Hybrid Digital Signatures with Strong Unforgeability

Lucas Prabel

Based on draft-prabel-cfrg-suf-hybrid-sigs, draft-devevey-cfrg-silithium and draft-josefsson-cfrg-mothma

Thanks to my SUF Hybrid co-authors for their feedback.

IETF 127, San Francisco

Context

EUF-CMA (*Existential Unforgeability under Chosen Message Attack*): no attacker can forge any valid signature for a ***new*** message.

SUF-CMA (*Strong Unforgeability under Chosen Message Attack*): no attacker can forge any new valid signature, even for a **previously signed** message.

- **ML-DSA**, **SLH-DSA** and **EdDSA** are designed to provide SUF-CMA security; **ECDSA** does not.
- Parallel composites (e.g. Composite ML-DSA) are SUF-CMA only if **both** components are, and one of them is **deterministic**.
- Therefore, the usual hybrid guarantee (secure as long as one component is) does not hold for SUF-CMA**: ML-DSA alone is SUF-CMA, but ML-DSA + ECDSA is not**.
- Currently, three CFRG drafts aim at addressing this gap:
  - Mothma (draft-josefsson-cfrg-mothma)
  - SUF Hybrid Signatures (draft-prabel-cfrg-suf-hybrid-sigs)
  - Silithium (draft-devevey-cfrg-silithium)

IETF 126 poll

Should CFRG explore PQ/T hybrid signatures?

Yes	**42**
No	**17**
No opinion	**8**

Follow-up on the list (Oct. 2026): should CFRG explore a **strongly unforgeable** PQ/T hybrid signature?

This talk explores those constructions and their properties, to help CFRG scope future work.

Two current approaches

The three current constructions bind the post-quantum and traditional components in one of **two ways**:

1. Nested signatures

Used in: Mothma, SUF Hybrid §3

Black box: both components, so any traditional scheme works.

```
s1 = T.Sign(m')
s2 = PQ.Sign(m' || s1)
sig = s1 || s2   // Mothma also sends r, h
```

The PQ signature covers the traditional one.

SUF-CMA as long as the PQ component is

2. PQ signature bound to a Schnorr signature

Used in: SUF Hybrid §4, Silithium

Black box: the PQ component.

Opened: the traditional one (Schnorr-type: EdDSA, EC-Schnorr).

```
R = T.Commit()          // Schnorr commitment
s2 = PQ.Sign(m || R)
c = f(s2)               // Schnorr challenge
z = T.Respond(R, c)     // Schnorr response
sig = s2 || z
```

The challenge c comes from s2: a hash of s2 (SUF Hybrid §4), or the ML-DSA challenge inside s2 (Silithium).

SUF-CMA if either component is (§4)

SUF-CMA if ML-DSA is (Silithium)

T traditional **PQ** post-quantum

Mothma

draft-josefsson-cfrg-mothma-01 (S. Josefsson)

A nested hybrid with fresh randomness per signature, sent with its message.

Sign — **T** traditional **PQ** post-quantum

```
r = H(fresh randomness)
h = H(r, H(pk), name, app, appctx, m)
s1 = T.Sign(r ‖ h)
s2 = PQ.Sign(s1 ‖ r ‖ h)  // ctx = name
output (s2, s1, r, h, m)  // H = SHA3-256
```

Verify

```
h == H(r, H(pk), name, app, appctx, m)
T.Verify(s1; r ‖ h) ∧ PQ.Verify(s2; s1 ‖ r ‖ h)
```

- **Components:** **EdDSA or ECDSA** with **ML-DSA, SLH-DSA, XMSS or LMS**, both as black boxes, in a fixed list of named variants
- **Binding:** the PQ component signs s1, and h binds both public keys and the message
- **Signature size:** |s1| + |s2| + 64 B (r, h) + the message
- **Analysis:** no formal analysis in the draft yet

EUF-CMA if either component is EUF-CMA

SUF-CMA if the outer PQ scheme is SUF-CMA

SUF Hybrid Signatures: black-box

draft-prabel-cfrg-suf-hybrid-sigs-03, §3 (Prabel, Wang, Janneck, Reddy, Preuß Mattsson)

A nested hybrid for any pair of schemes, sending only the two component signatures.

Sign — **T** traditional **PQ** post-quantum

```
m′ = Prefix ‖ Label ‖ len(ctx) ‖ ctx ‖ PH(m)
s1 = Sign_1(sk1, m′)
s2 = Sign_2(sk2, m′ ‖ s1)
output sig = s1 ‖ s2
```

Verify

```
Verify_1(pk1, m′, s1)
∧ Verify_2(pk2, m′ ‖ s1, s2)
```

- **Components:** **any traditional scheme** (ECDSA, EdDSA, RSA) and **any PQ scheme**, both as black boxes; also PQ/PQ
- **Binding:** the PQ component signs s1; m′ is built as in Composite ML-DSA
- **Signature size:** |s1| + |s2|
- **Analysis:** security rationale in the draft

EUF-CMA if either component is EUF-CMA

SUF-CMA if the 2nd (PQ) component is SUF-CMA

Silithium

draft-devevey-cfrg-silithium-00 (Devevey, Guerreau, Roméas)

ML-DSA inside an EC-Schnorr signature, reusing ML-DSA's own challenge as the Schnorr challenge.

T traditional **PQ** post-quantum

Sign

```
k random in ]0, q[
R = [k]G            // Schnorr commitment
s2 = ML-DSA.Sign(skPQ, M, ctx = R ‖ P)
c = s2[:c_len]      // challenge, taken from ML-DSA
x = k + d·c mod q // Schnorr response
output sig = s2 ‖ x
```

Verify

```
check 0 ≤ x < q; R = [x]G - [c]P
ML-DSA.Verify(pkPQ, M, s2, ctx = R ‖ P)
```

EUF-CMA if either component is

SUF-CMA if the PQ component is SUF-CMA

- **Components:** EC-Schnorr on P-256/384/521, opened; ML-DSA-44/65/87 as a black box
- **Binding:** ML-DSA signs R as context, and its challenge c is reused as the Schnorr challenge
- **Signature size:** |s2| + |x|
- **Analysis:** proofs in [DGR26]

SUF Hybrid Signatures: non-black-box

draft-prabel-cfrg-suf-hybrid-sigs-03, §4 (Prabel, Wang, Janneck, Reddy, Preuß Mattsson)

A PQ signature inside EdDSA; it stays SUF-CMA if either component is.

T traditional **PQ** post-quantum

Sign

```
m' = Prefix || Label || … || pk1 || pk2 || PH(m)
(com, st) = ID.Com(sk1)  // EdDSA commitment
s2 = Sign_2(sk2, PH(1 || m' || com))
chl = PH(2 || s2)  // EdDSA challenge
rsp = ID.Rsp(sk1, com, chl, st)
output sig = rsp || s2
```

Verify

```
check rsp in range  // EdDSA: S < L
com = ID.ExtCom(pk1, PH(2 || s2), rsp)
Verify_2(pk2, PH(1 || m' || com), s2)
```

EUF-CMA if either component is EUF-CMA

SUF-CMA if either component is SUF-CMA

- **Components:** **EdDSA**, opened; **any PQ scheme with MBS and RMV** (ML-DSA, SLH-DSA, FN-DSA) as a black box
- **Binding:** the PQ component signs the EdDSA commitment, and the challenge hashes the whole s2
- **Signature size:** |s2| + |rsp|
- **Analysis:** proofs in [Jan25]

Properties compared

	Property	Meaning	Use cases
Unforgeability	**EUF-CMA** existential unforgeability	No valid signature on a message the signer never signed	The basic security notion
	SUF-CMA strong unforgeability	No new valid signature, even on a message already signed	Signatures as identifiers (logs, replay caches); some protocol proofs
Non-separability	**SNS** strong non-separability	No valid component signature can be extracted from a hybrid signature (RFC 9955)	Stops stripping a component
BUFF properties	**EO** exclusive ownership	A signature is valid under only one public key	Stops key-substitution attacks
	MBS message-bound signatures	Under a given key, a signature is valid for only one message	A signature cannot be claimed for another message
	NR non-resignability	Without knowing the message, no one can make a valid signature on it under another key	Stops re-signing a message one has not seen

BUFF = “beyond unforgeability” (Cremers et al., 2021). A construction that lacks one of them can usually get it with a simple change.

Security properties

	Property	Mothma	SUF Hybrid §3 black-box	SUF Hybrid §4 non-black-box	Silithium
Unforgeability	EUF-CMA	**PQ or T**	**PQ or T**	**PQ or T**	**PQ or T**
	SUF-CMA	PQ	PQ	**PQ or T**	PQ
Strong non-separability	Traditional part	**No**	**No**	**Yes**	**Yes**
	PQ part	**No**	**No**	**No**	**No**
BUFF properties	Exclusive ownership	**Yes**	if PQ and T have EO	if PQ has EO and MBS	if PQ has EO and MBS
	Message-bound	**Yes**	**if PQ or T has MBS**	if PQ has MBS	if PQ has MBS
	Non-resignability	**Yes**	if T has NR and PQ has RMV	if PQ has RMV	if PQ has RMV

Holds if either component does, or always

Requires a specific component (or both)

Not provided

PQ or T: one component suffices (only PQ against a quantum adversary).
PQ conditions: all met by ML-DSA. Silithium is specified with ML-DSA only; §3 and §4 accept any PQ scheme that meets those conditions.

Questions for CFRG

1. Which SUF-CMA guarantee?

As long as the PQ component is SUF-CMA, or as long as either component is?

2. Black-box or Schnorr-type?

Both components as black boxes, or open up a Schnorr-type traditional scheme?

3. Which traditional schemes?

Are ECDSA or RSA required? Only the nested constructions support them today; for RSA, a dedicated construction could also give SUF-CMA if either component is [Jan25].

4. Combiner or instances?

A generic combiner, or a small set of named instances?

5. Beyond SUF-CMA?

Are strong non-separability, EO, MBS and NR required? Should existing keys be reusable?

Possible choices and motivations

1. Which SUF-CMA guarantee?	**Either component** if T is Schnorr-type, otherwise PQ	closest to the usual hybrid guarantee
2. Black-box or Schnorr-type?	**Both**, one scheme per approach	trade-off between security and deployability
3. Which traditional schemes?	**ECDSA, RSA** nested; **EdDSA, EC-Schnorr** Schnorr-based	ECDSA and RSA largely deployed in PKI
4. Combiner or instances?	**A generic combiner** with a few named instances	as for hybrid KEMs
5. Beyond SUF-CMA?	**Both public keys** in what PQ signs, no key reuse	EO, MBS and NR from PQ, at no size cost

Nested scheme

Black box: both components

Sign — **T** traditional **PQ** post-quantum

```
m' = Prefix || … || pk1 || pk2 || PH(m)
s1 = T.Sign(m')
s2 = PQ.Sign(m' || s1)
sig = s1 || s2
```

SUF-CMA if PQ is SUF-CMA

EO, MBS if PQ has them (always with ML-DSA)

Schnorr scheme

Black box: PQ only; the Schnorr signature is opened

Sign — **T** traditional **PQ** post-quantum

```
m' = Prefix || … || pk1 || pk2 || PH(m)
R = T.Commit()
s2 = PQ.Sign(m' || R)
z = T.Respond(R, H(s2))
sig = s2 || z
```

SUF-CMA if PQ or T is SUF-CMA

EO, MBS if PQ has them (always with ML-DSA)

References

draft-prabel-cfrg-suf-hybrid-sigs-03 – L. Prabel, G. Wang, J. Janneck, T. Reddy, J. Preuß Mattsson, “Hybrid Digital Signatures with Strong Unforgeability”
datatracker.ietf.org/doc/draft-prabel-cfrg-suf-hybrid-sigs/

draft-devevey-cfrg-silithium-00 – J. Devevey, M. Roméas, M. Guerreau, “Silithium – A Compact, Efficient and Non-separable Hybrid Signature”
datatracker.ietf.org/doc/draft-devevey-cfrg-silithium/

draft-josefsson-cfrg-mothma-01 – S. Josefsson, “Mothma”
datatracker.ietf.org/doc/draft-josefsson-cfrg-mothma/

draft-ietf-lamps-pq-composite-sigs – Composite ML-DSA
datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/

RFC 9955 – N. Bindel, B. Hale, D. Connolly, F. Driscoll, “Hybrid Signature Spectrums”
www.rfc-editor.org/rfc/rfc9955

[Jan25] J. Janneck, “Bird of Prey: Practical Signature Combiners Preserving Strong Unforgeability”, Eurocrypt 2026, ePrint 2025/1844
eprint.iacr.org/2025/1844

[DGR26] J. Devevey, M. Guerreau, M. Roméas, “Compact, Efficient and Non-Separable Hybrid Signatures”, PQCrypto 2026, ePrint 2025/2059
eprint.iacr.org/2025/2059

[CDF+21] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin, C. Janson, “BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures”, IEEE S&P 2021
eprint.iacr.org/2020/1525