Addendum

There is no Entente Cordiale in PQC yet!

Expanding the Solution Space: wolfSSL, Chimera Certificates, and the Post-Quantum Transport Gateway

Santosh Pandit

London, 22 March 2026

Addendum to: *"There is no Entente Cordiale in PQC yet!"* (21 March 2026)

Purpose of this Addendum

On 21 March 2026, I published *“There is no Entente Cordiale in PQC yet!”*, arguing that the regulatory divergence between the UK/USA preference for pure post-quantum signatures and the French/European requirement for hybrid signatures creates a practical interoperability gap for cross-border B2B authentication. I presented a dual-certificate workaround using RFC 9763 and identified that the absence of hybrid signature support in OpenSSL leaves the Linux/C ecosystem without a native solution.

For my technical readers, who wish to explore the solution space even further, I want to mention two other solutions. In addition, I will include a brief clarification on DNSSEC based risk mitigation.

wolfSSL and X9.146 Chimera Certificates

My original paper stated that hybrid or composite signature capability exists only in the Java ecosystem (Bouncy Castle), leaving the C and Linux world unable to satisfy ANSSI’s requirements natively. This is not entirely accurate.

wolfSSL, a C-native cryptographic library designed for embedded and high-performance environments, implements X9.146 Chimera certificates, which are single X.509 certificates containing two public keys and two signatures via three X.509 extensions defined in the 2019 edition of ITU-T X.509: Subject Alternative Public Key Info (SAPKI), Alternative Signature Algorithm, and Alternative Signature Value.

Critically, wolfSSL also implements the TLS 1.3 CKS (Certificate Key Selection) extension defined in the X9.146 banking standard draft, which allows the TLS handshake itself to negotiate which signature algorithm the peer should verify. This addresses the TLS-layer limitation I identified in my paper, where standard TLS 1.3 supports only a single certificate chain per handshake, forcing dual-certificate verification to the application layer.

Honest Qualifications

wolfSSL is not OpenSSL. It is not the default cryptographic library on Debian, Ubuntu, Red Hat, or most other Linux distributions. Organisations with existing infrastructure built on OpenSSL cannot substitute wolfSSL without significant integration work. For example, I use nginx, curl, and openssl. wolfSSL does provide integration with several of these components (curl, Apache, nginx, Lighttpd, Stunnel), but adopting it remains a deliberate infrastructure decision that goes beyond compliance timelines.

The CKS TLS extension is defined in the X9.146 draft, not yet in an IETF RFC, so its standardisation path is still in progress. Additionally, the Chimera certificate extensions are non-critical by design, meaning legacy clients will simply ignore the alternative signature rather than enforcing dual verification. This is elegant for

backward compatibility but may not satisfy the “both must hold” property that ANSSI requires without explicit application-layer enforcement.

Whether OpenSSL 3.5 can parse or verify Chimera certificate extensions is, to my knowledge, not yet established. If it cannot, then interoperability between wolfSSL endpoints and OpenSSL endpoints remains limited for hybrid authentication. I would welcome clarification from the OpenSSL community on this point.

A better way to look at the gap

My original paper presented two approaches: IETF composite signatures (Bouncy Castle, Java-only) and RFC 9763 dual certificates (OpenSSL, workaround). The solution space should properly include a third: X9.146 Chimera certificates (wolfSSL, C-native, with TLS 1.3 CKS extension support). This narrows the interoperability gap. The problem is no longer “no C library supports hybrid signatures” but rather “the dominant C library (OpenSSL) does not support hybrid signatures, and alternative C libraries require infrastructure migration.” I intend to test Chimera certificates in my lab.

The Post-Quantum Transport Gateway for QKD Infrastructure

My original paper focused on general B2B mutual authentication across the English Channel. For organisations operating in the higher-stakes niche where Quantum Key Distribution hardware is already deployed or budgeted, a separate but related vulnerability deserves mention.

The Post-Quantum Transport Gateway (PQTG), described in a whitepaper by Sylvain Cormier of Paraxiom Technologies (September 2025, revised February 2026), addresses a specific and well-defined problem: every QKD deployment compliant with the ETSI GS QKD 014 key delivery API uses classical TLS with RSA or ECDSA certificates to authenticate its control channel - the channel that transports quantum-derived keys between endpoints. A future quantum computer can forge these certificates and extract every key transported over that channel, regardless of the quantum channel’s information-theoretic security.

PQTG replaces this classical authentication with NIST-standardised post-quantum primitives (ML-KEM-768, Falcon-512, SPHINCS+-256f) via a transparent localhost gateway that requires no vendor firmware modifications. It deploys on existing QKD hardware from Nokia, ID Quantique, Toshiba, and QuantumCTek in under one hour per endpoint.

Relevance to the Entente Cordiale Problem

PQTG is niche by definition - it applies only where QKD hardware exists. But the environments deploying QKD (national quantum networks, banking infrastructure, defence communications) are precisely the environments where the Anglo-French regulatory divergence I described is most consequential. For these deployments, PQTG provides an additional layer of post-quantum authentication that is

independent of both the OpenSSL and wolfSSL ecosystems, operating as a self-contained gateway rather than a library substitution.

One design decision in PQTG is worth noting for its relevance to the monoculture risk I discussed in my paper. PQTG uses a dual-signature construction: Falcon-512 for fast online session authentication and SPHINCS+-256f for long-term certificate security. These rest on different mathematical foundations (lattice-based and hash-based respectively), providing algorithmic diversity within a single system. This is the same principle I would recommend for any PQC deployment that relies solely on ML-DSA: introduce a second signature scheme from a different mathematical family to hedge against a breakthrough in lattice cryptanalysis.

Scope and Limitations

Critical evaluation of PQTG reveals that its primary value is converting a passive, scalable, undetectable harvest-now-decrypt-later threat into one requiring expensive, targeted, detectable real-time intrusion. With well-configured TLS 1.3 using post-quantum ephemeral key exchange, AEAD, and short-lived certificates with disciplined rotation, the retrospective bulk decryption scenario is already substantially mitigated. PQTG's residual contribution is preventing real-time endpoint impersonation once quantum computers arrive - a narrower but still severe threat for high-value QKD infrastructure.

PQTG is open source under the MIT license and is available at github.com/Paraxiom/pq-transport-gateway. In my view it is worth exploring its adoption for deployments where QKD investment budgets exist.

Caveat on DNSSEC, DANE/TLSA, and CAA

In my original paper, I recommended DNSSEC, DANE/TLSA, and CAA as defences against man-in-the-middle attacks. This recommendation is sound against classical threats, and I continue to advocate for these technologies. However, in a paper specifically about post-quantum readiness, an important caveat must be stated.

The entire DNSSEC chain of trust (from the root Key Signing Key down through TLD and authoritative zone signatures) currently uses classical digital signatures (RSA-2048 or ECDSA). Shor's algorithm breaks these in the same way it breaks TLS certificate signatures. An adversary with a cryptographically relevant quantum computer could derive the DNSSEC root zone's private key from its published public key, forge zone-signing keys down the delegation chain, publish fraudulent TLSA records pointing to forged certificates, and present a consistent, valid chain to any verifier.

CAA is even less relevant in this scenario, as it is a policy signal enforced by CA compliance rather than cryptographic verification. An adversary who can derive a CA's private key via Shor's algorithm never interacts with any CA and therefore never encounters the CAA record.

DNSSEC, DANE, and CAA relocate trust from the CA model to the DNS hierarchy. Both hierarchies share the same fundamental quantum vulnerability. My recommendation of these technologies for MITM defence therefore carries an implicit

dependency: DNSSEC itself must migrate to post-quantum signature algorithms before a cryptographically relevant quantum computer becomes operational.

I am actively working with a few stakeholders (e.g. IS3C) to motivate this migration, but I should have stated this dependency explicitly in my original paper. The window for DNSSEC post-quantum migration is constrained by the same timeline uncertainty as all other PQC transitions, with the additional complication that DNSSEC infrastructure is globally coordinated and not under any single organisation's control.

Summary: Expanded solution space

If you wish to explore the dual certificate mTLS solution, see my original paper.

If you are happy to complement your openssl stack with wolfSSL, you should explore the CHIMERA certification solution. I will try that myself in the lab.

If your stakes are high and have the QKD budget, PQTG comes across as a no-brainer addition.

The core argument of my original paper remains unchanged: a modern PQC Entente Cordiale is needed, and the regulatory divergence between the UK/USA and France/Europe creates a harder problem for cross-border deployment than any technical limitation. But the technical landscape is richer than I initially described, and intellectual honesty requires saying so.

The solution space is evolving, and it is likely that even this Addendum will prove incomplete very soon. But let us not forget the elephant in the room – all engineers are burning their weekends – to solve one problem i.e. the lack of interoperability.

Acknowledgments

I thank Anthony Hu of wolfSSL for directing me to the X9.146 Chimera certificate work and the wolfSSL examples repository. I also acknowledge the analysis of the PQTG whitepaper by Sylvain Cormier of Paraxiom Technologies, which informed the discussion of QKD control channel vulnerabilities in this addendum.

Santosh Pandit

London, 22 March 2026

PS: I know this is a weekend.

References

Note: All links were working when I checked on 22 March 2026. If they stop working, please use the Internet Archive or 404wayback or contact the respective authors.

1. S. Pandit, "There is no Entente Cordiale in PQC yet!", London, 21 March 2026.

2. wolfSSL, "Chimera Certificate Standards Compliance," wolfssl.com, May 2025.

3. wolfSSL, "X9.146 Examples," github.com/wolfSSL/wolfssl-examples/tree/master/X9.146.

4. wolfSSL, "X.509 Alternative Public Key and Signature," wolfssl.com, January 2024.

5. ITU-T X.509 (2019 Edition), Information technology - Open systems interconnection - The Directory: Public-key and attribute certificate frameworks. (Note: This page has a big collection, and I suggest you bookmark it)

6. S. Cormier, "Post-Quantum Authentication for Quantum Key Distribution Control Channels," Paraxiom Technologies, September 2025 (revised February 2026).

7. qssl : GitHub page.

8. RFC 9763, Related Certificates for Use in Multiple Authentications within a Protocol, A. Becker, R. Guthrie, M. Jenkins, IETF, June 2025.

9. PKI Consortium, "X9.146 Quantum TLS," Austin, 2025.

10. RFC 9881, Internet X.509 Public Key Infrastructure - Algorithm Identifiers for ML-DSA, IETF, October 2025.

11. draft-ietf-lamps-pq-composite-sigs-15, Composite ML-DSA for use in X.509 Public Key Infrastructure, M. Ounsworth et al., February 2026.

12. ETSI GS QKD 014 V1.1.1, Quantum Key Distribution (QKD); Application Interface, 2023.