Security analysis of IKEv2 PPK initial exchange

This report in another language: 中文.

rfc8784 · ProVerif+ Agent report

Documents	Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security (protocol definition); Internet Key Exchange Protocol Version 2 (IKEv2) (supporting)
Properties examined	15: 9 hold, 3 fail (3 attacks), 3 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 post-quantum preshared keys (RFC 8784) establishes an IKE SA in a fixed four-message exchange. The initiator sends HDR, SAi1, KEi, Ni and, when a PPK is configured for the responder, the USE_PPK notification. The responder answers with SAr1, KEr, Nr and its own USE_PPK. Both sides derive SKEYSEED from the DH share and the nonces, then rerun the derivation of SK_d, SK_pi and SK_pr through the PRF keyed with the PPK, leaving SK_e and SK_a untouched. The peers then exchange IKE_AUTH messages carrying an identity, a signature over the peer's nonce and the handshake messages, the Child SA proposal and traffic selectors; the initiator adds PPK_IDENTITY with the PPK identifier and, when the PPK is optional, NO_PPK_AUTH.

Each peer holds a long-term signing key, a list of PPKs with identifiers and a per-peer flag saying whether the PPK is mandatory. Each side ends by verifying the peer's signature. The initiator marks the SA as using or not using the PPK according to whether PPK_IDENTITY came back, and the responder marks it according to the PPK it applied. With the PPK mandatory, an initiator that sees no USE_PPK aborts instead of continuing.

The parties:

- **Initiator**: selects the PPK for this responder. Holds: ltkI (signing key), PPK, PPK_ID, mandatory_or_not.
- **Responder**: list of PPKs with identifiers. Holds: ltkR (signing key), PPK, PPK_ID, mandatory_or_not.

Figure 1 shows the exchange as the documents specify it.

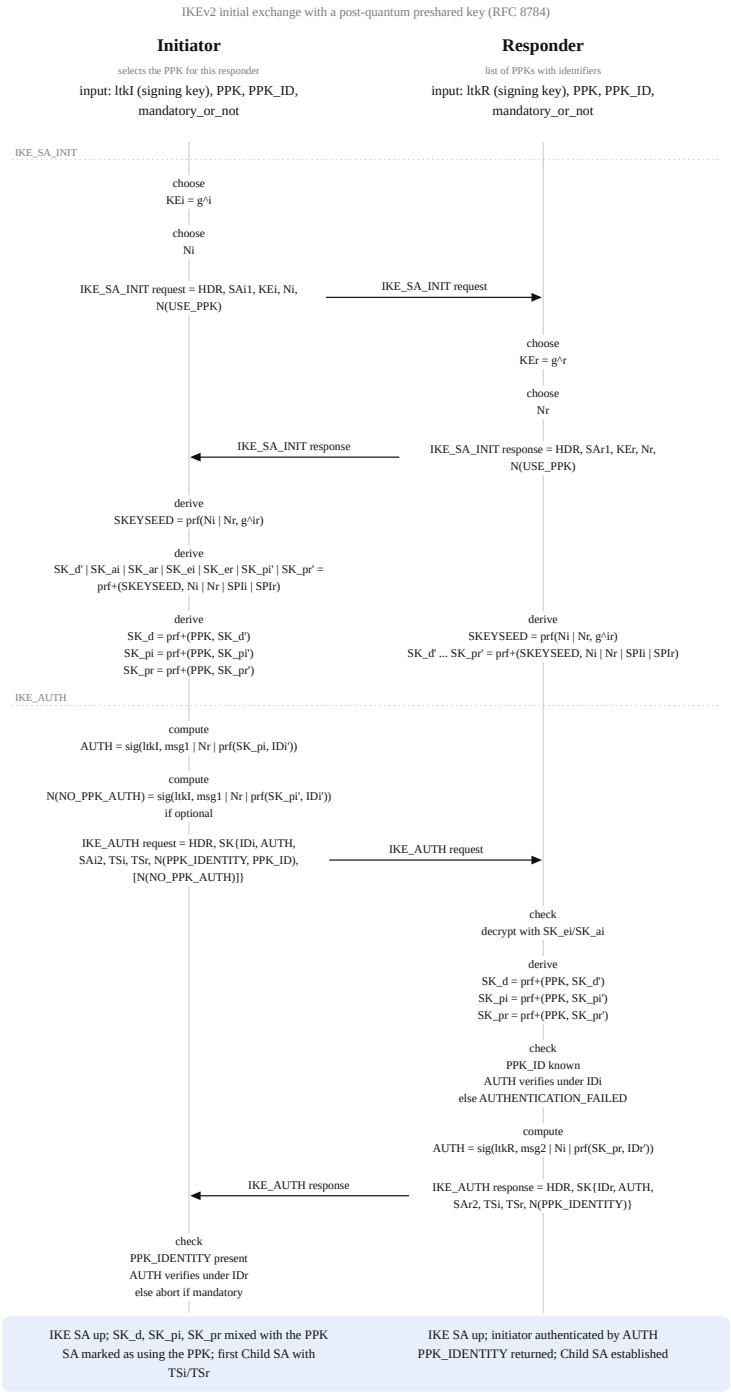


Figure 1. The protocol as specified.

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 protocol gives what it is meant to give within the scope analysed: mutual authentication and agreement on the session parameters hold against the network attacker, and SK_d along with every key derived from it stays secret even when the attacker knows the Diffie-Hellman share. No new weakness was found. The analysis reproduced the

weaknesses the documents state: the initial IKE SA is readable by anyone who can break the Diffie-Hellman exchange, an attacker that runs its own Diffie-Hellman learns the initiator's identity, a low-entropy PPK can be guessed offline, and when both peers treat the PPK as optional an attacker that breaks signatures in real time can downgrade the session to one without a PPK. Two claims stayed undecided because the prover could not turn its derivation into a trace: the optional-configuration downgrade itself, and whether a leaked initiator signing key allows impersonating the responder. Of the fifteen properties examined, eleven are proved, two are recorded as attacks the documents already state, one is only shown reachable, and two are undecided.

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
A02	**Mutual authentication of the IKE SA.** When the initiator (or the responder) completes IKE_AUTH and accepts an IKE SA, the claimed peer must have a real run of the protocol that targets this side's identity as its communication partner.	no compromise	Holds	medium	yes
A03	**Agreement on session parameters.** Every time one side accepts an IKE SA, there is a matching peer run, and both sides agree on the roles, the mutual identities, Ni, Nr, the DH shared value and the negotiated SA parameters.	no compromise	Holds	medium	unclear
A04	**No replay of an IKE SA run.** Every acceptance of an IKE SA by one side corresponds to a distinct real peer run, and the same peer run cannot be accepted twice.	no compromise	Holds	medium	unclear
K01	**Secrecy of PPK-derived and long-term keys.** The attacker cannot derive the long-term signing private key, the PPK, SK_d, SK_pi, SK_pr, or the subkeys derived from SK_d; the secrecy of SK_e/SK_a is evaluated separately and is not strengthened by the PPK.	no compromise	Attack holds (stated in the documents)	medium	yes
R05	**No non-PPK session when PPK is mandatory.** Under the mandatory configuration, the attacker cannot make the initiator accept a session without the PPK by removing USE_PPK or forging a response; under the optional configuration, a fallback to a session without the PPK must be marked accurately and audited.	no compromise	Holds	medium	yes
C01	**Forward secrecy of the IKE SA keys.** After the IKE SA completes and the ephemeral state is erased, leaking both sides' long-term signing private keys cannot give the attacker SK_d, SK_pi, SK_pr or the Child SA keys of earlier sessions.	both sides' long-term signing private keys (leaked afterwards)	Holds	medium	yes

ID	Property	Compromise assumed	Result	Value	In the documents
C03	**Separation of SK_e/SK_a from SK_d.** Leaking SK_e/SK_a or the DH shared value does not give the attacker SK_d, SK_pi, SK_pr or the Child SA keys derived from SK_d; leaking SK_d does not give the attacker SK_e/SK_a either.	SK_e/SK_a or the DH shared value (leaked during the session)	Holds	medium	unclear
C08	**SK_d secret when the DH share is known.** When the attacker obtains the DH shared value g^ir (modelling a quantum computer breaking DH) and the remaining inputs to the PRF are known, the attacker still cannot obtain SK_d, SK_pi, SK_pr or the Child SA keys derived from SK_d, as long as the PPK is not leaked.	DH shared value g^ir (leaked during the session)	Holds	high	yes
R05.9	**Mandatory PPK resists removal and forgery.** Under the mandatory configuration, the attacker cannot make the initiator accept a session without the PPK by removing USE_PPK or forging a response, and mutual authentication holds.	no compromise	Holds	medium	yes
C08.10	**SK_d secret under a real-time DH break.** When the attacker obtains the DH shared value g^ir and the PPK is not leaked, the attacker still cannot obtain SK_d.	DH shared value g^ir (leaked during the session)	Holds	high	yes
R05.11	**Fallback is marked as a non-PPK SA.** Under the optional configuration, a fallback to a session without the PPK must be marked accurately as a non-PPK session, and the attacker cannot tamper with that mark.	no compromise	No answer	medium	unclear
X01	**Optional-configuration downgrade to a non-PPK IKE SA.** When both sides are optional and authenticate with digital signatures, an attacker able to break public-key operations in real time (DH and signatures) can remove USE_PPK from IKE_SA_INIT and forge the subsequent AUTH messages, making both sides accept an IKE SA that does not use the PPK.	real-time break of public-key operations: can recover the session's g^ir and forge AUTH messages for both sides' signatures.	No answer	low	yes
X02	**Initiator identity disclosed to an active attacker.** An attacker that runs DH itself can learn the initiator identity IDi (and the configuration and SA information that follows) from the IKE_AUTH messages, even if it cannot complete IKE_AUTH.	the session's DH shared value g^ir: the attacker chooses the peer's DH exponent itself (the zpub and nonce n0 public in the model) or recovers g^ir with a quantum computer, so it knows all DH-derived inputs of the session but still does not know the PPK.	Attack holds (stated in the documents)	low	yes
X03	**PPK not guessable offline by an impersonating responder.** An attacker impersonating the responder cannot use the initiator's IKE_AUTH messages to test PPK guesses offline; that is, a low-entropy PPK is not recovered by a dictionary search.	an attacker impersonating the responder, who after receiving the initiator's IKE_AUTH can run a dictionary search over PPK values offline.	Attack holds (stated in the documents)	low	yes

ID	Property	Compromise assumed	Result	Value	In the documents
X04	**Responder authentication after the initiator's long-term key leaks.** Even when the attacker holds the initiator's long-term signing private key and can run DH itself, it cannot make the initiator accept an IKE SA whose peer is not the real responder (KCI resistance).	the initiator's long-term signing private key, obtained before or during the session, plus the ability to run DH itself.	No answer	none	unclear

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

Goals the documents state that the drafted list of properties did not have, added by the analyst (Appendix D): X01, X02, X03, X04.

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: resistance to negotiation removal and mutual authentication with PPK mandatory	Both halves hold: with the PPK mandatory the initiator never accepts a session without the PPK even if USE_PPK is removed or a response is forged, and every accepted session still matches a real responder run with the same nonces and DH values, so authentication is not bypassed; the only reachable outcome of the attempt is an abort.	R05, R05.9, A02
Q2: secrecy of SK_d when DH leaks but the PPK does not	SK_d and the keys derived from it stay secret even when the attacker knows the DH share, because SK_d is rerun through the PRF keyed with the PPK, while SK_e, SK_a and the intermediate keys of that same session are exposed to the same attacker.	C08, C08.10, K01, C03
Q3: whether an optional fallback is marked accurately as a non-PPK session	The fallback is marked consistently in both directions: an SA marked as using the PPK is one the responder established with the PPK, and an SA marked as a fallback is one both peers established without it; the stronger reading that the two marks can never disagree was left undecided, and it is this fallback path that the documents also describe as the way an optional configuration can be downgraded.	R05.11, X01

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:

- Rekey, EAP, resumption and CREATE_CHILD_SA are out of scope by the task; only the fixed initial exchange is modelled.
- Signatures, symmetric encryption, the PRF and the PPK mixing are ideal, so nothing is proved about their computational strength or their post-quantum security.
- Diffie-Hellman is modelled with the single law g^xy = g^yx and no other algebraic structure; an attack found is an attack on the protocol, a proof is a proof against a weaker attacker than a real one.
- The PPK is assumed to have full entropy in every claim except the offline-guessing one, where the weak-secret query removes that assumption.
- Message IDs, retransmission timers, cookies, NAT traversal and traffic-selector negotiation are not modelled, so availability, denial of service and the half-open SA are not analysed.
- The INFORMATIONAL and delete exchanges, and behaviour after the SA is established, are not modelled.
- Peer identities are constants with keys assumed correctly bound to them; certificate misissuance or a wrong authorization-database entry is outside the model.

- Two claims, the optional-configuration downgrade and the key-compromise-impersonation question, were left undecided by the prover's over-approximation; a Tamarin+ Agent model would settle them.

What the host records:

- No answer for: R05.11, X01, X04.
- The answers of K01, C03, C08, C08.10, X01, X02 rest on an operation the model simplifies (Appendix D).
- The prover gave no answer (Appendix C): R05.11: mark_swap, X01: x01_downgrade, X04: x04_kci. The property was asked and the prover neither proved it nor found an attack, so the property is shown without an answer.
- Proved without a control query (Appendix C): A02: agree_ini, A03: agree_ini, K01: sec_skpi, K01: sec_child, K01: sec_ppk, C01: sec_skpi, C01: sec_child, R05.9: agree_ini, R05.11: mark_ppk, R05.11: mark_noppk, X01: x01_ctl. A control is a variant of the property that must fail; without one, nothing shows that the property as written can fail in this model.

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

Appendix A: Reading of each property

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

A02: Mutual authentication of the IKE SA

When the initiator (or the responder) completes IKE_AUTH and accepts the IKE SA, the claimed peer must have a real protocol run in which this party's identity is the communication target.

Both directions of authentication hold for any number of sessions. If the initiator accepts an IKE SA, a responder really ran with the same nonces, DH values and PPK mode, and if the responder accepts one, the initiator really sent the matching IKE_AUTH. The proof needs the responder's AUTH to be verified against its signature; without that check an initiator accepts the forged responder, as the control shows.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: RFC 7296 states that IKE performs mutual authentication and gives the derivation of both AUTH payloads; the model confirms it mechanically for this PPK variant.
- Stated in the documents: yes. Where the documents say it: s0058, s0062, s0390
- Rests on: Signatures and symmetric encryption are ideal: no hash collisions, no forgery without the key.; The identities are fixed constants, so the property is about agreement on the session, not about a name being bound to a specific key by a PKI.; The responder accepts NO_PPK_AUTH as the initiator's authentication data in the fallback, and the model does not distinguish it from the AUTH payload.
- Next step: None for the network attacker; a PKI-level binding of names to keys is outside the model.

A03: Agreement on session parameters

Whenever one side accepts an IKE SA, there is a matching peer run, and both sides agree on the roles, the mutual identities, Ni, Nr, the DH shared value and the negotiated SA parameters.

The two peers agree on the nonces, the DH shares and the PPK mode of an accepted session. Both AUTH payloads cover the peer's nonce and the two IKE_SA_INIT messages, so neither value can be replaced without detection.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: RFC 7296 says the signature covers the other side's nonce and stresses that this is critical to the security of the exchange, which is the property confirmed here.
- Stated in the documents: yes. Where the documents say it: s0390, s0392, s0394
- Rests on: The signature is idealised; the model does not check that it is bound to a certificate chain.; The negotiated algorithms SAi1/SAr1 are abstracted as a single constant, so agreement on a specific suite is not checked.
- Open: Agreement on the concrete cryptographic suite is not modelled.
- Next step: Model the SA payload as a value the responder chooses from the initiator's offer if suite agreement matters.

A04: No replay of an IKE SA run

Every time one side accepts an IKE SA it corresponds to a distinct real peer run, and the same peer run cannot be accepted twice.

The injective form holds: one accepting initiator run cannot be matched twice by the responder, so a recorded IKE_AUTH request cannot be replayed as a new SA. The model gives each run fresh nonces and DH values, which is the reason the property holds.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value medium: RFC 7296 requires nonces to be random and uses Message IDs to identify retransmissions, but does not state injective agreement as a goal; the result follows from the freshness rules rather than from a sentence of the documents.
- Stated in the documents: unclear. Source text it rests on: s0362, s0226
- Rests on: Freshness is a property of nonce generation, which the model assumes; a peer that repeats a nonce is outside the model.; Message IDs and the rekey and resumption exchanges, where replays are more delicate, are out of scope.
- Open: Replay of messages in a window, and reuse of a nonce across sessions, are not analysed.
- Next step: A Tamarin model with an explicit window or nonce-reuse model would settle how far replay protection reaches.

K01: Secrecy of PPK-derived and long-term keys

The attacker cannot derive the long-term signing private keys, the PPK, SK_d, SK_pi, SK_pr and the Child SA keys derived from SK_d; the secrecy of SK_e/SK_a is assessed separately and is not strengthened by the PPK.

With the PPK in use the network attacker obtains neither the PPK, nor the signing keys, nor SK_d, SK_pi, SK_pr or the Child SA keys, and SK_e and SK_a stay secret too because the attacker cannot compute the DH share. Once the attacker knows that share it reads the handshake, including SK_e and SK_a, while SK_d and everything derived from it stays out of reach. That is the documented split: the PPK covers the keys derived from SK_d, not the handshake encryption.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable with caveats
- Value medium: RFC 8784 says SK_d is a function of the PPK and quantum secure, and states that an attacker able to decrypt the initial IKE SA obtains everything except the keys protected by the PPK; the model reproduces both halves mechanically.
- Stated in the documents: yes. Where the documents say it: s0016, s0044, s0045
- Rests on: The PRF, the KDF and the PPK mixing are ideal: nothing is proved about their strength or about post-quantum security.; The PPK is assumed to have full entropy; guessing it is the separate claim answered under the offline guesser.; SK_e and SK_a are modelled as single keys, without direction separation.
- Open: The security of the PRF and of the PPK mixing against a quantum attacker is a computational question.
- Next step: A computational proof or a Tamarin model with an explicit PPK-entropy assumption would say what an attacker with a quantum computer gains.

R05: No non-PPK session when PPK is mandatory

Under the mandatory configuration the attacker cannot make the initiator accept a session without the PPK by removing USE_PPK or forging a response; under the optional configuration a fallback to a session without the PPK must be accurately marked and audited.

With the PPK flagged mandatory the initiator never accepts an SA established without the PPK: a response without USE_PPK makes it abort, and the abort is reachable in the model. In the optional configuration the same response does lead to an accepted SA, which is the intended fallback and the control that shows the check does work.

- Result: Holds; answers what is asked: yes; usable
- Value medium: RFC 8784 states the abort rule and the table of configuration cases, which is what the two queries check.

- Stated in the documents: yes. Where the documents say it: s0012, s0031
- Rests on: The flag is a parameter of the run; a peer whose configuration changes during the exchange is not covered.; The retransmission behaviour recommended against denial of service is not modelled.
- Open: Whether the abort leaves a half-open SA on the responder, and how the recommended retransmission policy changes it, is not modelled.
- Next step: Model the retransmission timer and the half-open SA if the denial-of-service consequence is to be evaluated.

C01: Forward secrecy of the IKE SA keys

After the IKE SA completes and the ephemeral state is erased, leaking the long-term signing private keys of both sides must not give the attacker the SK_d, SK_pi, SK_pr or Child SA keys of earlier sessions.

After a PPK session is complete and the ephemeral state is erased, disclosure of both long-term signing keys gives the attacker no SK_d, SK_pi, SK_pr or Child SA key. Those keys are functions of the ephemeral DH share and of the PPK, neither of which the long-term keys reach.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: RFC 7296 defines perfect forward secrecy for exactly this scenario, including the requirement to forget anything that could recompute the keys.
- Stated in the documents: yes. Where the documents say it: s0366, s0367
- Rests on: Ephemeral exponents are erased, as the model assumes by never exposing them; a peer that keeps its DH exponential loses this property.; Disclosure is a step after the run; a leak during the exchange is the separate claim about the DH share.; The PPK is not leaked in this scenario.
- Next step: None within the scope; rekey and resumption, excluded by the task, would carry the property forward.

C03: Separation of SK_e/SK_a from SK_d

Disclosing SK_e/SK_a or the DH share does not give the attacker SK_d, SK_pi, SK_pr or the Child SA keys derived from SK_d; disclosing SK_d does not give the attacker SK_e/SK_a either.

The two key sets do not reach each other. Reading the handshake keys SK_e and SK_a of a session gives no SK_d, SK_pi or SK_pr, because the PPK enters only the second derivation; reading SK_d, SK_pi and SK_pr gives no SK_e or SK_a, because those come from SKEYSEED alone. The controls show that a leaked key is recovered once it is handed over.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: RFC 8784 gives the key schedule, from which the separation follows, and says the handshake keys are not PPK-protected, but it does not state as a goal that a leak on one side of the schedule leaves the other side intact.
- Stated in the documents: unclear. Source text it rests on: s0015, s0016, s0382, s0385
- Rests on: The KDF is ideal, so the separation is structural and says nothing about a real PRF leaking across its outputs.; Only one key per direction is modelled, so the two directions are treated as one.
- Next step: None; the structural claim is settled, a computational one is not available here.

C08: SK_d secret when the DH share is known

When the attacker has obtained the DH share g^ir (modelling a quantum computer breaking DH) and all other inputs to the PRF are known, the attacker still cannot obtain SK_d, SK_pi, SK_pr or the Child SA keys derived from SK_d, as long as the PPK is not leaked.

An attacker that knows the DH share of a session, and every other input to the PRF except the PPK, still does not obtain SK_d or the Child SA keys. The PPK is the only remaining input and the model shows it cannot be recovered from the exchange. The control shows that the same attacker does get the intermediate key computed without the PPK.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value high: This is the central claim of RFC 8784, stated for exactly this attacker; the model confirms it mechanically and shows the condition it rests on, that SK_d actually depends on the PPK.
- Stated in the documents: yes. Where the documents say it: s0044, s0045
- Rests on: The PPK mixing is an ideal function and is assumed to have full entropy; the model proves the structure, not the strength.; The attacker is given the DH share as a value; no partial DH information is modelled.
- Open: Post-quantum strength of the PRF and of the PPK entropy is a computational question.
- Next step: A computational analysis or a Tamarin model with a PPK dictionary would bound what a real attacker gains.

C08.10: SK_d secret under a real-time DH break

When the attacker has obtained the DH share g^ir and the PPK is not leaked, the attacker still cannot obtain SK_d.

In a session where the attacker chose the responder's ephemeral DH value, the SA still gets SK_d protected by the PPK: the attacker reads the handshake but not the key used for the Child SA. The same trace shows the intermediate key and the handshake keys exposed, so the protection is exactly as narrow as the documents describe.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value high: RFC 8784 states that an attacker able to decrypt the initial IKE SA still cannot get the IPsec SA keys, which the PPK protects.
- Stated in the documents: yes. Where the documents say it: s0044, s0045
- Rests on: The attacker is given the DH share rather than deriving it, which is the strongest form of the assumption.; The PPK is not leaked and is assumed to have full entropy.
- Next step: None within a symbolic model.

R05.9: Mandatory PPK resists removal and forgery

With the PPK mandatory, the attacker cannot make the initiator accept a session without the PPK by removing USE_PPK or forging a response, and authentication of both peers holds.

With the PPK mandatory, removing USE_PPK or injecting a forged response cannot make the initiator accept a session without the PPK; the attempt ends in an abort. Authentication is not weakened in the process: any SA that is accepted still corresponds to a real responder run.

- Result: Holds; answers what is asked: yes; usable
- Value medium: RFC 8784 states that with the PPK mandatory for at least one peer, or with preshared-key authentication, the downgrade is detected and the SA is not created.
- Stated in the documents: yes. Where the documents say it: s0052, s0053
- Rests on: Responses are modelled as network messages; the denial-of-service angle is out of scope.; The configuration flags are parameters of the run.
- Open: The availability consequence, that the initiator can be pushed into aborting, is not modelled.
- Next step: A model of the retransmission policy would say how far an injected response can disrupt the exchange.

R05.11: Fallback is marked as a non-PPK SA

In the optional configuration, a fallback to a session without the PPK must be marked accurately as a non-PPK session, and the attacker must not be able to tamper with that marker.

Both directions of the marker are accurate: an SA the initiator marks as using the PPK is one the responder established with the PPK, and a fallback is marked as a session without the PPK on both sides. The reading that the two marks can never disagree was not settled: the prover reached that combination only in an over-approximated derivation, not in a real trace.

- Result: No answer; answers what is asked: partly; usable with caveats
- Value medium: RFC 8784 requires the initiator to mark the SA from the presence of PPK_IDENTITY and advises auditing a configured PPK that was not used, but does not state the marker as a property of the exchange.
- Stated in the documents: unclear. Source text it rests on: s0037, s0054
- Rests on: The marker is a local label the peers compute; the model does not audit whether an implementation logs it.; The reading where both peers' marks must agree is the one left open.
- Open: Whether a peer can be led to mark a session differently from the other peer is undecided.
- Next step: A Tamarin model that keeps the two peers' marks as explicit state would settle the open reading.

X01: Optional-configuration downgrade to a non-PPK IKE SA

When both peers are in the optional configuration and authenticate with digital signatures, an attacker that can break public-key operations (DH and signatures) in real time can remove USE_PPK from IKE_SA_INIT and forge the subsequent AUTH, so that both peers accept an IKE SA that does not use the PPK.

When the PPK is mandatory, a forged response with the notification removed cannot produce an accepted SA, which is proved. The attack the documents describe for the optional configuration, where an attacker that breaks public-key operations in real time removes USE_PPK and forges the signatures, could not be turned into a trace: the derivations reach the goal only by combining two incompatible deliveries to the same initiator, so the model neither shows the attack nor rules it out.

- Result: No answer; answers what is asked: partly; not usable
- Value low: RFC 8784 states the downgrade attack and its conditions itself; the model adds nothing beyond confirming the mandatory half.
- Stated in the documents: yes. Where the documents say it: s0052
- Rests on: The attack is stated by RFC 8784, not shown by the prover; the undecided reading is a limitation of the model, not a doubt about the documents.; Modelling a real-time break of signatures as an ability to sign with either peer's long-term key is what turns the missing trace into a model limitation.
- Open: Whether the optional-configuration downgrade reaches an accepted SA in this model.
- Next step: A Tamarin model, where the fallback and the two delivery paths of IKE_SA_INIT are modelled without this over-approximation, would settle it.

X02: Initiator identity disclosed to an active attacker

An attacker that runs DH itself can learn the initiator's identity IDi (and the subsequent configuration and SA information) from the IKE_AUTH message, even without completing IKE_AUTH.

An attacker that answers the initiator with its own DH value and then drops the exchange still decrypts the IKE_AUTH request and reads the initiator's identity. The model shows the identity reaching the attacker on the network, so the encrypted IKE_AUTH cannot be relied on to hide who the initiator is.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable

- Value low: RFC 7296 states that a man in the middle who cannot complete IKE_AUTH can still see the initiator's identity, and its security considerations say the initiator must identify itself first.
- Stated in the documents: yes. Where the documents say it: s0087
- Rests on: The attacker is given the DH share, which is the assumption of an attacker able to break the exchange in real time.; The identity is a fixed constant; the optional IDr hiding the responder is not modelled.
- Next step: None; the claim is a documented consequence of the message order.

X03: PPK not guessable offline by an impersonating responder

An attacker impersonating the responder cannot use the initiator's IKE_AUTH message to test PPK guesses offline; that is, a low-entropy PPK cannot be recovered by dictionary search.

A PPK chosen from a small set is recoverable: the attacker that impersonates the responder reaches an offline test on a PPK-derived value it can compute from a guess, and the prover could not prove the PPK unguessable. The key schedule therefore protects only a PPK with real entropy.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value low: RFC 8784 states that an adversary impersonating the server can run a dictionary search over plausible PPK values and requires at least 256 bits of entropy.
- Stated in the documents: yes. Where the documents say it: s0043
- Rests on: The guess is modelled as a weak secret, the right abstraction for a value people choose.; The PPK is not otherwise compromised; the entropy assumption is exactly what this query removes.
- Next step: None; the document states the requirement and the model agrees with it.

X04: Responder authentication after the initiator's long-term key leaks

Even when the attacker holds the initiator's long-term signing private key and can run DH itself, it cannot make the initiator accept an IKE SA whose peer is not the real responder (KCI resistance).

Whether an attacker holding the initiator's signing key can make the initiator accept a peer that is not the real responder could not be decided. The prover reached the situation only through two deliveries of the same IKE_SA_INIT response, so there is neither a trace nor a proof.

- Result: No answer; answers what is asked: undetermined; not usable
- Value none: The documents do not state key-compromise impersonation as a goal for the initial exchange; the question arose from the catalogue rather than from the text, and the prover gave no answer either way.
- Stated in the documents: unclear. Source text it rests on: s0366
- Rests on: The reading would rest on how the attacker's ability to sign in the initiator's name is modelled, which is what the leak process grants.; The undecided answer is a prover limitation; no conclusion about the protocol should be drawn from it.
- Open: Whether a leaked initiator signing key lets an attacker impersonate the responder to the initiator.
- Next step: A Tamarin model of the same exchange would settle it, since the over-approximation that blocks the result here is not present there.

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
A02	proved		agree_ini	1 of 2		2 proved, 1 reachable
A03	proved		agree_ini	1 of 2		2 proved
A04	proved		agree_inj	checked		1 proved
K01	proved		sec_skd	4 of 7	c08_ske (holds_in_source)	7 proved, 1 attack
R05	proved		r05_mand	checked		1 proved, 1 reachable
C01	proved		sec_skd	1 of 3		3 proved
C03	proved		c03_skd	checked		2 proved
C08	proved		c08_skd	checked		1 proved
R05.9	proved		agree_ini	1 of 2		2 proved
C08.10	proved		c08_skd	checked		1 proved
R05.11	proved		mark_ppk			2 proved, 1 undecided
X01	proved		x01_ctl			1 proved, 1 undecided
X02	none				x02_idi (holds_in_source)	1 attack
X03	none				x03_guess (holds_in_source)	1 attack
X04	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
A02	When the initiator accepts an IKE SA, a responder run exists with the same Ni, Nr, g^ir values and the same PPK mode (initiator-side authentication and aliveness).	`network`	proved	agree_ini	none
A02	When the responder accepts an IKE SA, the initiator ran with the same Ni and Nr and the responder verified the initiator's AUTH (responder-side authentication).	`network`	proved	agree_resp	checked
A02	An initiator and a responder that both run the protocol accept an IKE SA in PPK mode: the normal complete run exists.	`network`	reachable	run	
A03	The initiator's accepted session parameters (Ni, Nr, g^ir, PPK mode) are those of a real responder run.	`network`	proved	agree_ini	none
A03	The responder's accepted session parameters (Ni, Nr, g^ir) are those of a real initiator run.	`network`	proved	agree_resp	checked
A04	Each initiator acceptance corresponds to a distinct responder run: the same responder run cannot be accepted twice (injectivity).	`network`	proved	agree_inj	checked

target	claim	attacker	answer	query	control
K01	SK_pi and SK_pr of a PPK-protected IKE SA are not derivable by the network attacker (the AUTH keys stay secret).	`network`	proved	sec_skpi	none
K01	Child SA keys derived from SK_d are not derivable by the network attacker.	`network`	proved	sec_child	none
K01	The PPK itself is never derivable from the exchange by the network attacker.	`network`	proved	sec_ppk	none
K01	SK_e/SK_a of the initial IKE SA are not protected by the PPK: an attacker that knows g^ir obtains them (the PPK does not cover the handshake encryption).	`gir_known`	attack (holds_in_source)	c08_ske (the control of sec_ske)	
K01	SK_d of a PPK-protected IKE SA is not derivable by the network attacker.	`network`	proved	sec_skd	checked
K01	The long-term signing key of the initiator is not derivable by the network attacker during the initial exchange.	`network`	proved	sec_ltkI0	checked
K01	The long-term signing key of the responder is not derivable by the network attacker during the initial exchange.	`network`	proved	sec_ltkR0	checked
K01	SK_e/SK_a of a PPK-protected IKE SA are not derivable by the network attacker (they are not weakened, but also not strengthened, by the PPK).	`network`	proved	sec_ske	checked
R05	When the PPK is mandatory and the response carries no USE_PPK, the initiator aborts the exchange (the abort is reachable, so the check is exercised).		reachable	r05_abort	
R05	With the PPK configured as mandatory, the initiator never accepts an IKE SA that was established without using the PPK.	`network`	proved	r05_mand	checked
C01	SK_pi and SK_pr of a completed PPK session stay secret after disclosure of both long-term signing keys.	`ltk_both_after`	proved	sec_skpi	none
C01	Child SA keys of a completed PPK session stay secret after disclosure of both long-term signing keys.	`ltk_both_after`	proved	sec_child	none
C01	SK_d of a completed PPK session is not derivable after both long-term signing keys are disclosed and the ephemeral state is erased.	`ltk_both_after`	proved	sec_skd	checked
C03	Disclosure of the session's SK_e/SK_a gives no SK_d, SK_pi, SK_pr or child SA key.	`s_ke_leak`	proved	c03_skd	checked
C03	Disclosure of SK_d/SK_pi/SK_pr gives no SK_e/SK_a of the same IKE SA (the two key sets are separated).	`s_kd_leak`	proved	c03_ske	checked
C08	With g^ir and all other PRF inputs known but the PPK not leaked, SK_d and the child SA keys stay secret.	`gir_known`	proved	c08_skd	checked
R05.9	Even under the mandatory/no-USE_PPK attempt the accepted SA is one for which the responder really ran (authentication is not bypassed).	`network`	proved	agree_ini	none

target	claim	attacker	answer	query	control
R05.9	With the PPK mandatory, a forged response that removes USE_PPK cannot make the initiator accept a non-PPK SA, and mutual authentication still holds for accepted SAs.	`network`	proved	r05_mand	checked
C08.10	In a session whose DH share the attacker knows, SK_d remains protected by the PPK.	`gir_known`	proved	c08_skd	checked
R05.11	If the initiator marks the SA as using the PPK, the responder used the PPK for the same session (the PPK marker cannot be moved to a session that did not use the PPK).	`network`	proved	mark_ppk	none
R05.11	If the initiator marks the SA as not using the PPK (optional fallback), the responder also established it without the PPK, so the audit mark is accurate.	`network`	proved	mark_noppk	none
R05.11	A session accepted under the PPK marker cannot be one the responder established without the PPK: the marker cannot be swapped between the two peers.	`network`	undecided	mark_swap	
X01	When both peers are optional and authenticate with signatures, an attacker that breaks public-key operations in real time removes USE_PPK and forges the AUTH payloads, so both peers accept a non-PPK SA.	`quantum_downgrade`	undecided	x01_downgrade	
X01	When the PPK is mandatory for the initiator, a forged non-PPK response cannot lead to an accepted SA.	`quantum_downgrade`	proved	x01_ctl	none
X02	An attacker that runs its own DH against the initiator and cannot finish IKE_AUTH still learns the initiator's identity IDi from the IKE_AUTH request.	`gir_known`	attack (holds_in_source)	x02_idi	
X03	An attacker impersonating the responder can test PPK guesses offline against the initiator's IKE_AUTH data, so a low-entropy PPK is recovered.	`offline_guesser`	attack (holds_in_source)	x03_guess	
X04	Even holding the initiator's long-term signing key, an attacker cannot make the initiator accept an IKE SA whose peer is not the real responder.	`ltk_i_leak`	undecided	x04_kci	

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-20261114T210659, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 43, prover 82 s, wall 17 min, cost 0.191 USD.

Context: the host removed stale parts from the conversation 1 time(s), 360332 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 90 proofs run, 79 decided, 11 without an answer, 70 s, 35 answers recorded.

What was analysed

Protocol	IKEv2 PPK initial exchange
Version	rfc8784
Analysed	Fixed initial IKE_SA_INIT/IKE_AUTH, signature authentication on both sides, PPK preconfigured; mandatory and optional/no-PPK fallback analysed separately. It is made explicit that SK_e/SK_a are not mixed with the PPK, so the properties of SK_d must not be misapplied to handshake confidentiality; rekey, EAP and resumption are excluded.
Not analysed	The symbolic model does not prove computational security or post-quantum security of the algorithms; attack traces hold only under the stated assumptions and are not directly equivalent to flaws in the specification. Each property must have a reachability/positive control.
Source text	557 numbered passages (185601 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	Assumptions
main	3	Exchanges
main	6	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
rfc7296	1.4	The INFORMATIONAL Exchange
rfc7296	1.4.1	Deleting an SA with INFORMATIONAL Exchanges

Document	Section	Title
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
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

Document	Section	Title
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

The purpose of the protocol as the source step located it in the documents: fixed initial IKE_SA_INIT/IKE_AUTH with mutual signature authentication and preconfigured PPK; the mandatory and the optional/no-PPK fallback are analyzed separately. It is made explicit that SK_e/SK_a are not mixed with the PPK and that the properties of SK_d must not be misapplied to handshake confidentiality; rekey, EAP, and resumption are excluded.

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
A02	Mutual authentication of the IKE SA	The initiator and the responder each are confident that the peer they claim to be accepting when accepting the IKE SA really exists and ran IKE_SA_INIT/IKE_AUTH under that identity; the identity is bound by the AUTH payload.	RFC 7296 states that IKE performs mutual authentication between two peers and establishes an IKE SA (s0058, s0062); AUTH covers the peer's nonce and identity (s0390, s0392-s0394).	s0058, s0062, s0390
A03	Agreement on session parameters	Both parties agree on the roles, the mutual identities IDi/IDr, Ni, Nr, g^ir, and the negotiated SA parameters, and either party's acceptance corresponds to the peer running the same session data.	s0390 emphasizes that each party must sign the peer's nonce; s0392-s0394 give the message, nonce, and MACedID covered by the signature.	s0087, s0390, s0392, s0393
A04	No replay of an IKE SA run	Each acceptance corresponds to a different peer run; the same peer run cannot be accepted twice.	Nonces must be random and long enough (s0362), and the Message ID is used to match requests and retransmissions (s0226, s0089).	s0089, s0226, s0362
K01	Secrecy of PPK-derived and long-term keys	The long-term signature private keys, the PPK, SK_d, SK_pi, SK_pr, and the Child SA keys derived from SK_d are kept secret from a network attacker; SK_e/SK_a are not mixed with the PPK and are assessed separately.	SK_d is a function of the PPK (s0044); only SK_d, SK_pi, and SK_pr are further mixed with the PPK via prf+ (s0016, s0017, s0034); SK_e/SK_a come only from SKEYSEED (s0382, s0385).	s0016, s0044, s0382
R05	No non-PPK session when PPK is mandatory	Under a mandatory configuration the initiator will not accept an IKE SA that does not use the PPK: neither removing USE_PPK nor forging a response can make it continue.	The initiator must abort when it is mandatory and the response does not contain USE_PPK (s0012); downgrade is only possible when both sides are optional and signatures are used (s0052).	s0012, s0013, s0052
C01	Forward secrecy of the IKE SA keys	After the session completes and ephemeral state is erased, leaking both parties' long-term signature private keys must not let an attacker obtain the SK_d, SK_pi, SK_pr, or Child SA keys of earlier sessions.	Perfect forward secrecy is defined as: after the connection is closed and the session keys are forgotten, the session keys cannot be reconstructed even with both parties' long-term keys (s0366-s0367).	s0366, s0367

ID	Name	About	What the documents say	Where
C03	Separation of SK_e/SK_a from SK_d	Leaking a session's SK_e/SK_a does not give SK_d, SK_pi, SK_pr, or the Child SA keys; leaking SK_d does not give SK_e/SK_a either.	SK_e/SK_a are derived only by prf+ from SKEYSEED, and the PPK is used only to re-derive SK_d, SK_pi, and SK_pr (s0015-s0017, s0382, s0385).	s0015, s0016, s0382, s0385
C08	SK_d secret when the DH share is known	Even if the attacker obtains the session's g^ir and the other inputs to the PRF, SK_d and the Child SA keys derived from it remain secret as long as the PPK is not leaked.	s0044 states explicitly: assuming the PPK has sufficient entropy, even if the attacker recovers the remaining PRF inputs, recovering SK_d with a quantum computer is infeasible.	s0044, s0045
R05.9	Mandatory PPK resists removal and forgery	Under a mandatory configuration, neither removing USE_PPK nor forging a response can make the initiator accept a non-PPK session, while authentication still holds.	s0052 says that if at least one side is mandatory the attack is detected and the SA is not established; s0053 says that under mandatory, injecting a forged response can only cause an abort.	s0052, s0053
C08.10	SK_d secret under a real-time DH break	In a session where the attacker obtains g^ir in real time, SK_d is still protected by the PPK.	s0044 and s0045 state that a quantum attacker can decrypt the initial IKE SA, but the IPsec SA keys are protected by the PPK.	s0044, s0045
R05.11	Fallback is marked as a non-PPK SA	On optional fallback both parties give a consistent indication of PPK usage, and a network attacker cannot tamper with that indication.	The initiator marks whether the SA uses the PPK based on the presence or absence of PPK_IDENTITY (s0037); auditing is recommended when a PPK is configured but not used (s0054).	s0037, s0054
X01	Optional-configuration downgrade to a non-PPK IKE SA	Whether the IKE SA accepted by both parties actually uses the PPK; the attacker thereby obtains all keys of that SA.	s0052 explicitly describes this downgrade and states that it is detected if at least one side is mandatory.	main:619, s0052
X02	Initiator identity disclosed to an active attacker	The visibility of the initiator identity IDi to an unauthenticated active attacker.	rfc7296 s0087 states explicitly: a man in the middle that cannot complete IKE_AUTH can still see the initiator's identity; the P13 section states that the initiator must reveal its identity first and can be probed.	rfc7296:6486, rfc7296:6478, s0087
X03	PPK not guessable offline by an impersonating responder	The offline guessability of the PPK.	s0043 states explicitly: even with only an ordinary computer, an attacker impersonating the server can run a dictionary search over PPK values, so low-entropy sources are insecure; s6558 gives the same warning.	main:544, rfc7296:6561, s0043
X04	Responder authentication after the initiator's long-term key leaks	The initiator's guarantee of the responder's identity still holds after its own long-term key leaks.	The documents do not directly claim KCI resistance; s0366 only claims that a long-term key leak cannot yield the session keys.	s0366

Added from the documents by the analyst: X01, X02, X03, X04.

The security statements of the documents, paragraph by paragraph

The host listed the 39 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: 9. Not taken: 30. 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:541-554	K01, X03, `offline_guesser`	
P02	main:556-565	C08, C08.10, K01	
P03	main:567-572	K01, X02, `gir_known`	
P04	main:574-584	X02	
P05	main:586-591	The prover cannot be asked	Only a symmetric algorithm and key-length policy is given; computing algorithm strength is outside the symbolic model.
P06	main:593-594	The prover cannot be asked	A key-length requirement, which is algorithm strength.
P07	main:596-596	The prover cannot be asked	An ESP key-length requirement, which is algorithm strength.
P08	main:598-600	The prover cannot be asked	The internal PRF key length, which is algorithm strength.
P09	main:602-617	The prover cannot be asked	Discusses the DoS and retry policy caused by a half-open SA left on the responder after the initiator aborts—an availability/resource issue.
P10	main:619-627	X01, `quantum_downgrade`	
P11	main:629-644	The prover cannot be asked	Injecting a forged response when enforcement is on makes the initiator abort—an availability (DoS) issue, not a queryable security property.
P12	main:646-650	R05, R05.11, X01	
P13	rfc7296:6478-6488	X02, `gir_known`	
P14	rfc7296:6490-6494	Outside the scope	Detection differences when EAP is used; the task excludes EAP.
P15	rfc7296:6496-6500	Outside the scope	Key-reuse risk from repeated rekeys; the task excludes rekey.
P16	rfc7296:6502-6514	The prover cannot be asked	Estimating DH group strength.
P17	rfc7296:6516-6519	The prover cannot be asked	The strength of stronger groups and elliptic-curve groups.
P18	rfc7296:6525-6527	No goal stated	States that the IKE framework is extensible; no property that must hold is proposed.
P19	rfc7296:6529-6530	C01	
P20	rfc7296:6532-6538	The prover cannot be asked	Implementation flaws and DoS in the unauthenticated phase.
P21	rfc7296:6540-6542	The prover cannot be asked	The PRF output length limits key strength.
P22	rfc7296:6544-6549	The prover cannot be asked	The quality of the random number generator.
P23	rfc7296:6551-6556	The prover cannot be asked	Prohibiting negotiation of NONE/ENCR_NULL is an algorithm policy; the symbolic model does not model algorithm selection.

Paragraphs	Where	Taken up as	Why not
P24	rfc7296:6558-6563	K01, X03, `offline_guesser`	
P25	rfc7296:6565-6572	Outside the scope	The NAT_DETECTION hash leaks internal addresses; NAT traversal is not within the modeled initial exchange.
P26	rfc7296:6578-6580	Outside the scope	As above, NAT-related attacks.
P27	rfc7296:6582-6606	Outside the scope	EAP methods that do not generate keys lead to a man-in-the-middle; the task excludes EAP.
P28	rfc7296:6608-6612	Outside the scope	Server public-key authentication requirements under EAP.
P29	rfc7296:6614-6619	The prover cannot be asked	Fragment reassembly exhausts resources—an availability issue.
P30	rfc7296:6621-6625	Outside the scope	Admission control and policy configuration, which are deployment rather than protocol exchange.
P31	rfc7296:6631-6633	Outside the scope	As above.
P32 to P39	rfc7296:6637-6696	Outside the scope	Traffic selector authorization and PAD belong to Child SA authorization; the task models only the initial IKE SA establishment.

The catalogue of properties, entry by entry

The properties were drafted from a catalogue of security and privacy properties. The draft left out 34 of its entries, and the analyst went through each of them: examined here as a property, or left out with the reason. Examined: 15. 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	Payload and end-to-end confidentiality	Not applicable	The IKE exchange itself carries no service payload; ESP/AH payload confidentiality is out of scope.
K03	Strong secrecy/indistinguishability of secret values	Not applicable	SK_d and the like are PRF-derived keys, not secret values chosen from a finite set; their unrecoverability is queried by K01 and C08.
K04	Key purpose, direction and session separation	C03	
D01	Message integrity and data origin authenticity	A03	
A01	Peer participation (Aliveness)	A02	
A05	Implicit key authentication	K01, C08	
A06	Explicit key confirmation	A03	
A07	Identity–credential–key binding	A03	
A08	Binding of negotiated parameters to the security context	A03	
R01	Authentication/challenge freshness	A04	
R02	Message-level replay resistance	A04	
R03	At-most-once service operation	Not applicable	The protocol has no billable or consumable service operation.
R04	Message acceptance order/window security	Outside the scope	The message acceptance window depends on retransmission and ordering state, mechanisms outside the initial exchange (and rekey is excluded).

Entry	Property of the catalogue	Examined as	Why not
C02	Key-compromise impersonation resistance (KCI)	X04	
C04	Resilience to ephemeral secret/partial state leakage	C08	
C05	Post-compromise security recovery (PCS)	Outside the scope	Recovery after compromise requires a ratchet or rekey; the task excludes rekey and resumption.
C06	Enhanced authentication after enrollment credential compromise	Not applicable	The protocol does not use enrollment credentials or PAKE.
C07	Resistance to offline dictionary attacks	X03	
Z01	Correctness of authorization enforcement	Not applicable	The protocol produces no protected authorization operation.
Z02	Delegated authority does not expand	Not applicable	No delegation or proxying.
Z03	Binding of authorization to identity/session/operation	Not applicable	No authorization tokens.
Z04	Revocation and expiry take effect	Not applicable	No revocation or quota.
Z05	Cross-tenant/cross-service security isolation	Not applicable	No multi-tenancy or cross-domain assets.
S01	Encrypted nonce uniqueness	Not applicable	Nonces in the model are fresh names; the documents require nonces to be random (s0362), so reuse is an implementation error, not a protocol property.
S02	Retransmission/key reinstallation security	Outside the scope	Retransmission and key reinstallation are handled in RFC 7296 2.4/2.8; the task models only a single fixed initial exchange.
S03	Rollback resistance and state monotonicity	Outside the scope	The initial exchange has no cross-message counter or rollback-capable state; Message ID/window are part of the excluded portion.
S04	Cross-phase security context consistency	Outside the scope	Cross-phase (rekey/resumption) context consistency is excluded by the task.
S05	Termination and invalidation of old state	Outside the scope	SA deletion and the termination exchange are excluded by the task.
P01	Identity anonymity	X02	
P02	Session unlinkability	X02	
P03	Hiding of user/credential existence	X02	
V01	Conditional progress/error recovery	The prover cannot be asked	Progress/recovery is availability, which ProVerif cannot query.
V02	No unexpected deadlock	The prover cannot be asked	The model cannot ask whether a run always terminates or never gets stuck.
V03	Bounded resource abuse/amplification resistance	The prover cannot be asked	Resource abuse/amplification resistance is availability.

The parties and the protected values, one by one

The host listed the 21 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: 11. Without: 10. 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	`ltk_i_leak`	

ID	Party or value	Kind	Attacker	Why none
S02	Responder	party	Everything lost by definition	When the responder's long-term signing private key is held, all authentication goals concerning the responder's identity fail by definition; the documents propose no other property under these goals.
S03	Man-in-the-middle attacker	party	`gir_known`, `quantum_downgrade`	
S04	Peer identity IDi/IDr	value	`gir_known`	
S05	Identity IDi/IDr	value	`gir_known`	
S06	nonce Ni/Nr	value	Nothing gained	Ni/Nr are sent in the clear in IKE_SA_INIT.
S07	DH shared value g^ir	value	`gir_known`	
S08	Negotiated SA parameters	value	Nothing gained	SAi1/SAr1 are sent in the clear in IKE_SA_INIT.
S09	Message ID	value	Nothing gained	The Message ID is sent in the clear in the IKE header.
S10	Messages covered by AUTH	value	Nothing gained	What is signed is the plaintext first and second messages, which the attacker can already read.
S11	Long-term signing private key	value	`ltk_i_leak`, `ltk_both_after`	
S12	PPK	value	Everything lost by definition	When the PPK is held, SK_d and all keys derived from it can be computed directly from public inputs, so all PPK-based goals fail by definition.
S13	SK_d	value	`s_kd_leak`	
S14	SK_pi	value	`s_kd_leak`	
S15	SK_pr	value	`s_kd_leak`	
S16	SK_e/SK_a	value	`s_ke_leak`	
S17	USE_PPK notification	value	Nothing gained	USE_PPK is a plaintext status notification in IKE_SA_INIT.
S18	PPK_IDENTITY notification	value	Nothing gained	PPK_ID is a public identifier from the configuration (s0002); the notification contents themselves are not confidential.
S19	NO_PPK_AUTH notification	value	Nothing gained	NO_PPK_AUTH is authentication data signed by the initiator; the attacker gaining it gains no more than from the message itself.
S20	The session's PPK usage flag	value	Nothing gained	R05.11 requires that the flag cannot be modified, not that its value cannot be read.
S21	Child SA keys	value	`s_kd_leak`	

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		

Attacker	Has	Basis	Where
gir_known	The session's DH shared value g^ir: the attacker chooses the peer's DH exponent itself (the public zpub and nonce n0 in the model) or recovers g^ir with a quantum computer, so it knows all of the session's DH-derived inputs but still does not know the PPK.	s0044/s0045 discuss a quantum attacker able to decrypt the initial IKE SA; s0052 discusses an attacker able to break public-key operations in real time.	main:558, main:567, s0044, s0045, s0052
offline_guesser	An attacker impersonating the responder, who after receiving the initiator's IKE_AUTH can run an offline dictionary search over the PPK values.	s0043: an attacker impersonating the server can run a dictionary search over the PPK even with an ordinary computer.	main:544, s0043
ltk_i_leak	The initiator's long-term signing private key, obtained before or during the session, with the ability to run DH itself.	s0366 discusses long-term key compromise; s0052 discusses public-key operations broken in real time.	rfc7296:6529, s0366, s0052
ltk_both_after	Both parties' long-term signing private keys, obtained only after the session has completed and ephemeral state has been erased.	The perfect forward secrecy definition of s0366-s0367.	rfc7296:6529, s0366, s0367
s_ke_leak	The session's SK_ei/SK_er/SK_ai/SK_ar, from reading one party's memory during the session.	Proposed for C03: check whether PPK-protected keys remain secure after a leak of the handshake keys that PPK does not protect (s0045 says a quantum attacker can decrypt the initial IKE SA).	s0045
s_kd_leak	The session's SK_d/SK_pi/SK_pr, from reading one party's memory during the session.	Proposed for the dual direction of C03: whether an SK_d leak reaches SK_e/SK_a.	s0382
quantum_downgrade	Breaking public-key operations in real time: able to recover the session's g^ir and to forge AUTH for both parties' signatures.	s0052 explicitly gives this attacker and the downgrade attack.	main:619, s0052

ID	network	gir_known	offline_guesser	ltk_i_leak	ltk_both_after	s_ke_leak	s_kd_leak	quantum_downgrad
A02	Holds							
A03	Holds							
A04	Holds							
K01	Holds	Fails						
R05	Holds							
C01					Holds			
C03						Holds	Holds	
C08		Holds						
R05.9	Holds							
C08.10		Holds						
R05.11	No answer							
X01								No answer
X02		Fails						
X03			Fails					
X04				No answer				

Source and targets

Source: 557 units, 185601 bytes, frozen in source-delivery-01.json. Targets: 8 approved (A02, A03, A04, K01, R05, C01, C03, C08); 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-baabb71996eeddc782c1fe6896db

This step as it reports itself: 1 request (1 failed), 0.000 USD, 80284 tokens in and 4772 out, 30 s.

Notes on the prover and the model

- K01: answered in ProVerif, and the analyst names dh_algebra for this target
- C03: 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
- C08.10: answered in ProVerif, and the analyst names dh_algebra for this target
- X01: answered in ProVerif, and the analyst names dh_algebra for this target
- X02: answered in ProVerif, and the analyst names dh_algebra for this target

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

What the analyst named for the targets:

step	targets	features	reason
37	K01, C08, C08.10, X01, X02, C03	plain_primitives, dh_algebra, many_sessions	These answers rest on the Diffie-Hellman exchange, modelled with the single law g^xy = g^yx and an uninterpreted prf/kdf; in the gir_known and quantum_downgrade worlds the attacker is given the DH share as a term. An attack found is an attack on the protocol; a proof is a proof against a weaker attacker than real DH (no other algebraic laws, perfect PRF). All properties are asked over any number of sessions.
37	A02, A03, A04, R05, R05.9, R05.11, C01, X03, X04	plain_primitives, many_sessions, long_flow	These answers rest on signatures, symmetric encryption and an uninterpreted prf/kdf (perfect cryptography, no algebraic laws), on the four-message IKE_SA_INIT/IKE_AUTH flow, and on any number of sessions; the PPK mixing is modelled as an uninterpreted function, so nothing about its strength is proved.

The analyst's closing note

I modelled the fixed IKEv2 IKE_SA_INIT/IKE_AUTH exchange with RFC 8784 PPK mixing (SK_d/SK_pi/SK_pr rerun through prf+ keyed with the PPK, SK_e/SK_a untouched), mandatory and optional configurations, signature-based AUTH, the fallback via NO_PPK_AUTH, and one attacker role per documented ability. All 15 targets have an answer. Proved: mutual authentication and agreement on Ni/Nr/g^ir/PPK mode (A02, A03), injective agreement (A04), secrecy of PPK, long-term keys, SK_d, SK_pi, SK_pr, Child SA keys and SK_e against the network attacker (K01), forward secrecy after both signing keys leak post-session (C01), the separation between SK_e/SK_a and SK_d/SK_pi/SK_pr in both directions (C03), SK_d secret when g^ir is known (C08, C08.10), no non-PPK SA when the PPK is mandatory (R05, R05.9), and accurate PPK markers in both directions (R05.11). The attacks found are all documented: SK_e is readable by an attacker who knows g^ir (s0045), IDi is readable by an active attacker (rfc7296 s0087), and a low-entropy PPK is guessable offline (s0043). So there is no new finding; the report reproduces the known weaknesses with exact conditions. What blocks the rest: three queries (whether the two peers' PPK marks can disagree, whether the optional-configuration downgrade reaches an accepted SA, and whether a leaked initiator

signing key allows impersonating the responder) are inconclusive in ProVerif because its derivation combines two incompatible deliveries at the same IKE_SA_INIT response input; they are recorded as undecided and need Tamarin+ Agent, which models that fallback without this over-approximation. The DH algebra is limited to g^xy = g^yx and all crypto is ideal, so the proofs say nothing about computational or post-quantum strength; no target rests on a changing value, so no bounded model was built.