Security analysis of Hybrid EAP-AKA prime full authentication

This report in another language: 中文.

draft-ietf-emu-hybrid-pqc-eapaka-02 · ProVerif+ Agent report

Documents	Enhancing Security in EAP-AKA' with Hybrid Post-Quantum Cryptography (protocol definition); Forward Secrecy Extension to the Improved Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS) (supporting)
Properties examined	13: 10 hold, 3 fail (3 attacks)
Findings	None

Contents

1 Protocol and scope
2 Findings
3 Properties examined
4 Limits of this analysis
Appendix A: Reading of each property
Appendix C: Claims and prover answers
Appendix D: How the analysis was run

1 Protocol and scope

EAP-AKA' full authentication with a hybrid post-quantum key establishment: the peer holds a USIM with the AKA long-term key K and an identity, the EAP server holds no long-term key of its own but obtains a fresh AKA vector for the subscriber from the home AAA server (AD), and the peer and the server then run the EAP-AKA' exchange in which the classical ECDHE group of RFC 9678 is replaced by a fixed ML-KEM-768 + X25519 hybrid. The peer first announces its identity, the server sends the challenge from the vector together with an AT_KDF_FS offer, the peer selects the hybrid value, the server sends its hybrid public key in AT_PUB_HYBRID together with the selected value and the original list, and the peer answers with its encapsulation and AT_RES.

When the exchange ends the server has checked AT_RES and AT_MAC and has decapsulated the peer's encapsulation, so it is sure of the subscriber's identity and of the hybrid secret it shares with it; the peer has checked the AUTN it got from its USIM and the AT_MAC on both challenges, so it is sure that the challenge and the hybrid public key come from a party that holds a valid AKA vector for its identity. Both sides hold the same MK_HYBRID and install MSK and EMSK from it; the peer's only evidence that the run succeeded on the server side is the unprotected EAP-Success.

The parties:

- **Peer**: EAP peer with USIM. Holds: K, identity, local policy (hybrid required optional).
- **EAP server**: EAP server of the serving network. Holds: no long-term key; one AKA vector per run from the AD.
- **AD**: home AAA server, AKA vector source (trusted). Holds: K, SQN counter, network name.

Figure 1 shows the exchange as the documents specify it.

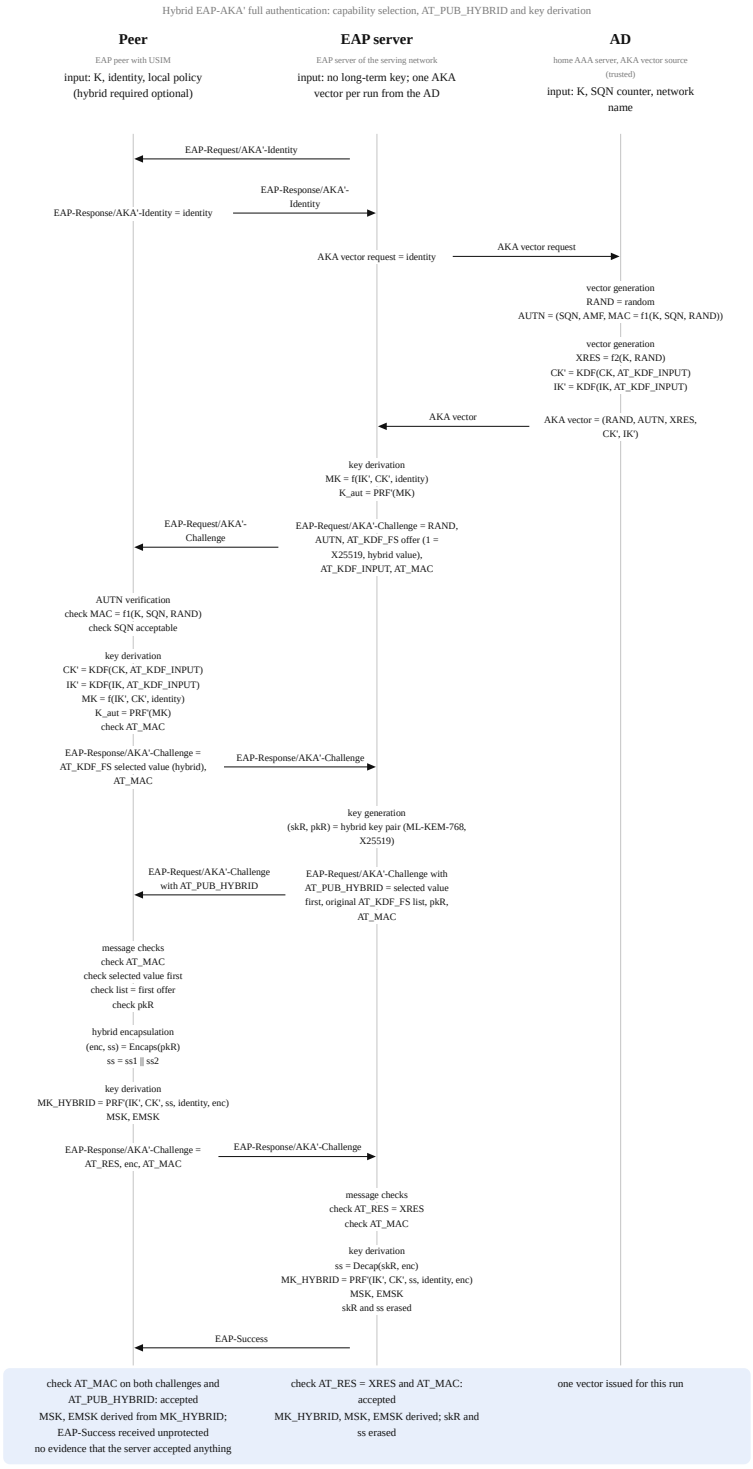


Figure 1. The protocol as specified.

The scope of the analysis is in Appendix D: what was analysed, what was left out, and the sections of the documents it is built from.

2 Findings

No finding. The analysis found nothing that the documents do not already state or set aside. The table in section 3 has the result for every property.

Within the scope analysed the draft gives what it aims for: a peer that requires a hybrid KEM cannot be pushed to a classic-only result by the network or by a dishonest server holding a valid vector, and the session keys survive both the loss of the AKA long-term key after the run and the break of one hybrid component. Agreement and key confirmation hold in both directions on the challenges, and each hybrid public key is used once. The one failing property is old and known: the peer derives its keys and finishes on an unprotected EAP-Success, so it can complete while the server accepted nothing, which RFC 4187 Section 12.8 already states. The replay of a challenge is an artifact of leaving the USIM sequence number out of the first model and is removed once the value is unrolled with its comparison, so no finding is new. All thirteen targets have an answer; twelve rest on idealised primitives, one, R01, on an unrolled model of two runs.

3 Properties examined

Each row is one property put to ProVerif. The result is ProVerif's answer on the analyst's model of the protocol; value and whether the documents already state the result are the analyst's judgement, with the reasons in Appendix A.

ID	Property	Compromise assumed	Result	Value	In the documents
A03	**Peer and server agree on the full run.** Whenever the peer accepts a full authentication, there exists a matching server run, and both sides agree on the peer identity, the server identity, the selected hybrid KEM value, pkR/enc and MK_HYBRID.	no compromise	Attack holds (stated in the documents)	medium	yes
A04	**Each server acceptance has its own peer run.** Every successful server acceptance corresponds to a distinct peer run; no two acceptances map to the same peer run.	no compromise	Holds	medium	no
A08	**AT_KDF_FS list bound to the true offer.** The AT_KDF_FS list the peer accepts must correspond to the server's real offer and to the selected hybrid KEM value, and must differ from the first list only by the requested change; AT_MAC covers that list.	no compromise	Holds	medium	no
K01	**Session keys stay secret.** The attacker cannot derive MK_HYBRID, MSK, EMSK or K_re unless it holds the corresponding long-term key or ss.	no compromise	Holds	medium	yes
C01	**Forward secrecy of the session keys.** If the long-term keys leak after the session completes, the attacker still cannot derive the MSK/EMSK/MK_HYBRID of that session.	long-term keys IK'/CK' (leaked after the session completes)	Holds	high	yes
C08	**Hybrid mix survives one component.** When either the X25519 component or the ML-KEM component is broken, the attacker still cannot derive the session keys MSK/EMSK.	a single hybrid KEM component (during or after the session)	Holds	high	no
R05	**Local hybrid policy prevents downgrade.** When the local policy requires a hybrid KEM, the accepted configuration must include a hybrid KEM; the attacker cannot downgrade the acceptance to classic ECDHE only by removing or modifying AT_KDF_FS.	no compromise	Holds	medium	yes
R01	**Fresh challenge, no reused evidence.** The authentication evidence the peer accepts must come from this challenge (a fresh RAND/AUTN); old evidence cannot be reused.	no compromise	Holds within a bound (the changing value unrolled to 2 steps)	medium	no

ID	Property	Compromise assumed	Result	Value	In the documents
A06	**Key confirmation in both directions.** When the server accepts, the peer has already derived the same K_aut and sent a correct AT_MAC/AT_RES; when the peer accepts, the server has already derived the same K_aut and sent a correct AT_MAC.	no compromise	Holds	medium	yes
K01.10	**Session keys kept after MK_HYBRID or ss leaks.** After MK_HYBRID or ss leaks, the attacker cannot derive MSK/EMSK.	MK_HYBRID or ss (leaked after derivation)	Attack holds (stated in the documents)	low	yes
R05.11	**Modified capability list cannot force classic result.** Under a mandatory hybrid policy, an attacker modifying the AT_KDF_FS list cannot downgrade the acceptance to classic ECDHE only.	no compromise	Holds	medium	yes
A03.12	**Identities and fresh challenge bound.** When the peer and the server accept, both sides agree on the identities, the fresh RAND/AUTN challenge and the successful outcome.	no compromise	Attack holds (stated in the documents)	medium	yes
X01	**Fresh hybrid KEM key pair per exchange.** No pkR value of AT_PUB_HYBRID is used in two authentication exchanges: two challenges of the same peer never carry the same pkR.	no compromise	Holds	medium	no

"Attack holds (stated in the documents)": the prover found the attack, and the documents themselves already state it. The property fails; this is a known result, not a finding (Appendix B).

Goals the documents state that the drafted list of properties did not have, added by the analyst (Appendix D): X01.

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: Secrecy under leakage of the hybrid MSK component	Leaking MK_HYBRID gives MSK and EMSK at once, because they are slices of MK_HYBRID, so nothing in the exchange can prevent it; leaking only the hybrid shared secret does not give them, because MK_HYBRID also takes CK' and IK' from the AKA run. The practical consequence: protect MK_HYBRID as the key material itself and erase skR, ss and the encapsulated secret after the run.	K01.10, C01, K01
Q2: A malicious modification of the capability list does not make a mandatory hybrid policy accept a classic result	No. An attacker that modifies or strips the AT_KDF_FS list cannot make a peer that requires a hybrid KEM accept a classic-only result, because the echoed list comes back under AT_MAC and is compared with the first offer, and the peer requires the hybrid value to be present and selected; neither the network attacker nor a dishonest server holding a valid vector succeeds. A peer that permits a fallback by policy does accept the stripped offer, so the protection is the policy, not the negotiation.	R05.11, R05, A08
Q3: Peer/server identities bound to the fresh authentication challenge/success	When the peer accepts, it agrees with the server on the RAND/AUTN challenge and on the hybrid public key, and the identity is bound into the same key derivation on both sides; so the identity and freshness half holds. Agreement on the outcome does not: the peer finishes on an unprotected EAP-Success and can therefore complete a run that the server never accepted, so the peer must not treat its own key derivation as evidence of the server's acceptance.	A03.12, A03, A06

4 Limits of this analysis

A result here is a result on a model of the protocol: it says what the specified messages and checks give or fail to give, with cryptography taken as ideal, and nothing about an implementation.

What this analysis does not cover, as the analyst states it:

- Fragmentation and reassembly (AT_FRAGMENT) and AT_CHECKCODE are not modelled: messages are taken whole, so nothing is said about MTU limits or about a fragmented challenge.
- The hybrid KEM is a symbolic encapsulation with two independent components: no computational, IND-CCA or post-quantum claim is made, so the strength of ML-KEM-768 and X25519 is outside the analysis.
- X25519 is not given its Diffie-Hellman algebra; it is one opaque component, so a property that needed algebraic reasoning about the classical part is not covered.
- The AKA vector is assumed fresh and trusted and the model without the sequence number has no freshness check on the challenge; the unrolled model covers two runs with sequence numbers 1 and 2 only.
- The sequence-number acceptance rule, any acceptance window and the 48-bit wrap are not in these documents; they would have to come from 3GPP TS 33.102 and be settled by Tamarin+ Agent with integer rules.
- The draft's requirement to delete skR, ss and the derived keys is not modelled as erasure; the forward-secrecy result assumes the deletion happens, and a retained skR is not covered.
- Concurrent runs of the same subscriber and races on a kept value are outside what ProVerif can be asked; the unrolled peer runs its two authentications in sequence.
- The failure answers of the peer (Synchronization-Failure against Authentication-Reject) are not queried for indistinguishability; the state review records them as a candidate finding without a model query.
- Identity request, notification and re-authentication exchanges are not modelled; the analysis is of one full authentication.
- The eight agreement and freshness proofs (A03, A04, A08, A06 both directions, A03.12, X01 and the unrolled R01) carry no control variant: the variant that would have to fail is one in which the network attacker also holds the AKA long-term key and signs its own challenge, which this model does not contain.

What the host records:

- The answers of R01 rest on an operation the model simplifies (Appendix D).
- Proved without a control query (Appendix C): A03: a03ch, A04: a04, A08: a08, R01: r01, A06: a06a, A06: a06b, A03.12: a03ch, X01: x01. A control is a variant of the property that must fail; without one, nothing shows that the property as written can fail in this model.

What stays open for each property, and what would settle it, is in Appendix A.

Appendix A: Reading of each property

The analyst's own reading of the prover's answers, not an answer of the prover: what the answers mean for the protocol, what the reading rests on, whether it can be used as it stands, and what it is worth.

A03: Peer and server agree on the full run

Whenever the peer accepts a full authentication, there is a matching server run, and both sides agree on the peer identity, the server identity, the selected hybrid KEM value, pkR/enc and MK_HYBRID.

A peer that accepts the hybrid challenge response can only have accepted a challenge and a pkR that the server really sent, because AT_MAC over the echoed list and pkR is checked with a key derived from the AKA vector. Agreement on the derived keys does not hold in the other direction: the peer derives MK_HYBRID, MSK and EMSK and finishes on an unprotected EAP-Success, so an attacker on the network can drop its response and the server never accepts anything. Agreement on the peer identity is carried by the key derivation itself, since the identity is an input of MK_HYBRID on both sides.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable with caveats
- Value medium: The positive half is the guarantee the draft is built for and it holds without a simplification that touches it apart from the symbolic KEM; the negative half is a weakness EAP-AKA' has always had and the documents already state, reproduced here for this construction.
- Stated in the documents: yes. Where the documents say it: rfc4187:3759, rfc4187:3766
- Rests on: The hybrid KEM is a symbolic encapsulation with two independent components; no computational or post-quantum claim is made, so agreement is agreement on terms, not on real key material.; The AKA vector is assumed fresh and trusted: the AD is honest and gives each vector to one server, and the peer's USIM is not modelled as refusing a stale challenge in this model.; Fragmentation and reassembly (AT_FRAGMENT, AT_CHECKCODE) are not modelled; messages are taken whole.; The server's identity is not a term in any message: the assurance the peer gets is that some holder of a valid vector for its identity sent the challenge.
- Open: Whether a peer that never sees a protected success indication should count as having completed the run is a policy question the draft leaves to the lower layer.
- Next step: Nothing further is needed for the protocol claim; a deployment question about protected result indications is outside what a protocol prover can settle.

A04: Each server acceptance has its own peer run

Every server acceptance corresponds to a distinct peer run; no two acceptances map to the same peer run.

Every server acceptance corresponds to a distinct peer run: a response cannot be replayed to make the server accept twice, because the server decapsulates the encapsulation of the single reply it receives and derives a new hybrid key pair for each challenge. This is injective agreement on the peer's response against the network alone.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: A central agreement property of the exchange, proved under the symbolic KEM abstraction; no simplification removes the mechanism it rests on, a fresh hybrid key pair per challenge. I searched the draft and RFC 9678 for an injective agreement statement about AT_RES and enc and found none.
- Stated in the documents: no

- Rests on: The hybrid KEM decapsulation is modelled so that only the encapsulation made to that pkR decapsulates; a real KEM's binding property is assumed, not proved.; The server's own state is not modelled as a kept value, so the target does not say what a server that retries with the same vector would do.
- Next step: None.

A08: AT_KDF_FS list bound to the true offer

The AT_KDF_FS list the peer accepts must correspond to the server's real offer of the selected hybrid KEM value, and must differ from the first list only by the requested change; AT_MAC covers that list.

The list and the hybrid value the peer accepts are exactly the ones the server offered, with the selected value moved to the head, because the peer compares the echo with the first offer and verifies AT_MAC over it. An attacker on the network cannot substitute a different list or make the peer accept a value that was not offered.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Proves the binding that makes the negotiation meaningful and that the draft relies on for its downgrade story; it holds only against an attacker without the long-term key, which is exactly the scope the draft claims.
- Stated in the documents: no
- Rests on: The source does not formalise the comparison beyond the selected value first and the rest as offered; the model follows Section 6.2 of RFC 9678 as the draft instructs.; The offered list is modelled as two elements, X25519 first and the hybrid value; a longer list with extensions is not covered.
- Next step: None.

K01: Session keys stay secret

The attacker cannot derive MK_HYBRID, MSK, EMSK, K_re unless it holds the corresponding long-term key or ss.

During the run an attacker on the network learns neither MK_HYBRID nor MSK and EMSK, and losing the long-term key after the run does not give them either, because the hybrid shared secret is fresh per exchange and never appears on the wire. The keys become known only where MK_HYBRID itself is disclosed, which is the leak the draft's derivation makes fatal.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The secrecy of the session keys is the draft's main promise and it survives both attackers the documents name for it; the proof is against an idealised KEM and PRF.
- Stated in the documents: yes. Where the documents say it: main:585
- Rests on: PRF', the AKA algorithms and the KEM are ideal functions; the result is a statement about the construction, not about their strength.; The long-term key is disclosed after the run and the ephemeral hybrid secrets are not; the draft's own warning about a retained skR is not modelled.; K_re is not derivable in the model beyond MSK; the claim about K_re rests on the same derivation step as MSK.
- Next step: None.

C01: Forward secrecy of the session keys

If the long-term key leaks after the session completes, the attacker still cannot derive that session's MSK/EMSK/MK_HYBRID.

Forward secrecy holds for the hybrid construction: an attacker that obtains the AKA long-term key, and with it CK', IK' and K_aut, after the session ended does not obtain that session's MK_HYBRID, MSK or EMSK, because the KEM secret is gone. This is the property the hybrid key establishment is added for and it is not touched by the disclosure of the classical long-term material.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value high: A central guarantee of the design, proved for the attacker the documents name, the long-term key after the session, without a simplification that removes the mechanism, the fresh hybrid secret per exchange; it is the reason the draft exists.
- Stated in the documents: yes. Where the documents say it: main:585
- Rests on: The hybrid private key and shared secret are erased after the run, as the draft requires; a retained skR plus the long-term key would give the session keys, as main:585 says.; Perfect forward secrecy is shown for the symbolic KEM only; no statement is made about a stored transcript plus a future break of ML-KEM or X25519.; The AD is honest and issues each vector once.
- Next step: None.

C08: Hybrid mix survives one component

When one of the X25519 component or the ML-KEM component is broken, the attacker still cannot derive the session keys MSK/EMSK.

One broken hybrid component does not break the session: an attacker that holds the X25519 shared secret of the session, or the ML-KEM component's secret, but not both, cannot derive MK_HYBRID, MSK or EMSK, because the combined secret needs both components and the AKA-derived CK' and IK' as well. The combination step, not the individual KEMs, is what the target rests on.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value high: The hybrid-composition guarantee is the whole point of the ML-KEM plus X25519 combination and it is proved for the attacker the target names; the caveat is that the primitives themselves are not modelled cryptographically.
- Stated in the documents: no
- Rests on: The KEM is symbolic: the model shows that the combination uses both components and the AKA material, not that ML-KEM-768 or X25519 resists a real attack.; Only one component is broken at a time; the disclosure selector is a modelling device so that exactly one disclosure happens per trace.; The draft does not define a component-compromise interface; the reading is ours, and it is the property that makes a hybrid worth deploying.
- Open: Whether a real ML-KEM-768 or X25519 break at the level the model assumes is a computational question the prover cannot be asked.
- Next step: A computational argument for the combination, as in the concrete hybrid KEM draft, would settle the part that is outside the symbolic model.

R05: Local hybrid policy prevents downgrade

When local policy requires a hybrid KEM, the accepted configuration must include a hybrid KEM, and an attacker cannot make acceptance degrade to classic ECDHE only by removing or modifying AT_KDF_FS.

A peer that requires a hybrid KEM by local policy never completes on a classic-only configuration, even when the attacker strips the hybrid value from the offer or a dishonest server holding a valid AKA vector offers only X25519 with a valid AT_MAC. The protection is the local policy alone: with the fallback allowed, the peer does accept the

stripped offer, which the control shows.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: This is the mitigation the draft itself prescribes for the downgrade it admits, shown mechanically for both attackers the documents name; it confirms rather than discovers.
- Stated in the documents: yes. Where the documents say it: main:593
- Rests on: The draft makes the hybrid requirement a MAY by local policy; the model selects the policy per peer session and both configurations appear.; The dishonest server is modelled as obtaining a valid vector and sending a classic-only list; it does not attempt the hybrid exchange.; The comparison the peer makes is modelled as the draft and RFC 9678 Section 6.2 describe it; a deployment with a longer list or extension attributes is not covered.
- Next step: None.

R01: Fresh challenge, no reused evidence

The authentication evidence the peer accepts must come from the current challenge (a new RAND/AUTN); old evidence cannot be reused.

On the model without a freshness check a recorded challenge is accepted twice, which is what the missing sequence number allows; with the check the draft relies on, and the vectors of two consecutive sequence numbers, the peer accepts no challenge twice. The replay is therefore not a flaw of the draft but the consequence of leaving the USIM's sequence number out; the half that ProVerif cannot do over any number of sessions was done on the unrolled model.

- Result: Holds within a bound (the changing value unrolled to 2 steps); answers what is asked: yes; usable with caveats
- Value medium: It tells a reader exactly what the replay result means and gives the bounded proof that the draft's reliance on the sequence number is justified for two runs; the acceptance-window question stays open.
- Stated in the documents: no
- Rests on: The unrolled model runs two authentications, with sequence numbers 1 and 2, and the peer hands the accepted value to the next run; concurrency between runs and the size of any acceptance window are outside it.; The sequence-number acceptance rule itself is not specified in these documents; the model follows the AKA rule that only a greater value is accepted, which the draft assumes rather than states.; The model without the check is the one the replay was found on; its attack is a model artifact, as recorded.
- Open: Whether an acceptance window in the real USIM lets an old but recent-enough vector through cannot be answered from these documents.
- Next step: Tamarin+ Agent with the 3GPP sequence-number rules written as integer constraints, and more runs, would settle the window and the wrap.

A06: Key confirmation in both directions

When the server accepts, the peer has derived the same K_aut and sent a correct AT_MAC/AT_RES; when the peer accepts, the server has derived the same K_aut and sent a correct AT_MAC.

Key confirmation holds in both directions on the AT_MAC: when the server accepts, the peer had derived the same K_aut and sent that response, and when the peer accepts the challenge, the server had derived the same K_aut and sent that challenge. The peer's assurance stops at the challenge, because nothing after the response is authenticated to it.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Confirms that the two AT_MAC checks give the mutual assurance the draft describes, with the asymmetry stated plainly; it is a proof of what the documents claim.

- Stated in the documents: yes. Where the documents say it: main:457
- Rests on: AT_MAC is modelled as a MAC over the whole message with a key derived from the AKA material; the exact coverage of each attribute in the encoding is taken from the draft, not checked bit for bit.; Identity request and notification exchanges are not modelled.
- Next step: None.

K01.10: Session keys kept after MK_HYBRID or ss leaks

After MK_HYBRID or ss leaks, the attacker cannot derive MSK/EMSK.

Disclosing the hybrid shared secret alone does not give MSK or EMSK, because MK_HYBRID also needs CK' and IK' from the AKA run. Disclosing MK_HYBRID gives both immediately, which is definitional: MSK and EMSK are slices of MK_HYBRID, so that half is fatal by the derivation itself and no change of the exchange can help.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value low: The failing half is definitional and the holding half follows from the derivation; it answers the user's question but says nothing surprising about the protocol.
- Stated in the documents: yes. Where the documents say it: main:585
- Rests on: The slice relation between MK_HYBRID, MSK and EMSK is modelled as a private derivation, as the draft defines it.; The two components' secrets are taken to be what an attacker with the hybrid shared secret would hold; the peer is modelled as disclosing it after the run.
- Next step: None.

R05.11: Modified capability list cannot force classic result

Under a mandatory hybrid policy, an attacker modifying the AT_KDF_FS list cannot make acceptance degrade to classic ECDHE only.

Modifying the AT_KDF_FS list cannot force a hybrid-required peer into a classic result: the echoed list is protected by AT_MAC and compared with the first offer, so a changed list either fails the check or is refused. This holds against a dishonest server with a valid AKA vector as well, because the vector lets it compute K_aut but the peer still requires the hybrid value to be present and selected.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It answers the user's question about list modification directly and under both attackers, and it shows that the protection comes from AT_MAC plus local policy rather than from the KEM.
- Stated in the documents: yes. Where the documents say it: main:590
- Rests on: AT_MAC is assumed to cover the whole AT_KDF_FS list and the selected value, as the draft states; the encoding of the attributes is not modelled.; The peer's policy is a MAY in the draft; the property is stated for a deployment that takes it.
- Next step: None.

A03.12: Identities and fresh challenge bound

When the peer and the server accept, both agree on the identities, the fresh RAND/AUTN challenge, and the successful result.

When the peer accepts, the RAND/AUTN challenge and the hybrid public key it accepted are the ones the server sent, and the identity is bound into the same key derivation on both sides. Agreement on the outcome does not hold the other way: the peer finishes on an unprotected EAP-Success even when its response was dropped and the server

accepted nothing.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable with caveats
- Value medium: It separates the two halves of what the run assures, with the recorded weakness in the second half; the identity part rests on an argument about the derivation rather than a separate query.
- Stated in the documents: yes. Where the documents say it: rfc4187:3759
- Rests on: Agreement on the identity is carried by the derivation of MK_HYBRID from the identity on both sides, not by a separate comparison; the model has no message in which the server states the identity to the peer.; The AKA vector is fresh and trusted; the sequence-number check is not in this model.
- Next step: None.

X01: Fresh hybrid KEM key pair per exchange

No pkR value of AT_PUB_HYBRID is used in two authentication exchanges: two challenges of the same peer never carry the same pkR.

No hybrid public key is used in two exchanges: each challenge carrying AT_PUB_HYBRID is answered with a freshly generated key pair, so two challenges of the same peer never carry the same pkR. This makes the encapsulation of one run useless in another.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It is the freshness guarantee the target from the documents asks for and it holds against the network alone; it is one of the two things a replay of a challenge must break, so it is worth having stated separately.
- Stated in the documents: no
- Rests on: Freshness of the key pair is modelled by generating the key pair inside each server run; a server that cached a key pair across exchanges would break the property and is not modelled.; The property is about the run that produces the message; a replayed challenge is not a second key-pair generation, which is why R01, not this target, carries the replay.
- Next step: None.

Appendix C: Claims and prover answers

The prover's record for each target: the level reached, the query that reached it, whether a control was checked, and the attacks with the analyst's verdict on each.

target	level	bound	lemma	control	attacks	claims
A03	proved		a03ch		a03keys (holds_in_source)	1 proved, 1 attack
A04	proved		a04			1 proved
A08	proved		a08			1 proved
K01	proved		c01	checked		2 proved
C01	proved		c01	checked		1 proved
C08	proved		c08	checked		1 proved
R05	proved		r05	checked		2 proved
R01	bounded	the changing value unrolled to 2 steps	r01		r01 (model_artifact)	1 bounded, 1 attack
A06	proved		a06a			2 proved
K01.10	proved		k0110b	checked	k0110a (holds_in_source)	1 proved, 1 attack
R05.11	proved		r05	checked		2 proved
A03.12	proved		a03ch		a03keys (holds_in_source)	1 proved, 1 attack
X01	proved		x01			1 proved

Claim by claim, in the analyst's wording of each claim; the answer to each is the prover's:

target	claim	attacker	answer	query	control
A03	The peer's acceptance of the challenge response (AT_MAC valid over rand, autn and the echoed list, pkR) implies that the EAP server sent that ch2 message for the same rand and pkR; the attacker has the network only.	`network`	proved	a03ch	none
A03	Both sides agree on MK_HYBRID/MSK/EMSK: a peer run that derived its keys has a matching server run that accepted the same enc and derived the same keys; the attacker has the network only.	`network`	attack (holds_in_source)	a03keys	
A04	Each successful server acceptance of an AKA'-Challenge response corresponds to a distinct peer run: no two server acceptances share one peer response (injective agreement on the response).	`network`	proved	a04	none
A08	The AT_KDF_FS list and the hybrid value the peer accepts are the ones the server offered (the selected value first, the rest the same as in the first offer), under an AT_MAC the peer verified; the attacker has the network only.	`network`	proved	a08	none
K01	An attacker that obtains the AKA long-term key and everything derived from it (CK, IK, CK', IK', MK, K_aut) after the session ended still cannot derive MK_HYBRID/MSK/EMSK of that session; the attacker has the long-term key after the run.	`ltk_after`	proved	c01	checked

target	claim	attacker	answer	query	control
K01	During the run itself (before any disclosure) an attacker on the network cannot learn MSK/EMSK of the session; the attacker has the network only.	`network`	proved	k01p0	checked
C01	A completed session keeps MSK/EMSK/MK_HYBRID secret when the AKA long-term key material is disclosed after the session; the attacker has the long-term key after the run.	`ltk_after`	proved	c01	checked
C08	When one hybrid component (the X25519 shared secret or the ML-KEM component) is broken and the other is intact, the attacker still cannot derive MK_HYBRID/MSK/EMSK; the attacker has one component's shared secret of the session.	`one_component`	proved	c08	checked
R05	A peer that requires a hybrid KEM by local policy never accepts a classic-only (ECDHE-only) configuration, even when the attacker removes the hybrid value from the AT_KDF_FS offer; the attacker has the network only.	`network`	proved	r05	checked
R05	A peer that requires a hybrid KEM by local policy never accepts a classic-only configuration offered by a dishonest server that holds a valid AKA vector and can sign any AT_KDF_FS list; the attacker is a dishonest AAA/EAP server with a valid vector.	`dishonest_server`	proved	r05	checked
R01	The evidence the peer accepts comes from a fresh challenge: the same RAND/AUTN challenge cannot be accepted by the peer in two different runs; the attacker has the network only.	`network`	attack (model_artifact)	r01	
R01	The evidence the peer accepts comes from a fresh challenge: with the USIM's sequence-number check in place and the AD issuing SQN 1 then SQN 2, no RAND/AUTN challenge is accepted by the peer in two runs; the attacker has the network only.	`network`	bounded (the changing value unrolled to 2 steps)	r01	none
A06	When the server accepts (AT_RES and AT_MAC valid), the peer had derived the same K_aut and sent that response; the attacker has the network only.	`network`	proved	a06a	none
A06	When the peer accepts the AKA'-Challenge (AT_MAC valid), the server had derived the same K_aut and sent that challenge; the attacker has the network only.	`network`	proved	a06b	none
K01.10	Disclosing the hybrid shared secret ss alone (both components' secrets, with IK'/CK' still secret) does not give MSK/EMSK; the attacker has ss of the session after the run.	`mk_hybrid_reveal`	proved	k0110b	checked
K01.10	After MK_HYBRID of the session is disclosed, the attacker still cannot derive MSK/EMSK; the attacker has MK_HYBRID after the run.	`mk_hybrid_reveal`	attack (holds_in_source)	k0110a (the control of k0110b)	
R05.11	A modified AT_KDF_FS list cannot make a hybrid-required peer accept a classic-only result; the attacker has the network only.	`network`	proved	r05	checked
R05.11	Even a dishonest server holding a valid AKA vector cannot make a hybrid-required peer accept a classic-only result by changing the AT_KDF_FS list; the attacker is a dishonest server with a valid vector.	`dishonest_server`	proved	r05	checked

target	claim	attacker	answer	query	control
A03.12	When the peer accepts, the RAND/AUTN challenge it accepted and the pkR it received are the ones the EAP server sent (agreement on the fresh challenge and on the hybrid key); the attacker has the network only.	`network`	proved	a03ch	none
A03.12	When the peer finishes a full run, the server has accepted the same run and both agree on the outcome; the attacker has the network only.	`network`	attack (holds_in_source)	a03keys	
X01	No pkR is used in two authentication exchanges: each challenge carrying AT_PUB_HYBRID uses a freshly generated hybrid key pair (injective correspondence between challenges and key-pair generation); the attacker has the network only.	`network`	proved	x01	none

Appendix D: How the analysis was run

An LLM analyst builds the model of the protocol from the documents and puts each property to ProVerif; every result in section 3 is an answer of the prover.

ProVerif+ Agent session session-20261114T214452, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 45, prover 58 s, wall 18 min, cost 0.194 USD.

The conversation was restarted from the analyst's notes 1 time(s). The trajectory file keeps everything.

ProVerif: 80 proofs run, 75 decided, 5 without an answer, 41 s, 21 answers recorded.

What was analysed

Protocol	Hybrid EAP-AKA' full authentication
Version	draft-ietf-emu-hybrid-pqc-eapaka-02
Analysed	Fixed ML-KEM-768+X25519 hybrid; capability selection first, then AT_PUB_HYBRID is sent; one full authentication; trusted AAA/AKA vector assumption. The supporting drafts are pinned to the in-repository -05/-12/-04, and the differences from the originally cited versions are stated explicitly. Includes negotiation-list echo, AT_MAC and final-key binding; the finite-fragment abstraction cannot be generalized to an arbitrary MTU.
Not analysed	The symbolic model does not prove computational or post-quantum security of the algorithms; attack traces hold only under explicit assumptions and are not directly equivalent to specification vulnerabilities. Each property must have a reachability/positive-example control.
Source text	147 numbered passages (33631 bytes) from the sections listed below. A reference such as s0001 in this report is one of these passages

The sections of the documents the analysis is built from:

Document	Section	Title
main	5	Hybrid Enhancements by Design
main	6	Message Fragmentation and Reassembly
main	7	Capability Negotiation
main	8	Protocol Construction
main	8.1	Protocol Call Flow
main	8.2	Key Steps in Protocol Construction
main	9	Extensions to EAP-AKA' FS
main	9.1	AT_PUB_HYBRID
main	10	Security Considerations
rfc9678	6	Extensions to EAP-AKA'
rfc9678	6.1	AT_PUB_ECDHE
rfc9678	6.2	AT_KDF_FS
rfc9678	6.3	Forward Secrecy Key Derivation Functions
rfc9678	6.4	ECDHE Groups

Document	Section	Title
rfc9678	6.5	Message Processing
rfc9678	6.5.1	EAP-Request/AKA'-Identity
rfc9678	6.5.2	EAP-Response/AKA'-Identity
rfc9678	6.5.3	EAP-Request/AKA'-Challenge
rfc9678	6.5.4	EAP-Response/AKA'-Challenge
rfc9678	6.5.5	EAP-Request/AKA'-Reauthentication
rfc9678	6.5.6	EAP-Response/AKA'-Reauthentication
rfc9678	6.5.7	EAP-Response/AKA'-Synchronization-Failure
rfc9678	6.5.8	EAP-Response/AKA'-Authentication-Reject
rfc9678	6.5.9	EAP-Response/AKA'-Client-Error
rfc9678	6.5.10	EAP-Request/AKA'-Notification
rfc9678	6.5.11	EAP-Response/AKA'-Notification

The purpose of the protocol as the source step located it in the documents: Fixed ML-KEM-768+X25519 hybrid; capability selection first, then AT_PUB_HYBRID is sent; one full authentication; trusted AAA/AKA vector assumption. The supporting drafts are pinned to the in-repository -05/-12/-04, and the differences from the originally cited versions are stated explicitly. Includes negotiation-list echo, AT_MAC and final-key binding; the finite-fragment abstraction cannot be generalized to an arbitrary MTU.

The properties against the documents

Before a query was written, the analyst set the drafted list of properties against what the documents themselves say the protocol is for. The name, what a property is about and what the documents say are the analyst's reading; the places named exist in the documents.

ID	Name	About	What the documents say	Where
A03	Peer and server agree on the full run	The peer and the EAP server agree, after a full hybrid authentication, on the peer identity, the KEM selection, pkR/enc and MK_HYBRID.	The call flow and key steps say the server derives ss = Decap(enc, skR) and checks AT_RES and AT_MAC, and the peer verifies AT_MAC and the AT_KDF_FS list; success requires both to be correct. They do not state that the peer learns the server accepted.	s0028, s0031, s0035, s0036, s0038
A04	Each server acceptance has its own peer run	Every successful server acceptance corresponds to a distinct peer run; no two acceptances map to one peer run.	The server MUST generate a new hybrid KEM key pair for each authentication exchange and key pairs MUST NOT be reused (Section 8.2); the response is checked against XRES and AT_MAC of that exchange.	s0040, s0028, s0036
A08	AT_KDF_FS list bound to the true offer	The list and hybrid KEM value the peer accepts are the ones the server really offered and selected, differing only by the requested change, and AT_MAC covers the list.	AT_MAC protects the AT_KDF_FS offer and the peer's selection because the final EAP-Request/AKA'-Challenge carries both the selected value and the original list (Section 10); RFC 9678 6.2 requires the peer to check that only the requested change occurred.	main:582, s0057, s0015, s0018, s0028

ID	Name	About	What the documents say	Where
K01	Session keys stay secret	MK_HYBRID, MSK, EMSK and K_re are kept from the attacker unless it holds the long-term key or the hybrid shared secret.	MK_HYBRID = PRF'(IK'|CK'|ss, ...) and MSK/EMSK/K_re are slices of it; RFC 9048 7 states that under the PRF assumption an attacker cannot derive any non-trivial information about these keys from the others.	rfc9048:1199, s0038, s0052
C01	Forward secrecy of the session keys	The MSK/EMSK/MK_HYBRID of a completed session stay secret when the AKA long-term key is disclosed after the session.	An attacker who obtains the long-term key cannot derive the session keys of a recorded exchange without also breaking the hybrid KEM (Section 8.1), and RFC 9678 7 says someone who gets all AKA long-term keys cannot reconstruct the keys used to protect a session once it ended.	rfc9678:929, s0028, s0031, s0056
C08	Hybrid mix survives one component	The session keys stay secret when one hybrid component (X25519 or ML-KEM) is broken and the other is intact.	Hybrid key exchange combines several algorithms so that security holds even if all but one component algorithm is broken (Section 8).	draft-irtf-cfrg-hybrid-kems:1031, s0021, s0052
R05	Local hybrid policy prevents downgrade	A party that requires a hybrid KEM by local policy never completes on a classic-only (ECDHE-only) configuration when the adversary removes or modifies the AT_KDF_FS offer.	A server that requires a hybrid KEM and finds that the peer did not select one MUST behave as if authentication failed, and a peer that requires one and finds no hybrid value MUST behave as if AUTN were incorrect (Section 7); deployments needing protection against an attacker that can remove the offer MUST require a hybrid KEM by local policy (Section 10).	main:585, s0019, s0057
R01	Fresh challenge, no reused evidence	The evidence the peer accepts comes from a fresh challenge (new RAND/AUTN); evidence from an old exchange cannot be accepted again.	On full authentication replay protection of the EAP exchange is provided by RAND and AUTN from the underlying AKA scheme, and a different K_aut is used in each full authentication exchange.	rfc4187:3707, s0028, s0035
A06	Key confirmation in both directions	When the server accepts, the peer had derived the same K_aut and sent a correct AT_MAC/AT_RES; when the peer accepts the challenge, the server had derived the same K_aut and sent a correct AT_MAC.	The server derives keys and verifies AT_MAC only when AT_RES is valid, and the peer verifies AT_MAC with K_aut derived from MK; RFC 9048 7.2 records that AKA would have benefited from additional key confirmation.	rfc9048:1402, s0028, s0031
K01.10	Session keys kept after MK_HYBRID or ss leaks	After MK_HYBRID or the hybrid shared secret ss of a session is disclosed, the attacker still cannot derive that session's MSK/EMSK.	MSK and EMSK are slices of MK_HYBRID and MK_HYBRID = PRF'(IK'|CK'|ss, ...); the documents do not discuss what a disclosure of one of these values gives, the user asks it.	s0038, s0056
R05.11	Modified capability list cannot force classic result	Modifying the AT_KDF_FS list cannot make a hybrid-required peer or server accept a classic-only result.	Both sides make a policy decision; a hybrid KEM can be required by local policy (Section 7, Section 10).	s0019, s0057
A03.12	Identities and fresh challenge bound	When peer and server accept they agree on the identity, on the fresh RAND/AUTN challenge and on the outcome of the run.	The peer identity enters MK and MK_HYBRID, AT_MAC covers RAND, AUTN and the list, and the server checks AT_RES and AT_MAC.	s0028, s0035, s0038

ID	Name	About	What the documents say	Where
X01	Fresh hybrid KEM key pair per exchange	The EAP server's hybrid KEM key pair (skR, pkR) and the pkR it sends in AT_PUB_HYBRID.	For each authentication exchange the server MUST generate a new hybrid KEM key pair; key pairs MUST NOT be reused across authentication exchanges (Section 8.2), and the draft's security considerations say a KEM key pair should be used in only one EAP session.	draft-ietf-emu-pqc-eapaka:1126, s0040, s0033

Added from the documents by the analyst: X01.

The security statements of the documents, paragraph by paragraph

The host listed the 405 paragraphs of the sections in which the documents say what the protocol protects and against what, and the analyst said of each what it made of it: a property, an attacker, or why neither. Taken up: 208. Not taken: 197. Which paragraphs are listed follows from the layout of the standard; where each went is the analyst's judgement.

Paragraphs	Where	Taken up as	Why not
P001	draft-ietf-emu-pqc-eapaka:1121-1123	The prover cannot be asked	Requirement on the quality of the random number generator, a property of the primitive.
P002	draft-ietf-emu-pqc-eapaka:1125-1129	C01	
P003	draft-ietf-emu-pqc-eapaka:1131-1136	A08, R05, `ltk_active`	
P004	draft-ietf-emu-pqc-eapaka:1138-1142	Outside the scope	Fragmentation and the total attribute length limit; the model carries attributes whole and does not model fragment reassembly or its cost.
P005	draft-ietf-emu-pqc-eapaka:1144-1149	K01, A08	
P006 to P007	draft-ietf-emu-pqc-eapaka:1158-1162	No goal stated	Page headers and draft boilerplate.
P008	draft-ietf-emu-pqc-eapaka:1167-1171	The prover cannot be asked	Security level of an ML-KEM variant, a property of the primitive.
P009 to P021	draft-irtf-cfrg-concrete-hybrid-kems:485-530	The prover cannot be asked	Security considerations of the hybrid KEM constructions themselves (generic combiner, PRG, binding goals); computational properties of primitives, not of the EAP exchange.
P022 to P023	draft-irtf-cfrg-hybrid-kems:1018-1027	The prover cannot be asked	Security considerations of the hybrid KEM constructions themselves (generic combiner, PRG, binding goals); computational properties of primitives, not of the EAP exchange.
P024 to P026	draft-irtf-cfrg-hybrid-kems:1031-1046	C08	
P027 to P040	draft-irtf-cfrg-hybrid-kems:1051-1140	The prover cannot be asked	IND-CCA, ciphertext second-preimage resistance and strong Diffie-Hellman are computational assumptions on the KEM components.
P041 to P046	draft-irtf-cfrg-hybrid-kems:1144-1181	A08, K01	

Paragraphs	Where	Taken up as	Why not
P047 to P059	draft-irtf-cfrg-hybrid-kems:1185-1262	The prover cannot be asked	Indifferentiability and PRG requirements of the KEM construction; properties of primitives.
P060 to P062	draft-irtf-cfrg-hybrid-kems:1266-1278	C08	
P063 to P073	draft-irtf-cfrg-hybrid-kems:1283-1316	The prover cannot be asked	IND-CCA goal of a hybrid KEM, a computational property of the primitive.
P074 to P079	draft-irtf-cfrg-hybrid-kems:1320-1352	A08	
P080	draft-irtf-cfrg-hybrid-kems:1356-1357	No goal stated	States what the hybrid KEM goal is not; no requirement on the EAP exchange.
P081 to P147	draft-irtf-cfrg-hybrid-kems:1361-1810	The prover cannot be asked	Analyses and lemmas for the binding and IND-CCA goals of the KEM constructions, and encoding rules; computational results on primitives that the symbolic model does not represent.
P148	main:564-567	The prover cannot be asked	Refers to the IND-CCA discussion of the KEM draft; a property of the primitive.
P149	main:569-570	The prover cannot be asked	Security levels and sizes of ML-KEM; a property of the primitive.
P150	main:578-578	The prover cannot be asked	Requirement on the random number generator.
P151	main:582-582	C01	
P152	main:585-588	A08, R05, `ltk_active`	
P153	main:590-595	No goal stated	Closing sentence of the same paragraph; no additional requirement.
P154 to P155	rfc3748:2182-2190	No goal stated	States that EAP itself has no security mechanisms and points to the other sections.
P156 to P168	rfc3748:2194-2252	A03, A08, `network`	
P169 to P192	rfc3748:2256-2427	A03, A06, A08, R01	
P193 to P196	rfc3748:2431-2455	Outside the scope	Identity protection rests on pseudonyms or SUCI, which the modelled flow and the selected source units leave out.
P197 to P204	rfc3748:2459-2502	A03	
P205 to P214	rfc3748:2506-2567	A08, A06	
P215 to P217	rfc3748:2571-2587	No goal stated	States that AKA is not a password protocol and has no dictionary-attack exposure; no property of this protocol to ask.
P218 to P220	rfc3748:2591-2611	Outside the scope	Protection of the user data by the lower layer, which is outside the EAP method.
P221 to P222	rfc3748:2615-2629	R05, A08	
P223 to P225	rfc3748:2633-2651	No goal stated	Advice about implementation idiosyncrasies.
P226 to P240	rfc3748:2659-2752	K01, C01	

Paragraphs	Where	Taken up as	Why not
P241 to P244	rfc3748:2765-2789	The prover cannot be asked	Strength of ciphersuites, a property of primitives.
P245 to P250	rfc3748:2793-2828	Outside the scope	Lower layer behaviour.
P251 to P257	rfc3748:2832-2880	R05, `dishonest_server`	
P258 to P260	rfc3748:2884-2898	No goal stated	No cleartext password in this protocol.
P261 to P267	rfc3748:2902-2945	Outside the scope	Channel binding to the authenticator is not provided by EAP-AKA' and is outside the modelled flow.
P268 to P280	rfc3748:2949-3056	A03	
P281	rfc4187:3594-3597	No goal stated	Introduction to the security section; no requirement of its own.
P282 to P283	rfc4187:3608-3629	Outside the scope	Identity protection by pseudonyms, outside the modelled flow.
P284 to P286	rfc4187:3633-3646	A03	
P287	rfc4187:3650-3659	The prover cannot be asked	Flooding the authentication centre; a resource-consumption property.
P288 to P289	rfc4187:3663-3672	K01	
P290	rfc4187:3676-3680	No goal stated	States that brute force on the 128-bit AKA key is not feasible; nothing to ask of the protocol.
P291 to P299	rfc4187:3684-3740	R01, A08, A03	
P300 to P302	rfc4187:3744-3757	R05, A08	
P303 to P304	rfc4187:3761-3777	A03	
P305 to P306	rfc4187:3781-3799	Outside the scope	Man-in-the-middle through tunnelled EAP; the tunnel is outside the modelled flow.
P307	rfc4187:3808-3811	The prover cannot be asked	Quality of the random number source.
P308 to P328	rfc9048:1131-1260	K01, C01, A08, R01	
P329 to P342	rfc9048:1264-1344	Outside the scope	Privacy of identifiers via SUCI, pseudonyms and fast re-authentication names, outside the modelled flow.
P343 to P354	rfc9048:1348-1448	A06, A03, `ltk_active`	
P355 to P359	rfc9048:1452-1482	C01, `ltk_after`	
P360 to P363	rfc9048:1486-1537	C01, `ltk_active`	
P364 to P375	rfc9678:868-954	C01, `ltk_after`	

Paragraphs	Where	Taken up as	Why not
P376 to P378	rfc9678:958-973	C01	
P379 to P387	rfc9678:977-1046	A08, R05, C01, `ltk_active`	
P388 to P394	rfc9678:1050-1096	The prover cannot be asked	Denial of service and the cost of public key operations; a resource property.
P395	rfc9678:1100-1102	Outside the scope	Identity privacy requirements for the Identity Response, outside the modelled flow.
P396 to P397	rfc9678:1106-1115	Outside the scope	Unprotected attributes and traffic analysis; metadata and privacy outside the modelled flow.
P398 to P400	rfc9678:1119-1136	C01	
P401 to P405	rfc9678:1140-1184	C08, C01, `one_component`, `ltk_active`	

The catalogue of properties, entry by entry

The properties were drafted from a catalogue of security and privacy properties. The draft left out 33 of its entries, and the analyst went through each of them: examined here as a property, or left out with the reason. Examined: 16. Left out: 17. That the documents do not speak of a property is not a reason the analyst may give.

Entry	Property of the catalogue	Examined as	Why not
K02	Payload and end-to-end confidentiality	Outside the scope	EAP-AKA' derives keys only; the payload it protects belongs to the lower layer, which the task leaves out.
K03	Strong secrecy / indistinguishability of the secret value	K01	
K04	Key usage, direction and session separation	K01	
D01	Message integrity and authenticity of the data origin	A08, A06	
A01	Peer aliveness	A03	
A02	Weak agreement authentication	A03	
A05	Implicit key authentication	A03, K01	
A07	Identity–credential–key binding	A03	
R02	Message-level replay resistance	A04, R01	
R03	At most one execution of a business operation	Not applicable	The protocol has no business operation to perform at most once.
R04	Message acceptance order / window safety	R01	
C02	Key-compromise impersonation (KCI) resistance	The prover cannot be asked	An attacker that holds the long-term key before or during the run cannot be expressed in this model, in which every disclosure is made after the session; the documents themselves state that such an attacker can impersonate any party.

Entry	Property of the catalogue	Examined as	Why not
C03	Known-key security / compromise isolation	K01	
C04	Resilience to ephemeral secret / partial state disclosure	C08	
C05	Post-compromise security (PCS)	The prover cannot be asked	Post-compromise recovery needs a run that starts after a long-term key disclosure; every disclosure in the model is made after the session, so no later run exists.
C06	Enhanced authentication after disclosure of the registration credential	Not applicable	There is no registration ceremony and no multi-part credential; the long-term key is shared by USIM and AD.
C07	Resistance to offline dictionary attacks	Not applicable	The AKA long-term key is a 128-bit random key, not a guessable password (RFC 4187 12.5).
Z01	Correct enforcement of authorization	Not applicable	Authorization, delegation, revocation and tenancy are not part of an EAP authentication method.
Z02	No expansion of delegated authority	Not applicable	Authorization, delegation, revocation and tenancy are not part of an EAP authentication method.
Z03	Binding of authorization to identity / session / operation	Not applicable	Authorization, delegation, revocation and tenancy are not part of an EAP authentication method.
Z04	Revocation and expiry take effect	Not applicable	Authorization, delegation, revocation and tenancy are not part of an EAP authentication method.
Z05	Cross-tenant / cross-service security isolation	Not applicable	Authorization, delegation, revocation and tenancy are not part of an EAP authentication method.
S01	Uniqueness of the cryptographic nonce	A04	
S02	Safety of retransmission / key reinstallation	R01, A04	
S03	Resistance to rollback and state monotonicity	R01	
S04	Consistency of the security context across phases	A03	
S05	Termination and invalidation of old state	R01	
P01	Identity anonymity	Outside the scope	Identity anonymity, unlinkability and existence hiding rest on SUCI, pseudonyms and fast re-authentication identities, which the modelled flow and the selected source units leave out.
P02	Session unlinkability	Outside the scope	Identity anonymity, unlinkability and existence hiding rest on SUCI, pseudonyms and fast re-authentication identities, which the modelled flow and the selected source units leave out.
P03	Hiding of user / credential existence	Outside the scope	Identity anonymity, unlinkability and existence hiding rest on SUCI, pseudonyms and fast re-authentication identities, which the modelled flow and the selected source units leave out.
V01	Conditional progress / error recovery	The prover cannot be asked	Liveness, absence of deadlock and resource abuse are outside what a symbolic trace property can express.
V02	No unexpected deadlock	The prover cannot be asked	Liveness, absence of deadlock and resource abuse are outside what a symbolic trace property can express.

Entry	Property of the catalogue	Examined as	Why not
V03	Bounded resource abuse / amplification resistance	The prover cannot be asked	Liveness, absence of deadlock and resource abuse are outside what a symbolic trace property can express.

The parties and the protected values, one by one

The host listed the 19 parties and protected values that the drafted properties name, and the analyst said of each which attacker controls the party or holds the value, or why no such attacker was set. With an attacker: 3. Without: 16. The list follows from the properties; what was made of each entry is the analyst's judgement.

ID	Party or value	Kind	Attacker	Why none
S01	peer	party	Nothing gained	A peer that is controlled by the attacker holds only its own USIM key and its own session keys; the attacker gains nothing about the server's assurances that the network attacker does not already have.
S02	server	party	`dishonest_server`	
S03	USIM/AD	party	`ltk_active`, `ltk_after`	
S04	MK_HYBRID	value	Everything lost by definition	Handing MK_HYBRID to the attacker is handing over the session keys themselves; the secrecy asked by K01 and C01 is lost by definition, and the documents do not ask it under this compromise.
S05	MSK	value	Everything lost by definition	MSK, EMSK and K_re are the session keys; holding them makes the secrecy question vacuous.
S06	EMSK	value	Everything lost by definition	MSK, EMSK and K_re are the session keys; holding them makes the secrecy question vacuous.
S07	K_re	value	Everything lost by definition	MSK, EMSK and K_re are the session keys; holding them makes the secrecy question vacuous.
S08	AT_MAC-covered negotiated list	value	Nothing gained	The MAC-covered list is a public field; an attacker that sees it gains nothing it does not see anyway, and changing it is caught by AT_MAC.
S09	AT_RES	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S10	enc	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S11	pkR	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S12	AT_KDF_FS list	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S13	selected hybrid KEM value	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S14	AT_MAC	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.

ID	Party or value	Kind	Attacker	Why none
S15	RAND	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S16	AUTN	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S17	RES	value	Nothing gained	AT_RES, enc, pkR, the AT_KDF_FS list, the selected value, AT_MAC, RAND, AUTN and RES travel on the wire or are public; holding them adds nothing to the network attacker.
S18	K_aut	value	`ltk_active`	
S19	identity	value	Nothing gained	The peer identity travels in the Identity exchange and is public in the modelled flow.

The properties under each attacker

The attackers are the analyst's, set before a query was written; each answer was recorded under one of them. A cell is the prover's answer for the property under that attacker; "No answer" is an attacker set for the property under which the prover left a claim undecided or nothing was recorded; an empty cell is an attacker the analyst did not set for the property.

Attacker	Has	Basis	Where
`network`	the network only: it reads, drops, changes and sends messages, and holds no secret of a party		
`ltk_after`	The AKA long-term key K and everything derived from it (CK, IK, CK', IK', MK, K_aut, K_encr) of the subscriber, obtained after the session under attack has ended.	The documents discuss a compromise of the AKA long-term keys on the USIM and ask that past sessions stay protected; s0056 notes that a retained skR plus the long-term key recovers the session keys.	rfc9678:929, s0056, s0028
`ltk_active`	The AKA long-term key of the subscriber before and during the run, used actively on the path.	Section 10: an attacker that holds the long-term key can still remove the hybrid KEM offer, and RFC 9678 7.2 says such an attacker can forge the negotiation and impersonate any party.	main:585, s0057
`dishonest_server`	A party on the network/AAA side that is dishonest or compromised, obtains a valid AKA authentication vector for the subscriber from the AD, and can therefore compute K_aut for that vector and send any AT_KDF_FS list with a valid AT_MAC.	RFC 3748 7.13 discusses the separation of authenticator and backend authentication server and the trust it needs; RFC 9048 7.2 describes a serving network that lets a run succeed and then misuses the keys. The task's scope declares the AAA/AKA vector source trusted, so this attacker is used only where the documents raise it.	rfc3748:2832, rfc9048:1358
`one_component`	The shared secret of one of the two hybrid components (the X25519 component or the ML-KEM component), with the other component intact.	Hybrid key exchange is meant to stay secure even if all but one component algorithm is broken; the hybrid KEM drafts state the safety properties component algorithms must have.	draft-irtf-cfrg-hybrid-kems:1031, s0021
`mk_hybrid_reveal`	MK_HYBRID, or the hybrid shared secret ss, of the session, disclosed after the run.	The user's question Q1 assumes the disclosure of the hybrid MSK component; the documents do not discuss it.	s0038

ID	`network`	`ltk_after`	`ltk_active`	`dishonest_server`	`one_component`	`mk_hybrid_reveal`
A03	Fails					

ID	network	ltk_after	ltk_active	dishonest_server	one_component	mk_hybrid_reveal
A04	Holds					
A08	Holds					
K01	Holds	Holds				
C01		Holds				
C08					Holds	
R05	Holds			Holds		
R01	Holds					
A06	Holds					
K01.10						Fails
R05.11	Holds			Holds		
A03.12	Fails					
X01	Holds					

Source and targets

Source: 147 units, 33631 bytes, frozen in source-delivery-01.json. Targets: 9 approved (A03, A04, A08, K01, C01, C08, R05, R01, A06); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

Decisions the host took by its documented default (task `decisions: auto`), for you to check:

- approve_package: approved the source package protocol-3ad9877a4632fe52f364bd17e792

This step as it reports itself: 1 requests (1 failed), 0.000 USD, 27975 tokens in and 4139 out, 24 s.

Notes on the prover and the model

- R01: answered in ProVerif, and the analyst names changing_state for this target

A ProVerif model that simplifies algebra (xor, dh_algebra, other_algebra) gives the attacker less than the real operation: a proof there is a proof against a weaker attacker, an attack there is an attack. A ProVerif model that leaves out a check on a value that changes (changing_state) lets parties accept more: a proof there holds, an attack there may be nothing but the missing check. Tamarin+ Agent models these as they are.

A bound in steps is an answer on bounded.pv: the analyst's second ProVerif model, in which the party that keeps a value that changes is unrolled to that many steps with its check in place. That model does less than the protocol does, so the property holds for those steps and no further, and an attack found there is an attack. The number of steps is the analyst's statement.

State review by the analyst

This part is the analyst's own reading of the source, not an answer of the prover. "yes" under shown by ProVerif means that ProVerif found a trace with these steps in the unrolled model, so the steps are possible; that they break a target is said only by an attack in the table of answers, and what else they cost is the analyst's reading. "nothing found" says which patterns were looked at and proves nothing.

value	kept by	changes when	checked when	refs
SQN (the AKA sequence number carried inside AUTN) and the highest SQN the subscriber has accepted	the peer (USIM/AD side: the USIM in the peer and the AD that issues the vector)	the USIM raises its stored value when it has verified an AUTN it accepts, i.e. once per accepted AKA'-Challenge; the AD raises the value it puts into a vector each time it issues one	at every AKA'-Challenge the peer compares the SQN recovered from AUTN with the value it keeps, after checking the AUTN MAC, and refuses when the SQN is not acceptable	s0021, s0028, s0035
the hybrid KEM private key skR and the encapsulation state (skx, skk) of the server, and the peer's encapsulation secrets (rx, rk)	the EAP server and the peer, for the length of one exchange	a new key pair is generated for every AT_PUB_HYBRID, so each challenge uses a different skR/pkR; the peer's encapsulation randomness is new per response	the server uses skR only to decapsulate the reply that matches the pkR it sent; the peer never reuses its secrets	s0033, s0040, s0056
the derived keys MK_HYBRID, MSK, EMSK and K_re of a completed run	the peer and the EAP server	each completed full authentication derives new values; the MSK is installed once into the lower layer	not compared; the peer's MSK is installed when it receives EAP-Success, without any check that the server derived the same value	s0031, s0038
the state of the AT_KDF_FS negotiation (offered list, selected value, echoed list)	the EAP server and the peer, for the length of one exchange	the selected value is inserted at the head of the echoed list in AT_PUB_HYBRID, which the peer compares with the list it was offered	the peer compares the echoed list with the first offer and requires the selected value first, under AT_MAC	s0015, s0018, s0028

pattern	result	target	shown by ProVerif	steps
replay	candidate		no: not tried with ProVerif	The attacker records EAP-Request/AKA'-Challenge for the subscriber, containing RAND, AUTN and AT_PUB_HYBRID with pkR. / The peer completes the run and sends its response; the attacker drops the response and delivers EAP-Success from its own store. / A second authentication for the same subscriber is started; the attacker delivers the recorded EAP-Request/AKA'-Challenge unchanged. / Without a sequence-number comparison the peer verifies AUTN again, encapsulates to the same pkR, accepts, and the attacker obtains a second run on one challenge.
repeat_one_time_action	nothing found			The one-time action of interest is deriving and installing the session keys and using pkR once. Two acceptances of one peer response are excluded on model.pv by A04 (each server acceptance has its own peer run, verified), and two acceptances of one challenge are excluded on bounded.pv by R01. A run of the same challenge delivering two server acceptances would need two peer responses under one pkR, which the server cannot obtain because it decapsulates the single reply it receives and A04 holds.
desynchronisation	nothing found			The AD raises the sequence number it issues and the peer only accepts a value greater than the one it keeps, so a refused run leaves the peer's value unchanged and the next vector still carries a greater value: no distance is created between the two and no recovery step is needed. bounded.pv shows the second run completing with SQN 2 after the first completed with SQN 1.
update_before_check	candidate		no: not tried with ProVerif	The USIM verifies the AUTN MAC of a challenge and accepts its SQN, raising the value it keeps; the peer has not yet sent its response. / The attacker drops the peer's AKA'-Challenge response, so the server never completes and never accepts. / The peer's stored sequence number has moved although no run completed; the same challenge can no longer be accepted by this peer.

pattern	result	target	shown by ProVerif	steps
no_rollback_on_failure	candidate		no: not tried with ProVerif	The peer verifies AUTN and accepts the challenge; its held sequence number moves forward. / The run is abandoned (the attacker drops the response, or the server stops), so no keys are installed on the server side. / The peer keeps the advanced value while no session exists; a recorded copy of the old challenge is now refused.
race_between_sessions	nothing found			Two concurrent runs of the same subscriber are what the replay candidate exploits, and with the sequence-number comparison the second acceptance of one challenge is excluded (R01, bounded.pv). EAP has one conversation at a time per peer; the model's two peer runs are sequential, one handing its value to the next, which is the stricter order. A genuinely concurrent pair of runs on the same stored value is outside what ProVerif can be asked here.
distinguishable_failure	candidate		no: not tried with ProVerif	A dishonest server that holds a recorded vector sends EAP-Request/AKA'-Challenge with it to a subscriber identity it suspects. / The AUTN MAC verifies at the peer and the sequence number is stale, so the peer answers EAP-Response/AKA'-Synchronization-Failure. / If the vector was not for that subscriber the peer answers EAP-Response/AKA'-Authentication-Reject instead, because its AUTN MAC does not verify under the key of that identity. / The two failure messages are different and carry an AT_MAC, so the answer tells the dishonest server that the recorded vector belongs to that subscriber.
wrap_or_window	nothing found			The draft does not define the acceptance rule for the sequence number and gives no window, so there is nothing in these documents to model; a 48-bit wrap needs 2^48 authentications and cannot be reached in any model. Settling it needs the AKA sequence-number rules from 3GPP TS 33.102, which is outside the source units, and would have to be done by Tamarin+ Agent with the integer rules written out.

What the analyst named for targets:

step	targets	features	reason
33	A03, A04, A06, A08, K01, C01, C08, R05, R05.11, A03.12, X01, K01.10	plain_primitives, many_sessions	These targets rest on PRF'/KDF, MAC, the AKA algorithms and a two-component KEM abstraction, all modelled as ideal functions; the hybrid KEM is a symbolic encapsulation with two independent components and no Diffie-Hellman algebra, so no computational or post-quantum claim is made. Every secrecy, agreement and downgrade query is asked over an unbounded number of peer, server and AD sessions, so the results are about any number of runs of the roles.
33	R01	changing_state	R01 asks that the evidence the peer accepts comes from a fresh challenge. Freshness rests on RAND/AUTN and on the sequence number the USIM keeps and updates and compares, which the model leaves out, so the peer accepts every challenge it can authenticate and the replay recorded for this target is the missing check. The half answer on a value that changes has to be given on an unrolled model.

The analyst's closing note

Model of the hybrid EAP-AKA' full authentication (draft-ietf-emu-hybrid-pqc-eapaka-02) with peer+USIM, EAP server, honest home AAA vector source and a dishonest server that holds a valid vector; model.pv runs and all thirteen targets have a recorded answer.

Established: the peer's accepted challenge and AT_PUB_HYBRID come from the server (A03 claim a, A03.12 claim a), the echoed AT_KDF_FS list is the offered one (A08), each server acceptance has its own peer run (A04), key confirmation holds both ways (A06), no hybrid public key is reused (X01). Session keys (MK_HYBRID/MSK/EMSK) stay secret during the run and against the AKA long-term key disclosed after it (K01,

C01), and survive the break of one hybrid component (C08) — all proved under idealised PRF'/KDF/MAC and a symbolic two-component KEM. A peer that requires a hybrid KEM cannot be pushed to ECDHE-only, either by the network or by a dishonest server with a valid vector (R05, R05.11), with the falsified control showing the peer that permits a fallback does accept the stripped offer.

Failing results, all recorded in the documents so no finding is new: the peer derives its keys and finishes on unprotected EAP-Success while the server accepted nothing (A03/A03.12 claim b, RFC 4187 Section 12.8); disclosing MK_HYBRID yields MSK/EMSK by definition, disclosing ss alone does not (K01.10). The replay of one challenge (R01) is an artifact of leaving the USIM sequence number out of model.pv; on bounded.pv with the AD issuing SQN 1 then SQN 2 and the peer comparing the last accepted value, the same query is verified for two unrolled runs.

Blocks: the sequence-number acceptance window and the 48-bit wrap are not in these documents and need Tamarin+ Agent with 3GPP TS 33.102 rules; the order of AUTN verification and the raise of the stored SQN is unspecified, so update_before_check and no_rollback_on_failure stay unconfirmed candidates; failure-message distinguishability (Synchronization-Failure vs Authentication-Reject revealing that a recorded vector is genuine) is a candidate finding without a model query; fragmentation, re-authentication and erasure of a retained skR are not modelled; the eight agreement and freshness proofs carry no control, because the variant that must fail needs an attacker holding the AKA long-term key actively, which the model does not contain.