# Securing your Web Server

LinuxFestNW 2025

Ted Matsumura

# 2025 Secure Sockets Layer (SSL) Statistics

**34% Of Sites Have Inadequate Security (SSL Pulse)**

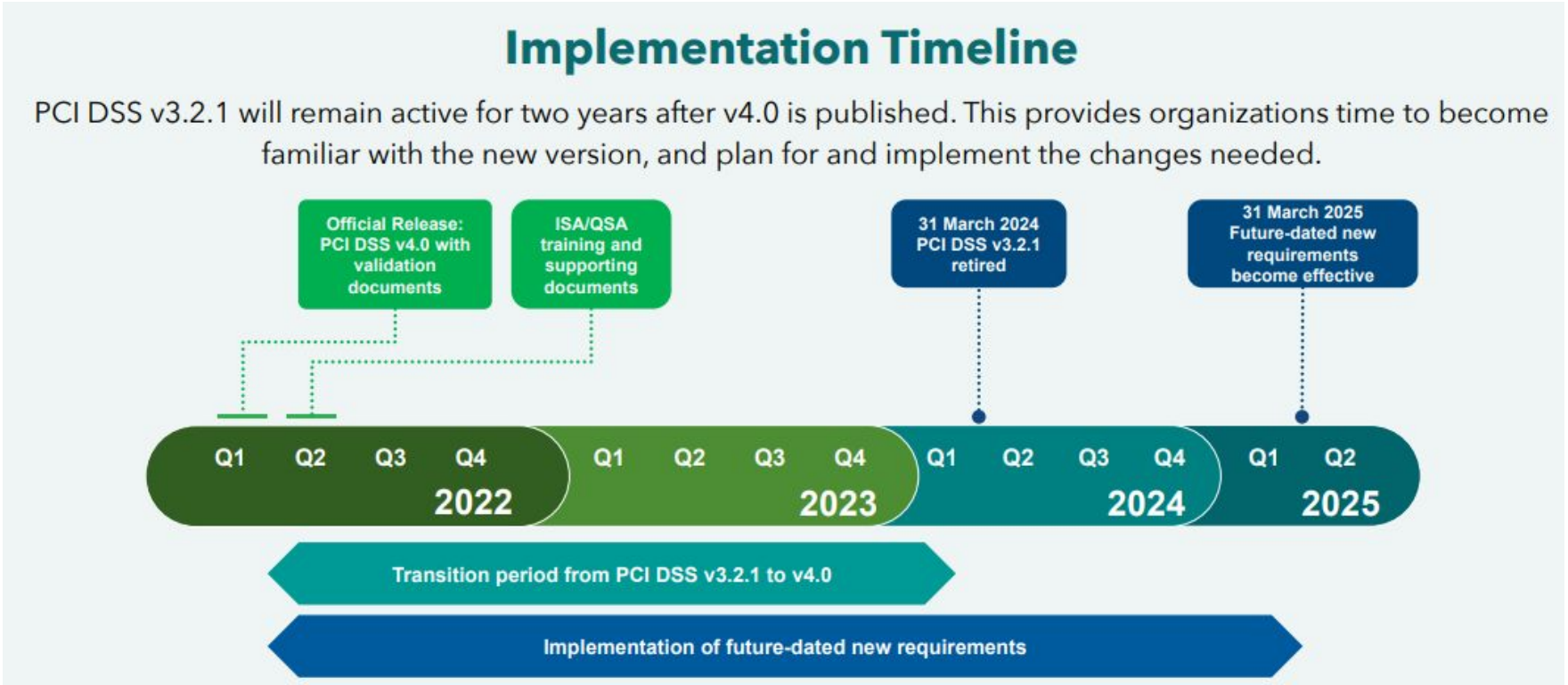
About 300M SSL Certificates on the Internet as of 3/2025, US - 27M, Germany 12M, Equatorial Guinea 16[1, 2]

90% of SSL Certs issued by 4 Cert Authorities (CAs): LetsEncrypt 60% GlobalSign 22%, Sectigo 5%, GoDaddy 4%, remaining spread broadly.[3]

Types of SSL certs are 94% Domain Validation (DV), 26% Organization Validated (OV) - 9, 13% Extended Validation (EV). Differences are that DV are validated simply by whois record email or admin with DNS access, OV have 9 additional business checks, EV have 16 additional checks over DV[4]

Depending on Geography, 2-9% of Content Delivery Network (CDN) traffic is still TLSv1.2. **Supporting TLSv1.3 is mandatory for good security, and PQ ciphers**, but TLSv1.2 currently meets most compliance requirements.

# Payment Card Industry (PCI) Data Security Standard v4.0



# Basic Assumptions for this ~40m talk

**Current well known best practices for website compliance and security include:**

**Part 1:** Secure websites using TLSv1.2, and TLSv1.3, disabling SSL and earlier versions of TLS (1.0, 1.1), and known weak ciphers. Example site (slatey.org) is an aws ec2 instance running AL2 (Fedora), with relatively old openssl version (1.0.2k-fips)

Use Strong ciphers supported by current and past generations of PC's and OS's, up to say 10 years back, assuming users are patching OS's and client components.

Use free or paid up-to-date tools to audit or self-audit websites for known vulnerabilities, independent of ciphers and protocols.

**Part 2:** Discussion of current state of PQ (Post Quantum) ciphers, NIST, OpenSSL v3.5

Time permitting - Discussion of relationship of known Industry Compliance standards (PCI DSS 4.0, FedRAMP, SOC II, HIPAA with NIST and FIPS guidelines)

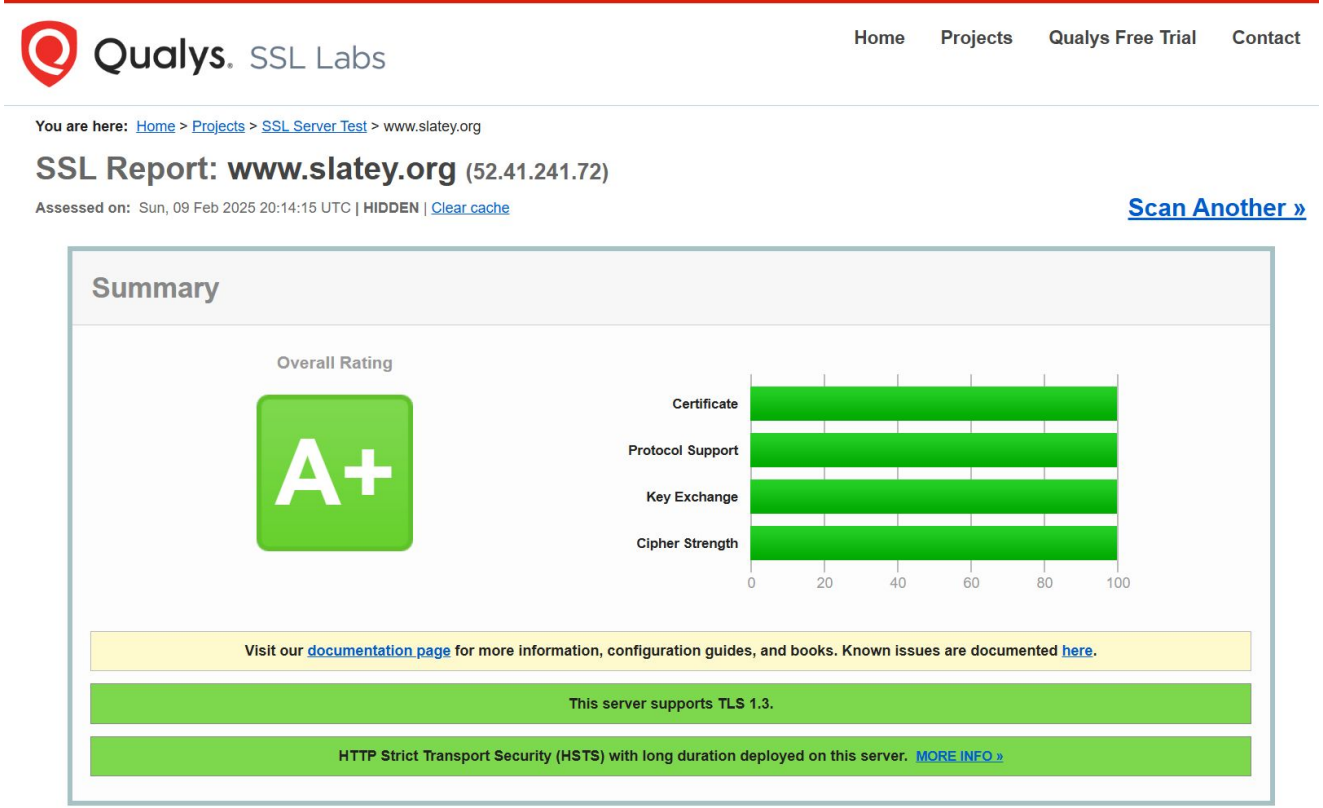
# Basic Assumptions (2)

Tools exist for scanning websites including the $scanssl tool, and various web based tools such as the Qualsys SSL test site tool.

Securing a website today, one can follow best practices easily found on the web, or on configurators like the Mozilla SSL Configuration tool, but some of the examples will provide even stronger security, while still ensuring compatibility back to ~10 year old (patched) clients.

The following example config file is for nginx, but similar parameters exist for other web servers. Use this resource for nginx specific documentation: https://nginx.org/en/docs/http/ngx_http_ssl_module.html

# Example of an online website scanning tool



# Example nginx.conf file (part 1 - setup with a 301 redirect)

## If port 80 is open, redirect it to 443:

```
server {
    listen 80; # optional - note that this redirect to https will be in plain text
    server_name slatey.org www.slatey.org;
    return 301 https://www.slatey.org$request_uri;
}
```

**Summary - unless specific reason for HTTP port 80, close it**

# Example nginx.conf file (part 2)

```
server {
        listen                  443 ssl http2;
        add_header Strict-Transport-Security "max-age=31536000;
        includeSubDomains" always;
        # above line tells browser clients to use https for one year
        server_name             slatey.org www.slatey.org; # ensure certs are for both names
         # use a SAN cert or wildcard cert from a trusted Certificate Authority (CA)
        root                    /usr/share/nginx/html;
        ssl_certificate             /etc/letsencrypt/live/slatey.org-0001/fullchain.pem;
        ssl_certificate_key         /etc/letsencrypt/live/slatey.org-0001/privkey.pem;
```

https://nginx.org/en/docs/http/ngx_http_ssl_module.htmlhttps://nginx.org/en/docs/http/ngx_http_ssl_module.html

# SSL/TLS Cipher Suite nomenclature

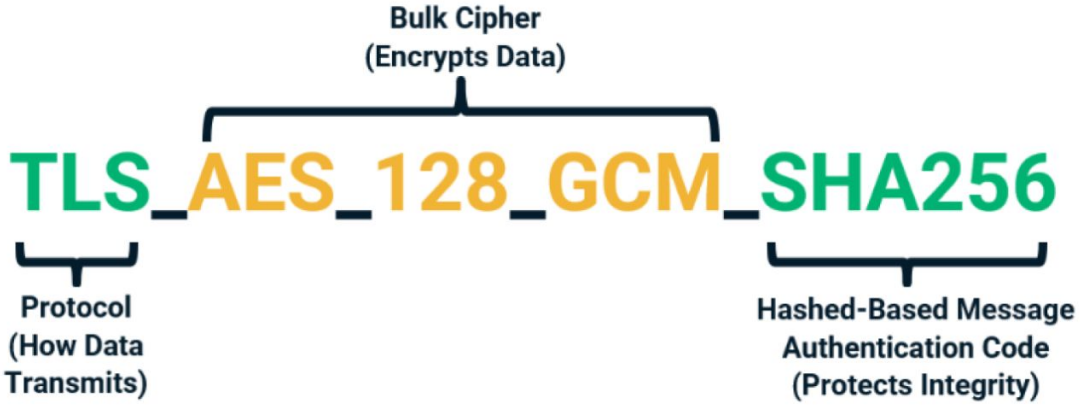
**OpenSSL vs IANA: https://www.ssl.org/cipher-suite-mapping**

**OpenSSL names may be shorter, IANA usually uses RFC name conventions**

**TLS** = Transfer Layer Security protocol (vs none or SSL)

**AES_128_GCM** = the bulk cipher encryption algorithm

**SHA256** = the Hash based MAC (Message Auth. Code) type



# Nginx.conf file part 3 (nginx ssl parameters)

**ssl_dhparam** /etc/ssl/certs/dhparam.pem; # create 4096-bit large prime for DH key exchange

**ssl_session_cache shared:SSL:1m;** # keep as short as possible, Mozilla default is 10m

**ssl_session_timeout 10m;** # about 40,000 sessions same as Mozilla config

**ssl_protocols TLSv1.2 TLSv1.3;** # if possible, use only TLSv1.3, but 5% of traffic still TLSv1.2

**ssl_ecdh_curve secp521r1:secp384r1;** # Mozilla defaults prime256v1, and secp384r1 # we want strong Elliptic Curve Diffie-Hellman Ephemeral curve negotiation

ssl_ciphers **ECDHE-RSA-CHACHA20-POLY1305**:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-CCM:DHE-RSA-AES256-CCM8:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256;

ssl_conf_command Ciphersuites TLS_AES_256_GCM_SHA384:**TLS_CHACHA20_POLY1305_SHA256**;

# Key Exchange in TLS - RSA vs DH Ephemeral (DHE) vs ECDSA

For key exchange[5], RSA is good, but static, and does not have forward secrecy. This means if a key were to be compromised on a web server using RSA key exchange, previous sessions would be able to be decrypted.

RSA uses large primes p and q, but if these become compromised, the private key can be calculated, and future (forward) communications could be decrypted RSA is still strong. 3072 or 4096 bit keys are stronger.

Diffie-Hellman (DHE) and Forward Secrecy (FS), also known as Perfect Forward Secrecy (PFS)

The client and server each generate a private secret number. This creates a shared secret, but without exposing it. Large DH keys can cause more CPU usage, making long DH keys less energy efficient, especially for mobile devices.

Elliptic Curves Diffie-Hellman Ephemeral (ECDHE) - smaller, less compute power, default for TLSv1.3, and supports Forward Secrecy

**Nginx setting: ssl_ecdh_curve secp521r1:secp384r1; # many default configs will use a curve with smaller bits like prime256v1 value here - to list the curves your version of openssl uses, run $openssl ecparam -list_curves**

# Chart of RSA, DHE, ECDHE key agreement protocols

| Protocol | Key Size (Equivalent Strength) | Performance Impact | Forward Secrecy | Modern Usage |
|---|---|---|---|---|
| RSA | 2048-bit | High | No | Legacy systems |
| DHE | 2048-bit | Moderate | Yes | Some applications |
| ECDHE | 256-bit ECC | Low | Yes | Default in TLS 1.3 |

# Example of Deprecated ciphers/protocols for FIPS 140-3

Most of the below are/were allowed in FIPS 140-2, although not recc.

SHA-1 hash, DH key sizes < 2048 bits, EC key sizes < 224 bits, RSA key sizes < 2048 bits

3DES

TLS 1.0, TLS 1.1

DH key sizes must be 2048-8192 bits

EC key sizes must be > 224 bits, P-192EC curve not allowed

# Good Starting Points:

To achieve 100% on scans like the Qualsys vs 90% or 95% do the following:

Support TLSv1.2 and TLSv1.3 with strong ciphers

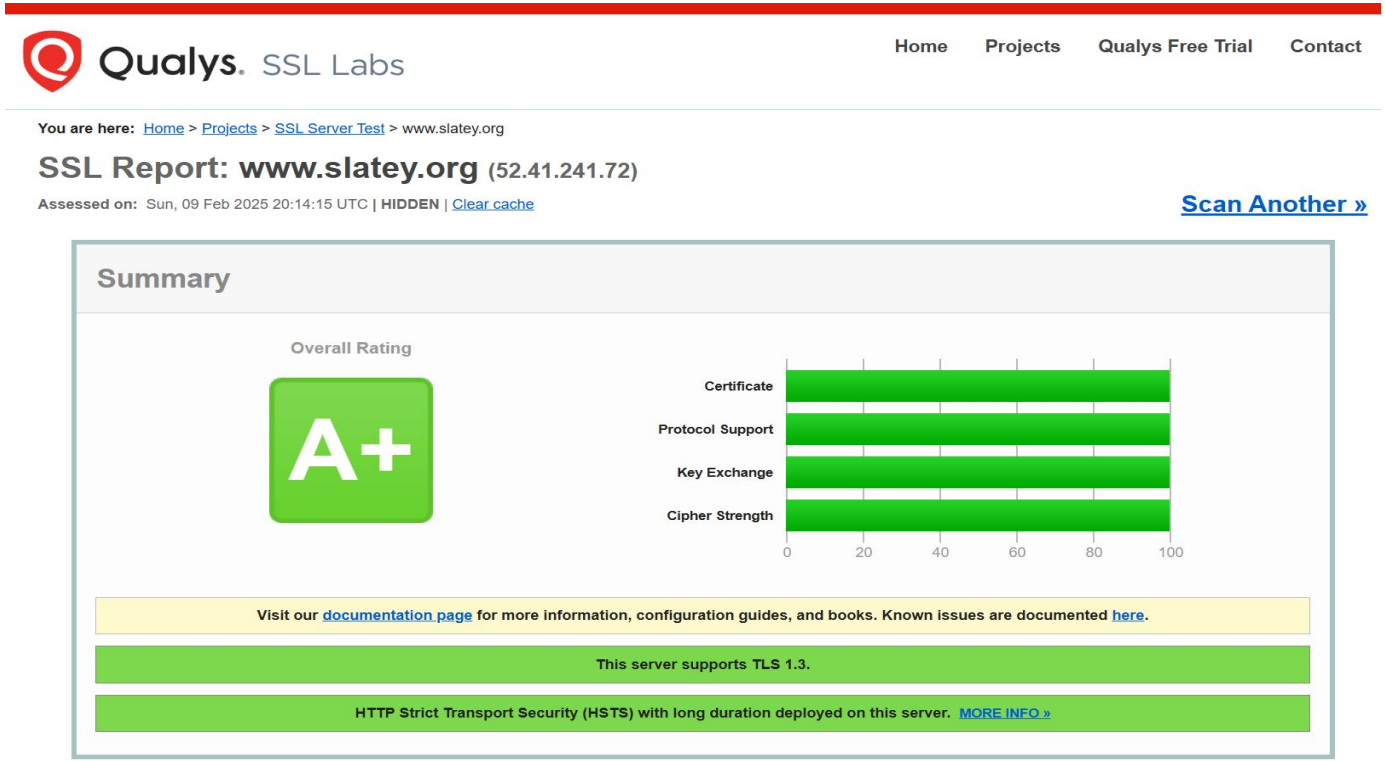
Implement HTTP Strict Transport Security (HSTS)

Use AES256 instead of AES128

If using ECDHE, use strong key sizes such as NIST P-384 and NIST P-521, these are supported by enterprise load balancers like the F5 BigIP

Use 4096-bit RSA certificates from CA, except for chained Intermediate (2048-bit will not affect security, and will speed up handshaking)

# Some scanning tools need those previous settings for A+



# Resources and Acronyms

Mozilla SSL Configuration Generator with Modern, Intermediate, and Old configs: https://ssl-config.mozilla.org/

Fedramp main page: https://www.fedramp.gov/

PCI DSS 4.0 Documents page: https://east.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss

NIST IR 8547 (Initial Public Draft) published Nov. 12, 2024:
“Transition to Post-Quantum Cryptography Standards”
https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf

# Part 2 - -

NIST documents can give us a guideline for time frames in which we will need to start moving to newer and stronger ciphers. Their guidelines indicate deprecation of some current ciphers in 2030, but this time frame could change depending on the timing of quantum computing deployments.

When Quantum Computers will be able to easily break existing ciphers and algorithms is unknown, but could be sooner than expected. Since data can be stored and decrypted later, it is always a good idea to secure data in motion and at rest in the strongest way feasible, and often mandated by Compliance standards and regulations.

Following are some timeframes from the NIST 800-171 guidelines document:

# Preparing for Quantum Computers (cont.)

As of Q1/2025, NIST has finalized it's PQ algorithms, after 9 years, including 3 years open to public input and comments.

1. **ML-KEM** (FIPS 203) - Module-Lattice-Based Key-Encapsulation Mechanism
2. **ML-DSA** (FIPS 204) - Module-Lattice-Based Digital Signature Algorithm
3. **SLH-DSA** (FIPS 205) - Stateless Hash-Based Digital Signature Algorithm
4. **FN-DSA** - (FIPS 206) - FN-DSA - FFT over NTRU-Lattice-Based DSA
5. Backup for ML-KEM in FIPS (Draft): **HQC** - non Lattice based, used ECCs

References for all are on public NIST pages (see example links below)

**** Memorize these 5, begin using them in your test environments ****

# Current known Quantum-vulnerable algorithms

488

**Table 2: Quantum-vulnerable digital signature algorithms**

| Digital Signature Algorithm Family | Parameters | Transition |
|---|---|---|
| **ECDSA** [FIPS186] | 112 bits of security strength | ***Deprecated*** after 2030<br>***Disallowed*** after 2035 |
| | ≥ 128 bits of security strength | ***Disallowed*** after 2035 |
| **EdDSA** [FIPS186] | ≥ 128 bits of security strength | ***Disallowed*** after 2035 |
| **RSA** [FIPS186] | 112 bits of security strength | ***Deprecated*** after 2030<br>***Disallowed*** after 2035 |
| | ≥ 128 bits of security strength | ***Disallowed*** after 2035 |

# Current known Quantum-vulnerable key-est. schemes

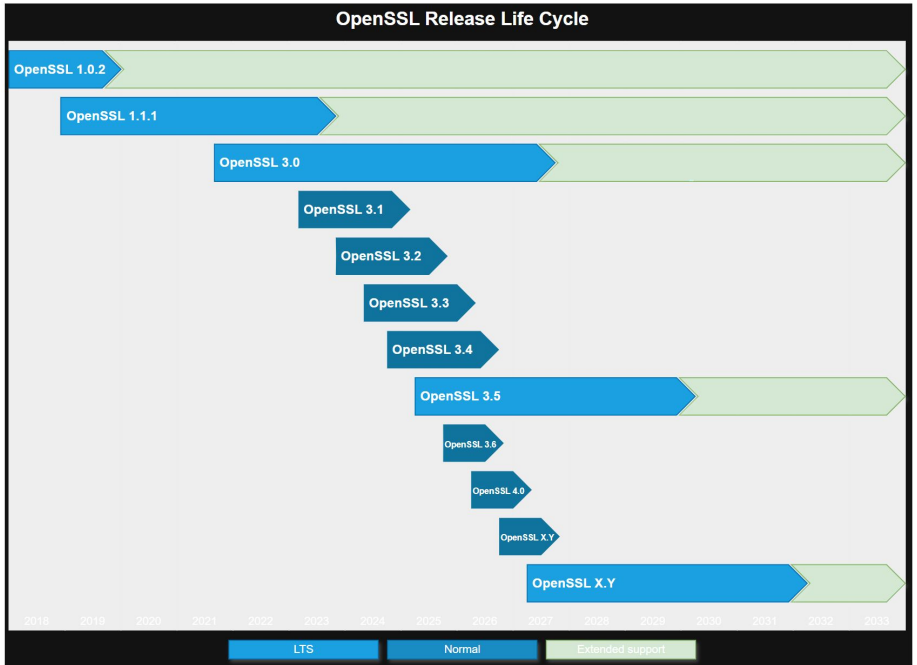
503

**Table 4: Quantum-vulnerable key-establishment schemes**

| Key Establishment Scheme | Parameters | Transition |
|---|---|---|
| **Finite Field DH and MQV** [SP80056A] | 112 bits of security strength | ***Deprecated*** after 2030<br>***Disallowed*** after 2035 |
| | ≥ 128 bits of security strength | ***Disallowed*** after 2035 |
| **Elliptic Curve DH and MQC** [SP80056A] | 112 bits of security strength | ***Deprecated*** after 2030<br>***Disallowed*** after 2035 |
| | ≥ 128 bits of security strength | ***Disallowed*** after 2035 |
| **RSA** [SP80056B] | 112 bits of security strength | ***Deprecated*** after 2030<br>***Disallowed*** after 2035 |
| | ≥ 128 bits of security strength | ***Disallowed*** after 2035 |

# Latest versions of OpenSSL support PQ

OpenSSL LTS 3.5 was released 4/8/2025, and supports several PQC algorithms: including: ML-KEM, ML-DSA, SLH-DSA, QUIC, being an LTS release, it will be supported until 4/8/2030.



# PQ Algorithms are Open

While I am focusing on OpenSSL, because it is comprehensive, free (Apache License), and widely available)). There are a variety of use cases where 3rd party SSL libraries may be preferred, e.g. LibreSSL, BoringSSL (Google), WolfSSL, Bouncy Castle, integrate specific ciphers into applications (e.g. Wireguard vpn), etc.

As of 4/2025, Cloudflare reports that 2% of web traffic (must be TLSv1.3) is using some form of hybrid conventional and PQ algorithms, and this will grow.

Certificate Authorities (CAs) will need to address new formats for pure PQ solutions, but hybrid solutions are available now, with existing certs, including LE.

https://pq.cloudflareresearch.com will show you if your particular browser connection key agreement is using X25519MLKEM76, for example, which is **post-quantum secure!**

# Thank you for attending!

The slides after this one are for reference, and info.

If you have any questions or corrections, you may email me at:

matsumura@gmail.com

## Thank you, and have a great LinuxFest NW 2025!!

# Key OpenSSL 3.5 PQ supported alrogithms

- ML-KEM (FIPS 203) — **Module Lattice-Based Key Encapsulation Mechanism (FIPS 203).** This is a PQC standard for Key Exchange.
- ML-DSA (FIPS 204) — **Module Lattice-Based Digital Signature Algorithm.** This is a PQC standard for digital signatures, and it uses the Dilithium signature method.
- SLH-DSA (FIPS 205) — **Stateless Hash-Based Digital Signature Algorithm.** This is a PQC standard for digital signatures and uses the SPHINCS+ signature method.

# Supported versions of Openssl

| Version | Released | LTS | Supported until |
|---|---|---|---|
| 3.5 | 8 April 2025 | Yes | 8 April 2030 |
| 3.4 | 22 October 2024 | No | 22 October 2026 |
| 3.3 | 10 April 2024 | No | 10 April 2026 |
| 3.2 | 23 November 2023 | No | 23 November 2025 |
| 3.1 | 7 March 2023 | No | 7 March 2025 |
| 3.0 | 7 September 2021 | Yes | 7 September 2026 |

# Listing PQ algorithms from OpenSSL 3.5 and later

Openssl commands (recent versions with post quantum):

#openssl list -kem-algorithms and #openssl list -signature-algorithms

1. #openssl list -kem-algorithms:

This command lists the supported Key Encapsulation Mechanism (KEM) algorithms in OpenSSL. In OpenSSL 3.5, you should see algorithms like "ML-KEM" which is part of the NIST FIPS 203 standard.

2. #openssl list -signature-algorithms:

This command lists the supported digital signature algorithms. In OpenSSL 3.5, you will find "ML-DSA" and "SLH-DSA", which are NIST FIPS 204 and "NIST FIPS 205 standards, respectively

# Tools - example sha3 commands from openssl

Check sha3 versions - $openssl list -digest-algorithms | grep sha3

Checksum a file - $openssl dgst -sha3-512 /bin/echo

Checksum a string - $printf “filename” | openssl dgst -sha3-512

There is a sha3sum command, for Debian, install libdigest-sha3-perl, in Fedora, install sha3sum

If available, install rhash - $rhash --sha3-512 /bin/echo

Busybox has sha3sum, $sha3sum -a 512 “filename”

# NIST SP 800-171

## What is the NIST SP 800-171?

The NIST SP 800-171 is shorthand for the National Institute of Standards and Technology Special Publication 800-171, Security and Privacy Controls for Federal Information Systems and Organization. NIST SP 800-171 provides recommended security requirements for protecting the confidentiality of controlled unclassified information (CUI) governed by the Federal Acquisition Regulation (FAR) and Defense Federal Acquisition Regulation Supplement (DFARS).

# NIST SP 800-171

## What is the purpose of NIST SP 800-171?

Created by computer security and privacy experts at the NIST, NIST Special Publication 800-171, Revision 2: Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations contains recommended security requirements for protecting the confidentiality of CUI when the information resides in nonfederal systems and organizations; when the nonfederal organization is not collecting or maintaining information on behalf of a federal agency or using or operating a system on behalf of an agency; and where there are no specific safeguarding requirements for protecting the confidentiality of CUI prescribed by the authorizing law, regulation, or governmentwide policy for the CUI category listed in the CUI Registry.

# CUI - Controlled Unclassified Information

## What are the requirements for NIST SP 800-171?

CUI, or controlled unclassified information, is government-created or owned information that requires safeguarding or dissemination controls consistent with applicable laws, regulations, and government-wide policies. It's also not corporate intellectual property unless created for or included in requirements related to a government contract.

Up until 2010, CUI wasn't even CUI — it appeared under an assortment of names mentioned previously, like "for official use only" and "sensitive but unclassified." More troubling was that no standardized guidelines existed for assessing CUI — one company could label information extremely sensitive while another could treat it as less sensitive. Related: Guide to NIST 800-53

# NIST SP 800-171

## Who Needs to Comply With NIST SP 800-171?

If you're a contractor for a federal agency and your organization is processing CUI, you may be contractually obligated by the agency to implement the requirements recommended in SP 800-171. To be clear, these security requirements would apply to the components of your environment that process, store, or transmit CUI or that provide security protection for such components.

# NIST SP 800-171

## Who Needs to Comply With NIST SP 800-171?

Here's the exact language from NIST SP 800-171 rev 2:
"The recommended security requirements contained in this publication are only applicable to a nonfederal system or organization when mandated by a federal agency in a contract, grant, or other agreements. The security requirements apply to the components of nonfederal systems that process, store, or transmit CUI, or that provide security protection for such components."

If you're seeking to secure a contract with a federal agency, during the evaluation process the agency is likely to ask you to submit a system security plan, a comprehensive document that describes in detail how security requirements in NIST SP 800-171 are met within your organization and how you plan to address known and anticipated threats. Federal agencies consider the submitted system security plans and associated plans of actions as critical inputs in their decision on whether it would be advisable to enter an agreement with the nonfederal organization.

# NIST 800-171 and FedRAMP

NIST SP 800-171 requirements cover about 35% of the broader NIST SP 800-53 controls needed for FedRAMP compliance. CSPs who aim to offer services to government agencies must meet FedRAMP standards to get their Authority to Operate (ATO) status.

**What are the security requirements of FedRAMP?**

CSPs need to follow security rules from NIST Special Publication 800-53. This guide lists all the security controls and statements CSPs should use in their systems. It covers things like who can access what, how to handle problems, and making sure systems and info are safe and sound.

**How does FedRAMP map to NIST controls?**

CSPs need to show they meet these security requirements from NIST, which include areas like who can access things, what to do if there’s a problem, and keeping systems and info safe.

# What about FIPS 140-2, 140-3 and CMMC?

FIPS 140-2 defines many cryptographic modules that are now considered weak and not recommended. FIPS 140-3 is better. Organizations that need to be compliant with NIST 800-171 must employ FIPS 140-2 or 140-3 crypto. Modules

The Cypersecurity Maturity Model Certification (CMMC) was released in 2020, v2.0 by DoD 11/2021. The CMMC has 3 levels:

Level 1 - (equivalent to CMMC 1.0) Foundational

Level 2 - Advanced

Level 3 - Expert

https://www.kratosdefense.com/-/media/k/pdf/s/c/nist-800-171-requirements-for-validated-cryptographic-modules.pdf

# FIPS 140-2 timeline

As of 4/1/2022, FIPS 140-3 supersedes FIPS 140-2 for new submissions, but 140-2 validated modules are valid until 9/21/2026.

Validated OpenSSL FIPS 140-2 modules include 3.0.8 and 3.0.9

Validated OpenSSL FIPS 140-3 versions include 3.1.2

FIPS/TLS - SHA1 is prohibited, no TLSv1.0/v1.1

# Reference Links


1. https://trends.builtwith.com/ssl/traffic/Entire-Internet
2. https://www.ssldragon.com/blog/ssl-stats/
3. https://w3techs.com/technologies/overview/ssl_certificate
4. https://www.digicert.com/difference-between-dv-ov-and-ev-ssl-certificates
5. https://community.citrix.com/tech-zone/build/tech-papers/key-exchange-in-ssl-tls/#_=_
6. https://blog.cloudflare.com/pq-2024/
7. https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption
8. https://blog.aegrel.ee/kyber-nginx.html
9. https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8545.pdf
10. https://www.helpnetsecurity.com/2025/04/09/openssl-3-5-0-released/
11. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf
12. https://openssl-corporation.org/post/2025-03-11-fips-140-3/