Security analysis of PQ EAP-AKA prime full authentication

This report in another language: 中文.

draft-ietf-emu-pqc-eapaka-05 · ProVerif+ Agent report

Documents	Post-Quantum Key Encapsulation Mechanisms (PQ KEMs) in EAP-AKA prime (protocol definition); Improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA') (supporting); Forward Secrecy Extension to the Improved Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS) (supporting)
Properties examined	15: 11 hold, 4 fail (8 attacks)
Findings	None; known results reproduced in Appendix B: 2

Contents

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

1 Protocol and scope

PQ EAP-AKA' adds an ML-KEM key encapsulation to EAP-AKA' FS. The peer and its USIM authenticate to the EAP server through an AKA vector obtained from the operator's backend, and the peer encapsulates a shared secret to the server's ML-KEM public key.

The exchange starts with a privacy-friendly identity, then an AKA'-Challenge that lists the forward-secrecy key derivation functions, a request from the peer to use ML-KEM, a repeated Challenge that carries the ML-KEM public key, and the peer's response with AT_RES and the KEM ciphertext. Both sides end with CK and IK from AKA plus the KEM shared secret, from which MK_PQ_SHARED_SECRET, MSK and EMSK are derived.

The parties:

- **Peer / USIM**: EAP peer with the USIM. Holds: K, identity, network name.
- **EAP server**: with the trusted AD behind it. Holds: K, network name.
- **AD**: authentication backend. Holds: operator key, subscriber data.

Figure 1 shows the exchange as the documents specify it.

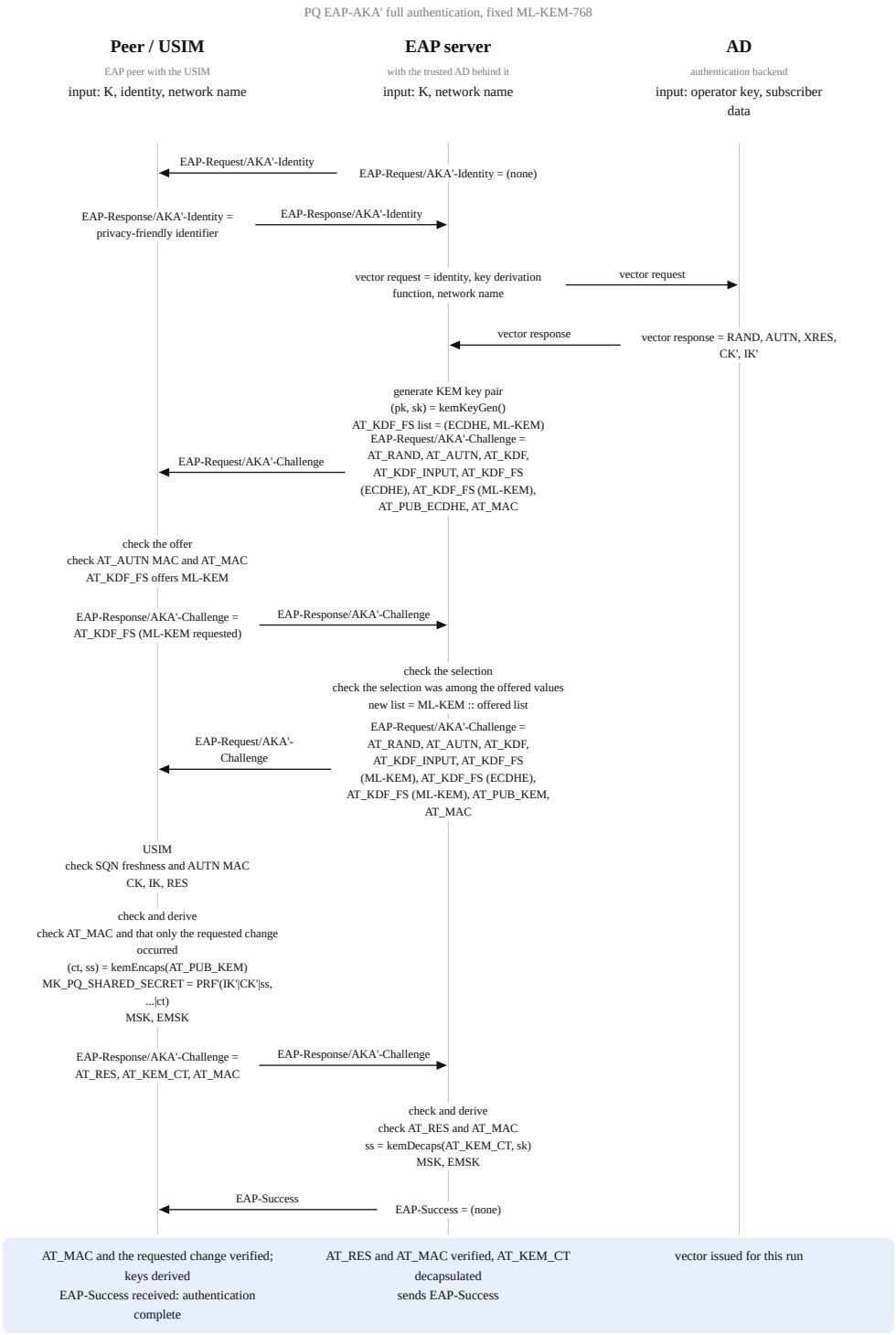


Figure 1. The protocol as specified.

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. Where a property fails, the documents know it: 2 such results are reproduced in Appendix B.

The protocol gives what it is meant to give in the modelled scope: the peer and the EAP server authenticate each other, the session keys stay secret, and the ML-KEM exchange gives forward secrecy against an attacker who later obtains the AKA long-term key. No result is new: both weaknesses the analysis reproduces, the unprotected EAP-Success and the impersonation possible with the long-term key, are stated in the documents. Replay protection rests on the AKA sequence number, which the model checks only after unrolling it to two steps. Fragmentation and the identity protection of AT_CHECKCODE were not modelled, so nothing is claimed for them. The remaining results are proofs over an idealised KEM and uninterpreted AKA and PRF functions.

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	**Mutual authentication of peer and EAP server.** Whenever the peer accepts a full authentication, there should exist a server run in which both identities agree and the MSK/EMSK derived by both sides agree with the negotiated ML-KEM parameters; likewise, when the server succeeds, this corresponds to a real run of the peer.	no compromise	Attack holds (stated in the documents)	medium	yes
A04	**Replay protection of the challenge exchange.** Each successful acceptance by the peer corresponds to a distinct server Challenge run; each success of the server corresponds to a distinct peer response run, and replaying old messages cannot produce a new success.	no compromise	Holds	medium	yes
K01	**Secrecy and forward secrecy of the session keys.** The attacker cannot derive MSK, EMSK, MK_PQ_SHARED_SECRET; after the AKA long-term key leaks, the previously recorded exchange still cannot yield MK_PQ_SHARED_SECRET.	AKA long-term key (leaked after the session completes)	Holds	medium	yes
C01	**Forward secrecy of a completed session.** If the AKA long-term key leaks after a session completes, the attacker still cannot derive the MSK/EMSK/MK_PQ_SHARED_SECRET of that past session.	AKA long-term key (leaked after the session completes)	Holds	medium	yes
R05	**Mandatory PQ KEM resists downgrade.** When both sides, or either side, require PQ KEM by local policy, the attacker cannot make authentication succeed without ML-KEM being negotiated.	AKA long-term key (leaked during negotiation)	Holds	medium	yes
A08	**AT_MAC binds KDF list, identity and RAND.** The ML-KEM parameters, network name, identity and challenge value accepted by the peer correspond to the values the server actually proposed and protected with AT_MAC.	no compromise	Attack holds (stated in the documents)	medium	yes
D01	**Integrity of AT_MAC-protected messages.** Every AT_MAC-protected message accepted by the peer and the server (including attributes after fragment reassembly) corresponds to a real send by an allowed sender, with payload and context fields consistent.	no compromise	Holds	low	yes

ID	Property	Compromise assumed	Result	Value	In the documents
R02	**Message instance accepted at most once.** The same fragmented message instance is accepted at most once in a single authentication; duplicate or out-of-order fragments cause an abort rather than repeated processing.	no compromise	Holds	low	yes
R01	**AUTN and KEM values of the current run.** The AUTN and KEM values accepted by the peer correspond to the current run; an old AUTN or old KEM value cannot be accepted as a new authentication.	no compromise	Holds within a bound (the changing value unrolled to 2 steps)	medium	yes
A06	**Key confirmation between peer and server.** At the confirmation point, the peer and the server have protocol evidence that the other side has derived the same session key.	no compromise	Attack holds (stated in the documents)	medium	yes
K01.11	**MSK secrecy and peer-server agreement.** The attacker cannot derive MSK; the peer and the server agree on MSK.	no compromise	Holds	medium	yes
A08.12	**AT_MAC binding of accepted challenge values.** The AT_KDF_FS list, identity, network name and challenge value accepted by the peer correspond to the values the server actually proposed and protected with AT_MAC.	no compromise	Attack holds (stated in the documents)	medium	yes
R05.13	**No downgrade under mandatory PQ policy.** Under a mandatory PQ configuration, the attacker cannot use replay or downgrade to make authentication succeed without ML-KEM being negotiated.	AKA long-term key (leaked during negotiation)	Holds	medium	yes
X01	**Key binding to ciphertext and KEM key.** when the peer and the server complete with the same MK_PQ_SHARED_SECRET, they agree on the ciphertext AT_KEM_CT and on the KEM public key AT_PUB_KEM that were used, against the attacker on the network	no compromise	Holds	low	yes
X02	**Forward secrecy for future sessions.** after the AKA long-term key has leaked, an attacker holding it cannot derive MK_PQ_SHARED_SECRET of a later session that the real peer and the real server complete	the AKA long-term key K shared by the USIM and the operator (and with it CK' and IK' for any RAND); obtained after an earlier authentication run and used against recorded earlier runs and against later runs	Holds	medium	yes

"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, X02.

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: MSK secrecy and peer/server key agreement	The MSK stays secret from the network attacker and the peer and server derive the same MK_PQ_SHARED_SECRET and MSK; the peer, however, is never confirmed that the server holds it.	K01.11, K01, A06
Q2: AT_KDF_FS list and identity/network name/challenge binding	The AT_KDF_FS list, identity, network name, RAND and AT_PUB_KEM the peer accepts are the server's AT_MAC-protected values against the network attacker, but not against an attacker holding the AKA long-term key.	A08.12, A08
Q3: behaviour of replay and downgrade policy under a mandatory PQ configuration	With a PQ KEM required by policy, a recorded challenge is refused once the SQN comparison is in place and neither replay nor downgrade makes either side complete without ML-KEM, even with the long-term key.	A04, R01, R05.13

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:

- ML-KEM is idealised as a public-key encapsulation of a fresh shared secret; AKA f1/f2, PRF' and AT_MAC are uninterpreted one-way functions. No computational or quantum claim is made.
- The SQN freshness comparison is modelled only in bounded.pv, with the issuer and the peer unrolled to two steps; the AKA acceptance window, SQN wrap and resynchronisation with AUTS are not modelled.
- Fragmentation (AT_FRAGMENT ordering, total length, the cross-packet AT_MAC, duplicate and out-of-order fragments) is not modelled; no claim is made about fragmented attributes.
- AT_CHECKCODE and the identity exchange are not modelled; the identity is bound only through K_aut and MK_PQ_SHARED_SECRET.
- Fast re-authentication, the ECDHE-only fallback, the no-policy downgrade path, channel binding and lower-layer or privacy properties are outside the scope.
- One operator key and one peer identity are modelled; a dishonest serving network, several identities and the real acceptance window are left out.

What the host records:

- The answers of A04, R02, R01 rest on an operation the model simplifies (Appendix D).

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: Mutual authentication of peer and EAP server

Whenever the peer accepts a full authentication, there should be a server run with matching identities on both sides and with the MSK/EMSK derived by both parties consistent with the negotiated ML-KEM parameters; likewise, a server success should correspond to a real peer run.

Under the network attacker the peer accepts the final challenge only if the server sent it, and the server accepts the response only if the peer sent it with the same values and key. The peer's completion is not tied to the server: EAP-Success is unprotected, so a peer can believe the server finished when it did not. An attacker holding the AKA long-term key impersonates either party, as the documents state.

- Result: Attack holds (stated in the documents); answers what is asked: partly; usable with caveats
- Value medium: It shows mechanically which part of mutual authentication holds and which does not, but both failures are already stated in the documents.
- Stated in the documents: yes. Where the documents say it: rfc4187:3772, rfc9678:1005
- Rests on: ML-KEM is idealised as a public-key encapsulation of a fresh shared secret; AKA functions, PRF' and AT_MAC are uninterpreted one-way functions; sessions in the model run before or after the long-term-key leak, never both; the identity exchange and AT_CHECKCODE are not modelled
- Open: whether AT_RESULT_IND is used in a deployment, which changes the peer's completion claim
- Next step: Model AT_RESULT_IND if a deployment is known to use it.

A04: Replay protection of the challenge exchange

Every successful acceptance by the peer corresponds to a distinct server Challenge run; every server success corresponds to a distinct peer response run, so replaying old messages cannot produce a new success.

With the sequence-number comparison in place, a recorded challenge is refused, and a recorded response cannot produce a second server success because the server's RAND is fresh per run. Without the comparison a replayed challenge is accepted, which is why the unrolled model is the one that answers this target.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It confirms the replay claim the documents make, at the price of a two-step unrolling and no window model.
- Stated in the documents: yes. Where the documents say it: rfc9048:1183, rfc4187:3707
- Rests on: the peer's SQN state is unrolled to two steps in bounded.pv; the AKA acceptance window, SQN wrap and resynchronisation are not modelled; the server stages in bounded.pv send their challenges without checking the response
- Open: a longer unrolling and the real acceptance window
- Next step: Check the window and resynchronisation with a tool that models the counter as it is.

K01: Secrecy and forward secrecy of the session keys

The attacker cannot derive MSK, EMSK or MK_PQ_SHARED_SECRET; after the AKA long-term key leaks, the previously recorded exchange still cannot derive MK_PQ_SHARED_SECRET.

MSK and EMSK are not derivable from the exchange by the network attacker, and an attacker who obtains the AKA long-term key after the run still cannot derive MK_PQ_SHARED_SECRET, because that needs the KEM shared secret. The control shows the leak is effective: values derived from the long-term key alone, such as K_aut, are recovered.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: A clean proof of the central secrecy and forward-secrecy claim, limited by the idealised KEM.
- Stated in the documents: yes. Where the documents say it: main:1131, rfc9678:929
- Rests on: the KEM is idealised; no computational or quantum claim is made; AKA and PRF' are uninterpreted functions; the attacker is given the long-term key only after the run
- Open: computational security of ML-KEM-768 itself
- Next step: None for a symbolic prover; the computational claim is the algorithm's.

C01: Forward secrecy of a completed session

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

A completed run's MK_PQ_SHARED_SECRET and its MSK and EMSK stay out of reach of an attacker who obtains the AKA long-term key afterwards: the KEM shared secret is encapsulated to a key pair that the session discards. This holds only because the server uses a fresh KEM key pair for the session, as the security considerations require.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The forward-secrecy claim of the documents reproduced mechanically, with the fresh-key-pair condition made explicit.
- Stated in the documents: yes. Where the documents say it: rfc9678:929
- Rests on: one KEM key pair per run in the model, as s0158 recommends; the KEM is idealised; the long-term key is revealed only after the run
- Open: what happens if a deployment reuses a KEM key pair
- Next step: Model key-pair reuse to show that forward secrecy is then lost, as s0158 implies.

R05: Mandatory PQ KEM resists downgrade

When both parties, or either party's local policy, mandate the PQ KEM, the attacker cannot make authentication succeed without ML-KEM being negotiated.

When the peer requires ML-KEM, it refuses an offer without it, and the server completes only after the peer selected ML-KEM; this holds even for an attacker holding the long-term key. The protection therefore comes from the policy check, exactly as the documents say.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It pins the protection to the policy check that the documents recommend, but the no-policy downgrade path is not modelled.
- Stated in the documents: yes. Where the documents say it: s0153, s0159

- Rests on: the policy is fixed to required in the model; the peer's check is modelled as requiring the canonical offer with ML-KEM and ECDHE; the fallback to EAP-AKA' or ECDHE without the policy is not modelled
- Open: the no-policy fallback path and the ECDHE-only deployment
- Next step: Add the fallback branch with the policy switched off to reproduce the documented downgrade.

A08: AT_MAC binds KDF list, identity and RAND

The ML-KEM parameters, network name, identity and challenge value that the peer accepts correspond to the values the server actually proposed and protected with AT_MAC.

The list, identity, RAND, network name and AT_PUB_KEM the peer accepts are the server's AT_MAC-protected values under the network attacker. An attacker holding the long-term key forges them, which the documents state for the ML-KEM offer.

- Result: Attack holds (stated in the documents); answers what is asked: partly; usable with caveats
- Value medium: Shows the binding and its exact limit, both already stated in the documents.
- Stated in the documents: yes. Where the documents say it: rfc9678:989, rfc9678:1005
- Rests on: the network name is a fixed constant on both sides; the identity exchange is not modelled; the identity is bound through K_aut and MK_PQ; the KDF list is the fixed ML-KEM-768 offer
- Open: identity binding through AT_CHECKCODE when the identity exchange is used
- Next step: Model AT_CHECKCODE if fragmented identity messages are in scope.

D01: Integrity of AT_MAC-protected messages

Every AT_MAC-protected message the peer and server accept (including attributes after fragment reassembly) corresponds to a real send by a permitted sender, with payload and context fields consistent.

For messages carried in one EAP packet, every AT_MAC-protected challenge and response accepted corresponds to a real send by the other party with the same content. The target also speaks of fragmented attributes, and fragmentation is not modelled, so nothing is claimed for them.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value low: The non-fragmented part is a direct consequence of the AT_MAC; the part the target stresses, fragmentation, is out of the model.
- Stated in the documents: yes. Where the documents say it: rfc9048:1175
- Rests on: AT_FRAGMENT ordering, total length and the cross-packet AT_MAC are not modelled; the AT_MAC is modelled over the message fields as a single term
- Open: integrity of reassembled fragmented attributes
- Next step: Model AT_FRAGMENT with the lock-step ack and the MAC over all fragments.

R02: Message instance accepted at most once

A given fragmented message instance is accepted at most once in a single authentication; duplicate or out-of-order fragments cause an abort rather than duplicate processing.

A challenge or response instance is accepted at most once in one authentication: the peer's sequence number rules out a repeat, the server's fresh RAND rules it out on its side. Duplicate or reordered AT_FRAGMENT handling is not modelled.

- Result: Holds; answers what is asked: partly; usable with caveats

- Value low: The message-level claim holds; the fragment-level claim the target also makes is untested.
- Stated in the documents: yes. Where the documents say it: s0014, s0060
- Rests on: two-step unrolling of the peer's SQN state; fragments are not modelled
- Open: duplicate and out-of-order fragments
- Next step: Model the fragment sequence and its acknowledgments.

R01: AUTN and KEM values of the current run

The AUTN and KEM values the peer accepts correspond to the current run; an old AUTN or old KEM value cannot be accepted as a new authentication.

An old AUTN is not accepted as a new authentication: the peer keeps the accepted sequence number and refuses a value that is not greater, so a recorded challenge is refused. An old KEM public key cannot be inserted either, because AT_MAC binds it to the RAND and the identity.

- Result: Holds within a bound (the changing value unrolled to 2 steps); answers what is asked: yes; usable with caveats
- Value medium: The freshness claim confirmed with the comparison in place, over a short unrolling.
- Stated in the documents: yes. Where the documents say it: rfc4187:3707
- Rests on: two-step unrolling of the SQN state; the acceptance window is not modelled
- Open: a longer unrolling and the real window
- Next step: Check with a tool that models the counter as an integer with a window.

A06: Key confirmation between peer and server

The peer and the server have protocol evidence at the confirmation point that the other side has derived the same session key.

The server obtains evidence that the peer completed: AT_RES and AT_MAC bind the peer's response and the ciphertext. The peer obtains no evidence that the server completed, because its only signal is EAP-Success, which the attacker can send. The documents note both the missing key confirmation and the unprotected success packet.

- Result: Attack holds (stated in the documents); answers what is asked: partly; usable with caveats
- Value medium: A concrete reproduction of a known gap, with the exact condition (the final EAP-Success).
- Stated in the documents: yes. Where the documents say it: rfc4187:3766, rfc9048:1402
- Rests on: AT_RESULT_IND is not modelled; AT_MAC is modelled with an uninterpreted MAC function
- Open: whether AT_RESULT_IND is used in a deployment
- Next step: Model AT_RESULT_IND and re-ask the peer-direction claim.

K01.11: MSK secrecy and peer-server agreement

The attacker cannot derive the MSK; the peer and the server agree on the MSK.

The MSK is not derivable by the network attacker, and the server's accepted response carries the same MK_PQ_SHARED_SECRET as the peer's, so the two parties derive the same MSK. This is a key-agreement result, not a confirmation to the peer that the server holds it.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Answers the user's first question with a proof over the idealised KEM.

- Stated in the documents: yes. Where the documents say it: s0091, s0107
- Rests on: the KEM is idealised; agreement is shown through the values carried by the accepted messages
- Open: confirmation of the agreement to the peer
- Next step: See A06.

A08.12: AT_MAC binding of accepted challenge values

The AT_KDF_FS list, identity, network name and challenge values that the peer accepts correspond to the values the server actually offered and protected with AT_MAC.

The accepted AT_KDF_FS list, identity, network name and challenge values are the server's protected values under the network attacker. They are not protected against an attacker who holds the long-term key, as the documents say.

- Result: Attack holds (stated in the documents); answers what is asked: partly; usable with caveats
- Value medium: Same content as A08 for the user's second question.
- Stated in the documents: yes. Where the documents say it: rfc9678:989
- Rests on: the network name is fixed on both sides; the offer is the fixed ML-KEM-768 list
- Open: identity binding through the identity exchange
- Next step: See A08.

R05.13: No downgrade under mandatory PQ policy

Under a mandatory PQ configuration, the attacker cannot, by replay or downgrade, make authentication succeed without ML-KEM being negotiated.

With the PQ KEM required by policy, neither a replay nor a downgrade makes the peer or the server complete without ML-KEM, even for an attacker holding the long-term key. Without the policy the documents say the ML-KEM offer can be removed, which this model does not exercise.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Answers the user's third question for the mandatory configuration.
- Stated in the documents: yes. Where the documents say it: s0153, s0159
- Rests on: policy fixed to required; the no-policy fallback path is not modelled
- Open: the no-policy configuration
- Next step: Add the fallback branch to reproduce the documented downgrade.

X01: Key binding to ciphertext and KEM key

when the peer and the server complete with the same MK_PQ_SHARED_SECRET, they agree on the ciphertext AT_KEM_CT and on the KEM public key AT_PUB_KEM that were used, against the attacker on the network

When the peer and the server complete with the same MK_PQ_SHARED_SECRET, they used the same ciphertext and the same KEM public key. This is a symbolic consequence of putting both into the derivation and of the encapsulation being injective.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: It restates the derivation structure; a computational binding claim needs a computational model.
- Stated in the documents: yes. Where the documents say it: main:1144
- Rests on: the KEM is a symbolic encapsulation; the binding is structural, not computational

- Open: computational MAL-BIND-K-CT/PK
- Next step: None for a symbolic prover.

X02: Forward secrecy for future sessions

after the AKA long-term key has leaked, an attacker holding it cannot derive MK_PQ_SHARED_SECRET of a later session that the real peer and the real server complete

After the long-term key leaks, an attacker holding it cannot learn the MK_PQ_SHARED_SECRET of a later session that the real peer and the real server both complete, because the shared secret is encapsulated to a fresh server key. An attacker that runs one of the parties itself does learn that session's key, which the documents also state.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Confirms the forward-secrecy statement for future sessions under the qualification the documents give.
- Stated in the documents: yes. Where the documents say it: rfc9678:935
- Rests on: the session key pair is fresh per run; the KEM is idealised; the claim is stated for sessions both real parties complete
- Open: an attacker acting as a party after the leak
- Next step: None; that case is the documented impersonation.

Appendix B: Known results reproduced

A property that fails as the documents already state, or that they set aside, is not a finding. What follows is of that kind: the analysis reproduces it on the model, which gives the exact conditions.

B.1 A forged EAP-Success makes the peer believe a failed run succeeded

Kind	Weakness of the protocol as specified
Stated in the documents	rfc4187:3772, rfc4187:3766
Value	medium
Properties	A03, A06 (section 3)
Shown by	ProVerif attack query `a03_peer_fin`

What happens. The peer's only signal that the server completed is the final EAP-Success, which carries no AT_MAC and no replay protection. After the peer sends its response, an attacker on the network sends EAP-Success; the peer derives its keys and reports success while the server never received the response. The keys themselves stay secret, so the loss is a false belief about the server, not a key compromise.

What the attacker needs. The attacker is on the network path between peer and server and can drop the peer's response and send EAP-Success afterwards. No key is needed.

What the attacker gains. The peer believes the server authenticated it and starts using a session key the server never accepted. Access is denied anyway, but the peer may act on a run that does not exist and its state can be desynchronised from the server.

Proposed change. Require the protected result indication AT_RESULT_IND for this method: the server sends an integrity-protected success indication keyed with K_aut, carrying the RAND and the derived MK_PQ_SHARED_SECRET, before EAP-Success, and the peer accepts EAP-Success and reports completion only after verifying it. Checked on the model: with that change the forged-success attack is gone, the normal run still exists, and all other recorded answers are unchanged.

Check of the change with ProVerif. The change was put into a copy of the model: The server sends an integrity-protected success indication keyed with K_aut (AT_RESULT_IND), carrying RAND and MK_PQ_SHARED_SECRET, before EAP-Success; the peer accepts EAP-Success and reports completion only after verifying that indication. On the changed model ProVerif proves `a03_peer_fin`: the attack is no longer possible there. The normal run of the protocol is still found. The other 32 recorded answers are unchanged. This is a first check on the model: it shows that the change removes this attack there, not that it introduces no other problem. Changed model: `records/fixes/0079-fix-succ-ind.pv`.

B.2 The AKA long-term key lets an attacker impersonate either party

Kind	Weakness of the protocol as specified
Stated in the documents	rfc9678:1005, main:1131
Value	medium
Properties	A03, A08, A08.12 (section 3)
Shown by	ProVerif attack queries `a03_peer_chal_ltk`, `a03_server_resp_ltk`, `a08_chal1_ltk`

What happens. An attacker who holds the AKA long-term key computes the AUTN MAC and AT_MAC for any RAND and therefore forges a Challenge to the peer; it can also compute AT_RES and AT_MAC and complete the exchange with the server without any peer. In the forged run the attacker uses its own KEM public key, so it also learns the session keys.

What the attacker needs. The attacker holds the long-term key K (and so CK' and IK' for any RAND) and is on the path for the run it attacks.

What the attacker gains. Full impersonation of the network to the peer or of the peer to the server for that run, with the session keys known to the attacker. Earlier and later runs that the real parties complete stay protected, as the forward-secrecy results show.

Proposed change. None. No change of messages, checks or derivations of this protocol removes it: the long-term key is the AKA credential and the AUTN and AT_MAC are computed from it by definition. The only available mitigation is the local policy that requires a PQ KEM, already asked as R05, so no protocol change was tested.

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		a03_peer_chal	checked	a03_peer_fin (holds_in_source); a03_peer_chal_ltk (holds_in_source); a03_server_resp_ltk (holds_in_source)	2 proved, 3 attack
A04	proved		a04_server_inj	checked	a04_peer_inj (model_artifact)	1 proved, 1 bounded, 1 attack
K01	proved		k01_msk	checked		3 proved
C01	proved		k01_kp	checked		1 proved
R05	proved		r05_peer	checked		3 proved
A08	proved		a08_chal1	checked	a08_chal1_ltk (holds_in_source); a03_peer_chal_ltk (holds_in_source)	2 proved, 2 attack
D01	proved		a03_server_resp	checked		3 proved
R02	proved		a04_server_inj	checked		1 proved, 1 bounded
R01	bounded	the changing value unrolled to 2 steps	a04_peer_inj	checked	a04_peer_inj (model_artifact)	1 bounded, 1 attack
A06	proved		a03_server_resp	checked	a03_peer_fin (holds_in_source)	1 proved, 1 attack
K01.11	proved		k01_msk	checked		2 proved
A08.12	proved		a08_chal1	checked	a08_chal1_ltk (holds_in_source); a03_peer_chal_ltk (holds_in_source)	2 proved, 2 attack
R05.13	proved		r05_peer	checked		4 proved
X01	proved		a03_server_resp	checked		1 proved
X02	proved		k01_future	checked		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	Peer completion (EAP-Success) implies a server completion with the same MK_PQ_SHARED_SECRET (attacker on the network).	`network`	attack (holds_in_source)	a03_peer_fin	
A03	Under an attacker that holds the AKA long-term key, the peer's accepted challenge corresponds to a server send with the same values.	`ltk_holder`	attack (holds_in_source)	a03_peer_chal_ltk (the control of a03_peer_chal)	

target	claim	attacker	answer	query	control
A03	Under an attacker that holds the AKA long-term key, a server completion corresponds to a real peer response with the same values.	`ltk_holder`	attack (holds_in_source)	a03_server_resp_ltk (the control of a03_server_resp)	
A03	Peer accepts the challenge only for a server send carrying the same identity, RAND, AT_KDF_FS list and AT_PUB_KEM (attacker on the network).	`network`	proved	a03_peer_chal	checked
A03	Server accepts a response only for a peer run with the same AT_RES, AT_KEM_CT, AT_PUB_KEM and MK_PQ_SHARED_SECRET (attacker on the network).	`network`	proved	a03_server_resp	checked
A04	No challenge instance is accepted by the peer twice (attacker on the network).	`network`	attack (model_artifact)	a04_peer_inj	
A04	No challenge instance is accepted by the peer twice: the peer keeps the accepted SQN and refuses a value that is not greater (attacker on the network).	`network`	bounded (the changing value unrolled to 2 steps)	a04_peer_inj	checked
A04	No response instance is accepted by the server twice; each server success corresponds to a distinct peer response (attacker on the network).	`network`	proved	a04_server_inj	checked
K01	MSK is not derivable by the attacker of a completed run.	`network`	proved	k01_msk	checked
K01	MK_PQ_SHARED_SECRET of a completed run is not derivable from the recorded exchange even after the AKA long-term key leaks.	`ltk_after_session`	proved	k01_kp	checked
K01	EMSK is not derivable by the attacker of a completed run.	`network`	proved	k01_emsk	checked
C01	MK_PQ_SHARED_SECRET and the MSK/EMSK of a completed run stay secret after the AKA long-term key leaks.	`ltk_after_session`	proved	k01_kp	checked
R05	The peer accepts a challenge only after an ML-KEM value has been negotiated.	`network`	proved	r05_peer	checked
R05	The server completes only after the peer selected an ML-KEM value.	`ltk_holder`	proved	r05_server	checked
R05	Under an attacker that holds the long-term key, no authentication succeeds in which ML-KEM was not negotiated.	`ltk_holder`	proved	r05_peer	checked
A08	Under an attacker that holds the long-term key, the peer's accepted list, identity and RAND are a server send under AT_MAC.	`ltk_holder`	attack (holds_in_source)	a08_chal1_ltk (the control of a08_chal1)	
A08	Under an attacker that holds the long-term key, the peer's accepted list, identity, RAND and AT_PUB_KEM are a server send under AT_MAC.	`ltk_holder`	attack (holds_in_source)	a03_peer_chal_ltk (the control of a03_peer_chal)	

target	claim	attacker	answer	query	control
A08	The peer's accepted AT_KDF_FS list, identity and RAND are the server's AT_MAC-protected values.	`network`	proved	a08_chal1	checked
A08	The peer's accepted AT_KDF_FS list, identity, RAND and AT_PUB_KEM are the server's AT_MAC-protected values.	`network`	proved	a03_peer_chal	checked
D01	Every AT_MAC-protected challenge and response the peer or server accepts corresponds to a real send by the other party with the same content (fragmented attributes are not modelled).	`network`	proved	a03_server_resp	checked
D01	Every AT_MAC-protected challenge the peer accepts corresponds to a real server send with the same content.	`network`	proved	a08_chal1	checked
D01	Every AT_MAC-protected final challenge the peer accepts corresponds to a real server send with the same content.	`network`	proved	a03_peer_chal	checked
R02	A challenge instance is accepted at most once within one authentication; a repeated transmission is refused (attacker on the network).	`network`	bounded (the changing value unrolled to 2 steps)	a04_peer_inj	checked
R02	A response instance is accepted by the server at most once.	`network`	proved	a04_server_inj	checked
R01	An old AUTN is not accepted as a new authentication: the peer keeps the last accepted SQN and refuses a value that is not greater.	`network`	bounded (the changing value unrolled to 2 steps)	a04_peer_inj	checked
R01	An old AUTN or old KEM value is not accepted as a new authentication.	`network`	attack (model_artifact)	a04_peer_inj	
A06	The peer has protocol evidence that the server completed the run and derived the same keys.	`network`	attack (holds_in_source)	a03_peer_fin	
A06	The server has protocol evidence that the peer derived the same MK_PQ_SHARED_SECRET: AT_RES and AT_MAC bind the peer's response.	`network`	proved	a03_server_resp	checked
K01.11	MSK is not derivable by the attacker and peer and server agree on MK_PQ_SHARED_SECRET for the run.	`network`	proved	k01_msk	checked
K01.11	The server's accepted AT_RES, AT_KEM_CT, AT_PUB_KEM and MK_PQ_SHARED_SECRET are the peer's (key agreement).	`network`	proved	a03_server_resp	checked
A08.12	Under an attacker that holds the long-term key, the accepted list and challenge values are the server's AT_MAC-protected values.	`ltk_holder`	attack (holds_in_source)	a08_chal1_ltk (the control of a08_chal1)	
A08.12	Under an attacker that holds the long-term key, the accepted list and AT_PUB_KEM are the server's AT_MAC-protected values.	`ltk_holder`	attack (holds_in_source)	a03_peer_chal_ltk (the control of a03_peer_chal)	

target	claim	attacker	answer	query	control
A08.12	The accepted AT_KDF_FS list, identity, network name and challenge values are the server's AT_MAC-protected values.	`network`	proved	a08_chal1	checked
A08.12	The accepted AT_KDF_FS list, identity and challenge values (with AT_PUB_KEM) are the server's AT_MAC-protected values.	`network`	proved	a03_peer_chal	checked
R05.13	Under a mandatory PQ KEM policy, the peer accepts a challenge only after ML-KEM was negotiated.	`network`	proved	r05_peer	checked
R05.13	Under a mandatory PQ KEM policy, the server completes only after ML-KEM was negotiated.	`network`	proved	r05_server	checked
R05.13	Under an attacker with the long-term key and a mandatory PQ KEM policy, neither replay nor downgrade makes the peer accept without ML-KEM.	`ltk_holder`	proved	r05_peer	checked
R05.13	Under an attacker with the long-term key and a mandatory PQ KEM policy, neither replay nor downgrade makes the server complete without ML-KEM.	`ltk_holder`	proved	r05_server	checked
X01	When peer and server complete with the same MK_PQ_SHARED_SECRET they agree on AT_KEM_CT and AT_PUB_KEM.	`network`	proved	a03_server_resp	checked
X02	After the long-term key leaks, an attacker holding it cannot derive MK_PQ_SHARED_SECRET of a later session that the real peer and the real server both complete.	`ltk_after_session`	proved	k01_future	checked

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-20261114T214415, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 81, prover 29 s, wall 17 min, cost 0.205 USD.

Context: the host removed stale parts from the conversation 1 time(s), 396579 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 108 proofs run, 107 decided, 1 without an answer, 12 s, 66 answers recorded.

What was analysed

Protocol	PQ EAP-AKA prime full authentication
Version	draft-ietf-emu-pqc-eapaka-05
Analysed	Fixed ML-KEM-768, one full non-reauthentication flow and a fresh AKA vector; the AKA algorithm and KEM are idealised, and a trusted AAA and operator keys are assumptions. The model includes AT_MAC, the KDF list check, AUTN freshness and the success condition; the order, total length and MAC of fragments must be kept as abstractions or explicitly left uncovered — security for fragmentation cannot be proved from a model without fragmentation.
Not analysed	A symbolic model does not prove the computational security or post-quantum security of the algorithms; attack traces hold only under explicit assumptions and are not directly equivalent to a spec vulnerability. Every property must have a reachability/positive control.
Source text	348 numbered passages (82641 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	6	PQ KEM Enhancements by Design
main	7	Message Fragmentation and Reassembly
main	7.1	Fragmentation Attribute
main	7.2	Fragmentation Procedure
main	7.3	Correlating Fragments and Acknowledgments
main	7.4	Reassembly
main	7.5	Fragment Ordering and Flow Control
main	7.6	Applicability
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_KEM
main	9.2	AT_KEM_CT

Document	Section	Title
main	9.3	AT_KDF_FS
main	10	Capability Negotiation
main	12	Security Considerations
main	12.1	Selecting ML-KEM Variants
rfc9048	3	EAP-AKA'
rfc9048	3.1	AT_KDF_INPUT
rfc9048	3.2	AT_KDF
rfc9048	3.3	Key Derivation
rfc9048	3.4	Hash Functions
rfc9048	3.4.1	PRF'
rfc9048	3.4.2	AT_MAC
rfc9048	3.4.3	AT_CHECKCODE
rfc9048	3.5	Summary of Attributes for EAP-AKA'
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
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, one complete non-reauthentication flow and a fresh AKA vector; the AKA algorithm and the KEM are idealized, and a trusted AAA and operator keys are assumed. The model includes AT_MAC, the KDF list check, AUTN freshness and the success condition; the ordering, total length and MAC of fragments must either be preserved in the abstraction or explicitly left uncovered, and the security of fragmentation cannot be proved from a model without fragmentation.

The properties against the documents

Before writing a query, 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	Mutual authentication of peer and EAP server	the peer is assured it is talking to the EAP server of its home operator and the server is assured of the peer's identity; both derive the same MSK/EMSK for the negotiated ML-KEM run	rfc9048 Section 7 claims mutual authentication for EAP-AKA'; the flow has the peer verify AT_AUTN/AT_MAC and the server check AT_RES/AT_MAC, success requiring both (s0088, s0091)	rfc9048:1170, s0088, s0091
A04	Replay protection of the challenge exchange	each peer acceptance corresponds to a fresh server challenge run and each server success to a fresh peer response; an old message cannot produce a new success	rfc9048 Section 7 makes the replay protection claim; rfc4187 Section 12.6 says full authentication replay protection comes from RAND and AUTN	rfc9048:1183, rfc4187:3707, s0091
K01	Secrecy and forward secrecy of the session keys	MSK, EMSK and MK_PQ_SHARED_SECRET kept from the network attacker, and from an attacker who obtains the AKA long-term key after the run	s0091: compromise of the AKA long-term key alone does not enable an adversary to derive MK_PQ_SHARED_SECRET from a previously recorded exchange; s0107 gives the derivation	s0091, s0107
C01	Forward secrecy of a completed session	MK_PQ_SHARED_SECRET and the MSK/EMSK of a completed run kept from an attacker who obtains the AKA long-term key afterwards	rfc9678 Section 7: someone who has recorded the authentication run and gets the AKA long-term keys cannot reconstruct the session keys; s0091 says the same for MK_PQ_SHARED_SECRET	rfc9678:929, s0091, s0158
R05	Mandatory PQ KEM resists downgrade	when the peer or the server requires a PQ KEM, no authentication completes without an ML-KEM value being negotiated	s0153: a server or peer that requires a PQ KEM and finds none MUST behave as if authentication failed; s0159: deployments needing protection against removal of the offer MUST require a PQ KEM by local policy	s0153, s0159
A08	AT_MAC binds KDF list, identity and RAND	the AT_KDF_FS list, identity, network name, RAND/AUTN and AT_PUB_KEM the peer accepts are the values the server sent under AT_MAC	s0159: AT_MAC protects the AT_KDF_FS offer and the peer's selection; rfc9678 Section 6.2: the peer must check that the requested change, and only the requested change, occurred (s0289)	rfc9678:989, s0159, s0289, s0152
D01	Integrity of AT_MAC-protected messages	every AT_MAC-protected message the peer or server accepts corresponds to a real send by the allowed sender with the same content	rfc9048 Section 7 makes the integrity protection claim for EAP packets; s0061/s0062 say a fragmented message's AT_MAC covers all fragments and data is processed only after AT_MAC verification	rfc9048:1175, s0061, s0062
R02	Message instance accepted at most once	a challenge or response instance is accepted at most once within one authentication; a repeated or reordered transmission aborts instead of being processed twice	s0014 and s0042 (one fragmented exchange, lock-step), s0047 and s0056 (retransmission reuses the Identifier), s0060 (inconsistent sequencing aborts the exchange)	s0014, s0042, s0047, s0056, s0060
R01	AUTN and KEM values of the current run	the peer accepts an AUTN and an AT_PUB_KEM that belong to the current run; an old AUTN or old KEM value is not accepted as a new authentication	s0099 (USIM validates the AUTN), s0125/s0141 (a fresh value is generated for each run), rfc4187 Section 12.6 (full authentication replay protection from RAND and AUTN)	rfc4187:3707, s0085, s0099, s0125, s0141

ID	Name	About	What the documents say	Where
A06	Key confirmation between peer and server	each side has protocol evidence at its confirmation point that the other derived the same session key	rfc9048 Section 7 notes that AKA would have benefited from key confirmation; AT_RES/AT_MAC give the server evidence about the peer, while the peer's only completion signal is EAP-Success, which rfc4187 Section 12.6 says is not integrity protected	rfc9048:1402, rfc4187:3724, s0091
K01.11	MSK secrecy and peer-server agreement	MSK is kept from the network attacker and peer and server derive the same MSK for a run (user question Q1)	user question Q1; s0107 defines MSK from MK_PQ_SHARED_SECRET, s0091 says the keys are based on CK, IK and the PQ KEM shared secret	s0107, s0091
A08.12	AT_MAC binding of accepted challenge values	the accepted AT_KDF_FS list, identity, network name and challenge values are the server's AT_MAC-protected values (user question Q2)	user question Q2; s0289 and s0159	s0289, s0159, s0186
R05.13	No downgrade under mandatory PQ policy	with a PQ KEM required by policy, neither replay nor downgrade makes an authentication succeed without ML-KEM (user question Q3)	user question Q3; s0153, s0159	s0153, s0159, s0289
X01	Key binding to ciphertext and KEM key	MK_PQ_SHARED_SECRET, AT_KEM_CT, AT_PUB_KEM	s0161: PRF' binds the shared secret to the ciphertext (MAL-BIND-K-CT) and ML-KEM binds it to the public key (MAL-BIND-K-PK)	main:1144
X02	Forward secrecy for future sessions	MSK, EMSK, MK_PQ_SHARED_SECRET of later sessions	rfc9678 Section 7: even if the long-term keys have been compromised, forward secrecy is still provided for all future sessions as long as the attacker does not become an active party to the run	rfc9678:935

Added from the documents by the analyst: X01, X02.

The security statements of the documents, paragraph by paragraph

The host listed the 260 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: 90. Not taken: 170. 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	main:1121-1123	The prover cannot be asked	quality of the random number generator is a property of the implementation, not a trace property
P002	main:1125-1129	C01, K01, `ltk_after_session`	
P003	main:1131-1136	A08, R05, `ltk_holder`	
P004	main:1138-1142	The prover cannot be asked	a locally configured maximum reassembly length and the DoS cost it bounds are resource questions outside a trace model
P005	main:1144-1149	X01	
P006 to P007	main:1158-1162	No goal stated	page furniture

Paragraphs	Where	Taken up as	Why not
P008	main:1167-1171	The prover cannot be asked	IND-CCA2 security and the choice of parameter set are computational claims
P009 to P010	rfc3748:2182-2190	No goal stated	general statement about the EAP threat model
P011 to P012	rfc3748:2194-2203	`network`	
P013	rfc3748:2205-2206	Outside the scope	discovering user identities concerns the identity exchange and the privacy-friendly identifier, which this flow carries as a fixed value
P014	rfc3748:2208-2208	D01, A08	
P015	rfc3748:2210-2213	A04, A06	
P016	rfc3748:2215-2216	No goal stated	the protocol uses no pass-phrase, so no dictionary attack arises
P017	rfc3748:2218-2219	A03	
P018	rfc3748:2221-2222	R05	
P019	rfc3748:2224-2225	K01	
P020	rfc3748:2233-2234	Outside the scope	ciphersuites used after the EAP conversation are outside the modelled flow
P021	rfc3748:2236-2238	R05	
P022	rfc3748:2240-2244	A08	
P023	rfc3748:2246-2252	No goal stated	observation that attacks may be remote; gives the network attacker only
P024 to P025	rfc3748:2256-2261	No goal stated	requirement that methods declare their claims
P026	rfc3748:2263-2271	A03, A04, D01, K01	
P027	rfc3748:2273-2276	The prover cannot be asked	key strength in bits is a computational estimate
P028	rfc3748:2286-2294	The prover cannot be asked	definition of key strength in bits
P029	rfc3748:2296-2299	The prover cannot be asked	note on the definition of key strength
P030	rfc3748:2301-2308	The prover cannot be asked	dependence of key strength on the algorithms
P031	rfc3748:2310-2313	K01	
P032	rfc3748:2315-2317	No goal stated	requirement that methods state which claims they do not make
P033	rfc3748:2321-2322	No goal stated	introduction to the claims terminology

Paragraphs	Where	Taken up as	Why not
P034	rfc3748:2324-2328	A08, R05	
P035	rfc3748:2339-2344	A03	
P036	rfc3748:2346-2351	D01	
P037	rfc3748:2353-2355	A04, R01, R02	
P038 to P039	rfc3748:2357-2368	K01	
P040	rfc3748:2370-2374	The prover cannot be asked	key strength in bits
P041	rfc3748:2376-2383	No goal stated	no password or other low-entropy secret is used
P042	rfc3748:2392-2396	Outside the scope	fast re-authentication is not part of the modelled full authentication
P043	rfc3748:2398-2405	Outside the scope	cryptographic binding is a property of tunneled EAP methods, not of EAP-AKA' itself
P044	rfc3748:2407-2411	C01	
P045	rfc3748:2413-2417	R02, D01	
P046	rfc3748:2419-2423	Outside the scope	channel binding to lower-layer identifiers is not part of the modelled flow
P047	rfc3748:2425-2427	No goal stated	note that the list of claims is not exhaustive
P048	rfc3748:2431-2434	Outside the scope	whether the identity exchange can be omitted concerns the identity exchange, not the challenge exchange
P049	rfc3748:2436-2439	Outside the scope	routing of the realm portion of the identity is not modelled
P050	rfc3748:2445-2449	A08	
P051	rfc3748:2451-2455	No goal stated	recommendation to use the authenticated identity for access decisions
P052	rfc3748:2459-2461	Outside the scope	man-in-the-middle through EAP tunneling is outside the modelled method
P053 to P054	rfc3748:2463-2481	Outside the scope	sequences of EAP methods and rogue authenticators through tunnels are outside the modelled method
P055 to P059	rfc3748:2483-2502	No goal stated	mitigations for tunneled deployments, not properties of this protocol
P060	rfc3748:2506-2508	D01	
P061	rfc3748:2510-2514	A04, R02	
P062 to P064	rfc3748:2516-2531	D01	

Paragraphs	Where	Taken up as	Why not
P065	rfc3748:2533-2536	Outside the scope	coverage of Identity, Notification and Nak messages is not modelled (AT_CHECKCODE is optional and absent from the modelled flow)
P066	rfc3748:2538-2542	Outside the scope	tunneling protection is outside the modelled method
P067 to P068	rfc3748:2551-2562	D01	
P069	rfc3748:2564-2567	A08, R02	
P070 to P072	rfc3748:2571-2587	No goal stated	no password or pass-phrase is used in AKA
P073 to P075	rfc3748:2591-2611	A03	
P076 to P077	rfc3748:2615-2629	R05	
P078 to P079	rfc3748:2633-2643	No goal stated	implementation notes about lower layers
P080	rfc3748:2645-2651	Outside the scope	retry counters of the implementation, not a protocol message
P081	rfc3748:2659-2665	K01	
P082	rfc3748:2667-2671	Outside the scope	use of the MSK/EMSK by other protocols is outside this method
P083	rfc3748:2673-2677	Outside the scope	AAA-Key derivation is outside this method
P084	rfc3748:2679-2682	K01	
P085 to P086	rfc3748:2684-2693	Outside the scope	export and media independence of the keying material, not a trace property of this flow
P087	rfc3748:2695-2699	R05	
P088 to P089	rfc3748:2701-2713	The prover cannot be asked	effective key strength in bits
P090 to P091	rfc3748:2715-2729	K01	
P092	rfc3748:2731-2735	Outside the scope	rules for how the EMSK may be used are outside the modelled flow
P093	rfc3748:2737-2740	Outside the scope	key lifetime handling is outside the modelled flow
P094 to P095	rfc3748:2742-2752	No goal stated	advice on how to design key derivation, no property of this protocol
P096 to P099	rfc3748:2765-2789	Outside the scope	ciphersuites and data protection after the EAP conversation are outside the modelled flow
P100 to P105	rfc3748:2793-2828	Outside the scope	link-layer indications (PPP, IEEE 802.1X, 802.11) are outside the modelled method
P106 to P107	rfc3748:2832-2842	No goal stated	description of the split between authenticator and backend server; the model assumes a trusted AD

Paragraphs	Where	Taken up as	Why not
P108	rfc3748:2844-2847	A03	
P109	rfc3748:2849-2855	Outside the scope	per-packet protection of the AAA protocol is outside the modelled method
P110 to P112	rfc3748:2857-2880	Outside the scope	secure association and AAA-Key transport after EAP are outside the modelled flow
P113 to P115	rfc3748:2884-2898	No goal stated	no cleartext password authentication is defined
P116 to P122	rfc3748:2902-2945	Outside the scope	channel binding to NAS identifiers is outside the modelled flow
P123 to P124	rfc3748:2949-2963	A06	
P125	rfc3748:2965-2969	No goal stated	statement that protected result indications are not required against rogue authenticators
P126	rfc3748:2975-2983	A06	
P127	rfc3748:2985-2989	Outside the scope	idle timeouts and resource freeing are resource questions
P128	rfc3748:2991-2997	A06	
P129 to P132	rfc3748:2999-3030	No goal stated	guidance on how result indications should be sent
P133 to P134	rfc3748:3032-3046	A06	
P135	rfc3748:3048-3056	No goal stated	example about EAP-TLS
P136	rfc4187:3594-3597	No goal stated	introduction to the EAP-AKA security considerations
P137 to P138	rfc4187:3608-3629	Outside the scope	identity protection in EAP-AKA rests on pseudonyms and the identity exchange, which the modelled flow carries as a fixed value
P139 to P140	rfc4187:3633-3641	A03	
P141	rfc4187:3643-3646	Outside the scope	channel bindings to the authenticator are outside the modelled method
P142	rfc4187:3650-3659	The prover cannot be asked	flooding the authentication centre is a resource question
P143	rfc4187:3663-3664	The prover cannot be asked	effective key strength in bits
P144	rfc4187:3666-3672	K01	
P145	rfc4187:3676-3680	No goal stated	AKA is not a password protocol, so dictionary attacks do not apply
P146	rfc4187:3684-3689	D01, A04	
P147	rfc4187:3691-3694	D01	

Paragraphs	Where	Taken up as	Why not
P148	rfc4187:3696-3701	K01	
P149	rfc4187:3707-3713	A04, R01	
P150	rfc4187:3715-3719	Outside the scope	fast re-authentication is not modelled
P151	rfc4187:3721-3722	A08	
P152 to P154	rfc4187:3724-3740	A06	
P155 to P157	rfc4187:3744-3757	R05	
P158 to P159	rfc4187:3761-3777	A06	
P160 to P161	rfc4187:3781-3799	Outside the scope	man-in-the-middle attacks through tunneling protocols are outside the modelled method
P162	rfc4187:3808-3811	The prover cannot be asked	quality of the random number generator
P163	rfc9048:1131-1136	No goal stated	introduction to the EAP-AKA' properties
P164	rfc9048:1138-1139	No goal stated	scope statement for the list of properties
P165	rfc9048:1141-1152	A08, R05	
P166	rfc9048:1154-1160	Outside the scope	bidding down from EAP-AKA' to EAP-AKA is a choice of EAP method, outside this method's negotiation
P167	rfc9048:1162-1168	No goal stated	outlook on future extensions
P168	rfc9048:1170-1173	A03	
P169	rfc9048:1175-1181	D01	
P170	rfc9048:1183-1186	A04, R01, R02	
P171 to P173	rfc9048:1188-1206	K01	
P174	rfc9048:1208-1210	K01, C01	
P175	rfc9048:1212-1213	No goal stated	description of the PRF construction
P176	rfc9048:1215-1216	The prover cannot be asked	key strength in bits
P177	rfc9048:1218-1221	No goal stated	no password is used
P178	rfc9048:1223-1228	The prover cannot be asked	key strength in bits

Paragraphs	Where	Taken up as	Why not
P179 to P181	rfc9048:1230-1247	Outside the scope	fast reconnect is not part of the modelled full authentication
P182 to P183	rfc9048:1249-1260	C01	
P184 to P197	rfc9048:1264-1344	Outside the scope	identity privacy rests on SUCI or pseudonyms and on the identity exchange, which the modelled flow carries as a fixed identifier
P198	rfc9048:1348-1351	No goal stated	statement that no attack is known against the original trust model
P199	rfc9048:1353-1356	No goal stated	statement that attacks need a different trust model
P200 to P201	rfc9048:1358-1380	Outside the scope	insider attacks by a serving network that misuses keys are outside the modelled trust assumptions (trusted AAA and operator)
P202	rfc9048:1382-1391	No goal stated	tunneling of traffic between operators, not a property of this method
P203	rfc9048:1393-1400	The prover cannot be asked	exhausting the sequence number space is a resource question
P204	rfc9048:1402-1404	Outside the scope	AKA without session keys and HTTP Digest AKAv1 are outside this flow
P205	rfc9048:1406-1411	A06	
P206	rfc9048:1413-1418	No goal stated	reports of compromised long-term keys; the attacker is entered as ltk_after_session
P207	rfc9048:1420-1433	Outside the scope	resynchronisation-based subscriber detection is outside the modelled full authentication
P208	rfc9048:1435-1442	Outside the scope	SQN resynchronisation and keystream reuse are outside the modelled flow
P209	rfc9048:1444-1448	Outside the scope	paging-protocol attacks are outside AKA
P210 to P211	rfc9048:1452-1458	No goal stated	general pervasive-monitoring considerations
P212 to P213	rfc9048:1460-1472	No goal stated	operational protection of key material
P214	rfc9048:1474-1482	C01, `ltk_after_session`	
P215 to P216	rfc9048:1486-1515	A08	
P217	rfc9048:1517-1526	A08, `ltk_holder`	
P218	rfc9048:1528-1537	No goal stated	how names are assigned to access networks
P219	rfc9678:868-870	No goal stated	scope statement of the section
P220	rfc9678:872-882	No goal stated	description of key compromise risks
P221	rfc9678:884-888	No goal stated	description of the consequences of missing forward secrecy
P222 to P223	rfc9678:890-894	No goal stated	consequences for recorded and future traffic

Paragraphs	Where	Taken up as	Why not
P224	rfc9678:896-898	No goal stated	listing of on-path attacks
P225 to P226	rfc9678:900-924	No goal stated	recommendation that forward secrecy be mandated
P227	rfc9678:926-933	C01, K01, `ltk_after_session`	
P228	rfc9678:935-937	X02, `ltk_after_session`	
P229	rfc9678:939-943	A03, A08, `ltk_holder`	
P230	rfc9678:945-954	No goal stated	description of re-running the exchange for protection in the other direction
P231 to P233	rfc9678:958-973	No goal stated	deployment guidance on destroying ephemeral keys and on key transport
P234	rfc9678:977-978	No goal stated	scope statement of the section
P235 to P236	rfc9678:980-986	A08, R05	
P237 to P238	rfc9678:988-1003	A08, R05, `ltk_holder`	
P239 to P240	rfc9678:1005-1025	R05	
P241 to P242	rfc9678:1027-1046	K01, C01, `ltk_after_session`	
P243 to P249	rfc9678:1050-1096	The prover cannot be asked	denial of service and computing cost are resource questions
P250	rfc9678:1100-1102	Outside the scope	identity privacy rests on the privacy-friendly identity mechanism, outside the modelled flow
P251 to P252	rfc9678:1106-1115	Outside the scope	traffic analysis and tracking are privacy properties outside the modelled flow
P253 to P255	rfc9678:1119-1136	Outside the scope	privacy of re-authentication pseudonyms inside AT_ENCR is outside the modelled flow
P256 to P258	rfc9678:1140-1161	The prover cannot be asked	quantum-computational assumptions
P259	rfc9678:1163-1170	The prover cannot be asked	quantum resistance of symmetric algorithms
P260	rfc9678:1172-1184	The prover cannot be asked	quantum threat to forward secrecy

The catalogue of properties, entry by entry

The properties were drafted from a catalogue of security and privacy properties. The draft left out 32 of its entries, and the analyst went through each of them: examined here as a property, or left out with the reason. Examined: 12. Left out: 20. 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	end-to-end payload confidentiality of the protocols that use the MSK is outside this method

Entry	Property of the catalogue	Examined as	Why not
K03	Strong secrecy / indistinguishability of secret values	Outside the scope	the method's claim is key derivation and secrecy of the keys, not indistinguishability of the allowed values; rfc3748 Section 7.2.1 does not claim it
K04	Key purpose, direction and session separation	K01	
A01	Peer aliveness	A03	
A02	Weak consistent authentication	A03	
A05	Implicit key authentication	A03, K01	
A07	Identity–credential–key binding	A08	
R03	Business operation at most once	Not applicable	EAP-AKA' performs no billable or state-changing business operation
R04	Message acceptance order / window security	R02	
C02	Resistance to key-compromise impersonation (KCI)	A08	
C03	Known-key security / compromise isolation	C01	
C04	Ephemeral secret / partial state compromise resilience	C01	
C05	Secure recovery after compromise (PCS)	X02	
C06	Enhanced authentication after compromise of enrolment credentials	Not applicable	there is no registration record or enrolment secret in EAP-AKA'
C07	Resistance to offline dictionary attacks	Not applicable	AKA uses a long-term key, not a password; rfc4187 Section 12.5 says EAP-AKA is not vulnerable to dictionary attacks
C08	Preservation of the goal under failure of a designated primitive	The prover cannot be asked	failure of a primitive is a computational assumption
Z01	Correctness of authorization enforcement	Not applicable	EAP-AKA' carries no authorization token, delegation, revocation or tenant model
Z02	Delegated authority does not expand	Not applicable	EAP-AKA' carries no authorization token, delegation, revocation or tenant model
Z03	Authorization bound to identity / session / operation	Not applicable	EAP-AKA' carries no authorization token, delegation, revocation or tenant model
Z04	Revocation and expiry take effect	Not applicable	EAP-AKA' carries no authorization token, delegation, revocation or tenant model
Z05	Cross-tenant / cross-service security isolation	Not applicable	EAP-AKA' carries no authorization token, delegation, revocation or tenant model
S01	Encryption nonce uniqueness	Not applicable	no nonce-based encryption primitive is used in the modelled flow
S02	Retransmission / key reinstallation security	R02	
S03	Resistance to rollback and state monotonicity	R01	
S04	Consistency of the security context across phases	Outside the scope	re-authentication and reconnection phases are outside the modelled full authentication

Entry	Property of the catalogue	Examined as	Why not
S05	Termination and invalidation of old state	Not applicable	the modelled flow ends with EAP-Success and no teardown state
P01	Identity anonymity	Outside the scope	identity privacy depends on the SUCI or pseudonym scheme and on the identity exchange, which the modelled flow carries as a fixed identifier
P02	Session unlinkability	Outside the scope	unlinkability depends on pseudonym generation outside the modelled flow
P03	Hiding of user / credential existence	Outside the scope	hiding whether a subscriber exists concerns the identity exchange and is outside the modelled flow
V01	Conditional progress / error recovery	The prover cannot be asked	progress, deadlock freedom and resource consumption are not trace properties ProVerif can decide
V02	No unexpected deadlock	The prover cannot be asked	progress, deadlock freedom and resource consumption are not trace properties ProVerif can decide
V03	Bounded resource abuse / resistance to amplification	The prover cannot be asked	progress, deadlock freedom and resource consumption are not trace properties ProVerif can decide

The parties and the protected values, one by one

The host listed the 24 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: 8. 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	`ltk_holder`	
S02	EAP server	party	`ltk_holder`	
S03	USIM	party	`ltk_after_session`	
S04	AD	party	`ltk_holder`	
S05	MSK	value	Everything lost by definition	MSK is precisely the value K01 keeps from the attacker; nothing further is claimed for a session whose MSK is lost
S06	EMSK	value	Everything lost by definition	EMSK is a protected value of K01; no further property is asked under its loss
S07	MK_PQ_SHARED_SECRET	value	Everything lost by definition	holding MK_PQ_SHARED_SECRET is exactly what K01 forbids; no property survives it
S08	K_aut	value	`ltk_holder`	
S09	K_encr	value	`ltk_holder`	
S10	K_re	value	`ltk_holder`	
S11	KEM shared secret ss	value	Everything lost by definition	holding ss of a session reveals that session's MK_PQ_SHARED_SECRET by definition; no other property is asked under it
S12	AT_KDF_FS list	value	Nothing gained	the AT_KDF_FS list is carried in the clear
S13	AT_PUB_KEM	value	Nothing gained	AT_PUB_KEM is a public key sent in the clear
S14	AT_KEM_CT	value	Nothing gained	AT_KEM_CT is sent in the clear and reveals nothing about ss without sk
S15	AT_MAC	value	Nothing gained	AT_MAC values are public and session-specific

ID	Party or value	Kind	Attacker	Why none
S16	AT_RES	value	Nothing gained	AT_RES is an AKA response value valid only for its own RAND; a replay of it is covered by the replay claim
S17	AKA long-term key	value	`ltk_after_session`, `ltk_holder`	
S18	Negotiated ML-KEM parameters	value	Nothing gained	the negotiated ML-KEM parameters are public
S19	AT_KDF_INPUT network name	value	Nothing gained	the network name is sent in the clear on purpose
S20	Identity	value	Nothing gained	the identity is carried in the clear; privacy of the identifier is outside the modelled flow
S21	AT_RAND/AT_AUTN	value	Nothing gained	RAND and AUTN are public values; an old AUTN is covered by R01
S22	AT_FRAGMENT fragments	value	Nothing gained	fragments are public data; their ordering and integrity are asked by R02 and D01
S23	EAP Identifier	value	Nothing gained	the EAP Identifier is a public octet field
S24	AUTN	value	Nothing gained	AUTN is public; replay of an old AUTN is the freshness claim R01

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_session`	the AKA long-term key K shared by the USIM and the operator (and with it CK' and IK' for any RAND); obtained after an earlier authentication run and used against recorded earlier runs and against later runs	s0091: compromise of the AKA long-term key alone does not enable deriving MK_PQ_SHARED_SECRET from a recorded exchange; s0158 says key-pair reuse across sessions would break forward security; rfc9678 Section 7 discusses attackers who obtain the long-term keys	rfc9678:935, s0091, s0158
`ltk_holder`	the AKA long-term key K before or during the authentication, held by an active attacker on the path	s0159: an attacker that holds the long-term key can still remove the ML-KEM offer; rfc9678 Section 7 says an attacker with the long-term key can impersonate any party and must be on-path	rfc9678:1005, s0159

ID	`network`	`ltk_after_session`	`ltk_holder`
A03	Fails		Fails
A04	Holds		
K01	Holds	Holds	
C01		Holds	

ID	network	ltk_after_session	ltk_holder
R05	Holds		Holds
A08	Holds		Fails
D01	Holds		
R02	Holds		
R01	Holds		
A06	Fails		
K01.11	Holds		
A08.12	Holds		Fails
R05.13	Holds		Holds
X01	Holds		
X02		Holds	

Source and targets

Source: 348 units, 82641 bytes, frozen in source-delivery-01.json. Targets: 10 approved (A03, A04, K01, C01, R05, A08, D01, R02, 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-92a4b07b3f694b3fa7e5a5e8b766

This step as it reports itself: 1 requests (1 failed), 0.000 USD, 47699 tokens in and 4814 out, 27 s.

Notes on the prover and the model

- A04: answered in ProVerif, and the analyst names changing_state for this target
- R02: answered in ProVerif, and the analyst names changing_state for this target
- 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 in AUTN (the sequence number the USIM compares to decide freshness)	USIM behind the peer; the home network (AD) generates the value in the authentication vector	the peer/USIM accepts an authentication and replaces the stored value with the accepted SQN	when the peer processes the final EAP-Request/AKA'-Challenge and validates AT_AUTN with the USIM	s0085, s0099, s0088

pattern	result	target	shown by ProVerif	steps
replay	nothing found			With the comparison in place, a recorded challenge is refused by the second step of the peer chain: ProVerif verifies that no challenge instance is accepted twice. Without the comparison (control a04_peer_ctl) the replay goes through, so the property depends on the check.
repeat_one_time_action	nothing found			Each acceptance advances the kept SQN to the value just accepted, so the same vector cannot be installed twice; the same check that rules out the replay rules this out.
desynchronisation	nothing found			The attacker can deliver a later vector before an earlier one, which makes the peer's stored value larger and the earlier vector fail, but no key or authentication result is gained and the home network can always issue a new vector. Resynchronisation with AUTS is outside the modelled flow.
update_before_check	nothing found			In the unrolled peer the kept value is handed to the next step only after the AT_MAC, list, AUTN and SQN checks have passed; a failure ends the step and the value stays.
no_rollback_on_failure	nothing found			A refused message leaves the kept SQN unchanged; there is no failure path that advances or resets it in the model.
race_between_sessions	nothing found			The unrolled peer holds one value in sequence, so two concurrent authentications cannot be staged in bounded.pv; whether two sessions of the same USIM can race on the stored SQN is a case for a tool that models the counter as it is (Tamarin+ Agent).
distinguishable_failure	nothing found			All peer failures are silent in the model: no answer is sent and the exchange stops, so no failure answer tells the attacker which value the peer keeps.
wrap_or_window	nothing found			The unrolled SQN takes three consecutive values and cannot wrap; the AKA acceptance window is not modelled here (a case for Tamarin+ Agent).

What the analyst named for targets:

step	targets	features	reason
48	A04, R01, R02	changing_state	The replay and at-most-once claims rest on the AKA sequence number the USIM keeps and compares with the AUTN it receives (s0085, s0099): the peer accepts a challenge only if the SQN is greater than the value it kept. model.pv leaves the comparison out; bounded.pv unrolls the issuer and the peer to two steps with the comparison in place.
48	A03, A06, K01, K01.11, C01, X01, X02	plain_primitives	The ML-KEM is idealised as a public-key encapsulation: the peer encrypts a fresh shared secret under AT_PUB_KEM and only the server can decapsulate it; the AKA functions f1/f2, the PRF' derivations and AT_MAC are uninterpreted one-way functions. No algebraic law beyond this is used, so a proof holds for this idealisation and an attack found against it is an attack on the protocol.

The analyst's closing note

Model of the PQ EAP-AKA' full authentication (fixed ML-KEM-768, one AKA vector, trusted AD) in model.pv, plus bounded.pv where the AKA sequence number is unrolled to two steps with the freshness comparison. The normal run exists; all 15 targets have answers under the attackers set for them, each proof with a control that fails.

Established: peer/server agreement under the network attacker; MSK/EMSK secrecy and MK_PQ forward secrecy after a later AKA long-term-key leak; forward secrecy for later sessions both real parties complete; key binding of MK_PQ to the ciphertext and KEM public key; no completion without ML-KEM under a required PQ policy, even with the long-term key; no challenge accepted twice once the SQN comparison is in place (bounded, two steps, control without it falsified); server-side injective agreement.

Failing and stated in the documents (reproduced, not new): the peer's completion rests on unprotected EAP-Success, and an attacker holding the AKA long-term key impersonates either party and forges the ML-KEM offer. The first has a checked remedy: adding an integrity-protected success indication (AT_RESULT_IND) keyed with K_aut removes the attack, keeps the normal run and changes no other answer.

Blocking the rest: fragmentation and AT_CHECKCODE were not modelled, so D01 and R02 are answered only for single-packet messages; the SQN acceptance window, wrap and two sessions racing on the stored value are outside the two-step unrolling and are a case for Tamarin+ Agent; the KEM and AKA/PRF functions are idealised, so no computational or quantum claim is made.