X.509 Compliant Hybrid Certificates for the Post-Quantum Transition

Nina Bindel, Johannes Braun, Luca Gladiator, Tobias Stöckert, and Johannes Wirth

Technische Universität Darmstadt
Cryptography and Computer Algebra

Summary

We provide an X.509 standard compliant Java implementation of hybrid certificates, which enable the parallel usage of two independent cryptographic schemes within public key infrastructures and related applications. This enables a stepwise transition to post-quantum secure and hybrid algorithms without the risk of incompatibility problems.

Motivation and statement of need

Public Key Infrastructures (PKIs) support the use of public-key cryptography by handling keys and providing public-key certificates. The most common approach is the use of hierarchical PKIs, where certificates are issued by Certification Authorities (CAs) according to the X.509 standard [7]. These certificates bind the key owner's identity (e.g. a name) to their public key and hence, enable the authentication of public keys. This is a basic prerequisite for the use of digital signatures and public key encryption in applications such as e-business or e-government that require secure electronic communication. The most prominent example is secure Internet communication using the Transport Layer Security (TLS) protocol.

The security of current public-key systems, e.g., RSA and elliptic curve cryptography, depends on the computational difficulty of factoring large numbers into their prime factors or computing discrete logarithms. These schemes are called classical in the remainder. While the security guarantees of classical schemes are sufficient today, large quantum computers could break almost all public-key algorithms currently used by applying Shors algorithm [9], rendering anything protected by them vulnerable to exploitation. Therefore, post-quantum cryptography, i.e. cryptography that is secure even in the presence of quantum computers, is required and needs to be integrated into applications.

To ensure uninterrupted cryptographic security, it is important to begin the transition to post-quantum cryptography today. Post-quantum secure algorithms already exist, e.g., qTESLA [5], and can be used as substitutes for classical schemes. However, to facilitate the transition, also the cryptographic infrastructure must be adapted. One approach for a secure and smooth transition is the use of hybrids–multiple algorithms in parallel that are combined such that the hybrid scheme is secure as long as at least one of the parallely used algorithms is secure. For the post-quantum transition a classical scheme is combined with a post-quantum scheme. This has two clear advantages compared to a direct switch to post-quantum secure algorithms: "hedging our bets" when the security of newer algorithms is not yet certain but the security of older primitives is already in question; and to achieve security and functionality both in post-quantum-aware and in a backwards-compatible way with not-yet-upgraded software.

Hybrid certificates

Hybrid certificates [6] are the basis for the use of hybrid cryptography. They are signed in parallel with two different signature schemes and additionally bind two independent public keys–one classical and one post-quantum key–to one identity. Thereby, the authentication of the public keys contained in the certificate is protected with the combined security of both signature schemes. Moreover, the secure parallel usage of two different schemes for authentication and key exchange purposes is enabled.

We realize these hybrid certificates fully compliant to the X.509 standard [7]. The standard signature and public-key fields of the X.509 certificate are used for one of the signature schemes. For the post-quantum transition, the standard fields are used for the classical scheme. This allows compatibility with clients not supporting hybrid signatures. The second signature scheme, using qTESLA as an example, is integrated using two non-critical X.509 extensions. One of the extensions contains the public key associated with the second scheme, while the other contains the second signature on the certified data. To fully support legacy entities in a controlled manner, the extension containing the second public key may optionally be left out. This explicitly states that the certified entity does not support post-quantum schemes yet, while the certificate contents themselves are still protected in a hybrid fashion.

Implementation and features

We provide a Java implementation for BouncyCastle [1] available at `https://github.com/CROSSINGTUD/bc-hybrid-certificates` [3] that comprises generation procedures required to issue standard compliant hybrid certificates as well as path validation procedures for certification chains. Note that non-upgraded software evaluates hybrid certificates just as classical X.509 certificates. Therefore, full backwards-compatibility is provided, while falling back on the security of the classical signature scheme in these cases. Further technical details on the definition of the extensions, certificate generation, and path validation procedures can be found in the technical documentation [8].

The submitted Java implementation is independent, but fully compatible to a C implementation for the OQS OpenSSL fork [4], which integrates LIBOQS [4], a C library for quantum-safe cryptographic algorithms, into OpenSSL. The

C version also implements hybrid certificates and can be accessed at https://github.com/CROSSINGTUD/openssl-hybrid-certificates [2].

Our implementation enables a stepwise transition to post-quantum secure and hybrid algorithms in compliance with existing standards and software. Hence, our software enables first uses and experiments with such algorithms in (parts of) real-life applications and systems without the risk of incompatibility problems due to unforeseen dependencies. These experiments in turn allow to identify limits and potential obstacles requiring adaptations for a smooth and secure transition.

Acknowledgments

This work has been co-funded by the DFG as part of project P1 within the CRC 1119 CROSSING.

References

1. The Legion of the Bouncy Castle. https://www.bouncycastle.org/, 2013. Online; accessed 2019-07.
2. Hybrid Certificates - C, OpenSSL integration. https://github.com/CROSSINGTUD/openssl-hybrid-certificates, 2019. Online; accessed 2019-07.
3. Hybrid Certificates - Java, Bouncy Castle integration. https://github.com/CROSSINGTUD/bc-hybrid-certificates, 2019. Online; accessed 2019-07.
4. Open Quantum Safe - software for prototyping quantum-resistant cryptography. https://openquantumsafe.org/, 2019. Online; accessed 2019-07.
5. N. Bindel, S. Akleylek, E. Alkim, P. S. L. M. Barreto, J. Buchmann, E. Eaton, G. Gutoski, J. Kramer, P. Longa, H. Polat, J. E. Ricardini, and G. Zanon. qTESLA. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/qTESLA-Round2.zip.
6. N. Bindel, U. Herath, M. McKague, and D. Stebila. Transitioning to a quantum-resistant public key infrastructure. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography*, pages 384–405, Cham, 2017. Springer International Publishing.
7. D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008. Updated by RFC 6818.
8. L. Gladiator. Hybrid Certificates for OpenSSL. https://github.com/CROSSINGTUD/openssl-hybrid-certificates/blob/OQS-OpenSSL_1_1_1-stable/HybridCert_technical_documentation.pdf, 2019. Online; accessed 2019-07.
9. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Nov 1994.