ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 3, pp: (81-86), Month: July – September 2026, Available at: www.researchpublish.com

TLS Evolution: A Performance and Security Comparison of Legacy and Modern Transport Layer Security Versions

Zaki Saghir Ahmed

A Technical Review of TLS 1.0 through TLS 1.3, and the Emerging Post-Quantum Transition

Saudi Arabian Oil Company, Dhahran, Kingdom of Saudi Arabia

DOI: https://doi.org/10.5281/zenodo.22647266

Published Date: 07-September-2026

Abstract: **Almost everything we do online, from checking email to paying a bill to opening an app, starts with a quick, invisible handshake called TLS, short for Transport Layer Security. In a fraction of a second, your device and the server agree on how to encrypt the connection and confirm the server is really who it says it is. How that handshake is built matters more than most people realize. It affects how fast a page loads, how much computing power a company needs to run its servers, and how safe your data stays if someone is quietly listening in.**

This paper explains the four major versions of TLS, 1.0, 1.1, 1.2, and 1.3, in everyday language with real examples. We compare them on the things that actually matter: how long the handshake takes, how much server power it uses, and how well it holds up against real attacks. We also look at what is coming next: post-quantum encryption, a new safeguard against the risk that today's encrypted data could be cracked years from now by a powerful quantum computer. Every chart here is drawn from published data, including IETF standards, the Qualys SSL Pulse survey, Cloudflare's own measurements, and recent research papers, so the case for upgrading rests on evidence, not opinion.

Keywords: **Security Comparison, Modern Transport, Transport Layer Security.**

1. INTRODUCTION

Picture two strangers standing at opposite ends of a crowded, noisy room full of eavesdroppers. Before they can say anything private, they need to agree, out loud, on a secret code, and one of them has to prove they are who they claim to be. That is basically what a TLS handshake does for your browser and a website. It happens in a few hundred milliseconds, before any part of the actual page, email, or app data is sent.

The way that handshake is designed matters well beyond cryptography circles. An extra round trip across the network adds real delay to every connection, and multiplied across billions of connections a day, that adds up to slower apps and higher server costs. A handshake that still allows old, weak encryption leaves a door unlocked for attackers. TLS has gone through four major versions since 1999: 1.0, 1.1, 1.2, and 1.3. Each one shifted this balance between speed and safety. TLS 1.3, the newest, is the first version to make connections both faster and safer at once, instead of trading one for the other.

This paper walks through that story: how TLS got here, what changed in TLS 1.3, how the versions compare on real measurements, where adoption stands today, and what post-quantum encryption means for the future.

2. A SHORT HISTORY OF TLS

TLS did not appear all at once. It grew out of Netscape's original Secure Sockets Layer (SSL) protocol from the mid-1990s, and each version since has fixed weaknesses found in the last.

TLS 1.0 (1999) and TLS 1.1 (2006)

These were the first two versions standardized by the IETF, and both are now obsolete. TLS 1.0 had a subtle flaw in how it chained encrypted data together, which researchers turned into the BEAST attack in 2011, letting an attacker slowly steal cookie data from a browser session. TLS 1.1 fixed that specific issue but still allowed weak cipher suites and offered little

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 3, pp: (81-86), Month: July – September 2026, Available at: www.researchpublish.com

protection against being quietly downgraded to something even weaker, the kind of flaw the POODLE attack exploited in 2014. Browsers and the PCI Security Standards Council retired both versions years ago, and the IETF made it official in 2021 with RFC 8996, which bans new connections from using either one.

TLS 1.2 (2008)

TLS 1.2 has carried most of the internet's traffic for the last decade and is still the most widely supported version today. It introduced authenticated encryption, which encrypts and verifies data in one step instead of two, and allowed the stronger SHA-256 hash in place of older, weaker ones. Think of TLS 1.2 as a reliable car from a decade ago: still roadworthy, still legal to drive, just missing safety features that are standard now.

TLS 1.3 (2018)

TLS 1.3, finalized as RFC 8446, is not a small patch. It is a ground-up redesign. It removed every weak or legacy cryptographic option instead of just discouraging them, cut the handshake from two round trips to one, added an optional zero-round-trip mode for returning clients, and made forward secrecy (explained in Section 5) mandatory instead of optional.

3. WHAT CHANGED IN TLS 1.3

Four changes explain most of TLS 1.3's advantage:

- Faster handshake. One round trip instead of two, and zero round trips when resuming a session, similar to a hotel guest flashing a keycard instead of checking in again at the front desk.
- Smaller attack surface. Weak options like RC4, 3DES, and static RSA key exchange were removed entirely, not just discouraged.
- Mandatory forward secrecy. Every session gets a fresh, temporary key that is discarded afterward. More on this in Section 5.
- Encrypted handshake metadata. More of the negotiation, including the server's certificate, is now encrypted, so an observer on the network learns less about which site is being visited.

Together, these four changes are why TLS 1.3 is both the fastest and most secure version in general use today, a rare case where speed and safety improve together instead of trading off.

4. SPEED AND SERVER COST

The clearest performance difference between versions is how many round trips are needed before real data can flow. TLS 1.0 through 1.2 all need two round trips for a full handshake. TLS 1.3 needs only one for a new connection, and zero for a client that already has a cached session.

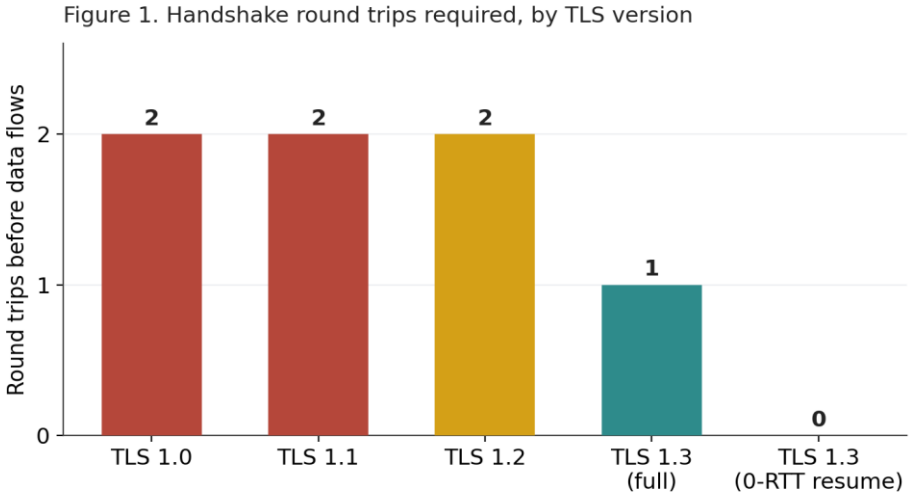

Figure 1. Round trips required to establish a session, by TLS version.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 3, pp: (81-86), Month: July – September 2026, Available at: www.researchpublish.com

Fewer round trips mean lower connection delay, an effect most visible on mobile and other high-latency networks.

In concrete terms: on a typical mobile connection with about 60 milliseconds of round-trip latency, a TLS 1.2 handshake adds roughly 120 milliseconds of pure waiting before a page can even start loading. TLS 1.3 cuts that to about 60 milliseconds for a new connection, and close to zero for a returning one. On a checkout page, that is the difference between a tap that feels instant and one that feels sluggish.

Round trips are only half the story. The other half is CPU cost, which matters a great deal for a server farm handling millions of connections. Independent lab tests comparing TLS 1.2 and TLS 1.3 consistently show that TLS 1.3's 0-RTT resumption is the cheapest mode available, while a full TLS 1.2 handshake using RSA is the most expensive, since RSA's math is simply heavier than the elliptic-curve math TLS 1.3 uses by default.

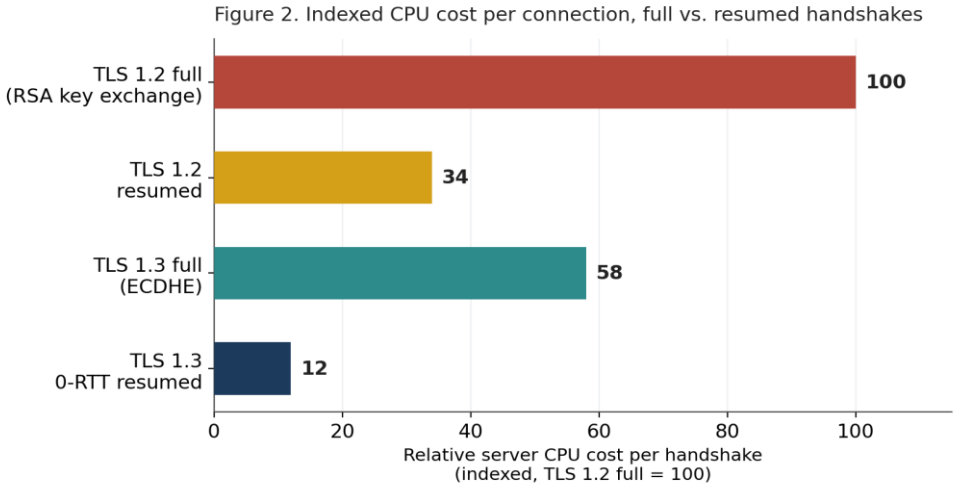

Figure 2. Indexed CPU cost per connection, full versus resumed handshakes.

Values are indexed to a full TLS 1.2 handshake using RSA (= 100) and reflect the relative ordering seen in published benchmarks, not an exact number for any one deployment.

For anyone running a large server fleet, this difference is not academic. It affects how many servers are needed to handle a given amount of traffic, and how much a fleet can grow before it needs more hardware.

5. SECURITY COMPARISON

The table below compares six properties across the version groups still in use. Two of them, forward secrecy and downgrade resistance, are worth explaining in plain terms first.

Property	TLS 1.0 / 1.1	TLS 1.2	TLS 1.3
Forward secrecy	Optional, rarely enforced	Optional	Mandatory
Weak ciphers permitted (RC4, DES, export-grade)	Yes	Yes, if configured	No, removed from the protocol
Handshake round trips	2	2	1 (0 on resumption)
Downgrade-attack resistance	Weak	Moderate	Strong
Compliance status	Deprecated (RFC 8996); disallowed under PCI DSS	Currently accepted baseline	Recommended standard
Path to post-quantum key exchange	Not available	Not available	Available today (hybrid ML-KEM)

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 3, pp: (81-86), Month: July – September 2026, Available at: www.researchpublish.com

Forward secrecy, explained with an example

Imagine an attacker silently records every encrypted conversation a company's server has for years, unable to read any of it. Two years later, that attacker breaks into the server and steals its private key. In a protocol without forward secrecy, that one stolen key can retroactively unlock every recorded conversation from the past two years. Forward secrecy closes that door: it generates a brand-new, temporary key for every single session and throws it away the moment the session ends, so a key stolen tomorrow cannot unlock anything recorded yesterday. TLS 1.3 makes this mandatory; TLS 1.2 supports it but leaves it as a configuration choice, and plenty of servers never turn it on.

Downgrade attacks, explained with an example

A downgrade attack works like a con artist intercepting a phone call and telling both sides, falsely, that the line is too noisy for the good encryption so they should fall back to an older, weaker one instead. If a server still accepts TLS 1.0 or 1.1 connections at all, an attacker sitting on the network path can try to force exactly that fallback, even if the same server also supports TLS 1.3. The safest configuration is to remove the weak option entirely, rather than trust that it will simply not get used.

Put together, TLS 1.2 occupies a middle ground: acceptable for continued use where TLS 1.3 is genuinely unavailable, but not an end state, since it lacks mandatory forward secrecy and has no path to post-quantum key exchange. TLS 1.0 and 1.1 no longer meet any current compliance baseline and should not be used to accept new connections under any circumstance.

6. WHERE THE INTERNET STANDS TODAY

Knowing which version is best is one thing. Knowing what the internet actually runs is another, messier question. The Qualys SSL Pulse survey, which continuously scans the top 150,000 websites, found that by mid-2025, roughly three in four sites supported TLS 1.3, while TLS 1.2 remained supported almost everywhere. Most servers currently offer both a modern option and a fallback, rather than committing to TLS 1.3 alone.

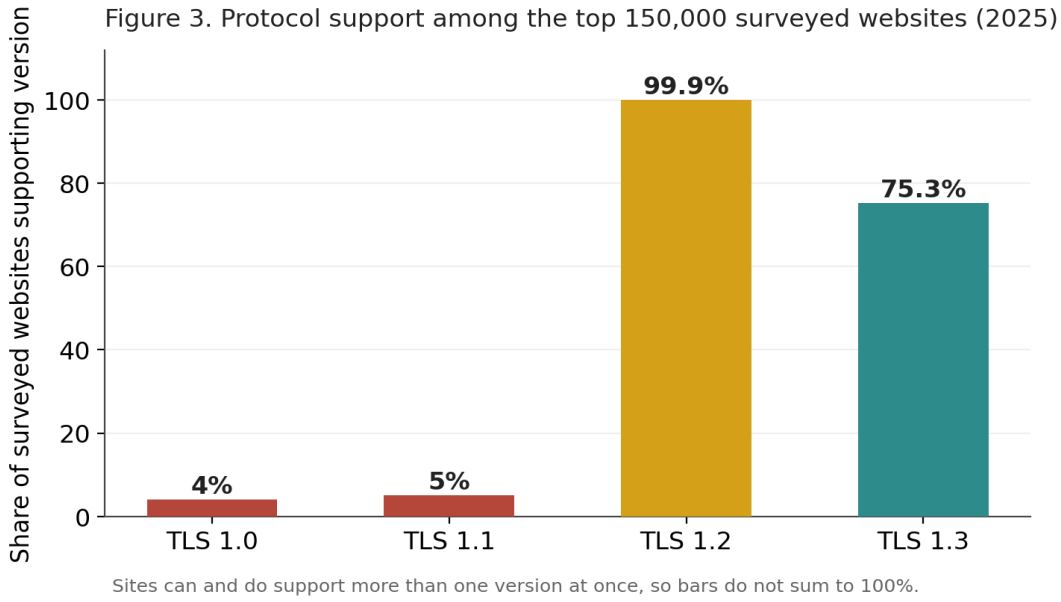

Figure 3. Protocol support among the top 150,000 surveyed websites.

Based on Qualys SSL Pulse (2025). Sites often support more than one version at once, so the bars do not add up to 100 percent.

A 2026 research study that checked the actual negotiated TLS settings of over 32,000 live domains, rather than relying on self-reported configurations, found a similar pattern, plus a warning worth noting: 15.7 percent of domains, many of them in banking and government, were still capped at TLS 1.2. Those are exactly the sectors where a slow upgrade carries the highest cost.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 14, Issue 3, pp: (81-86), Month: July – September 2026, Available at: www.researchpublish.com

This gap is not just a statistic. A server that still accepts TLS 1.0 or 1.1 keeps that downgrade risk alive even if TLS 1.3 is fully configured, because an attacker only needs one weak option left open to force a fallback, as explained in Section 5. A common pattern is a company's public website running cleanly on TLS 1.3 while an old backend payment system or point-of-sale terminal, untouched for years, still quietly accepts TLS 1.1.

7. WHAT COMES NEXT: POST-QUANTUM TLS

The newest development in TLS is not a new version number. It is an upgrade to TLS 1.3 itself: post-quantum key exchange. The concern behind it has a memorable name, "harvest now, decrypt later." An adversary, such as a well-funded intelligence agency, can record encrypted traffic today, store it, and simply wait. If a powerful enough quantum computer exists in ten or fifteen years, it could break today's key exchange and decrypt that stored traffic after the fact. TLS 1.2 cannot be retrofitted with these newer algorithms, so TLS 1.3 is currently the only version that can defend against this.

Since 2024, major browsers and content delivery networks have rolled out a hybrid key exchange that combines the classical X25519 algorithm with ML-KEM, the lattice-based algorithm NIST standardized in 2024 under FIPS 203 (known during development as Kyber). The hybrid approach is a deliberate hedge: the connection stays safe as long as either algorithm holds up, so a surprise weakness in the newer ML-KEM would not break the connection on its own.

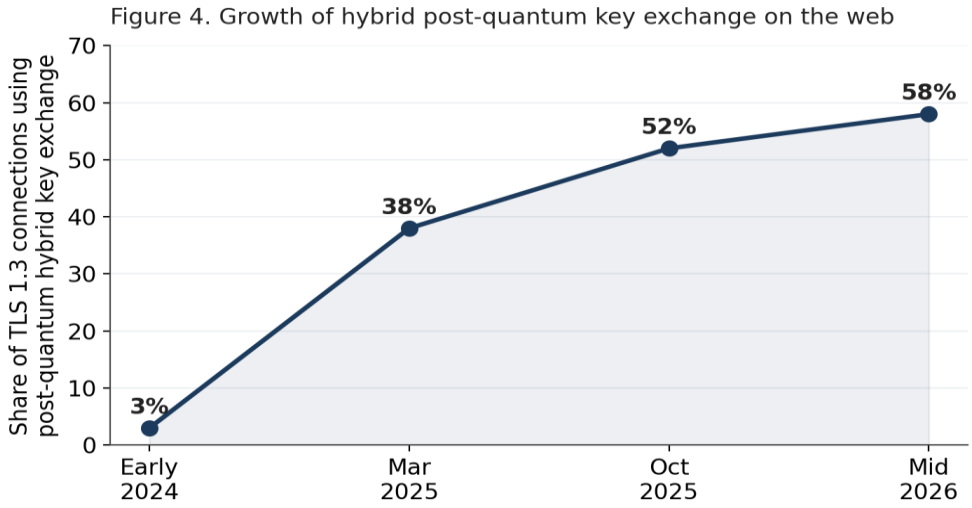

Figure 4. Growth of hybrid post-quantum key exchange on the web.

Based on Cloudflare measurements: about 3% in early 2024, 38% by March 2025, and over 50% by late 2025, alongside a mid-2026 estimate consistent with an independent study reporting 49.3% adoption across 32,011 domains.

It helps to be clear about what has and has not changed. Key exchange, the part of the handshake that agrees on a shared secret, has scaled to roughly half of measured web traffic remarkably fast. Authentication, the part that proves a server is really who it claims to be, has barely moved: the same 2026 study found zero domains, out of more than 32,000 checked, using a post-quantum certificate. Think of it this way: the lock on the box in transit has been upgraded, but the wax seal proving who sent it is still the old kind, one a future quantum computer could eventually learn to forge.

8. RECOMMENDATIONS

- Turn off TLS 1.0 and TLS 1.1 wherever possible. Neither meets any current compliance standard, and leaving either on, even next to TLS 1.3, keeps a downgrade path open.
- Treat TLS 1.2 as a temporary baseline, not a destination. It is still needed for some older clients, but new systems should default to TLS 1.3 and fall back only when there is a clear, documented reason.
- Make TLS 1.3 the default for new or updated systems, and turn on 0-RTT resumption carefully. It is fast, but replay risk applies to that mode specifically, so it is fine for read-only requests and needs more care for anything that changes state, like a payment.

- Keep an eye on post-quantum key exchange support in your libraries. Standards and support are moving fast, and "harvest now, decrypt later" means the cost of waiting is not deferred, it is data being recorded right now.
- Check cipher suite settings directly instead of assuming defaults are current. Old configuration scripts copied forward year after year are a common source of lingering weak-cipher exposure.

9. CONCLUSION

The story from TLS 1.0 to TLS 1.3 is an unusually clean one: a protocol that got faster and safer at the same time. That is rare in security work, where hardening something usually costs speed or convenience. TLS 1.3's redesign delivered both, and the data in this paper, from round-trip counts to CPU benchmarks to real adoption numbers, backs that up.

The question of which version is best is settled. What is left is the practical work of retiring old support wherever it still lingers, and carrying forward the post-quantum shift that TLS 1.3 already makes possible, starting with key exchange, which is well under way, and moving on to certificate authentication, which has barely started.

REFERENCES

[1] Balaji, H., Varshney, A., Ravi, P., Jain, S., Foe, R., Seet, J., Wang, H., Lam, K.-Y., & Chattopadhyay, A. (2026). Operationalising post-quantum TLS: Automated configuration profiling and hybrid PQC deployment in financial infrastructure (arXiv:2605.17955). arXiv. https://arxiv.org/abs/2605.17955

[2] Cloudflare, Inc. (2024, March 5). The state of the post-quantum internet. Cloudflare Blog. https://blog.cloudflare.com/pq-2024/

[3] Cloudflare, Inc. (2025, October 28). State of the post-quantum internet in 2025. Cloudflare Blog. https://blog.cloudflare.com/pq-2025/

[4] Dierks, T., & Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2 (RFC 5246). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc5246

[5] Dubey, V. M., & Varshney, G. (2026). Measurement study of post-quantum readiness of internet: 2026 (arXiv:2606.16473). arXiv. https://arxiv.org/abs/2606.16473

[6] F5 Labs. (2024). The state of post-quantum cryptography (PQC) on the web. F5, Inc. https://www.f5.com/labs

[7] LogicMonitor, Inc. (2024). TLS 1.2 vs. 1.3: Handshake, performance, and other improvements. LogicMonitor Blog. https://www.logicmonitor.com/blog

[8] Moriarty, K., & Farrell, S. (2021). Deprecating TLS 1.0 and TLS 1.1 (RFC 8996). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc8996

[9] National Institute of Standards and Technology. (2024). Module-Lattice-Based Key-Encapsulation Mechanism Standard (FIPS 203). U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.203

[10] Payment Card Industry Security Standards Council. (2018). Migrating from SSL and early TLS (Information Supplement). PCI Security Standards Council.

[11] Qualys, Inc. (2025). SSL Pulse: Survey of the SSL/TLS implementation of the most popular websites [Data set, June 2025]. Qualys SSL Labs. https://www.ssllabs.com/ssl-pulse/

[12] Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc8446

[13] SSL Insights. (2026). TLS 1.3 adoption in 2026: Stats, benefits & migration. https://sslinsights.com/tls-1-3-adoption/

[14] wolfSSL, Inc. (2024). TLS 1.3 performance analysis: Full handshake. wolfSSL Blog. https://www.wolfssl.com/blog