Security analysis of IKEv2 multiple intermediate key exchanges

rfc9370 · ProVerif+ Agent report

Documents	Multiple Key Exchanges 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)
Properties examined	15: 9 hold, 0 fail, 6 without an answer
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

IKEv2 with multiple intermediate key exchanges (RFC 9370, using the IKE_INTERMEDIATE exchange of RFC 9242) establishes an IKE SA between an initiator and a responder in three stages. IKE_SA_INIT negotiates a classical (EC)DH key exchange plus two additional post-quantum key exchange methods and carries the initial key exchange payload and nonces. Two IKE_INTERMEDIATE exchanges then run, each carrying a fresh ML-KEM public key from the initiator and a fresh encapsulation from the responder, and both extend the chain of IKE SA keys.

Each party holds a long-term identity and a signature private key; the initiator also holds its identity IDi, which travels only inside encrypted payloads. After the two intermediate exchanges IKE_AUTH carries the two signatures, and the signed octets of both parties include IntAuth, the cumulative value over the intermediate exchanges plus the IKE_AUTH message ID.

IKE_SA_INIT derives SKEYSEED(0) from the (EC)DH secret, the nonces and the SPIs, and splits it into SK_d, SK_ei, SK_er, SK_ai, SK_ar, SK_pi and SK_pr. Each intermediate exchange folds its own KEM shared secret into SKEYSEED(n), and the keys of generation n protect the messages of round n+1, so each extra exchange is authenticated with the SK_pi(n)/SK_pr(n) of the previous generation.

When the exchange ends each side is sure that the peer ran the same IKE SA, with the same negotiated additional methods in the same order, the same per-round key exchange payloads, and the same IntAuth value and IKE_AUTH message ID. The ML-KEM secrets of the two rounds are distinct, and the signatures of the last generation cover the whole run.

The parties:

- **Initiator**: starts the IKE SA; fresh DH exponent and nonce, fresh ML-KEM key pair per round. Holds: SKi, VKr, IDi.
- **Responder**: answers IKE_SA_INIT and both intermediate exchanges; encapsulates twice. Holds: SKr, VKi, IDr.

Figure 1 shows the exchange as the documents specify it.

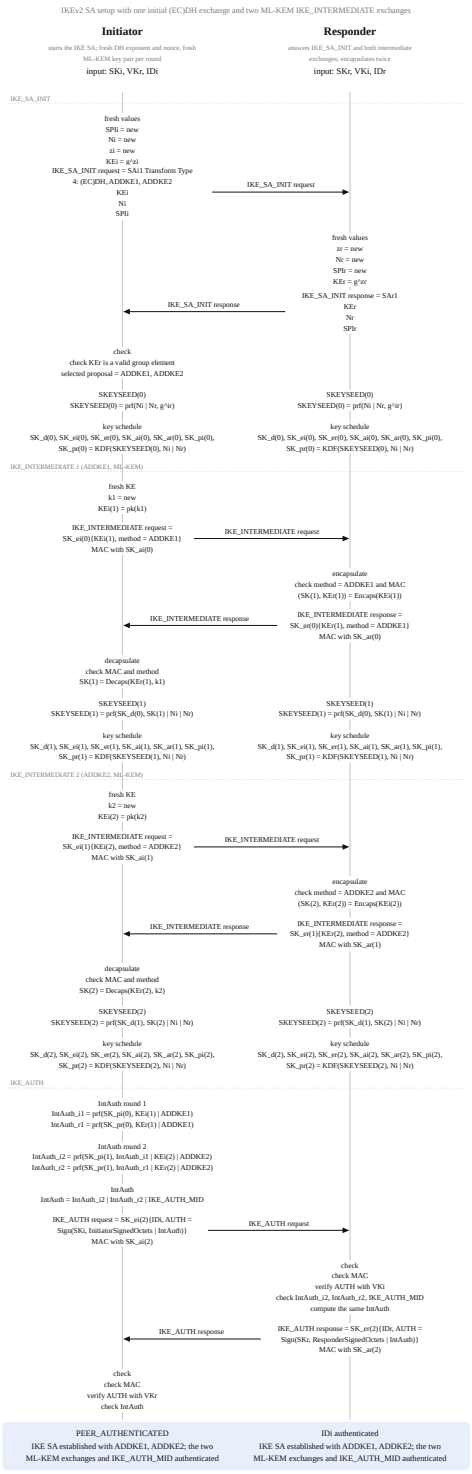


Figure 1. The protocol as specified.

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.

The modelled protocol gives the agreement, injectivity and transcript-binding guarantees it is meant to give: against an attacker on the network alone every accepted IKE_AUTH maps to a distinct peer run of the same session, and that run carried the negotiated additional key exchange methods, both key exchange payloads and the same IntAuth value. The prover established this for any number of sessions, and also established that the two IKE_INTERMEDIATE rounds never share one KEM secret. The key-secrecy and compromise questions got no answer: for the final IKE SA keys, for a leaked ML-KEM component, for a harvested classical secret, for one generation's keys, and for initiator identity confidentiality, the prover said it could not prove the property but found no trace either, so these are open rather than broken. No finding: nothing here is a weakness the documents do not already state. The results rest on a symbolic model with idealised KEM, prf, MAC and signatures, on (EC)DH with its one law only, and on a model that has no message ID counter, no window and no per-session state that changes.

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 theIKESArun.** Whenever the initiator (or responder) is presentIKE_AUTHAccept after completionIKESAThere's a real running of the opposite side, the character, the identity, the negotiation.ADDKEAlgorithm series, roundsKEi(n)/KEr(n) Content and finalitySKEYSEEDIt's all the same.	no compromise	Holds	medium	yes
A04	**no double acceptance of one peer run.** At each timeIKE_AUTHSuccessful acceptances correspond to the actual running of a different opposite end; One-to-one running cannot be accepted twice, MessageIDandIKE_AUTH_MIDIt must be in line with this operation.	no compromise	Holds	medium	yes
A08	**intermediate rounds bound to the transcript.** IKE_AUTHIt was accepted after success.ADDKEAlgorithm sequence, turn sequence andIntAuthThe transcribed records must be shared with both parties.IKE_SA_INITThe results of the consultations were consistent, and round nKEThe method field matches the n consultation algorithm.	no compromise	Holds	medium	yes
K01	**finalIKESAkeys stay secret.** FinalIKESAKey(s)SKEYSEED、SK_d、SK_ei/er、SK_ai/ar、SK_pi/pr) cannot be deduced by the attacker; Even if the assailant gets a follow-up.ML-KEMCovered shared secretsSK(n) The final key remains confidential as long as the remaining weights are not broken.	Individual follow-upML-KEMCover shared secretsSK(n) (inIKE_AUTHAfter completion of leak)	No answer	low	yes
K04	**key separation across use and generation.** Every roundIKE_INTERMEDIATEIt's a birth.SK_d(n)、SK_ai(n)/SK_ar(n)、SK_ei(n)/SK_er(n)、SK_pi(n)/SK_pr(n) Separation of use, direction and rotation; The same set of protection keys and authentication keys comes from the same generation.	no compromise	No answer	low	yes
C01	**forward secrecy after the session.** Yes.IKE_AUTHDisclose a long-term signature private key after completing and wiping out the session status (orPSKAnd the classics.EC)DHSharing secrets, güir, does not restore what the assailants have done.IKESAKey.	Long Sign Private Keys and ClassicsDHSharing Secrets (IKE_AUTHAfter completion and erasing)	No answer	low	yes

ID	Property	Compromise assumed	Result	Value	In the documents
C08	**one broken component does not break theSA.** When it's classic (EC)DHSize or a certain follow-upML-KEMWhen the weight is broken, as long as the rest is not broken, and eventually...IKESAThe key remains confidential.	ClassicDHSize or IndividualML-KEMSize (after birth)	No answer	low	yes
R01	**messageIDfreshness.** IKE_AUTHMessage on which success dependsIDandIKE_AUTH_MIDMust be the fresh value of this operation, and the evidence of the old running cannot be accepted.	no compromise	Holds	low	yes
R02	**one acceptance per message.** Same logicIKE_INTERMEDIATEorIKE_AUTHMessages are on a single basis.IKESAUp to once.	no compromise	Holds	medium	yes
R04	**intermediate rounds in order, none skipped.** IKE_INTERMEDIATEThe order of acceptance must match the Transform Type value order with the MessageIDIncremental rules, and every additional key exchange is completed before you start the next one.	no compromise	Holds	medium	yes
A08.11	**ADDKErounds bound to theAUTH.** IKE_AUTHIt was accepted after success.ADDKEAlgorithm sequence, turn sequence andIntAuthThe transcribed records must be shared with both parties.IKE_SA_INITThe results of the consultations were consistent, and round nKEThe method field matches the n consultation algorithm.	no compromise	Holds	medium	yes
K01.12	**final keys secret under oneKEMleak.** Even if the assailant gets a certain follow-upML-KEMCovered shared secretsSK(n) As long as the rest of the weight is not broken, eventuallyIKESAThe key remains confidential.	Individual follow-upML-KEMCover shared secretsSK(n) (inIKE_AUTHAfter completion of leak)	No answer	low	yes
A04.13	**no splicing of rounds or runs.** Each successful acceptance of the original IKE_AUTH corresponds to a different, complete, end-to-end reality; The aggressor cannot succeed in authentication by aggregating different sets of papers or skipping a given round.	no compromise	Holds	medium	yes
X01	**distinct KEM secrets per round.** The two IKE_INTERMEDIATE rounds of one IKE SA use fresh KEM key pairs and fresh encapsulation randomness, so their shared secrets SK(1) and SK(2) are different values; no duplicated algorithm selection can make one round's secret stand for the other's.	no compromise	Holds	medium	yes
X02	**initiator identity confidentiality.** A passive attacker on the network cannot tell which of two initiator identities the IKE_AUTH exchange carries, because IDi travels only inside the Encrypted payload.	no compromise	No answer	low	unclear

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: Intermediate order/calculation and finalityAUTHTie	Yes: both additional key exchange rounds and the IKE_AUTH message ID are bound into IntAuth, which both IKE_AUTH signatures cover, so an accepted run used the negotiated additional methods in the negotiated order with the key exchange payloads of those rounds, against an attacker on the network alone.	A08.11, A08, R04
Q2: Individual follow-upKEMSecret when leakedIKESAKey Confidentiality	This one the model does not answer: the attacker who holds one ML-KEM shared secret is not modelled, and the query could only be asked for the network attacker, where it came back inconclusive with no trace, so it must be settled by a bounded model with an explicit leak or by Tamarin+ Agent.	K01.12, K01, C08
Q3: Cross-cycle ciphering and missing rounds cannot produce successful authentication	No: acceptance is injective on the peer run and both rounds feed IntAuth, which both signatures cover, so an attacker on the network cannot splice messages of different rounds or runs, and cannot skip a round, and get an IKE_AUTH accepted.	A04.13, A04, A08

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:

- Symbolic model: no computational security and no post-quantum claim. The properties are properties of the idealised primitives, not of ML-KEM, the PRF, the MAC or the signature scheme.
- Diffie-Hellman is modelled with the single law g^xy = g^yx; no algebraic attack beyond that law is possible in the model.
- KEM encapsulation is a private function of the shared secret, the method and the public key with decapsulation as its inverse; implicit rejection, malformed ciphertexts and the ML-KEM encoding are not modelled.
- No SA payload negotiation, no proposal matching and no downgrade check: the two additional key exchange methods are fixed constants, so negotiation among candidates is not examined.
- No Message ID counter, no window of overlapping requests and no retransmission timers: freshness of IKE_AUTH is answered through injectivity over fresh SPIs and nonces, not through IKEv2 message ID state.
- No key update or rekeying of the IKE SA and no CHILD_SA; CREATE_CHILD_SA is outside the modelled scope.
- The compromise questions (one leaked ML-KEM secret, a harvested classical DH secret, one generation's keys, one encryption key, a leaked long-term signature key) could not be modelled: a leak inside the replicated responder made ProVerif diverge, so those queries were asked with the network attacker only.
- One initiator and one responder with a fixed pair of peer identities; cross-session linking and many-party settings are not examined beyond replication of the roles.
- Initiator identity confidentiality is asked as non-interference over two candidate identities, and ProVerif neither proved it nor produced a trace.

What the host records:

- No answer for: K01, K04, C01, C08, K01.12, X02.
- The prover gave no answer (Appendix C): K01: skd2, K04: skai1, K04: skd2, C01: sks2, C08: skd2, K01.12: skd2, X02: id_conf. The property was asked and the prover neither proved it nor found an attack, so the property is shown without an answer.

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: agreement on the IKE SA run

Whenever the initiating (or responding) isIKE_AUTHAccept after completionIKESAThere's a real running of the opposite side, the character, the identity, the negotiation.ADDKEAlgorithm series, roundsKEi(n)/KEr(n) Content and finalitySKEYSEEDIt's all the same.

Both directions hold: whenever the responder accepts the IKE_AUTH request an initiator run sent that same session, and whenever the initiator accepts the IKE_AUTH response a responder run sent it. The accepted IntAuth value chains both intermediate exchanges and the IKE_AUTH message ID, so the peer ran the same session and not merely one with the same identifiers. An attacker on the network alone cannot make either side accept a run the peer did not take part in.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: This is the central authentication property of the protocol as specified; the proof is mechanical but rests on the idealised signature and the IntAuth abstraction.
- Stated in the documents: yes. Where the documents say it: rfc9242:283, s0481, s0482, s0486
- Rests on: Signatures are idealised: verification succeeds only on the exact signed octets of the matching key.; IntAuth is abstracted from RFC 9242 as a prf chain over both rounds' key exchange payloads, the negotiated methods and the message ID; the optional A and P chunks are collapsed into that chain.; Encryption, MAC and prf are ideal primitives, so the model does not show that the real payload encoding is unambiguous.; Both roles are replicated and the peer identities are fixed, so agreement is shown for any number of sessions of this pair.

A04: no double acceptance of one peer run

Every time it's on its own.IKE_AUTHSuccessful acceptances correspond to the actual running of a different opposite end; One-to-one running cannot be accepted twice, MessageIDandIKE_AUTH_MIDIt must be in line with this operation.

Acceptance is injective on both sides: each accepted IKE_AUTH corresponds to a distinct run of the peer, so one peer run cannot be accepted twice. The message ID of IKE_AUTH is carried inside IntAuth and therefore inside both signatures, so a message of one run cannot be presented as belonging to another. An attacker on the network cannot replay an old IKE_AUTH into a second acceptance.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Replay protection of IKE_AUTH is a stated requirement; the proof is mechanical and its relation to the real message ID state is through the model's fresh values only.
- Stated in the documents: yes. Where the documents say it: rfc9242:205, s0487
- Rests on: Injectivity is proved over the accepting events; the peer run is identified by its fresh SPIi, SPIr and nonce.; The model has no message ID counter and no window, so double acceptance is excluded through fresh session values rather than through IKEv2 message ID state.; Signatures, prf and the IntAuth abstraction are idealised as in the agreement property.
- Open: Whether the real message ID rule rejects every replayed IKE_AUTH, including replays inside the acceptance window.

- Next step: A model with the message ID counter unrolled to a few steps would settle the window case.

A08: intermediate rounds bound to the transcript

IKE_AUTHIt was accepted after success.ADDKEAlgorithm sequence, turn sequence andIntAuthThe transcribed records must be shared with both parties.IKE_SA_INITThe results of the consultations were consistent, and round nKEThe method field matches the n consultation algorithm.

The IntAuth value that the responder accepts is a prf chain over the key exchange payload and ciphertext of round one, the key exchange payload and ciphertext of round two, and the two negotiated methods, so the accepted sequence, order and contents are the negotiated ones. Both IKE_AUTH signatures carry that value, and each role compares the method field of round n with the n-th negotiated method before use. An attacker on the network cannot substitute a method, reorder the rounds or swap a payload without the signatures failing.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Transcript binding of the additional exchanges is the point of IntAuth; the result is a mechanical proof under the model's idealisations.
- Stated in the documents: yes. Where the documents say it: rfc9242:286, s0483, s0488
- Rests on: Round-to-method binding is enforced structurally by both roles comparing the method field with the negotiated method, alg1 for round one and alg2 for round two; the model does not build a term for a different method.; The two additional methods are fixed constants, so no negotiation among several candidates is modelled.; Signatures and prf are idealised and the optional A and P chunks of IntAuth are collapsed.

A08.11: ADDKE rounds bound to the AUTH

IKE_AUTHIt was accepted after success.ADDKEAlgorithm sequence, turn sequence andIntAuthThe transcribed records must be shared with both parties.IKE_SA_INITThe results of the consultations were consistent, and round nKEThe method field matches the n consultation algorithm.

To the first question: yes, the accepted additional key exchange sequence and round order are bound to the AUTH exchange and to what was negotiated before. Both rounds' contents and the two negotiated methods enter IntAuth, IntAuth enters the signed octets of both parties, and the IKE_AUTH message ID is carried in the same value. An attacker on the network cannot get a run accepted whose intermediate contents differ from the ones negotiated.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Answers the user's question with a proved guarantee, under the same caveats as the transcript-binding property.
- Stated in the documents: yes. Where the documents say it: rfc9242:286, s0483, s0488
- Rests on: The same idealisations as the transcript-binding property, which answers the same question: idealised signatures and prf, IntAuth abstracted from RFC 9242, fixed additional methods.; The per-round method check is an equality test on the method field, not a full transform comparison.

A04.13: no splicing of rounds or runs

Each successful acceptance of the home side IKE_AUTH corresponds to a different, full-end real-life operation; The aggressor cannot succeed in authentication by aggregating different sets of papers or skipping a given round.

To the third question: no splicing and no skipped round. Both intermediate rounds feed IntAuth, IntAuth feeds both IKE_AUTH signatures, and acceptance is injective, so an accepted IKE_AUTH always corresponds to one complete peer run. An attacker on the network cannot combine messages of different rounds or different runs into an

acceptance.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Answers the user's question with a proved guarantee; the order of rounds is partly enforced by the structure of the roles rather than by a query.
- Stated in the documents: yes. Where the documents say it: rfc9242:286, s0487
- Rests on: Injectivity over the fresh session values, as for the replay property.; The rounds are sequential statements in each role, so the model cannot express an out-of-order round; the order argument is not a separate query.; Idealised signatures, prf, MAC and encryption.

K01: final IKE SA keys stay secret

Eventually.IKESAKey(s)SKEYSEED、SK_d、SK_ei/er、SK_ai/ar、SK_pi/pr) cannot be deduced by the attacker; Even if the assailant gets a follow-up.ML-KEMCovered shared secretsSK(n) The final key remains confidential as long as the remaining weights are not broken.

The final IKE SA keys are not shown to be secret and not shown to be derivable either. For an attacker on the network alone the prover could not prove the property and found no trace of an attack; for an attacker holding one ML-KEM shared secret the model could not even be asked, because a leak of a session component inside the replicated responder made the search diverge. What the documents argue stays an argument: SKEYSEED is uncompromised as long as at least one exchanged secret is unknown.

- Result: No answer; answers what is asked: undetermined; not usable
- Value low: Neither proof nor attack, and the decisive case of the target, one leaked KEM component, was not modelled at all.
- Stated in the documents: yes. Where the documents say it: main:910, s0106, s0108
- Rests on: The model gives the attacker no session component; the queries were asked with the network attacker only and the answers speak about that attacker.; Key derivation uses idealised prf and kdf; the argument that a fresh component protects the final keys is computational.; The inconclusive answer is the prover's over-approximation combining steps of different sessions; there is no trace behind it.
- Open: Whether one leaked ML-KEM shared secret leaves the final IKE SA keys secret.; Whether the network attacker alone can derive any final key.
- Next step: A bounded model with an explicit leak event for one component, or the same question in Tamarin+ Agent with a revealed component, would settle it.

K01.12: final keys secret under one KEM leak

Even if the assailant gets a follow-up.ML-KEMCovered shared secretsSK(n) As long as the rest of the weight is not broken, eventuallyIKESAThe key remains confidential.

To the second question there is no prover answer: the target is set for an attacker who holds one ML-KEM shared secret, that attacker is not in the model, and nothing recorded stands for it. The design argument, that the chain folds a fresh secret into each generation so one broken component does not break the others, is what the documents state and what a bounded or algebraic model would have to confirm. The same query for the network attacker alone was inconclusive.

- Result: No answer; answers what is asked: undetermined; not usable
- Value low: The question is not answered by the model; only the shape of the intended argument is reproduced.
- Stated in the documents: yes. Where the documents say it: main:910, s0106, s0108

- Rests on: The attacker of the target is not modelled: no leak of a session component exists in the model.; The inconclusive answer to the network-only version is an over-approximation without a trace.; Idealised prf and kdf.
- Open: The whole question, for the attacker the target names.
- Next step: Model the leak, bounded or in Tamarin+ Agent, and ask for the final keys again.

K04: key separation across use and generation

Every round.IKE_INTERMEDIATEIt's a birth.SK_d(n) 、 SK_ai(n)/SK_ar(n) 、 SK_ei(n)/SK_er(n) 、 SK_pi(n)/SK_pr(n) Separation of use, direction and rotation; The same set of protection keys and authentication keys comes from the same generation.

Key separation is built into the model by giving every use and every generation its own extractor of its own key block, so no key is syntactically another. Whether knowing one generation's keys or one encryption key reveals another is not answered: the model does not hand those values to the attacker and both queries came back inconclusive without a trace. The separation the documents require is therefore reproduced structurally but not tested against an attacker.

- Result: No answer; answers what is asked: undetermined; not usable
- Value low: No prover answer for either claim; separation is an assumption of the model rather than a result.
- Stated in the documents: yes. Where the documents say it: main:910, s0107
- Rests on: Distinct extractors and per-generation blocks give separation by construction, so the model cannot express a real derivation weakness.; No generation's keys are leaked in the model, so both queries were run for the network attacker only.; Idealised prf and kdf.
- Open: Whether an attacker holding generation-1 keys can derive generation 2.; Whether the encryption key of a generation reveals its integrity key.
- Next step: Leak one generation's keys to the attacker in a bounded model and ask both queries again, or use Tamarin+ Agent, which tolerates revealed values.

C01: forward secrecy after the session

Yes.IKE_AUTHDisclose a long-term signature private key after completing and wiping out the session status (orPSKAnd the classics.EC)DHSharing secrets, güir, does not restore what the assailants have done.IKESAKey.

Forward secrecy of a completed IKE SA can be reported neither as shown nor as broken. The intended design, that a later quantum computer learning the classical secret still cannot decrypt the archived session, is what the documents claim, but the query with the classical secret given to the attacker was not answered. For the network attacker alone the prover was inconclusive and produced no trace.

- Result: No answer; answers what is asked: undetermined; not usable
- Value low: The decisive attacker, one holding the harvested classical secret, is not in the model; nothing is established for or against the property.
- Stated in the documents: yes. Where the documents say it: main:925, s0106
- Rests on: The classical secret is not given to the attacker, so the query speaks about an attacker who does not hold it.; At least one ML-KEM component would have to stay hidden and the prf be quantum resistant, which the symbolic model only assumes.; Idealised prf and kdf.
- Open: Harvest-and-decrypt resistance for the modelled session.
- Next step: Ask the question with the classical secret revealed, in a bounded model or in Tamarin+ Agent.

C08: one broken component does not break the SA

When it's classic (EC)DHSize or a certain follow-upML-KEMWhen the weight is broken, as long as the rest is not broken, and eventually...IKESAThe key remains confidential.

The claim that one broken component does not break the SA is not answered: neither the classical-DH case nor the one-KEM-component case is in the model as an attacker, and both queries were inconclusive for the network attacker without a trace. What the model does show structurally is that each generation folds a fresh secret into the chain, the mechanism the documents rely on. Whether that mechanism holds against an attacker who really holds one component is open here.

- Result: No answer; answers what is asked: undetermined; not usable
- Value low: No prover answer; the compound claim needs revealed components.
- Stated in the documents: yes. Where the documents say it: main:910, s0104, s0106, s0107
- Rests on: No component is revealed; both queries were run for the network attacker only.; The two additional methods are fixed constants, so a case in which both rounds use the same algorithm, and therefore the same public key material, is not modelled.; Idealised prf and kdf.
- Open: Both readings of the claim: the classical component broken, and one KEM component broken.
- Next step: A bounded leak model, or Tamarin+ Agent with revealed values, as for the final-keys question.

R01: message ID freshness

IKE_AUTHMessage on which success dependsIDandIKE_AUTH_MIDMust be the fresh value of this operation, and the evidence of the old running cannot be accepted.

Freshness of IKE_AUTH is shown only in the form the model can carry: acceptance is injective over the session's fresh SPIi, SPIr and nonce, and the message ID of IKE_AUTH is inside the signed IntAuth value. The check of the message ID counter against the window is not in the model, so nothing is said about an old message that would still fit the window.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value low: The property as stated is about the message ID rule, which the model does not implement; what is proved is a weaker freshness property.
- Stated in the documents: yes. Where the documents say it: rfc9242:205, s0132, s0487
- Rests on: No message ID counter, no window and no retransmission logic: freshness rests on fresh session values and on injectivity.; Idealised signatures and prf; IntAuth abstracted from RFC 9242.
- Open: Whether the message ID rule rejects a replayed IKE_AUTH that fits inside the window.
- Next step: Unroll the message ID counter in a bounded model and replay an old request inside the window.

R02: one acceptance per message

Same logic.IKE_INTERMEDIATEorIKE_AUTHMessages are on a single basis.IKESAUp to once.

A replayed IKE_AUTH request is not accepted twice: the accepted signature covers this run's fresh SPIr and nonce and acceptance is injective on the peer run. The same holds for the intermediate messages, because each is protected with the keys of a distinct generation and carries this run's identifiers. The model has no message ID state, so the answer is about replay into a new or parallel session, not about replay inside one window.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value medium: A meaningful replay result under the model's assumptions, weaker than the full message ID rule.

- Stated in the documents: yes. Where the documents say it: rfc9242:205, s0132
- Rests on: Fresh session values and injectivity instead of message ID state.; Idealised encryption, MAC, prf and signatures.
- Open: Replay inside the acceptance window of one IKE SA.
- Next step: The same bounded message ID model as for message ID freshness.

R04: intermediate rounds in order, none skipped

The order of acceptance of IKE_INTERMEDIATE must conform to the Transform Type value order and the Message ID incremental rule, and the next one can only start after each additional key exchange has been completed.

IKE_AUTH is accepted only after both intermediate exchanges have completed: IntAuth chains the second round onto the first, both feed the signed octets, and the roles run the rounds in sequence, so a skipped round or a different order cannot produce an accepted run. The order relative to the negotiated methods follows from the per-round method check. No explicit message ID ordering is modelled, so the ordering of reordered messages within one round is not examined.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The stated binding of IKE_AUTH to the completed rounds is proved, with the caveat that ordering inside a round is structural.
- Stated in the documents: yes. Where the documents say it: rfc9242:286, s0022, s0468, s0469
- Rests on: Sequential role processes enforce the order; the model cannot express an out-of-order round.; The per-round method check is an equality test and both additional methods are fixed constants.; IntAuth abstracted from RFC 9242; idealised signatures and prf.

X01: distinct KEM secrets per round

The two IKE_INTERMEDIATE rounds of one IKE SA use fresh KEM key pairs and fresh encapsulation randomness, so their shared secrets SK(1) and SK(2) are different values; no duplicated algorithm selection can make one round's secret stand for the other's.

The two intermediate exchanges never share one ML-KEM secret: the initiator makes a fresh key pair for each round and the responder encapsulates afresh, so the two shared secrets are different values in every run. This blocks the failure the design warns about, where one shared secret would be counted twice as if it were two independent components. It holds for any number of sessions and any interleaving of the two rounds.

- Result: Holds; answers what is asked: yes; usable
- Value medium: A structural property of the design, proved mechanically: the distinctness of the two secrets the key chain and the hybrid argument depend on.
- Stated in the documents: yes. Where the documents say it: main:252, s0025, s0043
- Rests on: Encapsulation is a private function of the shared secret, the method and the public key; freshness comes from the model's new names.; Whether real ML-KEM encapsulation of one public key can ever return the same secret twice is a computational question the symbolic model assumes away.

X02: initiator identity confidentiality

A passive attacker on the network cannot tell which of two initiator identities the IKE_AUTH exchange carries, because IDi travels only inside the Encrypted payload.

The model shows neither that the initiator's identity is hidden from a passive attacker nor that it leaks. In the design IDi and the value derived from it travel only inside the encrypted IKE_AUTH payload, which is what the confidentiality rests on. The prover could not prove the non-interference; its derivation needed the classical secret and both KEM secrets at once and produced no trace, so this is not an attack.

- Result: No answer; answers what is asked: undetermined; not usable
- Value low: No answer either way; the strongest the model supports is that the identity is inside an encrypted payload.
- Stated in the documents: unclear. Source text it rests on: s0471, s0482
- Rests on: Encryption is idealised, so the answer is only about the symbolic hiding of IDi, not about traffic analysis such as sizes and patterns.; The prover never says a non-interference is false; the inconclusive answer means neither proof nor attack.; Only two candidate identities and one fixed responder identity are modelled.
- Open: Whether a passive attacker can tell two initiator identities apart.
- Next step: A model whose key schedule makes the IKE_AUTH generation key unreachable to the attacker, or Tamarin+ Agent, would settle the observation equivalence.

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		auth_resp	checked		2 proved
A04	proved		inj_resp	checked		2 proved
A08	proved		auth_resp	checked		1 proved
K01	none					2 undecided
K04	none					2 undecided
C01	none					1 undecided
C08	none					2 undecided
R01	proved		inj_resp	checked		1 proved
R02	proved		inj_resp	checked		1 proved
R04	proved		auth_resp	checked		1 proved
A08.11	proved		auth_resp	checked		1 proved
K01.12	none					1 undecided
A04.13	proved		inj_resp	checked		1 proved
X01	proved		same_kem	checked		1 proved
X02	none					1 undecided

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	Initiator side: whenever the initiator accepts the IKE_AUTH response, some responder run sent the same (SPIi, SPIr) and the same IntAuth value, which chains both IKE_INTERMEDIATE rounds and the IKE_AUTH message ID. Attacker has the network only.	`network`	proved	auth_init	checked
A03	Responder side: whenever the responder accepts the IKE_AUTH request, some initiator run sent exactly the same (SPIi, SPIr) and the same IntAuth value; the IntAuth value is a prf chain over both IKE_INTERMEDIATE rounds (pk(k1), ct1, alg1, pk(k2), ct2, alg2) and IKE_AUTH_MID, so the peer ran the same session. Attacker has the network only.	`network`	proved	auth_resp	checked
A04	Responder side: each accepted IKE_AUTH request corresponds to a distinct initiator run, so one peer run cannot be accepted twice; the IKE_AUTH_MID value carried inside IntAuth belongs to that run. Attacker has the network only.	`network`	proved	inj_resp	checked
A04	Initiator side: each accepted IKE_AUTH response corresponds to a distinct responder run, so the same responder run cannot be accepted twice. Attacker has the network only.	`network`	proved	inj_init	checked

target	claim	attacker	answer	query	control
A08	The accepted IntAuth is a prf chain over both rounds' contents (pk(k1), ct1, alg1, pk(k2), ct2, alg2) and the IKE_AUTH message ID, so the accepted ADDKE sequence, round order and KE payloads are the negotiated ones; the method field of round n is compared with the n-th negotiated algorithm. Attacker has the network only.	`network`	proved	auth_resp	checked
K01	The final IKE SA key material (SK_d and the SK_* of the last generation) cannot be derived by an attacker with the network alone. ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker has the network only.	`network`	undecided	skd2	
K01	With one IKE_INTERMEDIATE shared secret SK(n) leaked after IKE_AUTH, the final SK_d still stays secret. The model does not hand that component to the attacker (no leak process could be added without losing termination), so the query is asked with the network attacker only; ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds one ML-KEM shared secret of the session.	`kem_component`	undecided	skd2	
K04	Knowing only the generation-1 encryption key SK_ei(1) does not reveal the generation-1 integrity key SK_ai(1): keys of one generation are separated by distinct extractors of the same block. The model does not hand SK_ei(1) to the attacker; ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds SK_ei(1).	`enc1_key`	undecided	skai1	
K04	No generation-2 key follows from all generation-1 keys alone: SK_d(2) is derived through SKEYSEED(2) from the fresh SK(2) and the previous SK_d(1). The model does not hand the generation-1 keys to the attacker; ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds all generation-1 keys.	`gen1_keys`	undecided	skd2	
C01	After IKE_AUTH completes, learning the classical (EC)DH shared secret g^ir (harvested from the wire) does not give the completed session's SK_s. The model does not hand g^ir to the attacker; ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds g^ir of the session.	`classical_dh`	undecided	sks2	
C08	With the classical (EC)DH component broken, the final SK_d stays secret because the ML-KEM components still protect it. The model does not hand g^ir to the attacker; ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds g^ir of the session.	`classical_dh`	undecided	skd2	
C08	With one ML-KEM shared secret SK(n) broken, the final SK_d stays secret because the other components still protect it. The model does not hand SK(n) to the attacker; ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds one ML-KEM shared secret of the session.	`kem_component`	undecided	skd2	
R01	An accepted IKE_AUTH that carries a Message-ID-like value is fresh: acceptance is injective on the peer run, so an old run's IKE_AUTH evidence is not accepted again. The model has no Message-ID counter or window; the claim is answered by injectivity over the session's fresh SPIi/SPIr/Nr and the IKE_AUTH_MID inside IntAuth. Attacker has the network only.	`network`	proved	inj_resp	checked
R02	Responder side: a replayed or spliced IKE_AUTH message cannot be accepted by any responder instance, because the accepted signature covers SPIi, the fresh SPIr and Nr of this run, and acceptance events are injective. Message-ID/window state is not modelled; the freshness rests on the session's fresh SPIr/nr. Attacker has the network only.	`network`	proved	inj_resp	checked

target	claim	attacker	answer	query	control
R04	IKE_AUTH is accepted only after both IKE_INTERMEDIATE rounds have completed in order: IntAuth chains round 1 into round 2, both feed the signed octets, and the role processes issue the rounds as sequential statements, so no round can be skipped and none is accepted out of order. Attacker has the network only.	`network`	proved	auth_resp	checked
A08.11	(user question Q1) The accepted IntAuth transcript binds the two IKE_INTERMEDIATE rounds to the IKE_SA_INIT negotiation: both rounds' KE payloads, the negotiated ADDKE algorithms in order, and the IKE_AUTH message ID all feed IntAuth, which both IKE_AUTH signatures cover. Attacker has the network only.	`network`	proved	auth_resp	checked
K01.12	(user question Q2) With one IKE_INTERMEDIATE shared secret SK(n) leaked after IKE_AUTH, the final IKE SA keys stay secret. The model gives the attacker the network only (a leak process in the replicated responder made ProVerif diverge); ProVerif answered cannot be proved (inconclusive): it found no trace, so neither a proof nor an attack. Attacker holds one ML-KEM shared secret of the session.	`kem_component`	undecided	skd2	
A04.13	(user question Q3) One accepted IKE_AUTH is a distinct and complete peer run: no splicing of rounds or of runs and no skipped round, because both rounds feed IntAuth, IntAuth feeds both signatures, and the accepting events are injective. Attacker has the network only.	`network`	proved	inj_resp	checked
X01	The two IKE_INTERMEDIATE rounds of one IKE SA use fresh KEM key pairs and fresh encapsulation randomness, so no event ever pairs equal values as SK(1) and SK(2): one round's secret can never stand for the other's. Attacker has the network only.	`network`	proved	same_kem	checked
X02	A passive attacker cannot tell which of two initiator identities the IKE_AUTH exchange carries; IDi and prf(SK_pi, IDi) travel only inside the encrypted payload, and a reachable run exists in which the IKE_AUTH request is sent. ProVerif answered cannot be proved (inconclusive) for the non-interference: its derivation needs the DH secret and both KEM secrets at once and it produced no trace, so this is not an attack. Attacker has the network only.	`network`	undecided	id_conf	

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-20261115T080535, 2026-11-15. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 61, prover 607 s, wall 27 min, cost 0.331 USD.

Context: the conversation was restarted from the analyst's notes 2 time(s). The trajectory file keeps everything.

ProVerif: 54 proofs run, 35 decided, 19 without an answer, 474 s, 12 answers recorded.

What was analysed

Protocol	IKEv2 multiple intermediate key exchanges
Version	rfc9370
Analysed	InitialDHTwo different ones.ML-KEMSealed bounds.IKE_INTERMEDIATE, re-signature authentication; ReservationsADDKEConsultation, order number, key update and intermediate accreditation, never taking the same two long-termKEMSecret when independent; ExcludeCHILD_SA。
Not analysed	Symbolic models do not support the calculation of algorithms as safe or as resistant to quantum security; The trajectory of the attack is established only under clear assumptions and is not directly equivalent to a normative loophole. Each nature must be subject to accessibility/proportional controls.
Source text	500 numbered passages (177184 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	Multiple Key Exchanges
main	2.1	Design Overview
main	2.2	Protocol Details
main	2.2.1	IKE_SA_INIT Round: Negotiation
main	2.2.2	IKE_INTERMEDIATE Round: Additional Key Exchanges
main	2.2.3	IKE_AUTH Exchange
main	2.2.4	CREATE_CHILD_SA Exchange
main	2.2.5	Interaction with IKEv2 Extensions
main	2.2.5.1	Interaction with Childless IKE SA
main	3	IANA Considerations
main	4	Security Considerations
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

Document	Section	Title
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
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

Document	Section	Title
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

The promise of the program as the source step listed it in the documents:DHTwo different ones.ML-KEMSealed bounds.IKE_INTERMEDIATE, re-signature authentication; ReservationsADDKEConsultation, order number, key update and intermediate accreditation, never taking the same two long-termKEMSecret when independent; ExcludeCHILD_SA。

The properties against the documents

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

ID	Name	About	What the documents say	Where
A03	agreement on the IKE SA run	And responder: each side, when it comes to theIKESAafterIKE_AUTH, is assured that the peer ran the same session with the sameIKE_INTERMEDIATEcontents andIKE_AUTHmessageID.	The peers authenticate once, in IKE_AUTH, and AUTH covers the IKE_SA_INIT messages plus the IntAuth chain over every IKE_INTERMEDIATE message (s0481, s0482, s0486, s0488); until then the intermediate data is unauthenticated (rfc9242 5).	s0022, s0041, s0046, s0048, s0481, s0482, s0486, s0488
A04	no double acceptance of one peer run	One person runs at most once; IKE_AUTH_MID and the Message ID belong to this run.	Message IDs follow the RFC 7296 2.2 rule and IKE_AUTH's first Message ID is one more than the last intermediate one (s0469); IKE_AUTH_MID is part of IntAuth (s0487).	s0132, s0469, s0487
A08	intermediate rounds bound to the transcript	Sponsor and responder: the approvedADDKEsequence, round order andKEi(n)/KEr(n) contents are the negotiated ones, round n carrying the n-th negotiated method.	Rounds follow the order of their Transform Types and each completes before the next (s0022); the Key Exchange Method field MUST match the n-th negotiated exchange (s0041); IntAuth chains over each message (s0483, s0488).	s0022, s0025, s0041, s0483, s0487, s0488
K01	final IKE SA keys stay secret	and responder holding one ML-KEM shared secret of the session.	SKEYSEED(n) = prf(SK_d(n-1), SK(n) | Ni | Nr) and the seven keys are prf+ of SKEYSEED(n) (s0043, s0044, s0045); a compromised PQC algorithm must not compromise the final secret (s0104).	s0043, s0044, s0045, s0104, s0106
K04	key separation across use and generation	Starter and responder: the protection, integrity andAUTHkeys of a generation are separate, and nogeneration-2key follows from generation1alone.	The seven keys are distinct prf+ outputs distinguished by use (s0045); a round uses the most recently calculated keys and its protection and authentication keys are always the same generation (s0476, s0498).	s0045, s0046, s0476, s0498
C01	forward secrecy after the session	Sponsor and responder: a CompletedIKESA's keys stay secret when the classical (EC)DHsecret is harvested and although theAUTHsignatures are public.	A quantum attacker may learn SK(x) from the classical exchange and the post-quantum components keep SKEYSEED uncompromised (s0106, s0108); DH exponents are erased after use.	s0106, s0108, s0290

ID	Name	About	What the documents say	Where
C08	one broken component does not break the SA	And responder: with the final decision or one ML-KEM host broken the final keys stay secure.	Multiple key exchanges form the shared key so that a compromise of one does not compromise the result (s0104, s0106, s0107).	s0104, s0106, s0107
R01	message ID freshness	Response: the messageIDof an acceptedIKE_INTERMEDIATE/IKE_AUTHmessage is the next unused value.	Message IDs are 1 for the first IKE_INTERMEDIATE and increase, and implementations MUST verify it (s0469); IKE_AUTH_MID is that of the first IKE_AUTH round (s0487).	s0132, s0469, s0487
R02	one acceptance per message	Response: the sameIKE_INTERMEDIATEorIKE_AUTHmessage is accepted at most once within oneIKESA.	Replays must be recovered from (s0114) and the Message ID rule forbids reusing an ID (s0469).	s0114, s0132, s0469
R04	intermediate rounds in order, none skipped	Responder: round i is listed before round i+1, each round completes before the next, andIKE_AUTHis not accepted before the last round.	Rounds follow their Transform Type order and each MUST be fully completed before the next (s0022); they are sequential and complete before IKE_AUTH (s0468).	s0022, s0468, s0469
A08.11	ADDKE rounds bound to the AUTH	Sponsor and Respondent (user question)Q1): the acceptedADDKEsequence, round order andIntAuthtranscript are the negotiated ones.	Same as A08; the question asks whether the intermediate rounds are bound to the final AUTH.	s0022, s0041, s0483, s0488
K01.12	final keys secret under one KEM leak	Sponsor and Respondent (user question)Q2): with oneKEMsecret leaked the finalIKESAkeys stay secret.	Same as K01 for the component-compromise case (s0104, s0106).	s0043, s0045, s0106
A04.13	no splicing of rounds or runs	Starter and responder (user question Q3): rounds or runs cannot be spliced and a round cannot be sskiped.	IntAuth is a prf chain over the rounds in order plus IKE_AUTH_MID (s0483, s0488) and AUTH signs it (s0482).	s0022, s0468, s0482, s0483, s0488
X01	distinct KEM secrets per round	Launcher and responder: the valuesSK(1) andSK(2) that enterSKEYSEED(1) andSKEYSEED(2).	A fresh ephemeral key pair and ciphertext are REQUIRED for each ML-KEM exchange and responders MUST NOT reuse randomness (draft-ietf-ipsecme-ikev2-mlkem 3, lines 335-338); the responder's ADDKE choice MUST NOT contain duplicated algorithms (s0025).	draft-ietf-ipsecme-ikev2-mlkem:335, s0025, s0043
X02	initiator identity confidentiality	Launched by:IDiin theIKE_AUTHrequest, and prf(SK_pi,IDi) in the signed octets.	IKEv2 minimizes disclosure to unauthenticated peers, but the initiator identifies itself first and the responder, or someone impersonating it, can probe the initiator's identity (rfc7296 5, lines 6478-6488); IDi travels inside the Encrypted payload.	rfc7296:6478, s0471, s0482

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

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: 13. Not taken: 41. 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	draft-ietf-ipsecme-ikev2-mlkem:320-321	No goal stated	It only says that the security considerations of RFC 9242 and RFC 9370 apply; their paragraphs are listed separately.
P02	draft-ietf-ipsecme-ikev2-mlkem:323-338	X01, K01, C08	
P03 to P04	draft-ietf-ipsecme-ikev2-mlkem:340-346	X01	
P05	draft-ietf-ipsecme-ikev2-mlkem:350-350	No goal stated	Page footer and running header of the draft.
P06	draft-ietf-ipsecme-ikev2-mlkem:353-358	No goal stated	A pointer to another document; it states no goal for this protocol.
P07	draft-ietf-ipsecme-ikev2-mlkem:360-361	No goal stated	A pointer to implementation guidelines.
P08	draft-ietf-ipsecme-ikev2-mlkem:363-367	K01, C08, `kem_component`	
P09	draft-ietf-ipsecme-ikev2-mlkem:369-383	A08, C08, `classical_dh`	
P10	draft-ietf-ipsecme-ikev2-mlkem:385-395	No goal stated	Advice to operators about which proposals to disable.
P11 to P14	draft-ietf-ipsecme-ikev2-mlkem:397-429	Outside the scope	Childless IKE SAs, CREATE_CHILD_SA rekeying and IKE_FOLLOWUP_KE are excluded from this task.
P15	main:883-891	K01, C01, C08, `classical_dh`, `kem_component`	
P16	main:893-908	The prover cannot be asked	Grover on Transform Types 1-3 is the strength of a primitive.
P17	main:910-916	C01, C08, `classical_dh`	
P18	main:918-923	No goal stated	Order of the additional key exchanges does not matter for security; advice to implementers only.
P19	main:925-932	C01, `classical_dh`	
P20	main:934-942	The prover cannot be asked	That the authentication step stays classical is the strength of a primitive; the model signs and verifies perfectly, and limits records it.
P21	main:944-954	No goal stated	Discussion of why post-quantum authenticity is less urgent.
P22	main:956-961	The prover cannot be asked	Increased DoS exposure is a resource question.
P23	main:963-980	Outside the scope	Childless SA and CREATE_CHILD_SA/IKE_FOLLOWUP_KE are excluded.
P24	rfc7296:6478-6488	X02, A03	
P25	rfc7296:6490-6494	Outside the scope	EAP authentication is not part of the modelled run.
P26	rfc7296:6496-6500	Outside the scope	Repeated CREATE_CHILD_SA rekeying without new DH is excluded.
P27 to P29	rfc7296:6502-6527	The prover cannot be asked	The strength of Diffie-Hellman groups is the strength of a primitive.
P30	rfc7296:6529-6530	C01	

Paragraphs	Where	Taken up as	Why not
P31	rfc7296:6532-6538	The prover cannot be asked	DoS by unauthenticated peers is a resource question.
P32	rfc7296:6540-6542	The prover cannot be asked	The PRF output size is a primitive parameter.
P33	rfc7296:6544-6549	X01	
P34	rfc7296:6551-6556	No goal stated	Design rationale pointers plus a rule forbidding NONE integrity and ENCR_NULL encryption.
P35	rfc7296:6558-6563	Outside the scope	Pre-shared keys and low-entropy secrets are the alternative authentication mode; the modelled run uses signatures.
P36 to P37	rfc7296:6565-6580	Outside the scope	NAT detection is not part of the modelled run.
P38 to P39	rfc7296:6582-6612	Outside the scope	EAP authentication is not part of the modelled run.
P40	rfc7296:6614-6619	The prover cannot be asked	IP reassembly exhaustion is a resource question.
P41 to P42	rfc7296:6621-6633	Outside the scope	Admission control and the PAD concern Child SA authorization, excluded here.
P43 to P50	rfc7296:6637-6696	Outside the scope	Traffic selector authorization concerns Child SAs and the PAD, excluded here.
P51	rfc9242:463-470	A03, A08, K01	
P52	rfc9242:472-477	No goal stated	IKE fragmentation; the model treats messages as whole.
P53	rfc9242:479-499	The prover cannot be asked	Responder resource spending before authentication is a DoS question.
P54	rfc9242:501-514	A08, C08, `classical_dh`	

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: 13. Left out: 19. 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	Load and End-to-End Confidentiality	Outside the scope	Payload confidentiality belongs to Child SAs, excluded here.
K03	Indistinguishability of confidentiality/secretity	K01	
D01	Information integrity and authenticity of data sources	A03, A08	
A01	(Aliveness)	A03	
A02	Weak Consistency Certification	A03	
A05	Secret Key Authentication	K01, A03	
A06	Visible Key Confirmation	Not applicable	IKE_AUTH authenticates the peers; it does not confirm possession of the keys explicitly.

Entry	Property of the catalogue	Examined as	Why not
A07	Identity - Certificate - Key binding	A03	
R03	Operations at most once	Not applicable	No business operation, billing or resource allocation.
R05	Anti-degrading/security strategy maintained	Outside the scope	The modelled run fixes the negotiated algorithm set per run, so an on-path downgrade of the proposal cannot be asked here; limits records it.
C02	Anti-key disclosure fake (KCI)	Outside the scope	One long-term signature key per party and no way for a peer to sign as the other; the direction this entry speaks of needs a party's long-term key as an attacker input, which would let it impersonate that party in the agreement targets of the same file.
C03	Known key security/dissemination quarantine	K04	
C04	Temporary secret/partial state leaks resilience	K01, C08	
C05	Security restored after leak (PCS)	Not applicable	No ratchet or post-compromise recovery.
C06	Enhanced authentication after disclosure of a certificate of registration	Not applicable	No PAKE or registration record.
C07	Anti-offline dictionary attack	Outside the scope	The modelled run uses signatures; the pre-shared-key mode with a low-entropy secret is not modelled.
Z01	Authorization to execute correctness	Not applicable	No authorization, delegation, revocation or tenancy in an IKE SA establishment.
Z02	Delegation does not extend	Not applicable	No authorization, delegation, revocation or tenancy in an IKE SA establishment.
Z03	Authorisation and identity/meeting/operation binding	Not applicable	No authorization, delegation, revocation or tenancy in an IKE SA establishment.
Z04	Avoidance and expiry	Not applicable	No authorization, delegation, revocation or tenancy in an IKE SA establishment.
Z05	Cross Tenant/Interservice Safety	Not applicable	No authorization, delegation, revocation or tenancy in an IKE SA establishment.
S01	Encryption uniqueness	The prover cannot be asked	AEAD nonce derivation is not defined by these documents and not modelled.
S02	Re-transmit/key weight installation security	R02	
S03	Anti-roll and state monotony	R01, R04	
S04	Consistency in security context across phases	A08	
S05	Termination and old state lapses	R02	
P01	Identity anonymity	X02	
P02	Interconnectedness of sessions	Not applicable	Every run uses fresh SPIs and nonces and there is no stable per-user handle to link.

Entry	Property of the catalogue	Examined as	Why not
P03	User/evidence Existence Hide	Not applicable	The responder answers only after the exchange; no lookup reveals whether a user or credential exists.
V01	Conditional progress/ error recovery	The prover cannot be asked	Progress, deadlock and resource amplification are outside a trace prover.
V02	No unexpected death locks.	The prover cannot be asked	Progress, deadlock and resource amplification are outside a trace prover.
V03	Abuse of restricted resources/anti-magnification	The prover cannot be asked	Progress, deadlock and resource amplification are outside a trace prover.

The parties and the protected values, one by one

The host listed the 23 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: 10. Without: 13. 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	Sponsor	party	Everything lost by definition	A dishonest initiator knows the initiator's long-term key and every secret of its own run, so agreement and that run's key secrecy are lost by definition; the documents ask neither under a dishonest peer.
S02	Respondent	party	Everything lost by definition	Same as S01 for the responder.
S03	Attacker.	party	Nothing gained	The attacker is the participant whose ability the network already gives.
S04	IKESAKey Materials (%1)SKEYSEED(n)、SK_d、SK_ei/er、SK_ai/ar、SK_pi/pr）	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S05	Negotiating Algorithm series ADDKE	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S06	Rounds KE payload	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S07	Message ID	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S08	IKE_AUTH_MID	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.

ID	Party or value	Kind	Attacker	Why none
S09	IntAuth Chain	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S10	Algorithm series ADDKE	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S11	Rotation Order	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S12	IntAuth transcribed	value	Nothing gained	Algorithm sequence, round order, KE payloads, Message IDs, IKE_AUTH_MID and the IntAuth chain are values the network or the key holder already has; holding them alone gives no key.
S13	SKEYSEED	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S14	SK_d	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S15	SK_ei/SK_er	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S16	SK_ai/SK_ar	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S17	SK_pi/SK_pr	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S18	Rounds SK	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S19	GenerationsSK_d、SK_a、SK_e、SK_p	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S20	Completed IKE SA keys	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	
S21	Eventually.IKESAKey	value	`kem_component`, `classical_dh`, `gen1_keys`, `enc1_key`	

ID	Party or value	Kind	Attacker	Why none
S22	IKE_INTERMEDIATE Message	value	Nothing gained	The IKE_INTERMEDIATE and IKE_AUTH messages are what the network attacker holds; tampering is covered by the agreement targets.
S23	IKE_AUTH Message	value	Nothing gained	The IKE_INTERMEDIATE and IKE_AUTH messages are what the network attacker holds; tampering is covered by the agreement targets.

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		
`kem_component`	the shared secret SK(n) of one IKE_INTERMEDIATE round of a session, obtained after that session's IKE_AUTH completed; the other components and the long-term signature keys stay intact	The extension must survive the compromise of existing or future PQC key exchange algorithms (s0104), and such a compromise must not compromise the final SKEYSEED (s0106).	s0104, s0106
`classical_dh`	the classical (EC)DH shared secret g^ir of a session, recorded from the wire and computable later with a cryptographically relevant quantum computer	A quantum attacker may learn SK(x) when it is obtained by a classical key exchange (s0106); the main focus is to prevent harvest-and-decrypt (s0108).	s0106, s0108
`gen1_keys`	all generation-1 IKE SA keys of a session (SK_d(1), SK_ei(1), SK_er(1), SK_ai(1), SK_ar(1), SK_pi(1), SK_pr(1)), read out of a party's memory after the round	Each IKE_INTERMEDIATE produces a whole new generation of keys from SKEYSEED(n) (s0045) and the next round uses the most recently calculated keys (s0476).	s0045, s0476, s0498
`enc1_key`	only the generation-1 encryption key SK_ei(1) of a session	SK_ei protects the initiator's messages of the round (s0039, s0475), so a partial read of a party's state can expose it alone.	s0039, s0475

ID	`network`	`kem_component`	`classical_dh`	`gen1_keys`	`enc1_key`
A03	Holds				
A04	Holds				
A08	Holds				
K01	No answer	No answer			
K04				No answer	No answer
C01			No answer		
C08		No answer	No answer		
R01	Holds				
R02	Holds				
R04	Holds				
A08.11	Holds				

ID	network	kem_component	classical_dh	gen1_keys	enc1_key
K01.12		No answer			
A04.13	Holds				
X01	Holds				
X02	No answer				

Notes on the prover and the model

What the analyst named for targets:

step	targets	features	reason
52	A03, A04, A08, A08.11, A04.13, K01, K01.12, K04, C01, C08, R01, R02, R04, X01, X02	plain_primitives, many_sessions, other	The model treats ML-KEM encapsulation, prf/kdf, MAC, encryption and signatures as ideal primitives, and uses Diffie-Hellman only as a plain key agreement with the single law g^xy = g^yx; the properties are asked over any number of sessions of both roles. The IKEv2 Message ID counter, the window of overlapping requests and each party's one-time key stream are not modelled, so the R01/R02 answers rest on injectivity over fresh session values instead; no target depends on a counter or a replaced key, so no bounded model is used.

The analyst's closing note

All 15 targets are answered under every attacker set for them, and the report is complete (reading, figures, notes). Proved against the network attacker only: agreement in both directions (A03), injectivity of IKE_AUTH acceptance (A04, A04.13, R02), transcript binding of the two IKE_INTERMEDIATE rounds to the AUTH exchange (A08, A08.11, R04), freshness of IKE_AUTH through the session's fresh values (R01), and the distinctness of the two ML-KEM shared secrets (X01); each proof carries a reach lemma as its control.

Open, with no answer from ProVerif: the final IKE SA keys stay secret (K01, K01.12), one broken component does not break the SA (C08), harvest-and-decrypt forward secrecy (C01), key separation across generations and uses (K04), and initiator identity confidentiality (X02). For all of these ProVerif answered `cannot be proved` with no trace: neither a proof nor an attack.

What blocks those answers is the model, not the protocol. A leak of a session component inside the replicated responder made ProVerif diverge, so the compromise targets were asked for the network attacker only; the Message ID counter and window are not modelled, which limits R01/R02 to freshness through fresh session values. No bounded model or state review was needed, since no target rests on a counter or a replaced key. Tamarin+ Agent, which handles revealed values and the algebra directly, would settle K01, K01.12, C01, C08, K04 and X02.

There is no finding: no target was falsified, so nothing shows a weakness the documents do not already state, and the report prints "No finding". The proved results are guarantees the documents require, under a symbolic model with idealised ML-KEM, prf, MAC and signatures and with (EC)DH reduced to its one law.