
Messages receive a second signature from the TDI Co-Signing service to strengthen the integrity of the message and its authenticity. The TDI Co-Signing service retains its key pair as well as the public keys for the Fleet Server and Devices.

DEVICES

Devices are assigned to a fleet. They have the ability to create messages and sign with their private key. They also contain the public key of the fleet and of the co-signer to verify any messages being sent.

FLEET

Fleet PRIVATE key
Fleet PUBLIC key



FLEET SERVER

Server PRIVATE key
Server PUBLIC key

Fleet PRIVATE key
Device PUBLIC key
Co-Signer PUBLIC key

CO-SIGNING SERVICE

Co-signer PRIVATE key
Co-Signer PUBLIC key

Fleet Server PUBLIC key
Device PUBLIC key

DEVICE

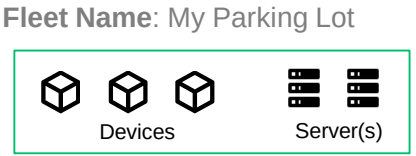
Device PRIVATE key
Device PUBLIC key

Fleet PUBLIC key
Co-Signer PUBLIC key
Reset PUBLIC keys

KEY CONCEPTS

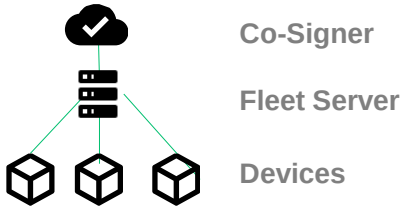
1. Devices and servers each have a unique key pair identity. We recommended generating the key based on a hardware root of trust (Arm Trust Zone or Intel TPM)

2. Devices and servers are assigned to a fleet



3. Fleet servers can verify trusted devices in the fleet, sign on behalf of the Fleet Signing Key, and request co-signature from the co-signing server using its Fleet Server Key.

   Devices send messages signed with their private keys which is validated by fleet server & co-signer



Components of TDI

REVIEW OF THE COMPONENTS OF TDI

TDI Co-Signing Engine

Hosts co-signing service, verifies devices and provides an aggregated view of devices, users & verification gateways

Open Source SDK

Client App

Regular application sending and receiving messages from devices

Open Source SDK

Fleet Server

Hosts Fleet signing service, manages & verifies devices assigned to this identity gateway

Open Source SDK

Gateway or Device Firmware/App

Identity Agent

Signs messages & verifies that messages are ok

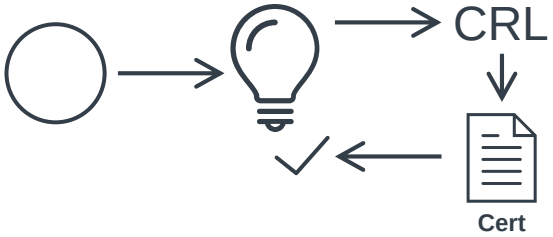
Open Source SDK

TECHNOLOGY

Cryptography	EC NSA Suite B, NIST p256 curve
Encryption	AES 256
APIs	RESTful
SDKs	Python, C, Node, Java
Communication Protocols	Agnostic
Device Requirements	Agnostic, <100KB + SSL library, 32-bit CPU, 32K RAM
Deployment	CLI and Reference Implementations
Hosting	Cloud (AWS), Private Cloud, On-Premise
Management	API hooks for NOC/SOC, CLI, Admin Portal

PKI vs TDI

PKI



In a typical PKI deployment an identity asserts itself to a recipient.

The recipient looks up the identity in CRL or via OCSP and then it’s own cert list to validate.

Burden on recipient!

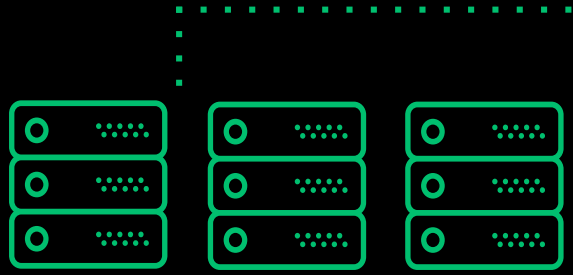
TDI

In a *TDI* deployment the Identity service validates identities and returns the validation to your application.

Only the validation is forwarded to the recipient.

Recipient does not need to do additional CRL/OCSP lookup.

PKI WITH CRL & SHARED SERVER KEYS



Device (x 500,000)

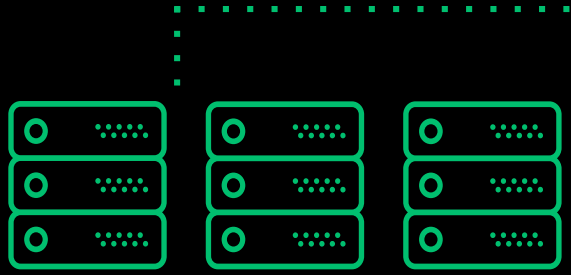
1. Single server key compromise causes all servers to be compromised

2. Compromised server keys are added to CRL

3. Devices have to lookup CRL to identify expired keys

4. Compromised keys need to be removed from every server and device that could communicate with them

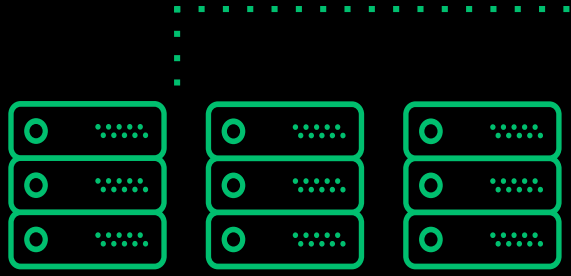
PKI WITH CRL & INDIVIDUAL SERVER KEYS



Device (x 500,000)

1. Only server(s) compromised need to be revoked
2. Compromised server keys are added to CRL
3. Devices have to lookup CRL to identify expired keys
4. Compromised keys need to be removed from every server and device that could communicate with them
5. Every device and server need an update with the public key every time a server is added or removed

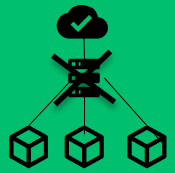
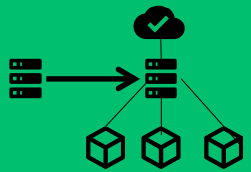
PKI WITH OCSP STAPLING & INDIVIDUAL SERVER KEYS



Device (x 500,000)

1. Only server(s) compromised need to be revoked
2. Compromised server keys no longer validated with OCSP
3. Devices don't have to lookup CRL, but still have public keys on them
4. Compromised keys still need to be removed from every server and device that could communicate with them
5. Every device and server still need an update with the public key every time a server is added or removed

SO WHAT DOES THIS GIVE US?

TDI 	Servers & devices can be instantly revoked.	**PKI**	Devices must check to see if server is revoked. Burden on end device for lookup.
TDI 	Devices only need to store and validate 2 keys to receive messages *(co-signer and fleet public keys)*	**PKI**	Devices must store a key(s) for each of the entities it is communicating with
TDI 	Servers can be quickly switched out without rekey of all devices because devices only respect the co-signer and fleet-signer keys	**PKI**	Potentially 1,000's of devices need to be re-keyed with the new server key

The Workflow

FLEET CREATION

Owner retrieves Fleet public key from HSM

HSM

1

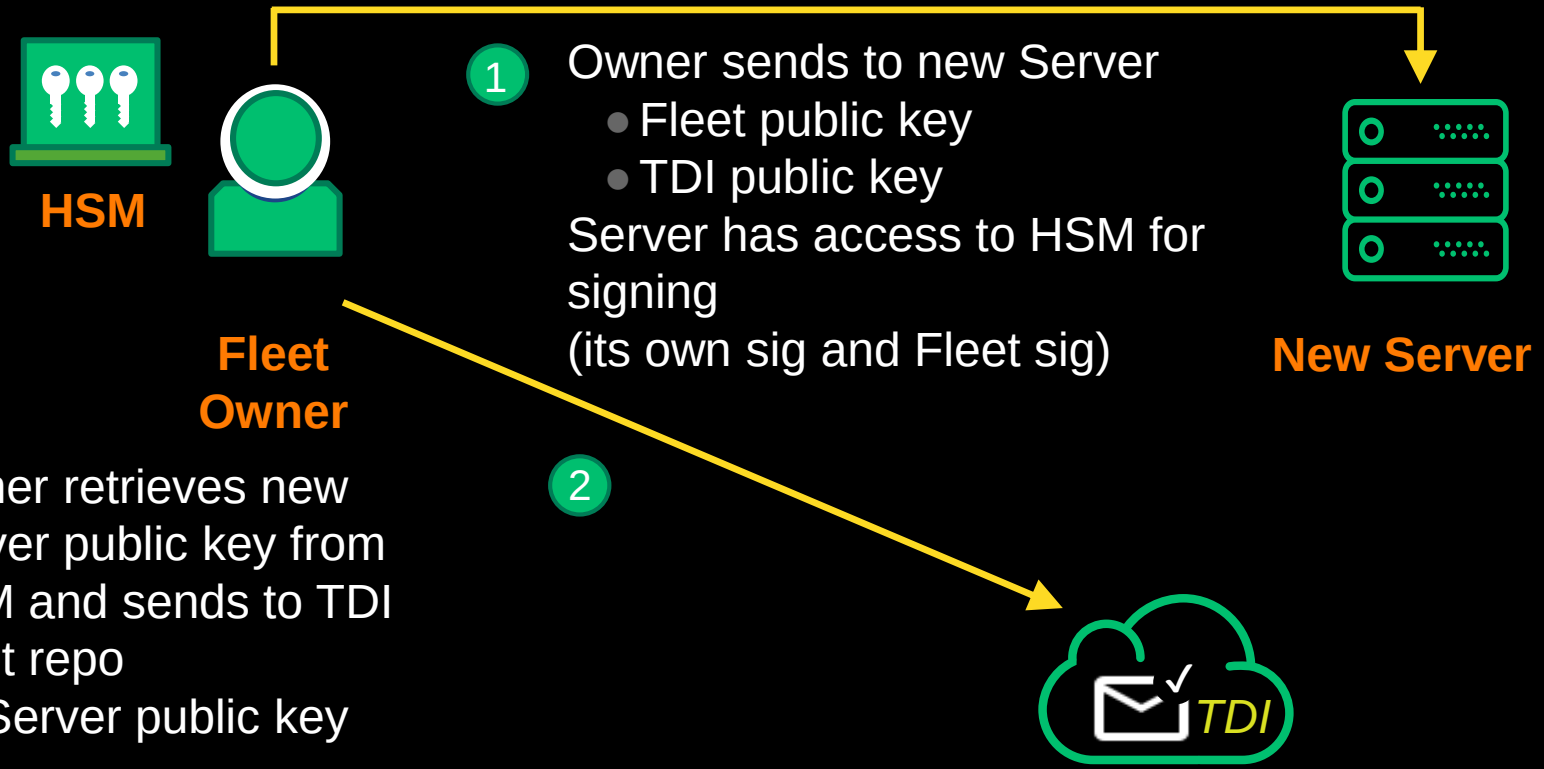
Fleet Owner

2

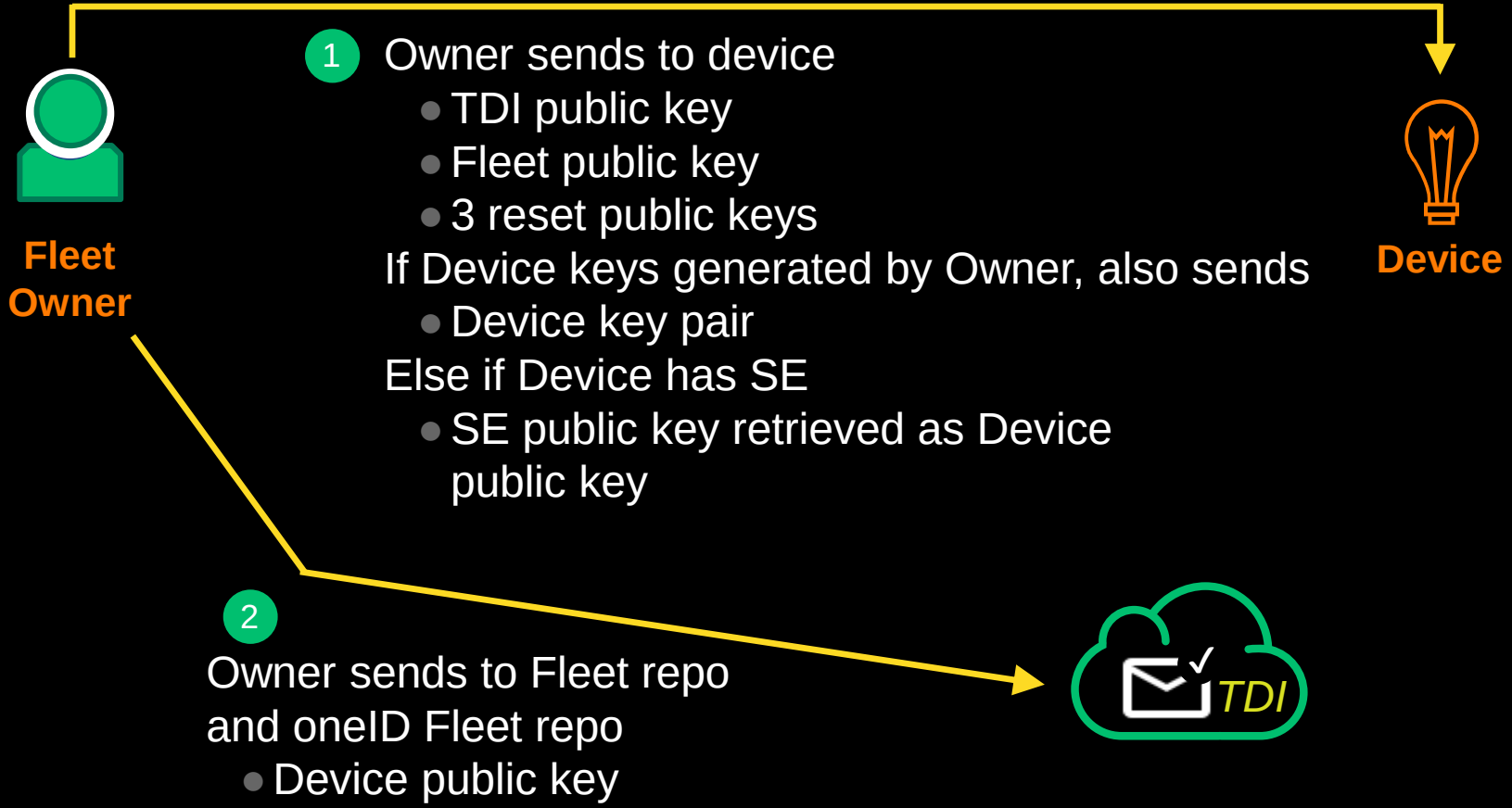
Owner creates a TDI admin account and creates a new Fleet in the TDI service (using public key from HSM)

3 TDI creates Fleet repo and TDI Fleet key pair

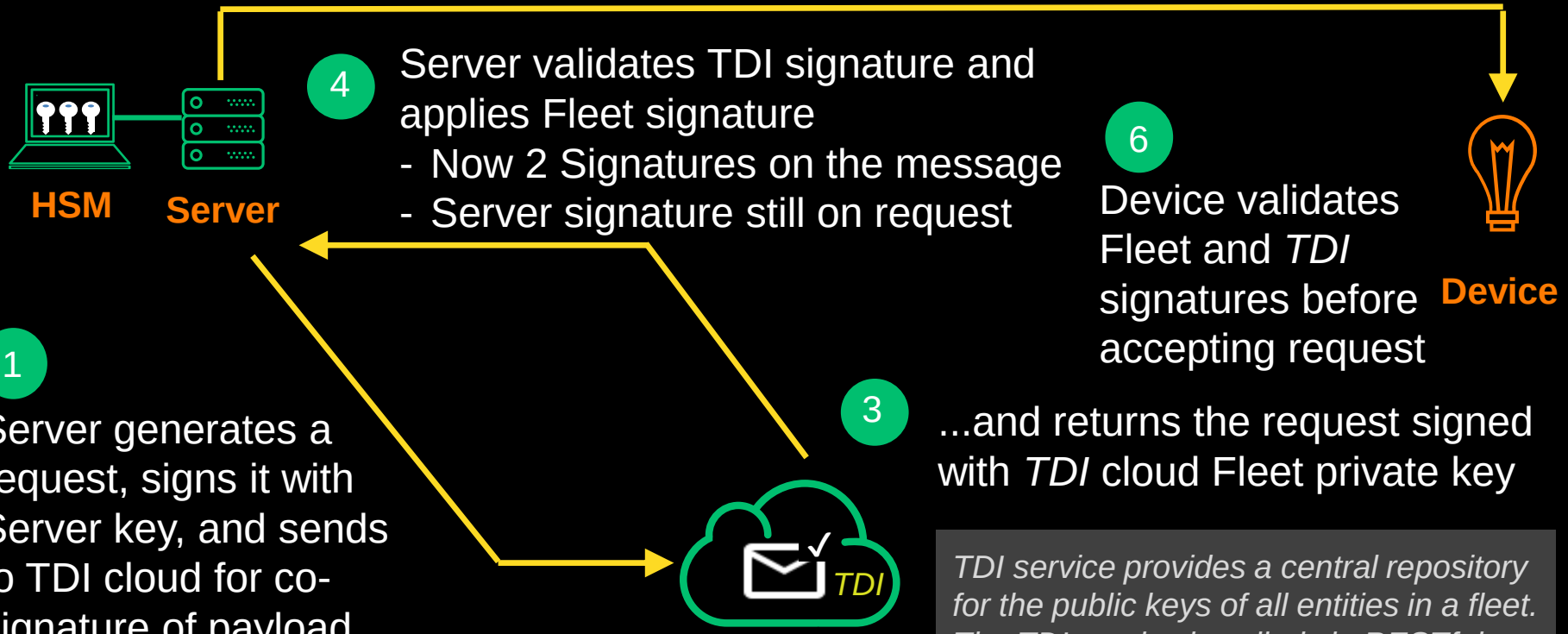
PROVISIONING SERVERS



PROVISIONING DEVICES

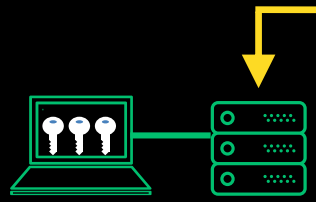


Server to Device Messaging



TDI service provides a central repository for the public keys of all entities in a fleet. The TDI service is called via RESTful APIs

Device to Server Messaging



HSM

Server

5 Server validates co-signer signature before accepting the message

1 Device generates a request and signs it with Device private key

Device

2 Server validates *Device Signature* with Fleet repo and forwards the request to co-signer service

4 ...and returns the request signed with TDI cloud Fleet private key

3 TDI service validates the device signature...

SIMPLIFIED MESSAGE FLOW: DEVICE SENDING A MESSAGE

Co-Signer

4

Verifies that neither the fleet server nor the device have been revoked

Signs the message

5

Sends the message back to the fleet server

Fleet Server

3

Verifies the signature against the public Device Verification Key.

Then sends the message to the Co-Signing Server to get an additional signature.

6

Verifies the co-signature key & Sends to destination app

App

2

App ingests the message and sends to the fleet server for validation and co-signatures

7

App does whatever it wants with the message

Device

1

The Device generates a message and signs the message with its private Device Signing Key.

SIMPLIFIED MESSAGE FLOW: APP SENDING A MESSAGE TO A DEVICE

Co-Signer

Fleet Server

App

Device

3 Verifies that neither the fleet server nor the app server have been revoked & signs the message

2 Fleet server verifies the app and sends to co-signature service

1 Message is generated by the app, signed with it's App/Device key, and sent to the fleet server – e.g. firmware

4 Sends the message back to the fleet server

5 Verifies the co-signature key and returns the dual signed message to the app

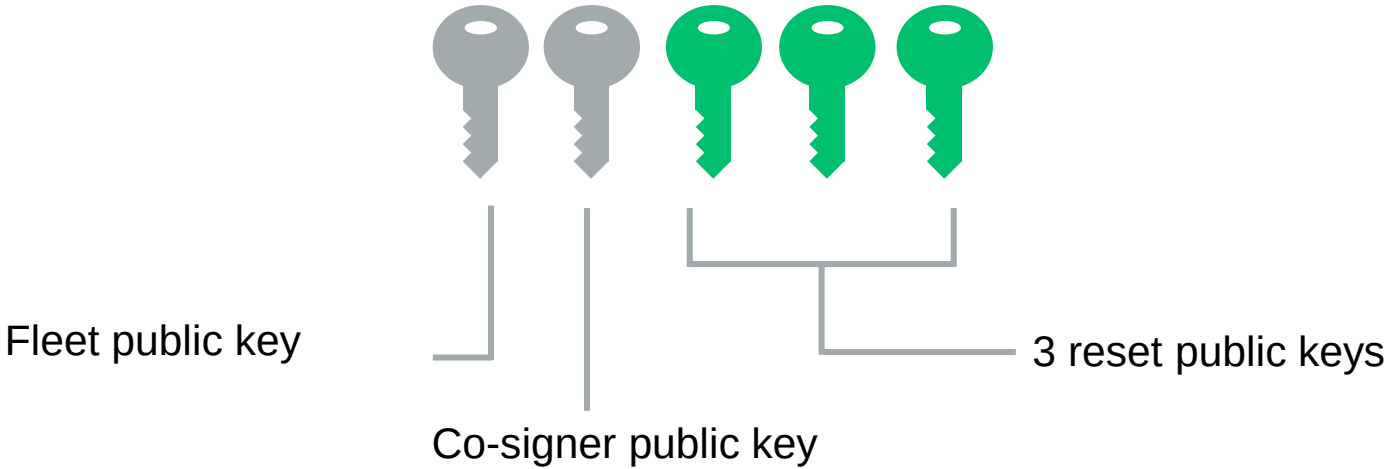
6 App sends message to device

7 Device verifies fleet signature & co-signer signature and carries out operation

TDI enables rapid breach recovery with minimal downtime

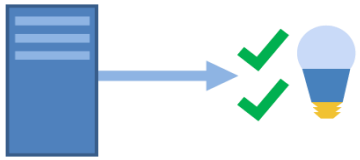
1. Admin revokes compromised Server(s) in their TDI fleet via CLI / dashboard or API
2. Admin provisions new Servers into the Fleet
3. On the next request the old servers are no longer valid and new servers are immediately validated by the TDI service
4. Recipients DO NOT require updates for servers being revoked or added

REKEY SCENARIO



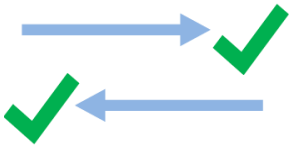
In the rare occurrence of needing to re-key, **3 reset public keys are stored with provisioned identities to enable backup MFA.** This means that both the fleet and co-signer private keys can be compromised and recovery can still be done remotely.

ROBUST AUTHENTICATION



AUTOMATED MFA

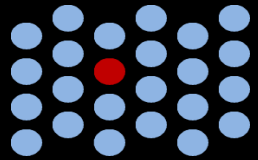
servers, devices, people, and applications automatically establish trust with one another using n-factor authentication.



BIDIRECTIONAL AND MUTUAL

requests are authenticated whether sent upstream or down, providing data provenance. TDI also supports mutual auth if required for your application.

FLEXIBLE MANAGEMENT



GRANULAR IDENTITY MANAGEMENT

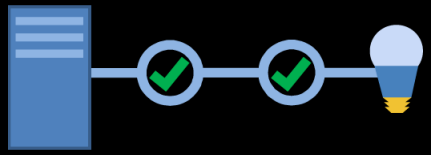
every server, device, user, and service is provisioned with a unique key, so each and every identity can be managed in real time.

REAL-TIME PROVISIONING / REVOCATION

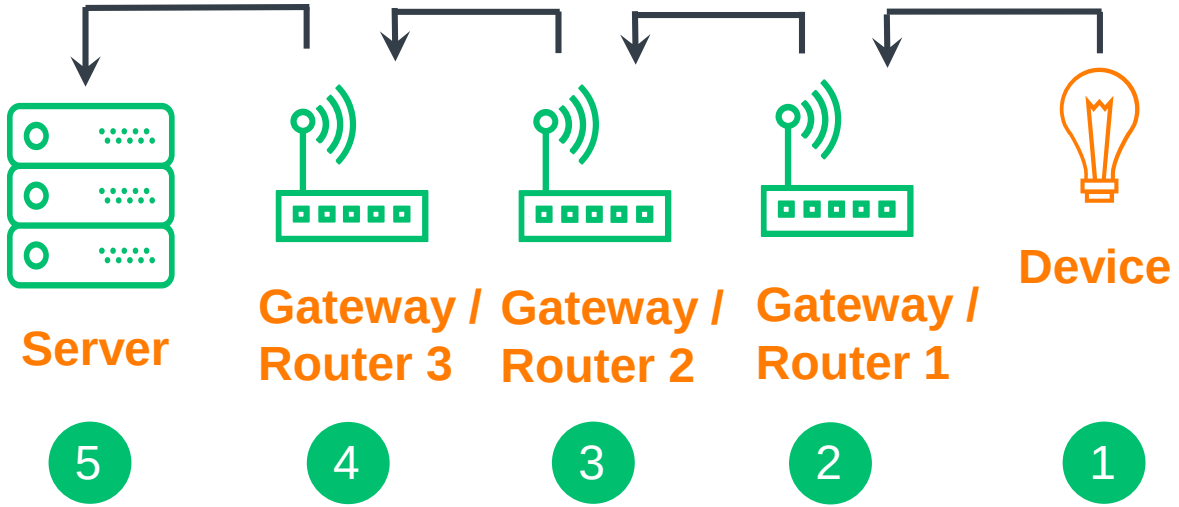
add identities in real-time without sharing public keys with every recipient. revoke in real time as well, without distributing crls or requiring ocsp calls by recipients.

THE GRAVY

ROUTE VALIDATION



Validate the path a message takes to ensure that it originated from a proper location in your network.



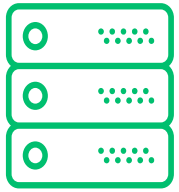
Devices, gateways, and routers each sign with their own keys.

Server validates the route, each individual key, and the order of keys

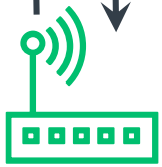
Future Framework Enhancements

DDOS PROTECTION WITH TDI

Update black/white lists at device or gateway, authorized by TDI



Server



Gateway

2

Gateway validates asset legitimacy, policy legitimacy, and passes site information upstream

1 Large sensor network identifies trends of sites targeted

Device(s)

6 Admin uses CLI/dashboard or API to monitor activity

3

Server sends to site trends to Neustar & TDI for asset verification, anomaly detection, & analysis

5 TDI pushes whitelist & blacklist updates down to gateway & device for mitigation

4 Neustar validates sites & identifies large changes in traffic patterns. TDI validates signature and behavior. TDI servers can be replicated to withstand DDoS.

OFFLINE / INTERMITTENT OFFLINE – VARIOUS USE CASE SPECIFIC OPTIONS

Hierarchal fleet signer & local co-signer to allow for offline TDI messages you can use the cloud co-signer to grant permission for a certain period of time

Cache messages for a period of time, so messages are pre authorized for a period of time

Device to device set a time to live for on ECDHE derived key message exchange between devices, gateways & users. This can be set up in Fleet server of cloud administration.

The Demo

The Code

THE CODE

OPEN SOURCE CODE

https://github.com/Neustar-TDI

DOCUMENTATION

https://oneid-connect.readthedocs.io/en/latest/index.html

MORE INFO ON TDI

http://bit.ly/NeustarTDI

QUESTIONS & SUPPORT

earlyaccess.iot @ team.neustar

ACKNOWLEDGEMENTS

- Casey Newton
- Steve Kirsch
- Neustar IoT Team
- Callie Holderman for editing the demo
- Countless security researchers who reviewed this
- My wife and kids for their endless patience & support

Q&A

Redesigning PKI To Solve Revocation, Expiration, & Rotation Problems

Brian Knopf @DoYouQA

neustar

- Do NOT rely on the router & firewall as your security model
- Trust nothing unless proven otherwise… constantly
- Enable your SOC or NOC to control the security rather than users or site managers

Thank You