Security analysis of IKEv2: ML-KEM additional key exchange and final authentication

This report in another language: 中文.

draft-ietf-ipsecme-ikev2-mlkem-09 · ProVerif+ Agent report

Documents	Post-quantum Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2) (protocol definition); Internet Key Exchange Protocol Version 2 (IKEv2) (supporting); Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2) (supporting); Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2) (supporting)
Properties examined	11: 11 hold, 0 fail
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

The documents describe IKEv2 with ML-KEM added as a second key exchange: two peers, an initiator and a responder, run IKE_SA_INIT, then one IKE_INTERMEDIATE exchange that carries an ML-KEM public key and ciphertext, and finally IKE_AUTH. The initiator holds its own signature key, the responder its own; the responder also holds the ML-KEM decapsulation secret for the exchange it answers.

Each session begins by agreeing a classical Diffie-Hellman secret and two nonces and derives the IKE SA keys from them. The initiator then sends an ML-KEM encapsulation key inside the protected IKE_INTERMEDIATE message, the responder sends back a ciphertext, and both rederive the keys so that they depend on the ML-KEM shared secret as well. IKE_AUTH then has each peer sign its first message, the other's nonce and the identity, together with a value covering all IKE_INTERMEDIATE messages, so both peers end the exchange knowing they talked to the named peer in this very run.

The parties:

- **Initiator**: starts the IKE SA. Holds: long-term signature key.
- **Responder**: answers and encapsulates. Holds: long-term signature key, ML-KEM decapsulation secret.

Figure 1 shows the exchange as the documents specify it.

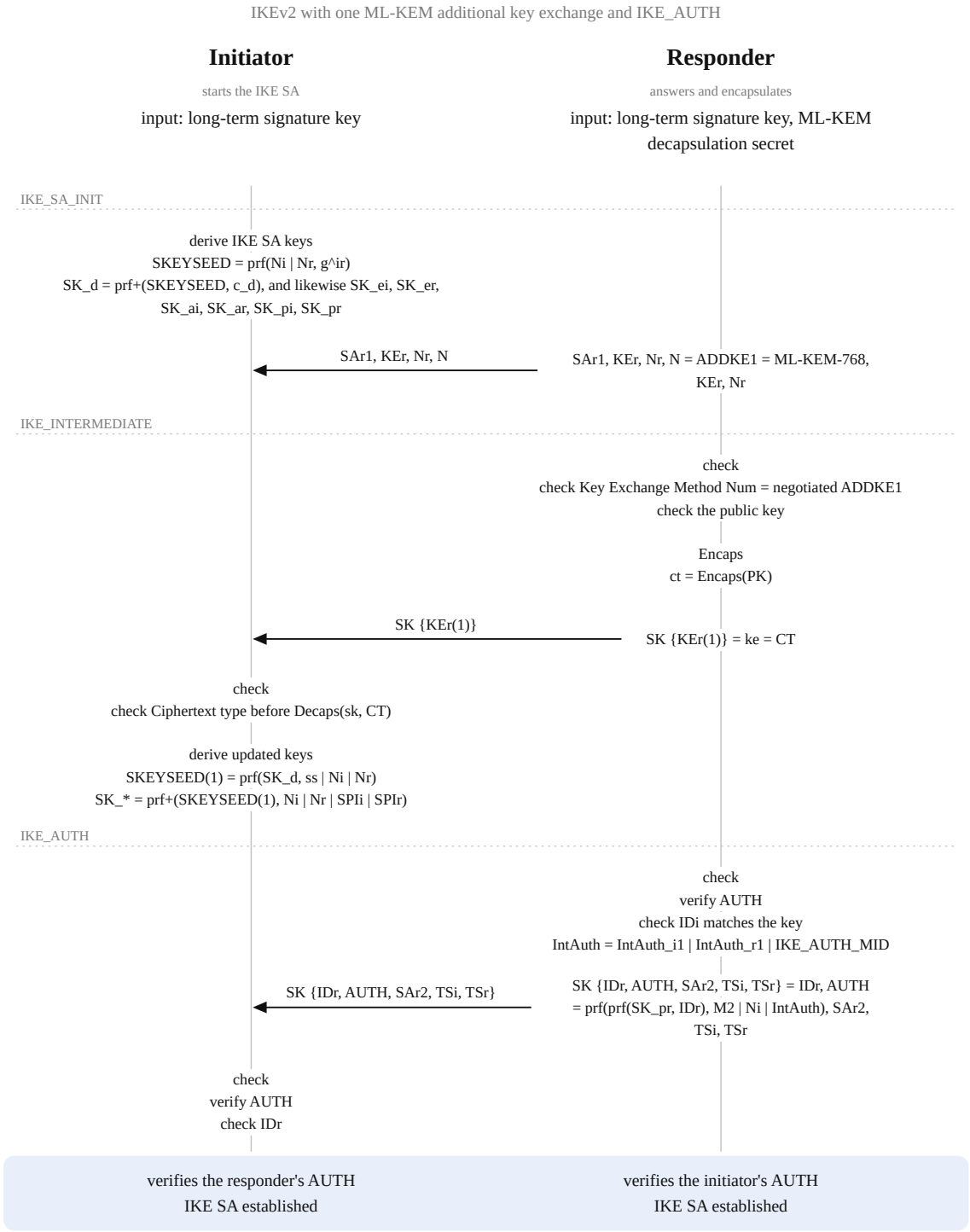


Figure 1. The protocol as specified.

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 analysed scope the protocol gives what it is meant to give, and the analysis found nothing the documents do not already state. Both peers are assured of the other's identity, of both nonces, of the SPIs and of the ML-KEM public key and ciphertext, and every acceptance corresponds to a distinct run of the peer. The IKE SA keys stay secret from the network attacker, they survive losing the classical Diffie-Hellman secret alone and the ML-KEM secret alone, and they survive a later leak of both long-term signature keys. The known weaknesses the documents themselves state were not turned into findings: the downgrade needs algorithm negotiation, which this scope fixes out, and the exposure of a broken classical exchange is shown to be caught in IKE_AUTH. All answers come from a model that replaces Diffie-Hellman by an abstract shared secret; Tamarin+ Agent, which models the algebra, would settle the Diffie-Hellman side.

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	**Agreement on peers and additional KE.** If the responder (or the initiator) accepts an IKE SA after IKE_AUTH completes, then there exists a real run of the peer whose role, both intended identities, Ni, Nr, SPIs and the additional KE method and shared secret negotiated in IKE_INTERMEDIATE are consistent.	no compromise	Holds	medium	yes
A04	**No replay into a second acceptance.** Every acceptance of an IKE SA by a party corresponds to a distinct real run of the peer; the same IKE_SA_INIT/IKE_INTERMEDIATE/IKE_AUTH message sequence cannot be replayed to produce a second acceptance.	no compromise	Holds	medium	yes
A08	**Additional KE bound to negotiation.** The additional KE method and order accepted in IKE_INTERMEDIATE, and the shared secret, correspond to the ADDKE transforms negotiated in IKE_SA_INIT and are bound to the SPI, nonces and AUTH of the same IKE SA.	no compromise	Holds	medium	yes
K01	**IKE SA key secrecy.** The attacker cannot derive SKEYSEED, SK_d, SK_ei/SK_er, SK_ai/SK_ar, SK_pi/SK_pr of the IKE SA (including every generation after an additional KE update), nor the long-term authentication keys of the two peers.	no compromise	Holds	medium	yes
C01	**Forward secrecy of the IKE SA keys.** After the IKE SA completes and the ephemeral secrets are erased, leaking the long-term authentication keys of both peers does not let the attacker obtain the SKEYSEED or SK of the earlier session.	both long-term authentication keys (after the session completes)	Holds	medium	yes
R05	**No downgrade to a KE without ML-KEM.** The attacker cannot make the peers accept a KE proposal without ML-KEM when both policies require ML-KEM; the accepted configuration satisfies the minimum policy.	no compromise	Holds	low	yes
C08	**Session keys survive one broken KE.** When the attacker obtains either the classical (EC)DH shared secret or the ML-KEM shared secret, the IKE SA session keys remain secret from the attacker.	classical (EC)DH shared secret g^ir or ML-KEM shared secret ss (after KE completes, after session keys are derived)	Holds	medium	yes
K01.8	**IKE SA key secrecy, user question.** The attacker cannot derive SKEYSEED and the SKs of the IKE SA (including every generation after an additional KE update).	no compromise	Holds	medium	yes

ID	Property	Compromise assumed	Result	Value	In the documents
A03.9	**Consistent binding at IKE SA acceptance.** Whenever a party accepts an IKE SA, both peers agree on the identities, Ni, Nr, SPIs, the additional KE result and AUTH.	no compromise	Holds	medium	yes
U01	**Tampering with IKE_INTERMEDIATE is caught.** When the attacker deletes, replaces or tampers with the IKE_INTERMEDIATE additional KE, the peers do not accept the IKE SA without noticing; the accepted additional KE corresponds to the negotiated result and is covered by AUTH.	no compromise	Holds	medium	yes
X01	**ML-KEM keypair and ciphertext freshness.** No two IKE SAs of one peer use the same ML-KEM encapsulation key or the same ML-KEM ciphertext: the initiator generates a fresh keypair and the responder a fresh ciphertext for each ML-KEM key exchange, and no randomness is reused.	no compromise	Holds	low	yes

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: IKE SA key secrecy	Yes. The IKE SA keys are secret from the network attacker both before and after the ML-KEM exchange; they also survive an attacker that obtains the classical Diffie-Hellman secret alone, the ML-KEM shared secret alone, and both long-term signature keys after the session.	K01, K01.8, C08, C01
Q2: consistent binding of both peers' KE results, identities and final AUTH	Yes. Each acceptance is matched by a real run of the peer with the same identities, Ni, Nr, SPIs, ML-KEM public key and ciphertext, and the AUTH signature covers the nonces, the identity and the IKE_INTERMEDIATE content.	A03, A03.9, A08, A04
Q3: whether deletion/replacement/negotiation tampering of the additional KE is detected	Deleting or replacing the IKE_INTERMEDIATE key-exchange data does not lead to an undetected acceptance: the accepted ciphertext is always the one the responder sent and signed over, even when the attacker has broken the classical Diffie-Hellman exchange and can read and forge that message.	U01, A08, A03

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:

- Algorithm negotiation is not modelled: the ML-KEM parameter set and the presence of the additional key exchange are fixed, so the downgrade attack the draft describes cannot be exhibited here.
- Diffie-Hellman is replaced by an abstract commutative shared secret, so nothing about the algebra of the exponents is proved; the classical-break attacker is modelled by handing it the session's g^ir.
- ML-KEM is an ideal key encapsulation mechanism: no decapsulation failure, no implicit rejection, no malformed key or ciphertext check from section 2.2 of the draft.
- Child SA keying material, CREATE_CHILD_SA, IKE_FOLLOWUP_KE, rekeying and Childless IKE SAs are outside the scope and not modelled.
- Message IDs, the window for overlapping requests and IKE fragmentation are not modelled; the replay property rests on fresh nonces and identities, not on the Message ID rule.

- Availability is not asked: the DoS exposure of the responder before authentication, the cost of many IKE_INTERMEDIATE exchanges and the reassembly limits are not covered.
- Identity confidentiality, NAT traversal, EAP and password-based authentication are not modelled.
- Erasure is modelled only as a phase boundary: everything of a run is over before a leak is released, so nothing is said about an attacker that obtains the ephemerals of a session that is still running.

What the host records:

- The answers of C01, R05, C08, U01 rest on an operation the model simplifies (Appendix D).
- Proved without a control query (Appendix C): A03: a03_i, A03: a03_r, A04: a04_r, A04: a04_i, A08: a08, K01: k01_final, K01: k01_secret, K01: k01_ltk, R05: r05, K01.8: k01_final, K01.8: k01_secret, A03.9: a03_i, A03.9: a03_r, U01: a08, U01: u01_dh, X01: x01_pki, X01: x01_ct. 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

A03: Agreement on peers and additional KE

If the responder (or the initiator) accepts an IKE SA after IKE_AUTH completes, then there exists a genuine run of the peer whose role, both parties' intended identities, Ni, Nr, SPIs, and the additional KE method and shared secret negotiated in IKE_INTERMEDIATE all agree.

Both peers end the exchange knowing which peer they talked to and which nonces, SPIs, ML-KEM public key and ciphertext that run used; each acceptance is matched by a complete run of the other side, so an attacker on the network cannot make one side accept values the other never used.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: A central authentication guarantee of the combined flow, proved for any number of sessions, but over idealised primitives and with Diffie-Hellman abstracted.
- Stated in the documents: yes. Where the documents say it: s0371, s0376, s0559, s0568
- Rests on: Diffie-Hellman is an abstract commutative shared secret, not the group algebra.; Algorithm negotiation is fixed by the scope.; Signatures are ideal: no key substitution, no certificate handling.; The session name inside the model is fresh, so the match is on identities, nonces and key-exchange data rather than on the IKE SPIs.

A03.9: Consistent binding at IKE SA acceptance

Every time one side accepts an IKE SA, both parties agree on the identities, Ni, Nr, SPIs, the additional KE result and AUTH.

The same agreement holds when stated as the user's question: at every acceptance both ends agree on the identities, both nonces, the SPIs, the additional key-exchange result and the AUTH value, in both directions.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Confirms the binding IKE_AUTH is meant to give once IKE_INTERMEDIATE is added, but it is the same proof as A03.
- Stated in the documents: yes. Where the documents say it: s0371, s0559
- Rests on: Same simplifications as A03: ideal signatures, abstract Diffie-Hellman, fixed algorithm suite.

A04: No replay into a second acceptance

Every acceptance of an IKE SA by one side corresponds to a distinct genuine run of the peer; the same IKE_SA_INIT/IKE_INTERMEDIATE/IKE_AUTH message sequence cannot be replayed to produce a second acceptance.

A recorded IKE_SA_INIT, IKE_INTERMEDIATE and IKE_AUTH sequence cannot be replayed to produce a second acceptance at the responder, nor can one responder run be counted twice by the initiator: the correspondence is injective in both directions.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: A property the documents require through the Message ID rule is here shown to follow from the cryptographic binding alone, which is worth knowing but is not the mechanism the documents rely on.
- Stated in the documents: yes. Where the documents say it: s0549, s0371

- Rests on: The Message ID rule is not modelled; injectivity comes from fresh nonces and identities and the signatures over them.; Retransmission and the overlap window are not modelled.

A08: Additional KE bound to negotiation

The additional KE method and order accepted in IKE_INTERMEDIATE correspond to the ADDKE transform negotiated in IKE_SA_INIT and are bound to the same IKE SA's SPIs, nonces and AUTH.

The ML-KEM ciphertext the initiator accepts is exactly the one the responder produced, and the method accepted is the negotiated one, so the additional key exchange cannot be substituted or re-bound to another run. The binding comes from the AUTH signature over the IKE_INTERMEDIATE content and over the identity-derived keys.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Shows that the IntAuth mechanism of RFC9242 does bind the additional key exchange into the authentication, which is the point of the draft's design.
- Stated in the documents: yes. Where the documents say it: s0621, s0623, s0559, s0568
- Rests on: Negotiation is a constant in the model, so the answer covers binding, not the choice of method.; ML-KEM is ideal: ciphertext and shared secret are related by an uninterpreted function.

K01: IKE SA key secrecy

The attacker cannot derive the IKE SA's SKEYSEED, SK_d, SK_ei/SK_er, SK_ai/SK_ar, SK_pi/SK_pr (including every generation after the additional KE update), nor can it derive either side's long-term authentication keys.

The IKE SA keys of both key generations, and the long-term signature keys, stay out of the network attacker's hands: it cannot derive SKEYSEED, the direction keys or the AUTH keys from what it sees.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: A central guarantee for the combined flow, but nothing a reader of IKEv2 would doubt, and it holds here against idealised primitives.
- Stated in the documents: yes. Where the documents say it: s0363, s0623, s0625
- Rests on: Diffie-Hellman is abstract, so the classical part of the secrecy is assumed rather than derived from the group.; The KEM is ideal and the PRF a random function; no strength of a concrete primitive is proved.

K01.8: IKE SA key secrecy, user question

The attacker cannot derive the IKE SA's SKEYSEED or any of the SK keys (including every generation after the additional KE update).

For the user's question about key secrecy the answer is the same as for K01: SKEYSEED and every derived key of both generations are kept from the network attacker, before and after the ML-KEM exchange.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It answers the user's first question directly, with the same caveats as K01.
- Stated in the documents: yes. Where the documents say it: s0363, s0626
- Rests on: Same simplifications as K01.

C01: Forward secrecy of the IKE SA keys

After the IKE SA completes and the ephemeral secrets are erased, leaking both peers' long-term authentication keys cannot give the attacker the SKEYSEED or SK of earlier sessions.

After a session has completed, an attacker that obtains both peers' long-term signature keys still cannot recover that session's keys. The control run, in which the session key itself is handed over, shows the property can fail, so the proof is not empty.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Reproduces the forward secrecy the documents state as a requirement, for any number of sessions, with the exact assumption that the ephemerals are erased.
- Stated in the documents: yes. Where the documents say it: rfc7296:2446, rfc7296:2454, s0411
- Rests on: Erasure is modelled as a phase boundary: the ephemerals of the run are gone before the leak is released, as the documents require but do not formalise.; Diffie-Hellman is abstract, so the forward secrecy of the classical component rests on the abstraction.

C08: Session keys survive one broken KE

When the attacker obtains either the classical (EC)DH shared secret or the ML-KEM shared secret, the IKE SA session keys remain secret from the attacker.

An accepted IKE SA carries keys an attacker cannot compute from the classical shared secret alone, nor from the ML-KEM shared secret alone: the derivation mixes both. Handing over both components at once does break it, as the control shows.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: This is the property the extension exists for, shown with both single-component breaks; since Diffie-Hellman's algebra is not modelled, Tamarin+ Agent is the tool that would settle that side.
- Stated in the documents: yes. Where the documents say it: rfc9370:910, rfc9370:925, s0029, s0583
- Rests on: Diffie-Hellman is abstract: the classical-break attacker is given the session's g^ir rather than the exponents.; The KEM is ideal, so an attacker that breaks ML-KEM is modelled by handing it ss.; Only one run is protected per attacker: the compromise is chosen once for the whole run.

R05: No downgrade to a KE without ML-KEM

The attacker cannot make both parties accept a KE proposal without ML-KEM when both parties' policy requires ML-KEM; the accepted configuration satisfies the minimum policy.

Every IKE SA the peers accept in this model is the ML-KEM hybrid one: the public key accepted is the one carried in IKE_INTERMEDIATE and the method is bound into AUTH. The downgrade the draft describes is not exhibited, because the model fixes the algorithm suite as the scope requires, so the property is answered for the fixed suite and not for the negotiation.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value low: It confirms that the accepted exchange is the hybrid one, which follows from the fixed scope; it does not answer the downgrade question, which the documents themselves state as a known issue.
- Stated in the documents: yes. Where the documents say it: main:369, s0030, s0031
- Rests on: Algorithm negotiation is not modelled at all: there is no alternative proposal an attacker could steer the peers to.; The policy that requires ML-KEM is the fixed scope, not a modelled check.

- Open: Whether an active attacker can force a proposal without ML-KEM when the peers also support classical-only sequences.
- Next step: Model the SA payload negotiation with two proposals, or reuse the analysis the draft cites, to exhibit the downgrade.

U01: Tampering with IKE_INTERMEDIATE is caught

When the attacker deletes, replaces or tampers with the IKE_INTERMEDIATE additional KE, the two sides do not accept the IKE SA without noticing; the accepted additional KE corresponds to the negotiated result and is covered by AUTH.

Deleting, replacing or modifying the IKE_INTERMEDIATE key-exchange data does not lead to an undetected acceptance: an accepted IKE SA always carries the ciphertext the responder sent and signed over. This holds even for an attacker that has the classical Diffie-Hellman secret and can therefore read and forge those messages, because the substitution surfaces in IKE_AUTH.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It mechanises the statement of RFC9242 that a forgery of IKE_INTERMEDIATE becomes evident in IKE_AUTH, with the exact attacker the documents name.
- Stated in the documents: yes. Where the documents say it: rfc9242:463, rfc9242:501, s0559, s0568
- Rests on: The attacker with g^ir is modelled by handing it that secret; it does not get the signature keys, which is the condition the draft names.; Deletion is modelled only in the sense that a session losing a message never completes; there is no failure answer in the model.; Diffie-Hellman is abstract.

X01: ML-KEM keypair and ciphertext freshness

No two IKE SAs of one peer use the same ML-KEM encapsulation key or the same ML-KEM ciphertext: the initiator generates a fresh keypair and the responder a fresh ciphertext for each ML-KEM key exchange, and no randomness is reused.

Each IKE SA uses its own ML-KEM encapsulation key and its own ciphertext; no two sessions share either, as the draft requires, so neither peer reuses the randomness of a key exchange.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: It checks a rule the draft states in one sentence; the interesting part, independence of the two seeds, is not in the model.
- Stated in the documents: yes. Where the documents say it: main:334, s0023
- Rests on: Freshness is a fresh name per run; ProVerif covers any number of sessions.; Independence of the ML-KEM seed from the Diffie-Hellman seed, also required by section 3 of the draft, is not modelled: there is no shared seed.

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_i			2 proved
A04	proved		a04_r			2 proved
A08	proved		a08			1 proved
K01	proved		k01_final			3 proved
C01	proved		c01	checked		1 proved
R05	proved		r05			2 proved
C08	proved		c08_dh	checked		2 proved
K01.8	proved		k01_final			2 proved
A03.9	proved		a03_i			2 proved
U01	proved		a08			2 proved
X01	proved		x01_pki			2 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	If the initiator accepts the IKE SA, the responder really completed the same run: identities, Ni, Nr, the ML-KEM ciphertext of IKE_INTERMEDIATE and the public key agree.	`network`	proved	a03_i	none
A03	If the responder accepts the IKE SA, the initiator really sent the same identities, Ni, Nr, additional-KE shared secret and ciphertext, and the ML-KEM public key, under its AUTH signature.	`network`	proved	a03_r	none
A04	Every responder acceptance corresponds to a distinct initiator run: the IKE_SA_INIT/IKE_INTERMEDIATE/IKE_AUTH sequence cannot be replayed into a second acceptance.	`network`	proved	a04_r	none
A04	Every initiator acceptance corresponds to a distinct responder run.	`network`	proved	a04_i	none
A08	The ML-KEM ciphertext the initiator accepts in IKE_INTERMEDIATE is the one the responder actually sent in KEr(1), covered by AUTH; the additional KE method is the negotiated constant and the KE data is bound to the SPIs and nonces.	`network`	proved	a08	none
K01	SKEYSEED, SK_d, SK_ai/SK_ar, SK_ei/SK_er and SK_pi/SK_pr of both key generations stay secret from the network attacker.	`network`	proved	k01_final	none
K01	The first generation SKEYSEED and the IKE_SA_INIT keys stay secret from the network attacker.	`network`	proved	k01_secret	none
K01	The long-term authentication private keys of both peers are not derivable by the attacker.	`network`	proved	k01_ltk	none
C01	After a session completes, an attacker that obtains both peers' long-term signature keys cannot recover that session's SKEYSEED or SK.	`ltk_leak`	proved	c01	checked

target	claim	attacker	answer	query	control
R05	An accepted IKE SA always results from an ML-KEM additional key exchange: the public key the initiator accepts was received in IKE_INTERMEDIATE and the ML-KEM method is the one carried in the clear and bound into AUTH.	`network`	proved	r05	none
R05	Even when the classical (EC)DH is broken, the peers cannot be made to accept an IKE SA that is not the ML-KEM hybrid one.	`dh_break`	proved	r05	none
C08	An attacker that obtains the classical (EC)DH shared secret g^ir cannot derive the final SKEYSEED or SK of an established IKE SA.	`dh_break`	proved	c08_dh	checked
C08	An attacker that obtains the ML-KEM shared secret ss cannot derive the final SKEYSEED or SK of an established IKE SA.	`kem_break`	proved	c08_kem	checked
K01.8	SKEYSEED and every SK of both key generations stay secret from the network attacker.	`network`	proved	k01_final	none
K01.8	The initial-generation keys (SKEYSEED, SK_d and the direction keys) stay secret from the network attacker.	`network`	proved	k01_secret	none
A03.9	If the initiator accepts, the responder completed the same binding: identical identities, Ni, Nr, ML-KEM ciphertext and public key.	`network`	proved	a03_i	none
A03.9	If the responder accepts, the initiator sent the same identities, Ni, Nr, shared secret and ciphertext under its AUTH signature.	`network`	proved	a03_r	none
U01	Modifying or replacing the IKE_INTERMEDIATE ciphertext is caught: an accepted IKE SA always carries the ciphertext the responder sent and signed over.	`network`	proved	a08	none
U01	Even when the attacker has the classical g^ir (and so can read and forge IKE_INTERMEDIATE messages), an accepted IKE SA still carries the ciphertext the responder sent; the substitution is caught in IKE_AUTH.	`dh_break`	proved	u01_dh	none
X01	No two IKE SAs use the same ML-KEM encapsulation key (initiator's KEi(1)).	`network`	proved	x01_pki	none
X01	No two IKE SAs use the same ML-KEM ciphertext (responder's KEr(1)); the responder does not reuse its randomness.	`network`	proved	x01_ct	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-20261114T184054, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 64, prover 132 s, wall 20 min, cost 0.251 USD.

Context: the host removed stale parts from the conversation 2 time(s), 557996 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 108 proofs run, 99 decided, 9 without an answer, 114 s, 32 answers recorded.

What was analysed

Protocol	IKEv2: ML-KEM additional key exchange and final authentication
Version	draft-ietf-ipsecme-ikev2-mlkem-09
Analysed	One fixed ML-KEM parameter set, one IKE_INTERMEDIATE additional KE and IKE_AUTH; a fixed authentication method; Child SA rekeying and general multi-round KE are not yet covered.
Not analysed	A combination of several RFCs, whose modelling cost exceeds that of a single service; the authentication method, key-leak scenarios and the PQ abstraction must be fixed. Only properties under the explicit model and attacker conditions are proved; missing checks, a fully secure channel or a fully honest NF must not be taken as a substitute for the conclusion to be proved.
Source text	658 numbered passages (222590 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	2	ML-KEM in IKEv2
main	2.1	Key Exchange Payload
main	2.2	Recipient Tests
main	3	Security Considerations
rfc7296	1	Introduction
rfc7296	1.1	Usage Scenarios
rfc7296	1.1.1	Security Gateway to Security Gateway in Tunnel Mode
rfc7296	1.1.2	Endpoint-to-Endpoint Transport Mode
rfc7296	1.1.3	Endpoint to Security Gateway in Tunnel Mode
rfc7296	1.1.4	Other Scenarios
rfc7296	1.2	The Initial Exchanges
rfc7296	1.3	The CREATE_CHILD_SA Exchange
rfc7296	1.3.1	Creating New Child SAs with the CREATE_CHILD_SA Exchange
rfc7296	1.3.2	Rekeying IKE SAs with the CREATE_CHILD_SA Exchange
rfc7296	1.3.3	Rekeying Child SAs with the CREATE_CHILD_SA Exchange

Document	Section	Title
rfc7296	1.4	The INFORMATIONAL Exchange
rfc7296	1.4.1	Deleting an SA with INFORMATIONAL Exchanges
rfc7296	1.5	Informational Messages outside of an IKE SA
rfc7296	1.6	Requirements Terminology
rfc7296	1.7	Significant Differences between RFC 4306 and RFC 5996
rfc7296	1.8	Differences between RFC 5996 and This Document
rfc7296	2	IKE Protocol Details and Variations
rfc7296	2.1	Use of Retransmission Timers
rfc7296	2.2	Use of Sequence Numbers for Message ID
rfc7296	2.3	Window Size for Overlapping Requests
rfc7296	2.4	State Synchronization and Connection Timeouts
rfc7296	2.5	Version Numbers and Forward Compatibility
rfc7296	2.6	IKE SA SPIs and Cookies
rfc7296	2.6.1	Interaction of COOKIE and INVALID_KE_PAYLOAD
rfc7296	2.7	Cryptographic Algorithm Negotiation
rfc7296	2.8	Rekeying
rfc7296	2.8.1	Simultaneous Child SA Rekeying
rfc7296	2.8.2	Simultaneous IKE SA Rekeying
rfc7296	2.8.3	Rekeying the IKE SA versus Reauthentication
rfc7296	2.9	Traffic Selector Negotiation
rfc7296	2.9.1	Traffic Selectors Violating Own Policy
rfc7296	2.9.2	Traffic Selectors in Rekeying
rfc7296	2.10	Nonces
rfc7296	2.11	Address and Port Agility
rfc7296	2.12	Reuse of Diffie-Hellman Exponentials
rfc7296	2.13	Generating Keying Material
rfc7296	2.14	Generating Keying Material for the IKE SA
rfc7296	2.15	Authentication of the IKE SA
rfc7296	2.16	Extensible Authentication Protocol Methods
rfc7296	2.17	Generating Keying Material for Child SAs
rfc7296	2.18	Rekeying IKE SAs Using a CREATE_CHILD_SA Exchange
rfc7296	2.19	Requesting an Internal Address on a Remote Network
rfc7296	2.20	Requesting the Peer's Version
rfc7296	2.21	Error Handling
rfc7296	2.21.1	Error Handling in IKE_SA_INIT
rfc7296	2.21.2	Error Handling in IKE_AUTH

Document	Section	Title
rfc7296	2.21.3	Error Handling after IKE SA is Authenticated
rfc7296	2.21.4	Error Handling Outside IKE SA
rfc7296	2.22	IPComp
rfc7296	2.23	NAT Traversal
rfc7296	2.23.1	Transport Mode NAT Traversal
rfc7296	2.24	Explicit Congestion Notification (ECN)
rfc7296	2.25	Exchange Collisions
rfc7296	2.25.1	Collisions while Rekeying or Closing Child SAs
rfc7296	2.25.2	Collisions while Rekeying or Closing IKE SAs
rfc9242	3	Intermediate Exchange Details
rfc9242	3.1	Support for Intermediate Exchange Negotiation
rfc9242	3.2	Using Intermediate Exchange
rfc9242	3.3	The IKE_INTERMEDIATE Exchange Protection and Authentication
rfc9242	3.3.1	Protection of IKE_INTERMEDIATE Messages
rfc9242	3.3.2	Authentication of IKE_INTERMEDIATE Exchanges
rfc9242	3.4	Error Handling in the IKE_INTERMEDIATE Exchange
rfc9370	2	Multiple Key Exchanges
rfc9370	2.1	Design Overview
rfc9370	2.2	Protocol Details
rfc9370	2.2.1	IKE_SA_INIT Round: Negotiation
rfc9370	2.2.2	IKE_INTERMEDIATE Round: Additional Key Exchanges
rfc9370	2.2.3	IKE_AUTH Exchange
rfc9370	2.2.4	CREATE_CHILD_SA Exchange
rfc9370	2.2.5	Interaction with IKEv2 Extensions
rfc9370	2.2.5.1	Interaction with Childless IKE SA

The purpose of the protocol as the source step located it in the documents: fix a single ML-KEM parameter set, one additional KE in IKE_INTERMEDIATE, and IKE_AUTH; fix the authentication method; for now, do not renegotiate the Child SA and do not do general multi-round KE.

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 concerns, and what the documents say are the analyst's reading; the places named do exist in the documents.

ID	Name	About	What the documents say	Where
A03	Agreement on peers and additional KE	The responder and the initiator are each assured of the other's identity, of Ni and Nr, of the SPIs and of the additional KE method and shared secret SK(1), through the AUTH payload and the IntAuth chunk.	RFC7296 2.15 defines the signed octets each peer signs (its own first message, the peer's nonce, the peer's ID); RFC9242 3.3.2 adds IntAuth over the IKE_INTERMEDIATE messages; RFC9370 2.2.2 puts SK(1) into SKEYSEED(1).	s0371, s0376, s0559, s0568, s0623, s0626
A04	No replay into a second acceptance	Every acceptance of an IKE SA corresponds to a different real run of the peer; the same IKE_SA_INIT/IKE_INTERMEDIATE/IKE_AUTH sequence cannot be replayed to produce a second acceptance.	RFC7296 2.2 and RFC9242 3.2 require Message IDs 1,2,... for IKE_INTERMEDIATE and the next value for IKE_AUTH and require them to be verified; the run is bound to fresh Ni/Nr and AUTH.	s0212, s0549, s0567
A08	Additional KE bound to negotiation	The additional KE method, its position and the shared secret SK(1) accepted in IKE_INTERMEDIATE correspond to the ADDKE transform negotiated in IKE_SA_INIT and are bound to the same IKE SA SPIs, nonces and AUTH.	RFC9370 2.2.2 requires the KE method field to match the n-th negotiated additional exchange and puts SK(n), Ni and Nr into SKEYSEED(n); RFC9242 3.3.2 puts the IKE_INTERMEDIATE content into AUTH.	s0602, s0621, s0623, s0559, s0568
K01	IKE SA key secrecy	SKEYSEED, SK_d, SK_ei/SK_er, SK_ai/SK_ar and SK_pi/SK_pr of both key generations, and the long-term authentication keys, are kept from the attacker.	RFC7296 2.14 derives all IKE SA keys from SKEYSEED, RFC9370 2.2.2 rederives them from SKEYSEED(n) after each additional key exchange; the keys protect the subsequent exchanges.	s0363, s0366, s0623, s0625, s0555
C01	Forward secrecy of the IKE SA keys	The SKEYSEED and SK of a completed session stay secret from an attacker that obtains the long-term authentication keys of both peers after the session, the ephemeral secrets having been erased.	RFC7296 2.12 states that perfect forward secrecy means that after a connection closes and its keys are forgotten, an attacker with all long-term keys cannot reconstruct the session keys, and requires each endpoint to forget the information that could recompute them.	rfc7296:2446, rfc7296:2454, s0347, s0411
R05	No downgrade to a KE without ML-KEM	The KE method a peer accepts when the local policy requires ML-KEM; the attacker must not be able to make the peers settle on a key exchange sequence that contains no ML-KEM.	RFC9370 and the draft state that an active on-path attacker that breaks the traditional (EC)DH can force the weakest common key exchange, and recommend rejecting proposals without ML-KEM.	s0030, s0031, s0032

ID	Name	About	What the documents say	Where
C08	Session keys survive one broken KE	SKEYSEED and the SK of an established IKE SA against an attacker that obtains one of the two key exchange components, the classical g^ir or the ML-KEM shared secret ss.	RFC9370 4 states that whereas a quantum attacker may learn the value obtained by a classical key exchange, the other SK(x) values from a post-quantum algorithm ensure the final SKEYSEED is not compromised; the draft calls the hybrid derivation IND-CPA but not proven IND-CCA2.	rfc9370:910, rfc9370:925, s0029, s0568, s0583
K01.8	IKE SA key secrecy, user question	SKEYSEED and every SK of both key generations are kept from the network attacker.	Same clauses as K01: RFC7296 2.14 and RFC9370 2.2.2.	s0363, s0369, s0623, s0626
A03.9	Consistent binding at IKE SA acceptance	At every acceptance both ends agree on the identities, Ni, Nr, the SPIs, the additional KE result and the AUTH value.	Same clauses as A03: the signed octets of RFC7296 2.15 with the IntAuth chunk of RFC9242 3.3.2.	s0371, s0376, s0559, s0568
U01	Tampering with IKE_INTERMEDIATE is caught	Deleting, replacing or modifying the IKE_INTERMEDIATE KE data must not lead to an undetected acceptance; the accepted additional KE is the negotiated one and is covered by AUTH/IntAuth.	RFC9242 5 states the IKE_INTERMEDIATE data is not authenticated until IKE_AUTH completes and that a forgery becomes evident in IKE_AUTH; RFC9242 3.3.2 binds the content into AUTH.	rfc9242:463, rfc9242:501, s0559, s0568, s0602, s0621
X01	ML-KEM keypair and ciphertext freshness	The ML-KEM public key the initiator sends in KEi(1) and the ciphertext the responder sends in KEr(1), across all sessions.	The draft requires generating an ephemeral keypair and ciphertext for each ML-KEM key exchange and forbids responders to reuse randomness in the generation of KEM ciphertexts.	main:334, s0023

Added from the documents by the analyst: X01.

The security statements of the documents, paragraph by paragraph

The host listed the 54 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: 16. Not taken: 38. 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
P01	main:320-321	No goal stated	It only says that the security considerations of RFC9242 and RFC9370 apply; those paragraphs are read here on their own.
P02	main:323-338	X01, K01	
P03 to P06	main:340-358	X01	

Paragraphs	Where	Taken up as	Why not
P07	main:360-361	No goal stated	A pointer to implementation guidance for KEMs, it states no property of this protocol.
P08	main:363-367	C08	
P09	main:369-383	R05, `dh_break`	
P10 to P11	main:385-400	R05	
P12 to P13	main:404-408	No goal stated	Page furniture of the Internet-Draft, no content.
P14	main:411-429	Outside the scope	It discusses Childless IKE SAs, CREATE_CHILD_SA rekeying and IKE_FOLLOWUP_KE; the task fixes one IKE_INTERMEDIATE additional exchange and leaves Child SA rekeying out.
P15	rfc7296:6478-6488	Outside the scope	It states the limits of identity protection in base IKEv2 (the initiator identifies itself first, an impersonating responder can probe it). The task fixes the modelled parties' public identities and asks nothing about identity confidentiality.
P16	rfc7296:6490-6494	Outside the scope	EAP authentication; the task fixes the authentication method and EAP is not it.
P17	rfc7296:6496-6500	Outside the scope	Repeated CREATE_CHILD_SA rekeying without new exponentiations; Child SA rekeying is outside the task's scope.
P18	rfc7296:6502-6514	The prover cannot be asked	The strength of the Diffie-Hellman groups, a property of the primitive and of the exponent size, not of the message flow.
P19	rfc7296:6516-6519	The prover cannot be asked	Same subject: the strength of the groups themselves.
P20	rfc7296:6525-6527	No goal stated	A remark that IKE encourages defining more groups; nothing that must hold or can be attacked.
P21	rfc7296:6529-6530	C01	
P22	rfc7296:6532-6538	The prover cannot be asked	Implementation robustness against DoS by unauthenticated peers, which is availability and not askable of a trace property.
P23	rfc7296:6540-6542	The prover cannot be asked	A minimum output size of the negotiated PRF, the strength of a primitive.
P24	rfc7296:6544-6549	The prover cannot be asked	The quality of the random number generator, a primitive assumption the symbolic model does not represent.
P25	rfc7296:6551-6556	Outside the scope	It forbids negotiating ENCR_NULL and NONE integrity; the task fixes the algorithm suite and does not model negotiation of encryption or integrity transforms.
P26	rfc7296:6558-6563	Outside the scope	Pre-shared keys derived from passwords; the task fixes digital-signature authentication and does not model password-based authentication.
P27 to P28	rfc7296:6565-6580	Outside the scope	Information leaked by NAT_DETECTION notifications; NAT traversal is not modelled.
P29	rfc7296:6582-6606	Outside the scope	Non-key-generating EAP methods; EAP is not the modelled authentication method.
P30	rfc7296:6608-6612	Outside the scope	A requirement on EAP deployments; EAP is not modelled.

Paragraphs	Where	Taken up as	Why not
P31	rfc7296:6614-6619	The prover cannot be asked	Reassembly buffer exhaustion by long messages, an availability property.
P32 to P33	rfc7296:6621-6633	Outside the scope	Admission control, trust anchors and the Peer Authorization Database: local policy configuration, outside the modelled exchange.
P34 to P41	rfc7296:6637-6696	Outside the scope	Traffic selector authorization for Child SAs; neither traffic selectors nor Child SA creation are in the modelled scope.
P42	rfc9242:463-470	U01, A03	
P43	rfc9242:472-477	The prover cannot be asked	A DoS by many large IKE_INTERMEDIATE exchanges, an availability property.
P44	rfc9242:479-499	The prover cannot be asked	The same DoS subject: a malicious initiator that aborts before IKE_AUTH and the cap on the number of exchanges.
P45	rfc9242:501-514	U01, C08, `dh_break`	
P46	rfc9370:883-891	C08, K01, `dh_break`	
P47	rfc9370:893-908	The prover cannot be asked	Grover's algorithm and symmetric key lengths, the strength of a primitive.
P48	rfc9370:910-916	C08, `dh_break`	
P49	rfc9370:918-923	C08	
P50	rfc9370:925-932	C08, C01, `dh_break`	
P51	rfc9370:934-942	The prover cannot be asked	It states that the authentication step remains classical, that is the strength of the signature primitive against a quantum computer; the symbolic model assumes signatures unforgeable and can only answer it by handing over the keys, which would void every other answer.
P52	rfc9370:944-954	No goal stated	Rationale on the urgency of post-quantum authenticity; it states no property of this protocol.
P53	rfc9370:956-961	The prover cannot be asked	Increased DoS susceptibility before the initiator is authenticated, an availability property.
P54	rfc9370:963-980	Outside the scope	The childless IKE SA and IKE_FOLLOWUP_KE alternative; rekeying and follow-up exchanges are outside the task's scope.

The parties and the protected values, one by one

The host listed the 31 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: 5. Without: 26. 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	Initiator	party	Everything lost by definition	A controlled initiator knows its own long-term key and its own session keys, so every secrecy property of its sessions is lost by construction; the documents ask no secrecy property under a dishonest party here.

ID	Party or value	Kind	Attacker	Why none
S02	Responder	party	Everything lost by definition	The same for the responder: its own keys and session secrets are known to whoever controls it.
S03	Active man-in-the-middle	party	Nothing gained	The active on-path attacker is the network attacker itself; it holds nothing beyond the network alone.
S04	IKE SA identity binding	value	Nothing gained	The identity binding is public protocol state carried in the handshake; holding it gives nothing the network attacker lacks.
S05	Ni, Nr, SPIi, SPIr	value	Nothing gained	Ni, Nr and the SPIs are sent in the clear.
S06	Additional KE method and shared secret SK	value	`kem_break`	
S07	AUTH payload	value	Nothing gained	The AUTH payload travels on the wire; with ideal signatures it reveals nothing.
S08	Message ID	value	Nothing gained	The Message ID is in the clear in every header.
S09	Ni, Nr	value	Nothing gained	Ni and Nr are sent in the clear.
S10	AUTH	value	Nothing gained	Same as S07: the AUTH payload is transmitted.
S11	ADDKE transform type and ID	value	Nothing gained	The ADDKE transform types and IDs are negotiated in the clear in the SA payload.
S12	KE method field	value	Nothing gained	The KE method field travels in the clear in the KE payload.
S13	SPIi, SPIr	value	Nothing gained	The SPIs are the first fields of every IKE header.
S14	SKEYSEED	value	Everything lost by definition	Holding SKEYSEED is holding the session key; every secrecy property about it is lost by definition and the documents ask none under it.
S15	SK_d	value	Everything lost by definition	Same for SK_d, from which all later keys are derived.
S16	SK_ei/SK_er	value	Everything lost by definition	Same for SK_ei and SK_er.
S17	SK_ai/SK_ar	value	Everything lost by definition	Same for SK_ai and SK_ar.
S18	SK_pi/SK_pr	value	Everything lost by definition	Same for SK_pi and SK_pr.
S19	Long-term authentication key	value	`ltk_leak`	
S20	Historical SKEYSEED and SK	value	Everything lost by definition	Asking forward secrecy while the attacker already holds the historical session keys is lost by definition.
S21	SA proposal and selected KE method	value	Nothing gained	The SA proposal and selection are carried in the clear; the network attacker can modify them, but RFC7296 and RFC9370 bind them into AUTH, which is what R05 asks.
S22	Policy configuration	value	Nothing gained	A local policy choice is not a secret and gives an attacker nothing it does not already try on the network.
S23	SKEYSEED and SK	value	Everything lost by definition	The same as S14 to S18.
S24	g^ir	value	`dh_break`	
S25	ML-KEM ss	value	`kem_break`	
S26	Identity	value	Nothing gained	The peer identities are public names.
S27	Ni, Nr, SPI	value	Nothing gained	Nonces and SPIs are public.

ID	Party or value	Kind	Attacker	Why none
S28	Additional KE result	value	kem_break	
S29	ADDKE negotiation result	value	Nothing gained	The negotiated transform list is carried in the clear.
S30	IKE_INTERMEDIATE KE data	value	Nothing gained	The ML-KEM public key and ciphertext are public by design; the shared secret they yield is covered by S25.
S31	AUTH/IntAuth	value	Nothing gained	AUTH and IntAuth values travel on the wire.

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		
dh_break	the classical (EC)DH ephemeral shared secret g^ir of a session, as an attacker with a cryptographically relevant quantum computer that breaks the IKE_SA_INIT exchange while it takes place would have it	The draft says an active on-path attacker with a CRQC may break the traditional (EC)DH ephemeral exchanges while the negotiation is taking place; RFC9370 says a quantum attacker may learn the value obtained by a classical key exchange.	main:369, rfc9370:910, s0030, s0583
kem_break	the ML-KEM shared secret ss of a session, as an attacker that breaks the post-quantum key exchange would have it	RFC9370 motivates multiple key exchanges by the potential compromise of existing or future post-quantum algorithms; the draft calls the hybrid derivation IND-CPA only.	rfc9370:883, main:363, s0029, s0583
ltk_leak	the long-term authentication (signature) private keys of both peers, obtained after the session has completed and its ephemeral secrets have been erased	RFC7296 2.12 states the forward secrecy goal against an attacker who gets access to all of the long-term keys of the two endpoints after the connection is closed; RFC9370 4 notes the authentication step remains classical.	rfc7296:2446, rfc9370:934

ID	network	dh_break	kem_break	ltk_leak
A03	Holds			
A04	Holds			
A08	Holds			
K01	Holds			
C01				Holds
R05	Holds	Holds		
C08		Holds	Holds	
K01.8	Holds			
A03.9	Holds			
U01	Holds	Holds		
X01	Holds			

Source and targets

Source: 658 units, 222590 bytes, frozen in source-delivery-01.json. Targets: 8 approved (A03, A04, A08, K01, C01, R05, C08, U01); 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 source package protocol-8fc5b1f5e64e7786ee00259b5460

This step as it reports itself: 1 requests (1 failed), 0.000 USD, 94170 tokens in and 4420 out, 29 s.

Notes on the prover and the model

- C01: answered in ProVerif, and the analyst names dh_algebra for this target
- R05: answered in ProVerif, and the analyst names dh_algebra for this target
- C08: answered in ProVerif, and the analyst names dh_algebra for this target
- U01: answered in ProVerif, and the analyst names dh_algebra 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.

What the analyst named for targets:

step	targets	features	reason
59	C08, C01, U01, R05	dh_algebra, many_sessions	The classical key exchange is not modelled with Diffie-Hellman algebra: the shared secret of the IKE_SA_INIT exchange is an abstract commutative function of the two exponents, so the attacker gets less than in the real protocol, and the dh_break attacker is modelled by handing that session's g^ir to the attacker after the run rather than by letting it compute the exponentials. The answers for the hybrid and the downgrade properties therefore rest on this simplification, and Tamarin+ Agent, which models the DH algebra as it is, would settle them.
59	A03, A03.9, A04, A08, K01, K01.8, X01	plain_primitives, many_sessions	These targets use only hashing, PRF/KDF, signatures, symmetric encryption and the ML-KEM abstraction as ideal primitives, and ProVerif covers any number of sessions in one run, so no bound is involved. The negotiation of the algorithm suite is fixed by the task's scope and is not modelled.

The analyst's closing note

Eleven targets, all with answers from ProVerif on a single model of the flow (IKE_SA_INIT, one IKE_INTERMEDIATE ML-KEM exchange, IKE_AUTH), plus one added goal from the security considerations.

Established, for any number of sessions and against the network attacker: both peers agree on the identities, Ni, Nr, the ML-KEM public key and ciphertext, and the additional key exchange is bound into AUTH through IntAuth; every acceptance corresponds to a distinct run of the peer (injective, both directions); the IKE SA keys of both generations and the long-term signature keys stay secret; the accepted ciphertext is always the responder's own, so substitution or modification of IKE_INTERMEDIATE is caught; each ML-KEM key and ciphertext is used once; an accepted IKE SA is always the ML-KEM hybrid one. Against a broken classical exchange (dh_break) and against a broken ML-KEM (kem_break) separately, the final session keys and SKEYSEED stay out of reach, with a control in which both components are handed over and the key falls. Against a later leak of both long-term signature keys (ltk_leak), completed sessions keep their keys, with a control that hands over the session key and fails.

No finding: every result is something the documents require or state, so nothing new is reported. The two weaknesses the documents name were reproduced as limits rather than findings: the downgrade to a non-ML-KEM sequence needs algorithm negotiation, which the scope fixes out (R05 is therefore answered only for the fixed suite), and the exposure of IKE_INTERMEDIATE to an attacker holding g^ir is exactly what u01_dh shows is caught in IKE_AUTH.

What blocks more: the model replaces Diffie-Hellman by an abstract commutative shared secret and ML-KEM by an ideal KEM, so the hybrid property and the downgrade property rest on a simplification; Tamarin+ Agent, which models the DH algebra, would settle the Diffie-Hellman side. Seventeen proofs are recorded without a control variant; the control structure of the report shows which. Notes are in notes.md.