Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 September 2022

A. Becker
R. Guthrie
M. Jenkins
NSA

22 March 2022

Non-Composite Hybrid Authentication in PKIX and Applications to Internet Protocols

draft-becker-guthrie-noncomposite-hybrid-auth-00

Abstract

The advent of cryptographically relevant quantum computers (CRQC) will threaten the public key cryptography that is currently in use in today's secure internet protocol infrastructure. To address this, organizations such as the National Institute of Standards and Technology (NIST) will standardize new post-quantum cryptography (PQC) that is resistant to attacks by both classical and quantum computers. After PQC algorithms are standardized, the widespread implementation of this cryptography will be contingent upon adapting current protocols to accommodate PQC. Hybrid solutions are one way to facilitate the transition between traditional and PQ algorithms: they use both a traditional and a PQ algorithm in order to perform encryption or authentication, with the guarantee that the given security property will still hold in the case that one algorithm fails. Hybrid solutions can be constructed in many ways, and the cryptographic community has already begun to explore this space. This document introduces non-composite hybrid authentication, which requires updates at the protocol level and limits impact to the certificate-issuing infrastructure.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The advent of cryptographically relevant quantum computers (CRQC) threatens the public key cryptography that is currently in use in today's secure internet protocol infrastructure. To address this, organizations such as the National Institute of Standards and Technology (NIST) will standardize new post-quantum cryptography (PQC) that is resistant to attacks by both classical and quantum computers. After PQC algorithms are standardized, the widespread implementation of this cryptography will be contingent upon adapting current protocols to accommodate PQC. Hybrid solutions are one way to facilitate the transition between traditional and PQ algorithms: they use both a traditional and a PQ algorithm in order to perform encryption or authentication, with the guarantee that the given security property will still hold in the case that one algorithm fails. Hybrid solutions can be constructed in many ways, and the cryptographic and Internet engineering communities have already begun to explore this space. This document provides background on the

current hybrid solution space, introducing a framework within which to view these solutions with a focus on authentication. It defines a new solution for a hybrid authentication, and considers changes that would be required in PKIX and Internet protocols in order to accommodate this solution.

This work is complementary to, and an extension of, other efforts, such as those as documented in [X509] and [I-D.draft-ounsworth-pq-composite-sigs]. Where existing hybrid authentication solutions attempt to leave protocol logic unaffected and instead invoke changes to cryptographic structures (e.g. certificate format) and processes (certificate validation), non-composite hybrid authentication takes the opposite approach: it minimizes changes required to PKIX and cryptographic libraries, instead limiting the scope of major changes to protocol logic in order to accommodate multiple authentications, each with separate signature and certificate objects.

2. Hybrid Solution Space

There are unique challenges to migration efforts poised at updating current infrastructure to be quantum-secure, and hybrid designs are emerging as a common solution in this space [I-D.draft-ietf-tls-hybrid-design], [I-D.draft-ounsworth-pq-composite-sigs], [I-D.draft-ounsworth-pq-composite-keys], [I-D.draft-ietf-ipsecme-ikev2-multiple-ke]. A hybrid solution is the use of two or more algorithms simultaneously such that the desired security property (e.g., encryption or authentication) still holds in the event that a component algorithm is broken. Hybrid designs can generally be sorted into two categories:

* Composite solutions are those in which both algorithms function together within a protocol, as one entity. Composite solutions alter cryptographic structures and processes (certificate structures, digital signature structures, key share structure,

validation processes, etc.) in order to minimize changes to protocol logic.

* Non-composite solutions are those in which each algorithm functions discretely within a protocol, as an individual entity. Non-composite solutions effect changes at the protocol level in order to minimize changes to cryptographic structures and processes.

These categories are meant to be broad, and are applicable to any type of cryptographic process. However, the remainder of this document focuses on authentication. In the context of hybrid

authentication, composite solutions employ single signature and certificate objects with cryptographic structures that are modified in order to encode multiple algorithms. Non-composite solutions use separate signatures and certificates for each algorithm, and protocol logic is modified instead of signature and certificate structure, in order to accommodate the sending, processing, and receiving of multiple signature and certificate objects.

In practice, the authentication context will be a determining factor in which approach is considered optimal. Applications with a small or closed user community may be more apt to undertake composite solutions (because the number of libraries and amount of hardware affected is limited). On the other hand, contexts where verifiers have the ability to express their capabilities and preferences lend themselves to non-composite solutions. Additionally, non-composite solutions are well-suited toward protocols that can already handle multiple certificate chains and/or digital signatures. Last, non-composite solutions offer forward compatibility to implementations that are difficult to update or that employ certificates with long lifetimes.

3. General Considerations for Non-Composite Hybrid Authentication

In non-composite hybrid authentication, multiple authentications are performed, where each authentication uses a digital signature with a distinct certificate chain. The use of multiple end-entity certificates introduces new security considerations. In particular, a peer that authenticates another peer via non-composite hybrid authentication may desire, in addition to the typical certificate validity checks that are performed, some form of assurance that the end-entity certificates used in each authentication are owned by the same entity. This assurance can be accomplished at the protocol level, through an additional check that compares Subjects or SANs of each end-entity certificate, or more strongly at the PKI level, through a certificate extension defined in [draft-becker-guthrie-cert-binding-for-multi-auth-00] .

3.1. PQ Migration Use Case

The main use case for non-composite hybrid authentication is a circumstance that may arise during the PQ migration period. In particular, an entity may decide to pursue non-composite hybrid authentication, in which authentication is performed once with a traditional signature and certificate chain that it already possesses, and then again with a PQ signature and certificate chain it obtains in order to migrate to PQC.

In order to provide the security promised by hybrid authentication, protocols should validate all digital signatures received before communication is considered authenticated. There may be exceptions where communication can still be considered authenticated if a subset of authentications is either not performed or not successful. For example, if peers are using this approach to facilitate the transition between traditional algorithms and PQ algorithms, the peers may agree that successful validation of the PQ digital signature is sufficient to provide authentication.

The solutions posed in the following sections are backwards compatible with currently existing traditional algorithms-based certificates. Additionally, these solutions allow the PQ-based certificates used for this hybrid solution to also be used for future authentication solutions when users may choose to rely upon PQ signatures and certificates unaccompanied by traditional cryptography, i.e., without needing to issue new PQ certificates when users no longer wish to support or deploy hybrid solutions for authentication. Note that, while these solutions are broad, they are presented in this document with respect to the PQ migration use case, in which a single traditional algorithm and a single PQ algorithm are used simultaneously.

3.2. Protocol-Level Ownership Assertion

End-entity certificates used in non-composite hybrid authentication should only be used in the same protocol if they are owned by the same entity. It may be sufficient to check the ownership of each certificate at the protocol level by verifying that the Subject or SAN is the same on every certificate, prior to validating the signatures. This check provides a low cost, flexible mechanism with which to indicate that end-entity certificates used in a given protocol belong to the same entity.

However, there are scenarios where this method is not sufficient to determine ownership. For example, if the Subject or SAN of an end-entity certificate has changed between the issuance of certificates, the Subject or SAN certificate fields may not match even though the end entity is the same.

3.3. PKI-Level Ownership Assertion

In the event described in the previous section, or in any event when an endpoint involved in authentication desires additional assurance of certificate ownership, users can support the bindingRequest CSR attribute and the X.509v3 certificate extension, BoundCertificates, as defined in [draft-becker-guthrie-cert-binding-for-multi-auth-00]. These mechanisms provide additional assurance at the PKI level that

multiple end-entity certificates each belong to the same entity.

For example, when an end entity is already in possession of a traditional algorithm-based certificate and wishes to obtain a PQ certificate, it submits a CSR containing the bindingRequest attribute. The CA that receives the CSR validates the signature field in the attribute using the public key of the traditional certificate. The CA can then issue the PQ certificate containing the BoundCertificates extension, which contains information identifying the traditional certificate that the PQ certificate is being bound to.

If the receiving peer does not support the BoundCertificates extension, it can ignore the extension (as it is non-critical), and fall back to protocol-level enforcement of certificate ownership.

4. Non-Composite Hybrid Authentication in Internet Protocols

4.1. IKEv2

In order to extend the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296] to support non-composite hybrid authentication, peers must be able to send multiple AUTH payloads, which contain signed bits, and corresponding CERT payloads, which contain certificates. Because IKEv2 currently supports use of only a single AUTH payload by each peer, this approach requires the introduction of a Notify Payload that indicates to the receiving peer that the sender:

1 how many authentications it would like to perform, and

2 what algorithms it would support and/or prefer using for each authentication.

To achieve the initial set of requirements and the following set of desired outcomes, the Notify Payload, N(SUPPORTED_AUTH_METHODS), defined in [I-D.draft-smyslov-ipsecme-ikev2-auth-announce] can be leveraged. N(SUPPORTED_AUTH_METHODS), which is first sent by the responder in IKE_SA_INIT, announces the authentication methods that the sender supports. This, in conjunction with a new Notify Payload that alerts the receiver of the sender's intent to do multiple authentications, along with information about how many authentications can be performed and instructions for how to delineate the list of announcements into choices for each authentication, provides sufficient information for both peers to perform multiple authentications with dually supported algorithms. After the responder sends these Notify Payloads in IKE_SA_INIT, the initiator can choose to oblige the responder's request for multiple

authentications by sending additional AUTH payloads and corresponding CERT payloads in its IKE_AUTH message. If the initiator would also like for the responder to authenticate itself multiple times, it sends the same set of Notify Payloads. The responder can then also opt in and send additional AUTH payloads and corresponding CERT payloads in its IKE_AUTH message.

A peer that supports the BoundCertificates extension, upon receipt of certificates, will check to see if the BoundCertificates extension is present in the end-entity certificate corresponding to the PQ digital signature algorithm. If the extension is present, the peer will perform the check specified in [draft-becker-guthrie-cert-binding-for-multi-auth-00]. If the extension is not present or not supported, the peer should check that the Subjects/SANs listed in each end-entity certificate match. It may be possible to use the PAD [RFC4301] to assist with this check.

4.2. TLS 1.3

In order to facilitate non-composite hybrid authentication in TLS 1.3 [RFC8446], several alterations are necessary. First, the Key Exchange messages must be enabled to negotiate the use of multiple certificates (for simplicity, this description focuses on two certificates). To indicate its request for hybrid authentication, the client can include a flag via the "tls_flags" extension [I-D.draft-ietf-tls-tlsflags] in the ClientHello that alerts the server to its desire to use two certificate chains for authentication.

The client can include two extensions that effect negotiation of multiple signature algorithms; for example, the "signature_algorithms" extension and an appropriately named duplicate of this extension, where each list negotiates a different type of algorithm. (Similarly, client can optionally include the "signature_algorithms_cert" extension and it's appropriately named duplicate for PQ algorithms.) Note that TLS 1.3 implementations are currently designed to only send, receive, and process a single "signature_algorithms" extension and a single "signature_algorithms_cert" extension, so the use of additional "signature_algorithms(_cert)" extensions will require renaming any additional "signature_algorithms(_cert)" extensions so that they are distinct from the original "signature_algorithms(_cert)" extensions.

If the server is willing to use non-composite hybrid authentication for this connection, it responds by sending the "tls_flags" extension with the bit set for the hybrid_auth flag in the ServerHello to acknowledge its support for this feature. This flag extension can also be used in a CertificateRequest message from the server, and if

it is requesting hybrid authentication from the client, then the

CertificateRequest must also include the two extensions for negotiating signature algorithms.

The Authentication Messages also require changes to accommodate non-composite hybrid authentication, namely via duplication of several existing extensions. If non-composite hybrid authentication is negotiated, then the server sends two Certificate messages, where each conveys a distinct certificate chain to the peer (i.e., one traditional chain and one PQ chain). This requires the server to send two individual CertificateVerify messages to the client, where the signature algorithms used in each CertificateVerify message are selected from the "signature_algorithm" extensions sent by the client. The content covered under the signature is the same in each CertificateVerify message, but the Transcript-Hash is computed once for each signature with the corresponding Certificate included in the appropriate hash.

If the implementation requires the BoundCertificates extension, then the server must check that the BoundCertificates extension is present in the appropriate end-entity certificate, and verify ownership as detailed in [draft-becker-guthrie-cert-binding-for-multi-auth-00]. If the implementation does not require this certificate extension, then the server should check that the Subject/SAN listed in each end-entity certificate is the same.

The Finished message contains verification data built from a hash of all handshake messages, which includes both sets of Certificate and CertificateVerify messages in the case of non-composite hybrid authentication.

5. Security Considerations

The use of non-composite hybrid authentication introduces an additional security consideration, in that peers in receipt of multiple end-entity certificates need to confirm that each certificate is owned by the sender. This document outlines two schemes to perform such a check, one at the protocol level, and the other at the PKI level. Technical details for the latter approach can be found in [draft-becker-guthrie-cert-binding-for-multi-auth-00].

It is worth noting that any hybrid solution introduces complexity into a protocol. This complexity can manifest in different ways, including but not limited to: extensions to protocols, changes to public key infrastructure, or modifications to cryptographic libraries. Depending on the implementation, it may be advantageous to limit the areas in which alterations are made, in order to mitigate increases in complexity and streamline further security assessments that may be required as a result of such changes.

All hybrid implementations are vulnerable to a downgrade attack in which a malicious peer does not express support for PQ algorithms, resulting in an exchange that can only rely upon traditional algorithms for security.

6. References

[draft-becker-guthrie-cert-binding-for-multi-auth-00]
Becker, A., Guthrie, R., Jenkins, M., and D. Nisbeth, "Binding Certificates for Multiple Authentications within a Protocol", March 2022.

[I-D.draft-ietf-ipsecme-ikev2-multiple-ke]
Tjhai, C., Tomlinson, M., Bartlett, G., Fluhrer, S., Van Geest, D., Garcia-Morchon, O., and V. Smyslov, "Multiple Key Exchanges in IKEv2", September 2021, <https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-multiple-ke/04/>.

[I-D.draft-ietf-tls-hybrid-design]
Stebila, D., Fluhrer, S., Gueron, S., and U. Haifa, "Hybrid key exchange in TLS 1.3", January 2022, <https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/04/>.

[I-D.draft-ietf-tls-tlsflags]
Nir, Y., "A Flags Extension for TLS 1.3", March 2022, <https://datatracker.ietf.org/doc/draft-ietf-tls-tlsflags/>.

[I-D.draft-ounsworth-pq-composite-keys]
Ounsworth, M. and M. Pala, "Composite Public and Private Keys For Use In Internet PKI", February 2022, <https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-keys/>.

[I-D.draft-ounsworth-pq-composite-sigs]
Ounsworth, M. and M. Pala, "Composite Signatures For Use In Internet PKI", February 2022, <https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/06/>.

[I-D.draft-smyslov-ipsecme-ikev2-auth-announce]
Smyslov, V., "Announcing Supported Authentication Methods in IKEv2", August 2021, <https://datatracker.ietf.org/doc/draft-smyslov-ipsecme-ikev2-auth-announce/>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[X509] ITU-T, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", October 2019, <https://www.itu.int/rec/T-REC-X.509>.

Authors' Addresses

Alison Becker
National Security Agency
Email: aebecke@uwe.nsa.gov

Rebecca Guthrie
National Security Agency
Email: rmguthr@uwe.nsa.gov

Michael Jenkins
National Security Agency
Email: mjjenki@cyber.nsa.gov