CFRG
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

L. Prabel
G. Wang
Huawei
J. Janneck
Ruhr University Bochum
T. Reddy
Nokia
J. Preu Mattsson
Ericsson AB

20 October 2025

# Hybrid Digital Signatures with Strong Unforgeability
draft-prabel-cfrg-suf-hybrid-sigs-00

## Abstract

This document proposes a generic hybrid signature construction that achieves strong unforgeability under chosen-message attacks (SUF-CMA), provided that the second component (typically the post-quantum one) is SUF-CMA secure. The proposed hybrid construction differs from the current composite hybrid approach by binding the second (post-quantum) signature to the concatenation of the message and the first (traditional) signature. This approach ensures that hybrid signatures maintain SUF-CMA security even when the first component only provides EUF-CMA security.

In addition to this general hybrid construction, this document also proposes a non-black-box variant specifically tailored for schemes built from the Fiat-Shamir paradigm. This variant is SUF-CMA secure as long as only one component is SUF-CMA secure.

## About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-prabel-cfrg-suf-hybrid/.

Discussion of this document takes place on the Cryptography Forum Research Group mailing list (mailto:cfrg@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cfrg/. Subscribe at https://www.ietf.org/mailman/listinfo/cfrg/.

Source for this draft and an issue tracker can be found at https://github.com/lucasprabel/draft-cfrg-suf-hybrid-sigs.

Status of This Memo



Copyright Notice



Table of Contents

## 1. Introduction

With the emergence of post-quantum (PQ) digital signatures, several groups (including ETSI CYBER and IETF LAMPS, TLS JOSE, SSHM) have explored hybrid constructions combining traditional and PQ algorithms. The main goal is to ensure long-term security during the transition to post-quantum cryptography, acknowledging that traditional algorithms are more mature than post-quantum ones and that the latter still raise uncertainty about their security.

Current composite hybrid schemes typically provide existential unforgeability under chosen-message attacks (EUF-CMA), but do not ensure strong unforgeability. SUF-CMA extends EUF-CMA by requiring that it be computationally infeasible to produce a new valid signature even for a message-signature pair previously observed. This distinction has practical implications in preventing message replay, transaction duplication, and log poisoning.

Although several recent algorithms such as EdDSA, ML-DSA, and SLH-DSA claim to achieve SUF-CMA security, some popular traditional schemes (RSA, ECDSA) only achieve EUF-CMA. Therefore, constructing a hybrid digital signature scheme maintaining SUF-CMA when one component does not is of particular interest.

To address this concern, this document specifies a generic hybrid construction that guarantees SUF-CMA security when the second underlying component (e.g. the PQ scheme) is SUF-CMA. The construction is quite simple and can be applied generically across PQ/T signature combinations. It is originally proposed in [BH23], though its SUF-CMA is not analyzed in the article.The construction could also be used for a hybrid PQ/PQ security, relying on two post-quantum components.

## 2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document follows the terminology for post-quantum hybrid schemes defined in [I-D.draft-ietf-pquip-pqt-hybrid-terminology].

This section recalls some of this terminology, but also adds other definitions used throughout the whole document:

_EUF-CMA_: Existential Unforgeability under Chosen Message Attack.

_SUF-CMA_: Strong Unforgeability under Chosen Message Attack.

_Post-Quantum Asymmetric Cryptographic Algorithm_: An asymmetric cryptographic algorithm that is intended to be secure against attacks using quantum computers as well as classical computers. They can also be called quantum-resistant or quantum-safe algorithms.

_PQ/T Hybrid Digital Signature_: A multi-algorithm digital signature scheme made up of two or more component digital signature algorithms where at least one is a post-quantum algorithm and at least one is a traditional algorithm.

_Post-Quantum Traditional (PQ/T) Hybrid Composite Scheme_: A multi-algorithm scheme where at least one component algorithm is a post-quantum algorithm and at least one is a traditional algorithm and the resulting composite scheme is exposed as a singular interface of the same type as the component algorithms.

_Component Scheme:_ Each cryptographic scheme that makes up a PQ/T hybrid scheme or PQ/T hybrid protocol.

## 3. Proposed Construction

The proposed construction ensures that the second (nested) signature binds the first (nested) signature, making the overall scheme SUF-CMA as long as the (typically PQ) component is SUF-CMA secure. The hybrid signature construction is defined in the following subsections.

Before signing a message m, the hybrid scheme derives a message representative m’ from m to address specific security concerns, and in particular to achieve non-separability, following a similar approach to [I-D.draft-ietf-lamps-pq-composite-sigs].

### 3.1. Hybrid Key Generation

Generate component keys

- Generate ‘(pk1, sk1)‘ for the traditional scheme.
- Generate ‘(pk2, sk2)‘ for the post-quantum scheme.
- The hybrid public key is ‘pk = (pk1 || pk2)‘.

### 3.2. Hybrid Sign

The Hybrid.Sign algorithm consists in signing a message m’ derived from m with the first component, and then signing the concatenation m’ || s1 of the derived message with the first signature with the second component.

Generate the message representative

- Compute m’ = Prefix || Label || len(ctx) || ctx || PH(m)

Generate hybrid signature

- Compute s1 = Sign_1(sk1, m’)
- Compute s2 = Sign_2(sk2, m’ || s1)
- Output the hybrid signature s = (s1 || s2)

In the computation of the message representative: - Prefix is the byte encoding of the string "SUFHybridSignature2025", which in hexadecimal is "53554648796272696453696 76E617475726532303235". - Label: a label which is specific to the particular component algorithms being used. - len(ctx): a single byte representing the length of ctx. - ctx: the context bytes. - PH(m): the hash of the message to be signed.

### 3.3. Hybrid Verify

Verify hybrid signature

- Compute m’ = Prefix || Label || len(ctx) || ctx || PH(m)
- Parse s as (s1, s2)
- Compute Verify_1(pk1, m’, s1)
- Compute Verify_2(pk2, m’ || s1, s2)
- Accept if both verifications succeed.

### 3.4. Related works

The hybrid construction in [I-D.draft-ietf-lamps-pq-composite-sigs] only provides SUF-CMA security if both components is providing SUF-CMA security and one of the components are deterministic. As traditional signatures do not provide any security against quantum attackers, when [I-D.draft-ietf-lamps-pq-composite-sigs] is used for PQ/T hybrid scheme, it does not provide SUF-CMA security against quantum attackers. In this document, only the second component needs to be SUF-CMA so that the hybrid scheme achieves SUF-CMA security.

In contrast to [I-D.draft-ietf-lamps-pq-composite-sigs], the signing process of the hybrid construction proposed in this document cannot be parallelized. Indeed, computing the hybrid signature s = (s1 || s2) requires to compute s1 = Sign_1(sk1, m') first in order to compute s2 = Sign_2(sk2, m' || s1).

## 4. Non-black-box Construction

The proposed construction of this section ensures that the overall scheme is SUF-CMA as long as only one component is SUF-CMA secure. The hybrid signature construction is defined in the following subsections.

The hybrid can be used for signature schemes that are built from the Fiat-Shamir paradigm as the first component and from any signature scheme as the second compoenent. Hence, they use a canonical identification scheme (ID) underlying a Fiat-Shamir construction and a signature scheme (Sig_2). This applies to combining EdDSA and any post-quantum signature scheme, for example ML-DSA.

Before signing a message m, the hybrid scheme derives a message representative m' from m to address specific security concerns, and in particular to achieve non-separability, following a similar approach to [I-D.draft-ietf-lamps-pq-composite-sigs].

### 4.1. Hybrid Key Generation

Generate component keys

- Generate '(pk1, sk1)' for the (traditional) ID scheme.
- Generate '(pk2, sk2)' for the (post-quantum) signature scheme.
- The hybrid public key is 'pk = (pk1, pk2)'.

### 4.2. Hybrid Sign

The Hybrid.Sign algorithm consists of applying the Fiat-Shamir paradigm for the first signature component. During the process (after the commitment has been computed), the second component is applied by signing the message and the commitment. The remainder of the Fiat-Shamir signature is computed using the second signature component instead of the message and the commitment as usual.

Generate the message representative

- Compute m’ = Prefix || Label || len(ctx) || ctx || pk’s || PH(m)

Generate hybrid signature

- Compute (com, st) = ID.Com(sk1)
- Compute m’’ = PH(1 || m’ || com)
- Compute s2 = Sig.Sign_2(sk2, m’’)
- Compute chl = PH(2 || s2)
- Compute rsp = ID.Rsp(sk1, com, chl, st)
- Output the hybrid signature s = (rsp || s2)

In the computation of the message representative: - Prefix is the byte encoding of the string "SUFHybridSignature2025", which in hexadecimal is "53554648796272696453696 76E617475726532303235". - Label: a specific label which is specific to the particular component algorithms being used. - len(ctx): a single byte representing the length of ctx. - ctx: the context bytes. - pk’s: the concatenation of pk1 and pk2. - PH(m): the hash of the message to be signed.

### 4.3. Hybrid Verify

Verify hybrid signature

- Compute m’ = Prefix || Label || len(ctx) || ctx || pk’s || PH(m)
- Parse s as (rsp || s2)
- Compute chl = PH(2 || s2)
- Compute com = ID.ExtCom(pk1, ch, rsp)
- Compute m’’ = PH(1 || m’ || com)
- Compute Verify_2(pk2, m’’, s2)
- Accept if verification succeeds.

### 4.4. Security and Applicability

The hybrid is SUF-CMA if one of the underlying signatures is SUF-CMA secure. Additionally, the ID scheme must have unique responses and the second signature component (post-quantum component) must fulfill message-bound security (MBS) [BUFF] and random-message validity (RMV) [Jan25].

The first requirement (on the traditional scheme) is fulfilled by EdDSA which is built from an ID scheme with unique responses. The second requirement (on the post-quantum scheme) is fulfilled by any of NIST standards/winners, i.e. ML-DSA, SLH-DSA, Falcon (to be FN-DSA).

## 5. Why the Binding Hybrid is Required

Hybrid constructions will have to provide SUF-CMA at the artifact level to ensure single-signature semantics and non-repudiation. In many real-world deployments the artifact signing use case is central: software releases, firmware images, signed logs, and legal/financial documents are all artifacts that rely on a single, unambiguous signature to prove provenance and integrity. A hybrid design achieves SUF-CMA only if one signature component is cryptographically bound to the other, forming a binding hybrid rather than signing the same message independently.

Any successful forgery of a binding hybrid must fall into one of two categories:

* New second signature on a new input:
  The attacker generates a new traditional signature s1* that the legitimate signer never produced. The attacker would then need to forge a valid s2* over the concatenation m’ || s1*. Producing such an s2* is a forgery against the PQC algorithm.

* Different second-signature on an already-signed input:
  The attacker reuses an existing (m’, s1) but fabricates a distinct s2* for the same (m’ || s1), yielding two valid second signatures for one message.

Both outcomes constitute a SUF-CMA forgery against the second component: the first case for a new message, the second for a second valid signature on an existing message. If the second component is SUF-CMA secure, neither case is computationally feasible, and the combined hybrid inherits SUF-CMA security.

### 5.1. Loss of Non-Repudiation in Parallel Hybrids under CRQC

As described in [I-D.draft-ietf-lamps-pq-composite-sigs], composite hybrids produce multiple component signatures independently over the same message.
Once a CRQC can forge the traditional component, an attacker can create an alternate classical signature s1* for a message that already has a valid hybrid signature (s1, s2). Because the PQC signature s2 remains valid independently of the classical signature, the modified pair (s1*, s2) also verifies successfully.

While authenticity of the PQC component remains intact, non-repudiation cannot be guaranteed: multiple distinct hybrid signatures (s1, s2) and (s1*, s2) can exist for the same message. Therefore, once the classical algorithm becomes breakable, parallel hybrids no longer provide single-signature semantics, the assurance that each message corresponds to exactly one, unique signature from the signer.

On the contrary, this document's hybrid construction, by binding the second signature s2 to the first signature s1, ensures single-signature semantics and preserves non-repudiation.

### 5.2. ECDSA vs EdDSA in Hybrid Constructions

Even though both ECDSA (secp256r1/secp384r1) and EdDSA (Ed25519/Ed448) become mathematically breakable once a CRQC can derive private keys from public keys, their behaviour in hybrid constructions differs significantly:

* ECDSA is randomized and non-deterministic, producing multiple distinct valid signatures for the same message. After CRQCs arrive, an attacker can generate arbitrarily many valid classical signatures, and hence multiple valid hybrids, destroying non-repudiation.

* Ed25519 and Ed448, in contrast, are deterministic and provide SUF-CMA security in their standard formulations, yielding a unique valid signature per message for a given key. This determinism eliminates malleability and preserves non-repudiation even if a CRQC later compromises the private key. In parallel hybrids, this property avoids ambiguity about which signature is authentic. In binding hybrids, EdDSA's fixed, deterministic format enables unambiguous inclusion of s1 in the PQC input (m' || s1), simplifying verification and ensuring consistent interpretation across implementations.

Consequently, ECDSA can only be used in a binding hybrid to preserve non-repudiation, and cannot be used in a parallel hybrid, because it is not SUF-CMA and becomes forgeable and repudiable once a CRQC can recover its private key.

# 6. Security Considerations

## 6.1. Security Model and Motivation

The hybrid construction described in this document aims to guarantee strong unforgeability of the composite signature whenever the second component is SUF-CMA secure. This is in contrast to the composite construction in [I-D.draft-ietf-lamps-pq-composite-sigs], where SUF-CMA of the composite generally requires both components to be SUF-CMA. The design proposed here strengthens that property: SUF-CMA of the overall construction depends only on the SUF-CMA of the second component, regardless of the security level of the first one.

## 6.2. SUF-CMA Security

### 6.2.1. Why SUF-CMA matters

While EUF-CMA security could be sufficient in several use cases, weaknesses in EUF-only schemes allow "re-signing" the same message, enabling real-world exploits such as replay of messages, double receipts, and log poisoning. Moreover, many current deployed systems implicitly assume that all digital signatures are SUF-secure, and that a single unique signature exists per message.

For this reason, the construction ensures that if the second component is SUF-CMA, the hybrid automatically resists replay and duplication attacks, aligning with best practices in recent signature standards (EdDSA, ML-DSA, SLH-DSA, etc.).

### 6.2.2. Security Rationale

Intuitively, an adversary attempting to forge (m*, s1*, s2*) must either:

* Forge s2* on (m* || s1*), which is infeasible if the second scheme is SUF-CMA;

or

* Reuse an existing (m, s1) pair with a modified s2, which again breaks SUF-CMA of the second scheme.

Consequently, if the second component is SUF-CMA secure, the hybrid construction remains SUF-CMA secure even when the first component provides only EUF-CMA security.

In contrast, if the second scheme were only EUF-CMA, the second attack (re-signing the same message differently) would no longer be excluded, and the hybrid construction would not be SUF-CMA secure.

This contrasts with classical composite hybrids (e.g. trad(M) || PQ(M)) where the PQ signature does not authenticate the output of the traditional signature, leaving possible avenues for replay or signature substitution.

### 6.3. Non-Separability

The document [I-D.draft-ietf-pquip-hybrid-signature-spectrums] defines both notions of Weak Non-Separability (WNS) and Strong Non-Separability (SNS).

The hybrid construction in this document achieves WNS because the Prefix of the message representative m’ is an evidence that a verifier may be able to identify, preventing the validation of a component signature which would have been removed from the composite signature.

However, SNS is not achieved, as s1 stripped from a composite signature s = (s1 || s2) is a valid component signature of the message m’ and s2 is a valid component signature of the message m’ || s1.

## 7. IANA Considerations

This document has no IANA actions.

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

### 8.2. Informative References

[BH23] Bindel, N. and B. Hale, "A Note on Hybrid Signature Schemes", July 2023, <https://eprint.iacr.org/2023/423.pdf>.

[BUFF] Cremers, C., Dzl, S., Fiedler, R., Fischlin, M., and C. Janson, "BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures", 2021, <https://ieeexplore.ieee.org/document/9519420>.

[I-D.draft-ietf-lamps-pq-composite-sigs] Ounsworth, M., Gray, J., Pala, M., Klauner, J., and S. Fluhrer, "Composite ML-DSA for use in X.509 Public Key Infrastructure", Work in Progress, Internet-Draft, draft-ietf-lamps-pq-composite-sigs-12, 10 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-sigs-12>.

[I-D.draft-ietf-pquip-hybrid-signature-spectrums] Bindel, N., Hale, B., Connolly, D., and F. D, "Hybrid signature spectrums", Work in Progress, Internet-Draft, draft-ietf-pquip-hybrid-signature-spectrums-07, 20 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-hybrid-signature-spectrums-07>.

[I-D.draft-ietf-pquip-pqt-hybrid-terminology] D, F., P, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-ietf-pquip-pqt-hybrid-terminology-06, 10 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqt-hybrid-terminology-06>.

[Jan25] Janneck, J., "Bird of Prey: Practical Signature Combiners Preserving Strong Unforgeability", October 2025, <https://eprint.iacr.org/2025/1844.pdf>.

## Authors' Addresses

Lucas Prabel
Huawei
Email: lucas.prabel@huawei.com

Guilin Wang
Huawei
Email: wang.guilin@huawei.com

Jonas Janneck
Ruhr University Bochum
Email: jonas.janneck@rub.de

Tirumaleswar Reddy
Nokia
Bangalore
Karnataka
India
Email: kondtir@gmail.com

John Preu Mattsson
Ericsson AB
Email: john.mattsson@ericsson.com