Post Quantum Cryptography (KEM and Authentication): Some Real World Data

Presented by: Nalini Elkins

NALINI ELKINS

OUTSIDE THE STACKS, INC.

NALINI.ELKINS@OUTSIDETHESTACKS.COM

At IETF (and NIST, etc)

Algorithms are being developed

Protocols are being modified (TLS, SSH, etc)

Many arguments are being had!

	Algorithm	Public Key (bytes)	Signature (bytes)
	ECDSA P-256	65	~72
	ECDSA P-384	97	~104
	RSA-2048	270	256
	RSA-3072	398	384
	RSA-4096	526	512
	ML-DSA-44	1,312	2,420
New	**ML-DSA-65**	1,952	3,309
	ML-DSA-87	2,592	4,627

3

Enterprises and Certificates?

- Can an Merkle Tree Certificates (MTC) relying party continue to access and process certificate information—such as Subject DN, SAN, EKU, Certificate Policies, Name Constraints, and enterprise-specific extensions—in the same way it does today?
- If application, middleware, or validation library changes are required, what migration guidance should be provided?
- What information should enterprises archive for long-term audit, compliance, forensic, and evidentiary purposes?
- How should private PKIs deploy and operate MTC?
- What enterprise deployment patterns or best practices should be documented?

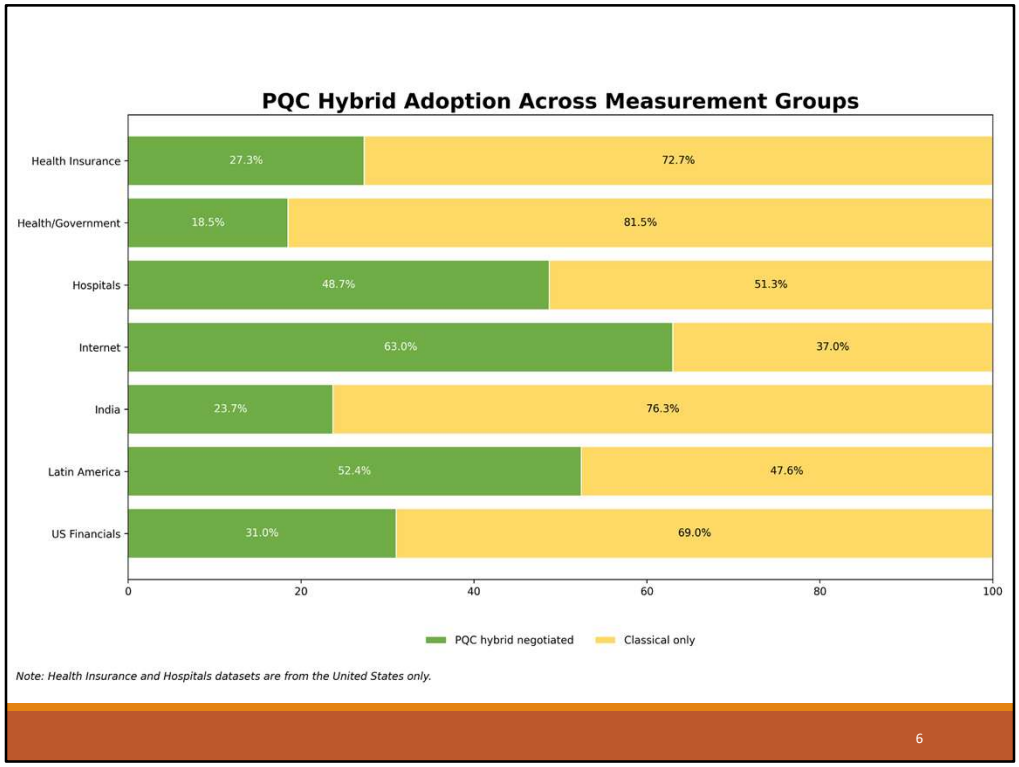
Focusing on Real World Adoption at Enterprises

Financials – horizontally

- US
- India
- Latin America

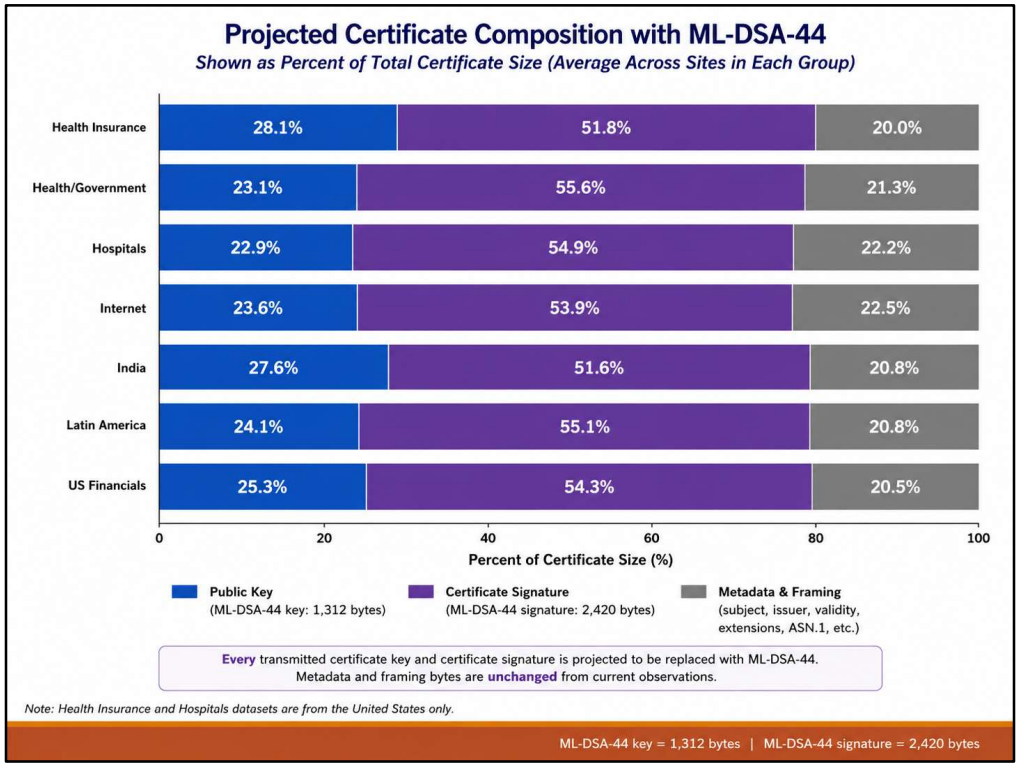
Health Care – vertically (US only)

- Insurance
- Hospitals
- Health care related government agencies (Medicare, HHS, etc)



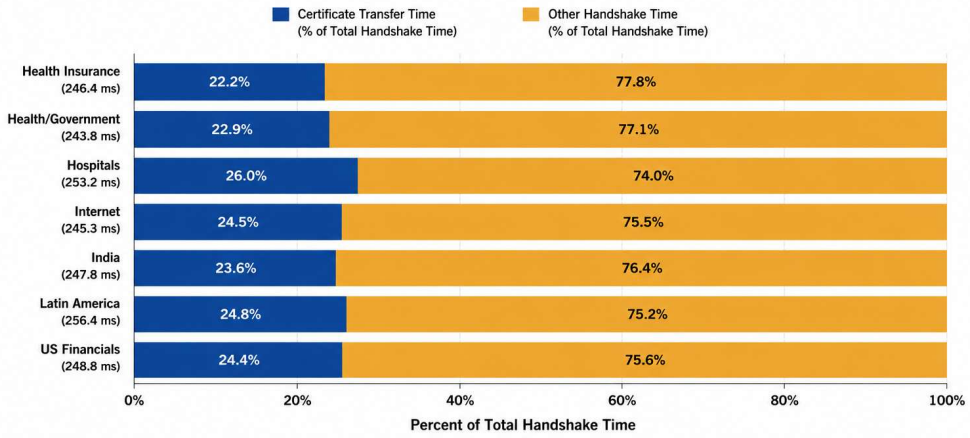
Note: Health Insurance and Hospitals datasets are from the United States only.

6


Projected Certificate Composition with ML-DSA-44
Shown as Percent of Total Certificate Size (Average Across Sites in Each Group)
Health Insurance
28.1%
51.8%
20.0%
Health/Government
23.1%
55.6%
21.3%
Hospitals
22.9%
54.9%
22.2%
Internet
23.6%
53.9%
22.5%
India
27.6%
51.6%
20.8%
Latin America
24.1%
55.1%
20.8%
US Financials
25.3%
54.3%
20.5%
0
20
40
60
80
100
Percent of Certificate Size (%)
Public Key
(ML-DSA-44 key: 1,312 bytes)
Certificate Signature
(ML-DSA-44 signature: 2,420 bytes)
Metadata & Framing
(subject, issuer, validity, extensions, ASN.1, etc.)
Every transmitted certificate key and certificate signature is projected to be replaced with ML-DSA-44.
Metadata and framing bytes are unchanged from current observations.
Note: Health Insurance and Hospitals datasets are from the United States only.
ML-DSA-44 key = 1,312 bytes | ML-DSA-44 signature = 2,420 bytes

Average Total Handshake Time vs. Percent Certificate Transfer Time (Current Observed)

Across Measurement Groups



***Overall Average* Certificate Transfer Time: 23.5% of Total Handshake Time (Average Total Handshake Time: 246.4 ms)**

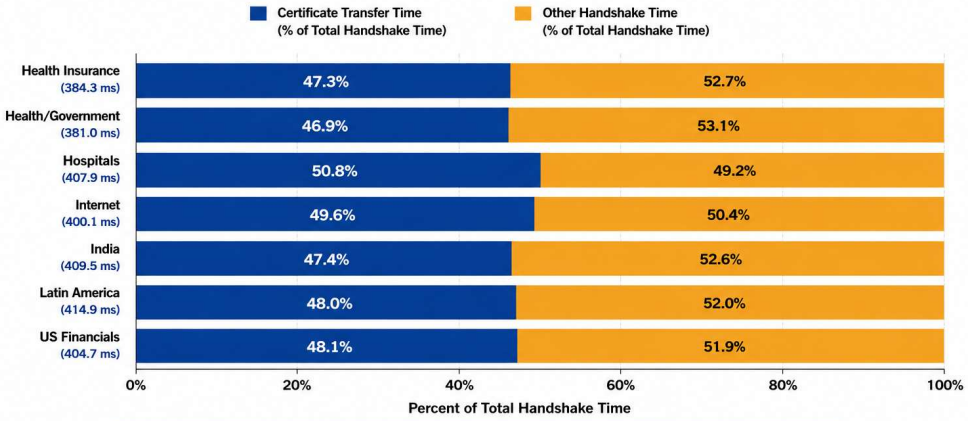
Methodology

- Each TraceIdentifier contributes one successful handshake observation.
- Total handshake time is measured from the first ClientHello to the selected Finished frame.
- For TLS 1.3 and TLS 1.1, certificate interval is measured from the first certificate byte to the peer Finished frame.
- For TLS 1.2, certificate interval is measured from the first certificate byte to ClientKeyExchange.
- Statistics are calculated from individual observations, not from averages of site-level averages.
- Standard deviation is the population standard deviation. Percentiles use linear interpolation.
- Normalized values divide milliseconds by observed certificate-list size in KiB (1024 bytes).

Note: Health Insurance and Hospitals datasets are from the United States only.

Average Total Handshake Time vs. Percent Certificate Transfer Time (ML-DSA-44 Projected)

Across Measurement Groups – ML-DSA-44 replaces every transmitted certificate key and signature



Overall Average Certificate Transfer Time: 48.3% of Total Handshake Time
(Average Total Handshake Time: 404.6 ms, Increase: +158.2 ms, +64.2%)

Methodology

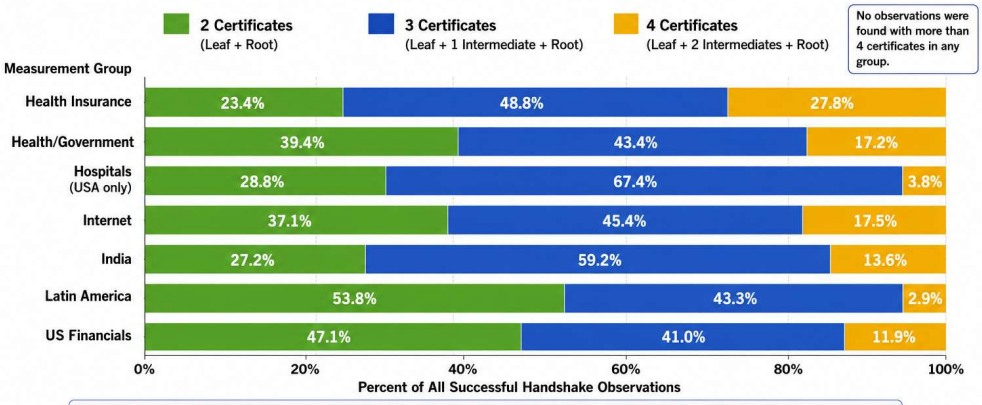
- Each TraceIdentifier contributes one successful handshake observation.
- Total handshake time is measured from the first ClientHello to the selected Finished frame.
- For TLS 1.3 and TLS 1.1, certificate interval is measured from the first certificate byte to the peer Finished frame.
- For TLS 1.2, certificate interval is measured from the first certificate byte to ClientKeyExchange.
- Statistics are calculated from individual observations, not from averages of site-level averages.
- Standard deviation is the population standard deviation. Percentiles use linear interpolation.
- Normalized values divide milliseconds by observed certificate-list size in KiB (1024 bytes).

Note: Health Insurance and Hospitals datasets are from the United States only.

9

Observed Certificate Chain Composition Across Measurement Groups

Shown as Percent of All Successful Handshake Observations (By Number of Certificates in Transmitted Chain)



Overall Average (All Groups): 2 Certificates: 36.7% | 3 Certificates: 49.8% | 4 Certificates: 13.5%

Average certificates per chain (all groups): **2.77**

Methodology

- Each TraceIdentifier contributes one successful handshake observation (one certificate chain).
- The number of certificates is the total transmitted in the TLS certificate-list for that handshake (leaf through root/observed root).
- Statistics are calculated from individual observations, not from averages of site-level averages.

Measurement Group	n = Successful Handshake Observations
Health Insurance	525
Health/Government	463
Hospitals (USA only)	633
Internet	491
India	529
Latin America	561
US Financials	571

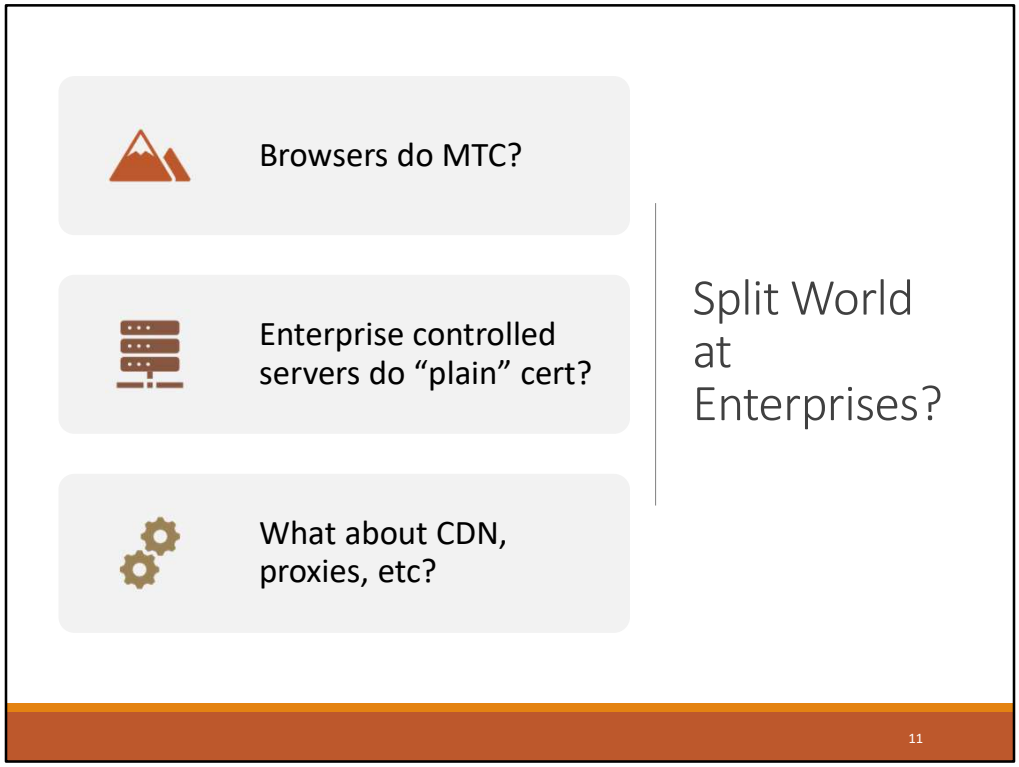
Interpretation

Most handshakes transmit 3 certificates, followed by 2 certificates. Chains with 4 certificates are less common.

No chains with more than 4 certificates were observed in any measurement group.

Note: Health Insurance and Hospitals datasets are from the United States only. *Percentages may not sum to 100% due to rounding.*

10


Browsers do MTC?
Enterprise controlled servers do “plain” cert?
What about CDN, proxies, etc?
Split World at Enterprises?
11

Questions?

- What else is interesting?
  - Definitely when there is some MTC in the real world
  - Post on web site?
- Discussion in PLANTS
  - We (our non-profit, Industry Network Technology Council) want to put up some more public test points

12

Questions

Contact:

info@industrynetcouncil.org

Nalini.elkins@insidethestack.com

NIST Standards

Standard	Algorithm	Based On
FIPS 203	ML-KEM	CRYSTALS-Kyber
FIPS 204	ML-DSA	CRYSTALS-Dilithium
FIPS 205	SLH-DSA	SPHINCS+

ML-KEM (Module-Lattice-Based Key Encapsulation Mechanism)

- **NIST's standardized post-quantum key establishment algorithm** (FIPS 203)
- Based on the **CRYSTALS-Kyber** algorithm selected by NIST in 2022
- Designed to resist attacks from both **classical and quantum computers**
- Intended to replace classical public-key key establishment mechanisms such as RSA key transport (legacy) and elliptic-curve/Diffie-Hellman based key exchange.
- Available in three security levels:
  - **ML-KEM-512**
  - **ML-KEM-768** (commonly selected as the general-purpose balance of security and performance)
  - **ML-KEM-1024** (highest security)
- Intended to protect against **"Harvest Now, Decrypt Later"** attacks
- Efficient enough for deployment in Internet protocols and TLS

ML-DSA (Module-Lattice-Based Digital Signature Algorithm)

- **NIST's standardized post-quantum digital signature algorithm** (FIPS 204)
- Based on the **CRYSTALS-Dilithium** algorithm selected by NIST in 2022
- Designed to resist attacks from **quantum computers**
- Used for **authentication and digital signatures**, not key establishment
- Intended to replace classical digital signature algorithms such as RSA and ECDSA.
- Available in three security levels:
  - **ML-DSA-44**
  - **ML-DSA-65** (commonly considered a general-purpose balance of security and performance)
  - **ML-DSA-87** (highest security)
- Designed for high performance and practical deployment across Internet protocols

ML-KEM in TLS

- Standardizing **hybrid key exchange** for TLS 1.3
- Hybrid combines:
  - **Classical ECDHE (e.g., X25519)**
  - **ML-KEM**
- Provides:
  - Security against today's computers
  - Protection if future quantum computers become practical
- Negotiated using new **Named Groups** in the TLS handshake
- Examples:
  - **X25519MLKEM768**
  - **SecP256r1MLKEM768**
- If either the classical or post-quantum component remains secure, the hybrid exchange remains secure

17

ML-DSA in TLS

Developing support for post-quantum digital signatures in TLS

- ML-DSA is used for authentication, not key exchange
- Can be used in:
  - X.509 certificates
  - CertificateVerify message
- TLS authentication framework
- Negotiated using the TLS signature_algorithms and signature_algorithms_cert extensions

• Examples:
  - ML-DSA-44
  - ML-DSA-65
  - ML-DSA-87

SLH-DSA Variant	Public Key (bytes)	Signature (bytes)
SHA2-128s	32	7,856
SHA2-128f	32	17,088
SHA2-192s	48	16,224
SHA2-192f	48	35,664
SHA2-256s	64	29,792
SHA2-256f	64	49,856
SHAKE-128s	32	7,856
SHAKE-128f	32	17,088
SHAKE-192s	48	16,224
SHAKE-192f	48	35,664
SHAKE-256s	64	29,792
SHAKE-256f	64	49,856

WG	Internet-Draft	Topic	Status (July 2026)
TLS	**draft-ietf-tls-mlkem**	ML-KEM key establishment for TLS 1.3	**WG Last Call**
TLS	**draft-ietf-tls-mldsa**	ML-DSA authentication for TLS	**IESG Processing**
PLANTS	**draft-ietf-plants-merkle-tree-certs**	Merkle Tree Certificates (MTC)	**Active WG Draft**

Merkle Tree Certs (MTC)

draft-ietf-plants-merkle-tree-certs-05

This document introduces Merkle Tree Certificates (MTCs), a **new form of X.509 certificate** that integrates logging with certificate issuance. Each CA maintains logs of everything it issues, signing views of its logs to assert it has issued the contents. The CA signature is combined with cosignatures from other parties who verify correct operation and optionally mirror the logs. These signatures, together with an inclusion proof for an individual entry, constitute a certificate.

STUXNet

The malware has both user mode and kernel mode rootkit ability under Windows, and its device drivers have been digitally signed with the private keys of two public key certificates that were stolen from separate well-known companies, JMicron and Realtek, both located at Hsinchu Science Park in Taiwan.

The driver signing helped it install kernel mode rootkit drivers successfully without users being notified, and thus it remained undetected for a relatively long period of time. Both compromised certificates have been revoked by Verisign.

Examples of Sensitive Data

Personal Information
- Social Security numbers
- Passport and driver's license numbers
- Date of birth
- Home address
- Personal phone numbers
- Email addresses (in some contexts)

Financial Information
- Credit and debit card numbers
- Bank account and routing numbers
- Tax records
- Investment account information
- Payroll and salary information

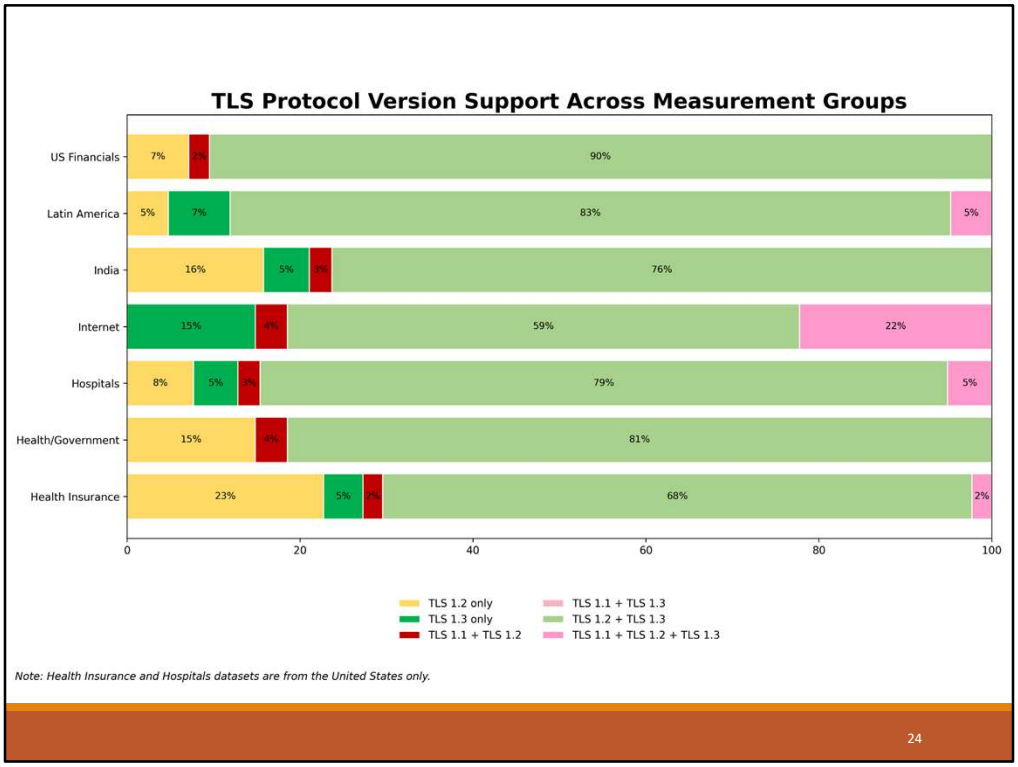
Healthcare Information
- Medical records
- Health insurance information
- Prescription history
- Laboratory results
- Protected Health Information (PHI)

Authentication Credentials
- Passwords
- Cryptographic private keys
- Authentication tokens
- Multi-factor authentication (MFA) secrets
- API keys

Business Information
- Intellectual property
- Trade secrets
- Source code
- Product designs
- Customer databases
- Financial forecasts
- Merger and acquisition plans

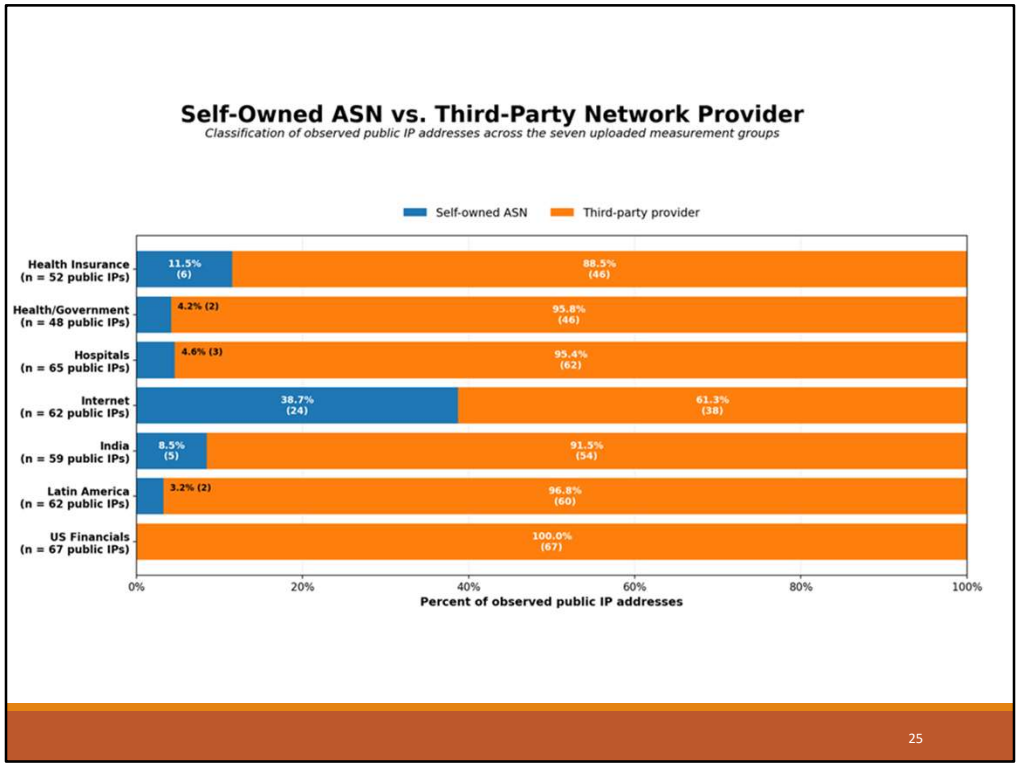
Government / National Security
- Classified information
- Military plans
- Intelligence reports
- Critical infrastructure data
- Election systems information
- Law enforcement investigations

23

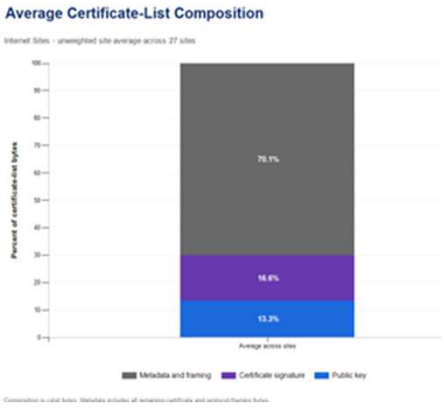


Note: Health Insurance and Hospitals datasets are from the United States only.

24


Self-Owned ASN vs. Third-Party Network Provider
Classification of observed public IP addresses across the seven uploaded measurement groups
Self-owned ASN
Third-party provider
Health Insurance (n = 52 public IPs)
11.5% (6)
88.5% (46)
Health/Government (n = 48 public IPs)
4.2% (2)
95.8% (46)
Hospitals (n = 65 public IPs)
4.6% (3)
95.4% (62)
Internet (n = 62 public IPs)
38.7% (24)
61.3% (38)
India (n = 59 public IPs)
8.5% (5)
91.5% (54)
Latin America (n = 62 public IPs)
3.2% (2)
96.8% (60)
US Financials (n = 67 public IPs)
100.0% (67)
0%
20%
40%
60%
80%
100%
Percent of observed public IP addresses

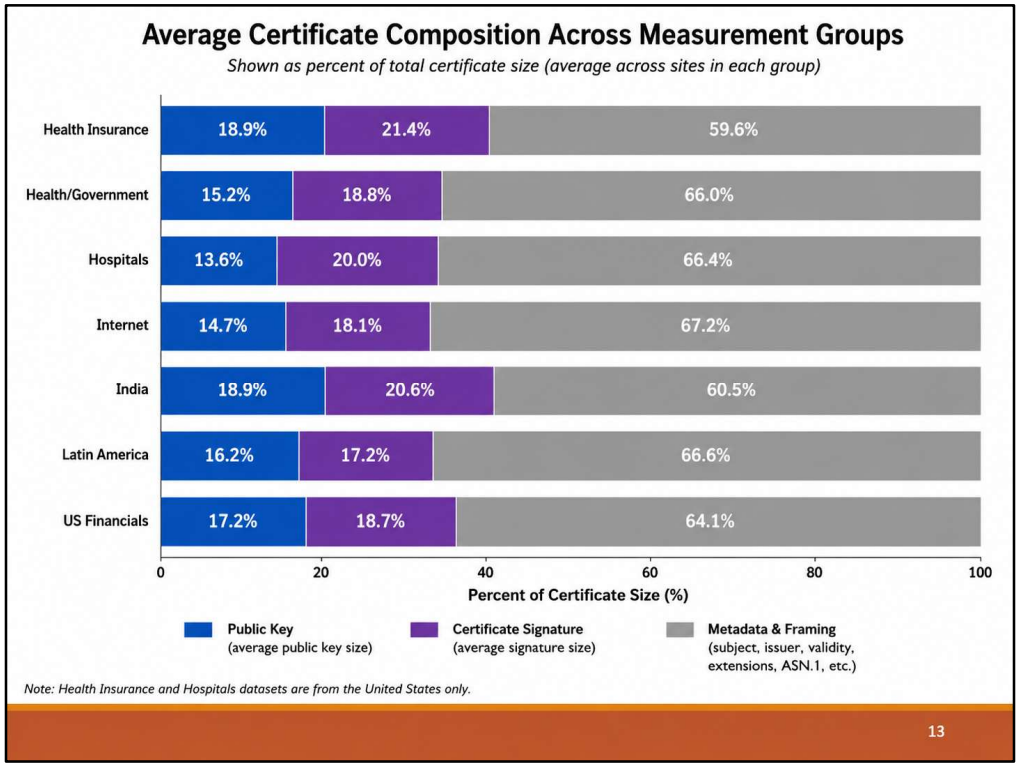
Internet Sites (top 27)

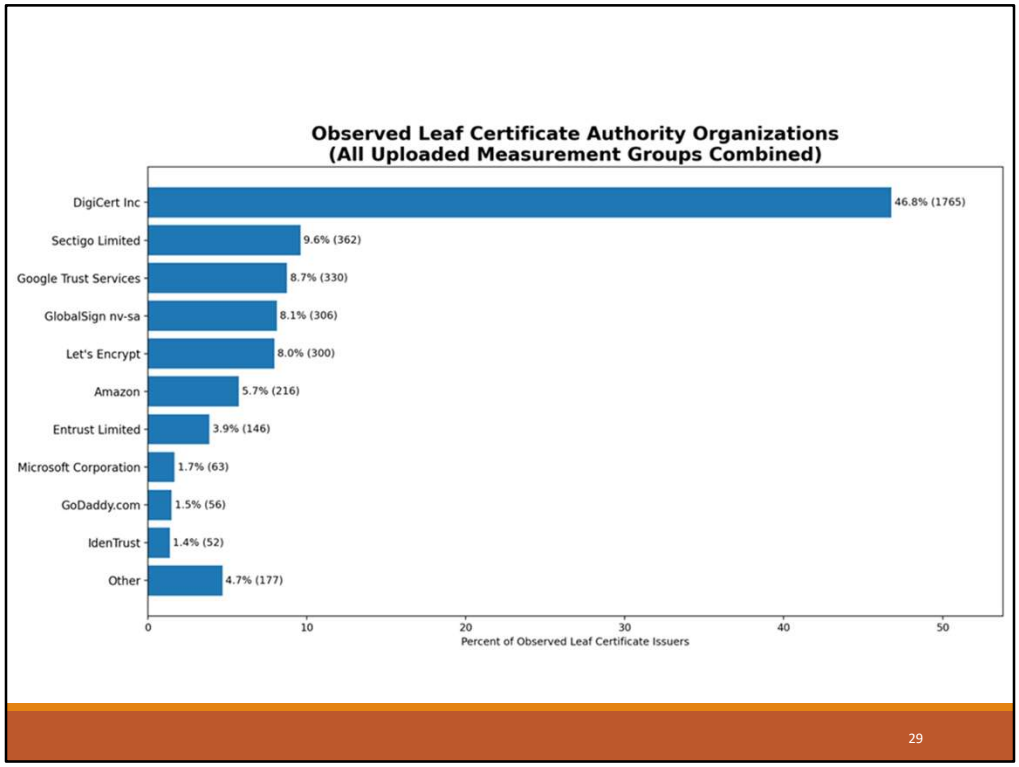


Observed Algorithm Set	Sites	Percent
RSA	**14**	**51.9%**
EC	**9**	**33.3%**
EC + RSA	**4**	**14.8%**
Total	**27**	**100.0%**

In the old days …

Component	Purpose	Typical Contribution
Public Key	Contains the server's public key (RSA or EC)	~10–15%
Digital Signature	CA signature protecting the certificate	~10–20%
Subject & Issuer	Organization and CA identity	~5–10%
Validity	Not Before / Not After dates	<2%
Extensions	Key Usage, EKU, SAN, SKI, AKI, Policies, AIA, CRL, SCTs, etc.	**40–60%**
Encoding & ASN.1 framing	DER structure, lengths, object identifiers	~10–20%


Average Certificate Composition Across Measurement Groups
Shown as percent of total certificate size (average across sites in each group)
Health Insurance
18.9%
21.4%
59.6%
Health/Government
15.2%
18.8%
66.0%
Hospitals
13.6%
20.0%
66.4%
Internet
14.7%
18.1%
67.2%
India
18.9%
20.6%
60.5%
Latin America
16.2%
17.2%
66.6%
US Financials
17.2%
18.7%
64.1%
0
20
40
60
80
100
Percent of Certificate Size (%)
Public Key
(average public key size)
Certificate Signature
(average signature size)
Metadata & Framing
(subject, issuer, validity, extensions, ASN.1, etc.)
Note: Health Insurance and Hospitals datasets are from the United States only.
13


Observed Leaf Certificate Authority Organizations
(All Uploaded Measurement Groups Combined)
DigiCert Inc
Sectigo Limited
Google Trust Services
GlobalSign nv-sa
Let's Encrypt
Amazon
Entrust Limited
Microsoft Corporation
GoDaddy.com
IdenTrust
Other
46.8% (1765)
9.6% (362)
8.7% (330)
8.1% (306)
8.0% (300)
5.7% (216)
3.9% (146)
1.7% (63)
1.5% (56)
1.4% (52)
4.7% (177)
0
10
20
30
40
50
Percent of Observed Leaf Certificate Issuers

Next Steps

- First, validate use of certificate payload / options at enterprises
- Next, put up MTC! (Maybe we have everything we need!)
- Maybe we don't need intermediate certs?
- How about certificate compression?
- Are private PKIs fundamentally different than public?

30