Security analysis of TLS certificate plus external PSK

This report in another language: 中文.

rfc9973 · ProVerif+ Agent report

Documents	TLS 1.3 Extension for Using Certificates with an External Pre-Shared Key (protocol definition); The Transport Layer Security (TLS) Protocol Version 1.3 (supporting)
Properties examined	17: 14 hold, 2 fail (2 attacks), 1 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

The extension lets a client and a server that share a pre-configured external pre-shared key run a TLS 1.3 handshake in which that key enters the key schedule while the parties still authenticate each other with certificates. The client holds its certificate, its signing key, the external PSK and its identity; the server holds its certificate, its signing key and the same PSK. Each side sends its hello with a fresh key share, the PSK identity and, from the client, a binder; the server adds EncryptedExtensions, a CertificateRequest, its Certificate, a CertificateVerify over the transcript and its Finished, and the client answers with its Certificate, CertificateVerify and Finished. When the exchange ends each side knows the peer's certified identity, the selected PSK identity and that the peer derived the same handshake keys.

The parties:

- **Client**: holds the pair's external PSK and its certificate. Holds: cert(idC, spk(skC)), skC, external PSK, identity pid.
- **Server**: holds the same external PSK and its certificate. Holds: cert(idS, spk(skS)), skS, external PSK, identity pid.

Figure 1 shows the exchange as the documents specify it.

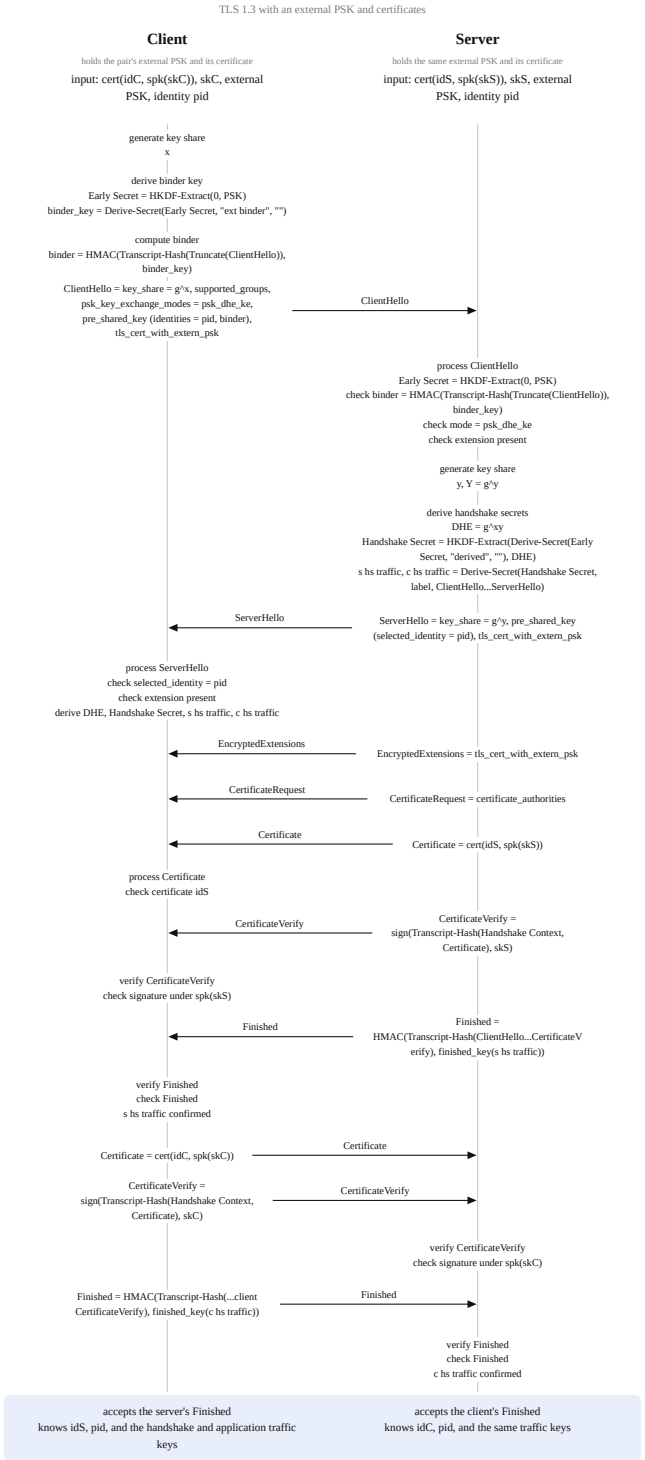


Figure 1. The protocol as specified.

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.

No finding: every property in scope that the documents present as a goal was proved on the model, and the two properties that fail are exactly the weaknesses the specification states. Mutual certificate authentication, transcript agreement, single use of Finished, binding of the identity to the certificate key, enforcement of psk_dhe_ke, secrecy of the PSK and of the session keys, forward secrecy against a later long-term leak, and survival of the session keys when the (EC)DH secret is computable all hold. The failures reproduce what the documents say: a PSK holder who can also compute the (EC)DH secret decrypts recorded traffic, and the plaintext PSK identity lets an observer link connections. One claim, that an accepted binder was produced by a client for the same partial ClientHello, was asked directly and did not terminate. The results rest on a symbolic model that idealises HMAC, HKDF and the AEAD and keeps only the law g^xy = g^yx for (EC)DH. Seventeen targets were examined.

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	**server and client authentication by certificate.** Whenever a client accepts the server identity S at the end of the handshake, there exists a run of the server with that client as its intended peer; the converse (when the server requests and verifies the client certificate) likewise.	no compromise	Holds	high	yes
A03	**agreement on handshake parameters.** Whenever a party accepts, there exists a matching run of the peer with the same role, mutual intended identities, selected PSK identity, key_share group and cipher suite.	no compromise	Holds	medium	no
A04	**no replay of binder or Finished.** Each acceptance corresponds to a distinct peer run; the same ClientHello (including the binder) or the same Finished cannot be accepted twice.	no compromise	Holds	medium	no
A06	**key confirmation by binder and Finished.** At the point of confirmed acceptance, the expected peer has genuinely derived and matched confirmation of the same key for this session (instantiated separately for the binder and for Finished in each direction).	no compromise	No answer	medium	no
A07	**identity bound to certificate key.** The peer identity accepted by the receiver must be supported by a signature under the private key of that identity's certificate, not merely by knowledge of the external PSK.	no compromise	Holds	medium	yes
A08	**extension negotiation cannot be downgraded.** The placement of the extension accepted by the receiver, the accompanying set of extensions, the key exchange mode, the selected_identity and the key_share group all correspond to an approved peer proposal and to the current security context.	no compromise	Holds	medium	no
K01	**secrecy of PSK and traffic keys.** The attacker cannot derive the external PSK or this session's handshake/application traffic keys; the set of legitimate knowers is fixed per instance.	no compromise	Holds	medium	no

ID	Property	Compromise assumed	Result	Value	In the documents
C01	**forward secrecy across the session.** After the session completes and the ephemeral (EC)DH private key is erased, leaking the long-term secrets (the certificate private key, and the external PSK's scope of use, fixed per instance) leaves the traffic keys and payloads of previously completed sessions confidential.	no compromise	Holds	high	no
C08	**PSK protects when (EC)DH is broken.** Under the assumption that the attacker obtains the (EC)DH shared secret, the session secrets remain underivable as long as the external PSK is unknown and key derivation is unbroken.	(EC)DH key agreement (during the session (the attacker has obtained the shared secret))	Holds	high	yes
C03	**PSK knowledge does not enable eavesdropping.** Under the assumption that the external PSK is known to the attacker (or to another member of the group), the session traffic keys and payloads protected by the (EC)DH shared secret remain confidential.	external PSK holder (group member or attacker) (before or during session establishment)	Holds	medium	yes
K04	**session key separation.** The derivation contexts of the external PSK and the resumption PSK are separated; traffic keys of different sessions differ from one another.	no compromise	Holds	low	no
R05	**accepted configuration policy.** The configuration accepted by the receiver satisfies the minimum policy: psk_dhe_ke mode, no resumption PSK, no early_data, certificate authentication present.	no compromise	Holds	low	no
U01	**separate signing key and PSK leaks.** Identity authentication and session key secrecy are checked separately under a signing private key leak and under a PSK leak.	certificate signing private key or external PSK (separately) (before or during session establishment)	Holds	medium	no
U02	**binder binds identity and transcript.** The binder and the PSK identity are bound to this transcript and to the certificate identity.	no compromise	Holds	medium	no
U03	**combined acceptance implies negotiated extension.** Combined authentication acceptance implies that the extension was negotiated on both sides.	no compromise	Holds	medium	no
X01	**recorded traffic secret from CRQC.** If the attacker obtains the external PSK and can also compute the (EC)DH shared secret, it must not be able to derive the session traffic keys of a recorded session.	psk_leak plus a broken (EC)DH key agreement	Attack holds (stated in the documents)	low	yes
X02	**external PSK identity unlinkability.** A passive observer of two connections must not be able to tell, from the external PSK identity carried in plaintext in the ClientHello, that the two connections come from the same client.	no compromise	Attack holds (stated in the documents)	medium	yes

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

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

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: identity authentication and key secrecy when the signing key and the PSK each leak separately	Neither leak alone is enough: a signing-key leak alone does not let the attacker impersonate the server because the client also needs the binder, and it gives no session key; a PSK leak alone gives neither impersonation nor the session keys. Only the PSK together with a broken (EC)DH agreement exposes recorded traffic, which is what the documents state.	U01, A07, X01, C03
Q2: binding of the PSK identity/binder to the certificate identity and the transcript	The binder is tied to the identity and to the partial ClientHello through the transcript, and the two sides are proved to agree on that transcript; a query that speaks about the binder alone did not terminate, so this part rests on transcript agreement rather than on a direct proof.	U02, A03, A06
Q3: without this extension negotiated, a combined authentication result must not be accepted wrongly	Yes: a certificate-authenticated acceptance under an external PSK happens only in a session whose transcript carries the extension marker on both sides, so one side cannot be led to treat a handshake as combined when it was not.	U03, 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:

- The model idealises HMAC, HKDF and the AEAD as ideal keyed functions, so a computational break of these primitives is outside it.
- Diffie-Hellman is modelled with the single law g^xy = g^yx; the attacker has no other algebraic ability in the group, so an attack found is real and a proof holds only against that weaker attacker.
- Certificate validation is a pinned certificate per identity: the chain to a CA, revocation and key usage are not modelled.
- The transcript is an idealised hash of the handshake messages, not the byte-exact encoding of RFC 9846.
- No resumption PSK, no 0-RTT, no PSK provisioning, no group PSK and no HelloRetryRequest are modelled, per the scope of the task.
- One external PSK per pair is modelled; the effect of reusing a PSK across many sessions is not.
- The ephemeral (EC)DH private key is assumed destroyed at the end of the handshake.
- No value that a party keeps and changes across messages is in scope, so no bounded unrolling and no state review were needed.

What the host records:

- No answer for: A06.
- The answers of A02, A03, A04, A06, A07, A08, K01, C01, C08, C03, K04, R05, U01, U02, U03, X01, X02 rest on an operation the model simplifies (Appendix D).
- The prover gave no answer (Appendix C): A06: a06b. 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: a02, A08: r05, K01: k01psk, K04: k04, R05: r05. 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: server and client authentication by certificate

Whenever a client accepts the server identity S when the handshake completes, there exists a server run whose intended peer is that client; the reverse (when the server requests and verifies a client certificate) is the same.

The client accepts a finished handshake only with a server that ran the same transcript and signed it with the key in its certificate, and the server does the same for the client. This holds when the attacker only has the network, when it holds the external PSK, and when it can compute the (EC)DH shared secret. Removing the signature check makes both directions fail, so the result depends on that check.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value high: Mutual authentication is the central promise of this extension, and the proof shows it holds even with the PSK known or the (EC)DH secret computable.
- Stated in the documents: yes. Where the documents say it: s0034, s0035, s0052
- Rests on: Certificate validation is modelled as a pinned certificate per identity; the chain to a CA and key usage are outside TLS and are not modelled.; Signatures are idealised: one can be made only with the private key.; The client certificate is always sent, since the scope fixes certificate authentication on both sides.

A03: agreement on handshake parameters

Whenever one side accepts, there exists a matching peer run with the same roles, mutual intended identities, selected PSK identity, key_share group and cipher suite.

Both sides end with the same transcript, so they agree on the roles, the selected PSK identity, the key shares and the cipher suite. The agreement is carried by the binder, by the signatures and by Finished, all of which cover the transcript.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Agreement on the transcript is what makes every other binding meaningful; the proof holds on an idealised transcript.
- Stated in the documents: no. Source text it rests on: s0031, s0032, s0419
- Rests on: The transcript is an idealised hash of the handshake messages rather than the byte-exact encoding of RFC 9846.; The HelloRetryRequest path is not modelled.

A04: no replay of binder or Finished

Every acceptance corresponds to a different peer run; the same ClientHello (including the binder) or the same Finished cannot be accepted twice.

Neither the server nor the client can be made to accept the same finished handshake twice. The freshness comes from the key share that the transcript covers, not from a counter.

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

- Value medium: Replay of a finished handshake is the first thing to test once a PSK is added, and the extension keeps the protection.
- Stated in the documents: no. Source text it rests on: s0441, s0442
- Rests on: The model keeps no state between messages, so acceptance is bounded only by the transcript and the freshness of the key shares.; The transcript is an idealised hash.

A06: key confirmation by binder and Finished

At the point of confirmed acceptance, the intended peer has genuinely derived the same key for this session and the confirmation matches (the binder and Finished directions are instantiated separately).

The finished messages confirm that the peer derived the same handshake traffic key, which is shown by proving that an accepted Finished has a matching sender, and this survives a PSK leak. The binder side is answered indirectly: the binder is part of the transcript the two sides are proved to agree on, so an accepted binder belongs to a client run of this identity, but a query speaking about the binder alone did not finish.

- Result: No answer; answers what is asked: partly; usable with caveats
- Value medium: Key confirmation is what makes the PSK useful at all, but one of its two claims has no direct answer.
- Stated in the documents: no. Source text it rests on: s0029, s0332, s0441
- Rests on: One claim is answered through transcript agreement, not by a direct query.; HMAC is idealised as a keyed one-way function.
- Open: The direct claim that an accepted binder was computed by a client for the same partial ClientHello was asked on the binder value and did not terminate; the answer rests on transcript agreement instead.

A07: identity bound to certificate key

The peer identity accepted by a party must be backed by the private key of that identity's certificate, not merely by knowledge of the external PSK.

A party that knows the external PSK still cannot be accepted as the server: the client keeps the certificate and the CertificateVerify. The documents require that the PSK not be the sole basis for authentication, and the model confirms it.

- Result: Holds; answers what is asked: yes; usable
- Value medium: It reproduces a requirement of the specification mechanically and shows the extension does not weaken authentication for PSK holders.
- Stated in the documents: yes. Where the documents say it: main:465, s0052, s0058
- Rests on: Certificate validation is a pinned certificate per identity.; The attacker with the PSK is given the PSK of the pair only, not a certificate private key.

A08: extension negotiation cannot be downgraded

The extension occurrences accepted by the receiver, the accompanying extension set, the key exchange mode, selected_identity and the key_share group all correspond to an approved peer proposal and the current security context.

No accepted ClientHello used the plain psk_ke mode: the server refuses a ClientHello whose mode field is not psk_dhe_ke. The combined handshake itself is reached and accepted, so the query is not vacuous.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Downgrade to psk_ke would remove the (EC)DH contribution, so enforcing the mode is the point of the extension.
- Stated in the documents: no. Source text it rests on: s0022
- Rests on: The mode field is a single constant compared by the server; the list of modes a client offers is not modelled.; The HelloRetryRequest path is not modelled.

K01: secrecy of PSK and traffic keys

The attacker cannot derive the external PSK or this session's handshake/application traffic keys; the set of legitimate knowers is fixed per instance.

An attacker on the network learns neither the external PSK nor the handshake traffic keys of a session, because the keys depend on both the PSK and the (EC)DH secret and it has neither.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Session secrecy against the network is the basic expectation, and the answer shows the extension keeps it.
- Stated in the documents: no. Source text it rests on: s0042, s0051, s0060
- Rests on: HMAC and HKDF are idealised; a break of HMAC as a PRF is outside the model.; DH carries only the law g^xy = g^yx, which gives the attacker less than a real group.

C01: forward secrecy across the session

After the session completes and the ephemeral (EC)DH private key has been erased, a later leak of the long-term secrets (the certificate private key, and the per-instance fixed external PSK usage scope) leaves the traffic keys and payloads of previously completed sessions confidential.

Traffic keys of a completed session stay secret when the certificate private keys and the external PSK leak afterwards, because the long-term values do not give the ephemeral (EC)DH secret.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value high: This decides whether an operator can keep a static PSK safely, and it is proved for the single-use case the documents describe.
- Stated in the documents: no. Source text it rests on: s0053, s0054
- Rests on: The ephemeral (EC)DH private key is assumed destroyed once the handshake is over; the leak is placed in a later phase.; One external PSK per pair, with no restriction on how many sessions may use it.

C08: PSK protects when (EC)DH is broken

Under the assumption that the attacker obtains the (EC)DH shared secret, session secrets remain underivable as long as the external PSK is unknown and the key derivation is unbroken.

When the attacker can compute the (EC)DH shared secret but does not know the PSK, the session traffic keys stay out of reach, because the handshake secret needs both inputs. The documents say this and the model confirms it.

- Result: Holds; answers what is asked: yes; usable
- Value high: The claim that the PSK keeps traffic safe from a future quantum attacker is the main motivation of the extension, and here it is checked rather than argued.
- Stated in the documents: yes. Where the documents say it: main:474, s0053

- Rests on: Broken (EC)DH is modelled by a session whose shared secret is public; the rest of the derivation is idealised.; The attacker is not given the PSK of that session.

C03: PSK knowledge does not enable eavesdropping

Under the assumption that the external PSK is known to the attacker (or to other members of the group), the session traffic keys and payloads protected by the (EC)DH shared secret remain confidential.

A party that knows the external PSK but not the (EC)DH shared secret cannot read the session traffic, since the traffic keys need the (EC)DH secret as well. Only a session whose (EC)DH secret is also computable falls, which is the case the documents describe.

- Result: Holds; answers what is asked: yes; usable
- Value medium: This is the argument the documents give for allowing a group PSK, and the model shows where it stops holding.
- Stated in the documents: yes. Where the documents say it: main:524, s0059
- Rests on: The PSK holder is modelled as an attacker with the PSK only; the failing case adds a session whose (EC)DH secret is public.; Group members are not modelled as recording parties in their own right.

K04: session key separation

The derivation contexts of the external PSK and the resumption PSK are separated; the traffic keys of different sessions differ from one another.

A handshake traffic key is used in one transcript only, so two sessions do not share keys. The separation comes from the fresh key shares in each transcript.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: Key separation is a supporting property, and the answer says nothing about the external-versus-resumption label, which the model does not have.
- Stated in the documents: no. Source text it rests on: s0055, s0520
- Rests on: External and resumption PSKs are not separated by a label, since no resumption PSK exists in scope.; The transcript is an idealised hash.

R05: accepted configuration policy

A configuration the receiver accepts satisfies the minimum policy: psk_dhe_ke mode, no resumption PSK, no early_data, certificate authentication present.

An accepted handshake used psk_dhe_ke and required certificate authentication on both sides. The exclusion of resumption PSKs and of early_data is meta: those messages are absent from the model, so no run carrying them can be shown.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value low: The answer confirms enforcement of the mode choice, but the rest of the policy is satisfied by what the model leaves out.
- Stated in the documents: no. Source text it rests on: s0004, s0022, s0023
- Rests on: Resumption PSKs and early_data are left out by the scope of the task, so the corresponding policy items are not tested.; The client's mode list is a single constant.

- Open: The items that no resumption PSK and no early_data appear are true by construction, since the model has neither message; those claims are answered by the model, not by the protocol.

U01: separate signing key and PSK leaks

Check identity authentication and session key secrecy separately under a signing private key leak and under a PSK leak.

A leak of the signing private key alone does not let the attacker impersonate the server, because the client also checks the binder, which needs the PSK; and neither key alone gives a session key. A leak of the PSK alone gives neither impersonation nor the session keys.

- Result: Holds; answers what is asked: yes; usable
- Value medium: It answers which single credential matters for which goal and shows the two factors are genuinely two.
- Stated in the documents: no. Source text it rests on: s0042, s0052, s0058
- Rests on: The two leaks are modelled one at a time, before the session.; One certificate private key per identity, without revocation or expiry.

U02: binder binds identity and transcript

The binder and the PSK identity are bound to this transcript and to the certificate identity.

The binder is covered by the transcript the two sides are proved to agree on, so an accepted binder belongs to a client run with the same PSK identity and the same partial ClientHello, and knowing the PSK does not let the attacker bind a handshake of its own. A direct query on the binder alone gave no answer.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The binder is what ties the PSK to the handshake; the answer is sound but indirect for one of its two claims.
- Stated in the documents: no. Source text it rests on: s0031, s0032
- Rests on: The binding is answered through transcript agreement, not by a direct query on the binder.; HMAC is idealised.
- Open: A query that speaks about the binder value alone, without the rest of the transcript, did not terminate.

U03: combined acceptance implies negotiated extension

Combined authentication acceptance implies the extension was negotiated on both sides.

A certificate-authenticated acceptance under an external PSK happens only inside a session whose transcript carries the extension marker on both sides, so a party cannot be led to accept a combined result for a handshake that did not negotiate the extension.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It rules out the free-rider case in which one side thinks certificates plus PSK were agreed and the other does not.
- Stated in the documents: no. Source text it rests on: s0010, s0012
- Rests on: The extension is an empty flag in the transcript, and the placement rules are modelled by requiring it in both messages.; The refusal a peer sends when the extension is misplaced is not modelled.

X01: recorded traffic secret from CRQC

If the attacker obtains the external PSK and can also compute the (EC)DH shared secret, it must not be able to derive the session traffic keys of a recorded session.

An attacker that holds the external PSK and can also compute the (EC)DH shared secret derives the session traffic key of a recorded session; with only the broken agreement and no PSK it cannot. This is the weakness the documents state for a PSK holder with a quantum computer: the PSK adds no protection to its holder once the key agreement falls.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value low: The result is what the documents already say; it is reproduced here as a check that the model sees it.
- Stated in the documents: yes. Where the documents say it: main:530, s0059, s0053
- Rests on: A broken (EC)DH agreement is modelled by publishing one session's shared secret.; The attack is about recorded traffic and needs both the PSK and the broken agreement.

X02: external PSK identity unlinkability

A passive observer of two connections must not be able to tell, from the external PSK identity carried in plaintext in the ClientHello, that the two connections come from the same client.

The external PSK identity travels in plaintext in the ClientHello, so an observer of two connections of the same client can tell they come from the same device. ProVerif cannot prove the identity unlinkable and produces the test the attacker makes.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value medium: The documents warn of it, and the model shows the exact leak point, which is useful for the mitigation they recommend.
- Stated in the documents: yes. Where the documents say it: main:555, main:566, s0555, s0561
- Rests on: The ClientHello always carries the single PSK identity of the pair, as the scope fixes one pre-configured PSK.; ProVerif never calls a non-interference false; the answer is that it cannot be proved, with a trace of the attacker's test.

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		a02	3 of 4		4 proved
A03	proved		a02	checked		2 proved
A04	proved		a04	checked		2 proved
A06	proved		a02	checked		2 proved, 1 undecided
A07	proved		a07	checked		1 proved
A08	proved		r05			1 proved, 1 reachable
K01	proved		k01psk	1 of 2		2 proved
C01	proved		keysec_lt	checked		1 proved
C08	proved		c08	checked		1 proved
C03	proved		keysec_psk	checked		1 proved
K04	proved		k04			1 proved
R05	proved		r05			1 proved
U01	proved		u01auth	checked		3 proved
U02	proved		a02	checked		2 proved
U03	proved		a02r	checked		1 proved
X01	proved		c08	checked	x01 (holds_in_source)	1 proved, 1 attack
X02	none				track (holds_in_source)	1 attack

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	Server authentication: whenever the client accepts the server Finished, a server run exists with the same transcript; nothing beyond the network.	`network`	proved	a02	checked
A02	Server authentication when the attacker can compute the (EC)DH shared secret but does not hold the PSK.	`dh_broken`	proved	a02	none
A02	Client authentication: whenever the server accepts the client Finished, a client run exists with the same transcript; nothing beyond the network.	`network`	proved	a02r	checked
A02	Server authentication when the attacker holds the external PSK of the pair.	`psk_leak`	proved	a02	checked
A03	Agreement on the transcript (roles, PSK identity, key shares, cipher suite) in the client's view.	`network`	proved	a02	checked

target	claim	attacker	answer	query	control
A03	Agreement on the transcript in the server's view.	`network`	proved	a02r	checked
A04	The client accepts a given server Finished transcript at most once (no replay to the client).	`network`	proved	a04	checked
A04	The server accepts a given client Finished (and its binder-bearing ClientHello) transcript at most once.	`network`	proved	a04r	checked
A06	Finished confirms the handshake traffic key: an accepting party has a peer that computed the same traffic secret.	`network`	proved	a02	checked
A06	The binder confirms a shared binder key for the selected external PSK identity: an accepted binder was computed by a client for the same identity and partial ClientHello.	`network`	undecided	a06b	
A06	Finished confirms the handshake traffic key even when the attacker holds the external PSK: an accepted Finished has a peer that derived the same traffic secret.	`psk_leak`	proved	a02	checked
A07	With the external PSK known, the client still accepts only a server run with the same transcript: the PSK is not the sole basis for authentication.	`psk_leak`	proved	a07	checked
A08	A combined handshake (extension, psk_dhe_ke, binder, key share) is reached and accepted by the server.	`network`	reachable	a08	
A08	No accepted ClientHello carried the plain psk_ke mode: servers select psk_dhe_ke for the initial handshake.	`network`	proved	r05	none
K01	The external PSK cannot be derived by an attacker that only has the network.	`network`	proved	k01psk	none
K01	The handshake traffic keys of a session stay secret from an attacker that only has the network.	`network`	proved	keysec_net	checked
C01	Traffic keys of a completed session stay secret from an attacker that obtains the certificate private keys and the external PSK after the session.	`longterm_after_session`	proved	keysec_lt	checked
C08	Session traffic keys stay secret from an attacker that can compute the (EC)DH shared secret, as long as it does not know the external PSK.	`dh_broken`	proved	c08	checked
C03	Session traffic keys stay secret from a party that knows the external PSK but not the (EC)DH shared secret.	`psk_leak`	proved	keysec_psk	checked
K04	A handshake traffic key is used in one transcript only: keys of different sessions are distinct.	`network`	proved	k04	none
R05	An accepted combined handshake used psk_dhe_ke; a resumption PSK and early_data are absent from the model and certificate authentication is required.	`network`	proved	r05	none

target	claim	attacker	answer	query	control
U01	A leak of the server's signing private key alone does not let the attacker impersonate the server: the client also checks the binder, which needs the PSK.	`sigkey_leak`	proved	u01auth	checked
U01	A leak of the external PSK alone gives no impersonation and no session traffic key.	`psk_leak`	proved	a07	checked
U01	A leak of the signing private key alone gives no session traffic key.	`sigkey_leak`	proved	keysec_sk	checked
U02	An accepted binder belongs to a transcript that a client of this identity produced (the binder is covered by the agreed transcript).	`network`	proved	a02	checked
U02	An accepted binder belongs to a client transcript of this identity even when the attacker holds the external PSK, so knowing the PSK does not let it bind a handshake of its own to the identity.	`psk_leak`	proved	a02	checked
U03	A certificate-authenticated acceptance under an external PSK happens only in a session whose transcript carries the extension marker on both sides.	`network`	proved	a02r	checked
X01	With the external PSK and a computable (EC)DH shared secret, the attacker derives the session traffic key of a recorded session; this is what the documents state a PSK holder with a CRQC can do.	`psk_leak`	attack (holds_in_source)	x01 (the control of c08)	
X01	With only the (EC)DH shared secret and not the external PSK, the recorded session traffic keys stay secret: this attacker alone does not reach them.	`dh_broken`	proved	c08	checked
X02	A passive observer sees the external PSK identity in plaintext in the ClientHello and can link two connections of the same client.	`network`	attack (holds_in_source)	track	

The prover never says that a property of this kind (the attacker cannot tell which of two values is used) is false. It gave no proof and a trace on which the attacker tells the values apart, and the attack is recorded on that trace: track.

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-20261114T220941, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 46, prover 797 s, wall 17 min, cost 0.156 USD.

Context: the host removed stale parts from the conversation 1 time(s), 442515 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 135 proofs run, 107 decided, 28 without an answer, 784 s, 33 answers recorded.

What was analysed

Protocol	TLS certificate plus external PSK
Version	rfc9973
Analysed	A fixed set of high-entropy external PSKs preconfigured independently per peer, certificate authentication and psk_dhe_ke; the full binder, CertificateVerify and Finished; it is made explicit that the external PSK is not a resumption PSK; 0-RTT, PSK provisioning and group PSKs are excluded.
Not analysed	The symbolic model does not prove the computational security or the quantum resistance of the algorithms; attack traces hold only under explicit assumptions and are not directly equivalent to a specification flaw. Every property must have a reachability/positive control.
Source text	555 numbered passages (150113 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	4	Extension Overview
main	5	Certificate with External PSK Extension
main	5.1	Companion Extensions
main	5.2	Authentication
main	5.3	Keying Material
main	7	Security Considerations
rfc9846	4	Handshake Protocol
rfc9846	4.1	The Transcript Hash
rfc9846	4.2	Key Exchange Messages
rfc9846	4.2.1	Cryptographic Negotiation
rfc9846	4.2.2	Client Hello
rfc9846	4.2.3	Server Hello
rfc9846	4.2.4	Hello Retry Request
rfc9846	4.3	Extensions
rfc9846	4.3.1	Supported Versions

Document	Section	Title
rfc9846	4.3.2	Cookie
rfc9846	4.3.3	Signature Algorithms
rfc9846	4.3.4	Certificate Authorities
rfc9846	4.3.5	OID Filters
rfc9846	4.3.6	Post-Handshake Certificate-Based Client Authentication
rfc9846	4.3.7	Supported Groups
rfc9846	4.3.8	Key Share
rfc9846	4.3.8.1	Diffie-Hellman Parameters
rfc9846	4.3.8.2	ECDHE Parameters
rfc9846	4.3.9	Pre-Shared Key Exchange Modes
rfc9846	4.3.10	Early Data Indication
rfc9846	4.3.11	Pre-Shared Key Extension
rfc9846	4.3.11.1	Ticket Age
rfc9846	4.3.11.2	PSK Binder
rfc9846	4.3.11.3	Processing Order
rfc9846	4.4	Server Parameters
rfc9846	4.4.1	Encrypted Extensions
rfc9846	4.4.2	Certificate Request
rfc9846	4.5	Authentication Messages
rfc9846	4.5.1	Certificate
rfc9846	4.5.1.1	OCSP Status and SCT Extensions
rfc9846	4.5.1.2	Certificate Selection
rfc9846	4.5.1.3	Receiving a Certificate Message
rfc9846	4.5.2	Certificate Verify
rfc9846	4.5.3	Finished
rfc9846	4.6	End of Early Data
rfc9846	4.7	Post-Handshake Messages
rfc9846	4.7.1	New Session Ticket Message
rfc9846	4.7.2	Post-Handshake Authentication
rfc9846	4.7.3	Key and Initialization Vector Update
rfc9846	7	Cryptographic Computations
rfc9846	7.1	Key Schedule
rfc9846	7.2	Updating Traffic Secrets
rfc9846	7.3	Traffic Key Calculation
rfc9846	7.4	Asymmetric Shared Secret Calculation
rfc9846	7.4.1	Finite Field Diffie-Hellman

Document	Section	Title
rfc9846	7.4.2	Elliptic Curve Diffie-Hellman
rfc9846	7.5	Exporters

The purpose of the protocol as the source step located it in the documents: a fixed set of high-entropy external PSKs preconfigured independently for each peer, certificate authentication, and psk_dhe_ke; full binder, CertificateVerify and Finished; an explicit statement that an external PSK is not a resumption PSK; exclusion of 0-RTT, PSK provisioning, and group PSKs.

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	server and client authentication by certificate	the client is assured that the server named by its certificate ran the handshake; the server is assured of the client's certified identity when it verifies the client certificate	Authentication depends on a signature validated with the certificate public key and is unchanged by the PSK (s0007, s0034, s0035, s0052).	s0007, s0034, s0035, s0052
A03	agreement on handshake parameters	both parties end with the same transcript: same roles, identities, selected PSK identity, key_share group, cipher suite	Key schedule, binder and signatures bind the transcript (s0006, s0031, s0032, s0037, s0038).	s0006, s0031, s0032, s0037, s0038
A04	no replay of binder or Finished	the server accepts each binder-bearing ClientHello and each client Finished at most once; the client each server Finished at most once	Binder and Finished bind the PSK and the current handshake (s0032, s0441, s0442); transcripts carry fresh key shares (s0003, s0010).	s0003, s0010, s0032, s0441, s0442
A06	key confirmation by binder and Finished	an accepting party has a peer that derived the same binder key and the same handshake traffic key for this session	Binder validation confirms a shared key for the identity (s0029, s0032); Finished authenticates the handshake and the computed keys (s0441, s0443).	s0029, s0032, s0441, s0443
A07	identity bound to certificate key	the identity an accepting party settles on is supported by a signature under the certified private key, not by PSK knowledge	The external PSK MUST NOT be the sole basis for authentication; PSK knowledge does not enable impersonation (s0052, s0058).	s0052, s0058
A08	extension negotiation cannot be downgraded	completion only when the extension was in ClientHello and ServerHello with its companion extensions and psk_dhe_ke	Companion extensions required; extension not in ServerHello unless in ClientHello; servers select psk_dhe_ke (s0003, s0005, s0010, s0022).	s0003, s0005, s0010, s0022
K01	secrecy of PSK and traffic keys	the external PSK and the handshake and application traffic keys are kept from an attacker on the network	Protect the external PSK (s0042); confidentiality goals continue to hold (s0051, s0060).	s0042, s0051, s0060
C01	forward secrecy across the session	traffic keys and payload of a completed session stay secret from an attacker that obtains the long-term secrets afterwards	Ephemeral key destroyed before the session ends; forward-secrecy discussion (s0053, s0054, s0055).	s0053, s0054, s0055

ID	Name	About	What the documents say	Where
C08	PSK protects when (EC)DH is broken	session secrets stay underivable when the attacker can compute the (EC)DH shared secret but not the PSK	The attacker cannot derive session secrets even if it computes the (EC)DH shared secret, as long as it lacks the PSK (s0053).	s0053
C03	PSK knowledge does not enable eavesdropping	session traffic keys and payload kept from parties holding the external PSK but not the (EC)DH secret	Confidentiality depends on the (EC)DH shared secret, so PSK knowledge does not enable eavesdropping (s0059).	s0059
K04	session key separation	traffic keys of different sessions are distinct; the external PSK is kept apart from resumption PSKs	PSK bound to one hash and KDF (s0008, s0061); binder label separates external from resumption (s0520); single-use PSK keeps session separation (s0055).	s0008, s0061, s0520, s0055
R05	accepted configuration policy	an accepted combined handshake used psk_dhe_ke, no resumption PSK, no early_data, certificate authentication	psk_dhe_ke required (s0022); resumption PSK must abort (s0023); early_data must abort (s0004); PSK not sole basis (s0052).	s0022, s0023, s0004, s0052
U01	separate signing key and PSK leaks	what a leak of the signing private key alone and of the PSK alone gives on authentication and session key secrecy	PSK not sole basis for authentication (s0052); PSK compromise exposes content only to a CRQC (s0042); PSK knowledge does not enable impersonation (s0058).	s0042, s0052, s0058
U02	binder binds identity and transcript	an accepted binder is one the client computed, for the same identity and the same partial ClientHello	The binder confirms a shared key for the identity over a partial transcript up to the identities field (s0029, s0031, s0032).	s0029, s0031, s0032
U03	combined acceptance implies negotiated extension	a certificate-authenticated acceptance under an external PSK happens only in a session where the extension was negotiated by both	The extension is in the ServerHello only if it was in the ClientHello and the server held an offered PSK (s0010, s0012, s0018).	s0010, s0012, s0018
X01	recorded traffic secret from CRQC	recorded session traffic of the client and server	The documents state the opposite: with the external PSK and a CRQC an attacker decrypts stored traffic like a group member (s0059). Added as the property that weakness breaks.	s0059
X02	external PSK identity unlinkability	the external PSK identity carried in plaintext in the ClientHello	An identity reused across connections lets an observer track clients; plaintext PSK identities aid tracking (s0555, s0561). Added as the property that weakness breaks.	s0555

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

The security statements of the documents, paragraph by paragraph

The host listed the 26 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: 23. Not taken: 3. 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:392-392	A02, A03, K01, C01, C08, C03, `network`, `psk_leak`, `dh_broken`, `longterm_after_session`	
P02	main:394-399	A02, A08, R05, U03	
P03	main:401-405	K01, C01, C08, C03, `psk_leak`	
P04	main:407-410	X02	

Paragraphs	Where	Taken up as	Why not
P05	main:412-421	The prover cannot be asked	s0044: PRNG strength for PSK generation; entropy is not a trace property
P06	main:423-437	The prover cannot be asked	s0045: symmetric key size and Grover's algorithm; computational strength
P07	main:439-441	The prover cannot be asked	s0046: minimum length of external PSKs; key size is not a trace property
P08	main:443-452	A02, A03, A06, A07, K01, C08, C03	
P09 to P11	main:454-458	K01, C08	
P12	main:460-463	K01, C08, C03	
P13	main:465-472	A07, A02, `psk_leak`	
P14	main:474-487	C08, `dh_broken`	
P15 to P18	main:489-512	X01, C01, K04, `psk_leak`, `dh_broken`, `longterm_after_session`	
P19	main:514-522	A07, C03, `psk_leak`	
P20	main:524-532	C03, X01, `psk_leak`, `dh_broken`	
P21	main:534-538	K01, C08, C03	
P22	main:540-547	K04	
P23	main:551-553	X02	
P24	main:555-559	X02	
P25	main:561-568	X02	
P26	rfc9846:4676-4677	A02, A03, K01, C01, C08, `network`, `psk_leak`, `sigkey_leak`, `longterm_after_session`, `dh_broken`	

The parties and the protected values, one by one

The host listed the 39 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: 24. Without: 15. 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	客户端	party	`sigkey_leak`	
S02	服务器	party	`sigkey_leak`	
S03	PSK 组内其他成员	party	`psk_leak`	
S04	PSK 组内成员	party	`psk_leak`	
S05	证书身份	value	Nothing gained	certificate identities are public
S06	external PSK 身份	value	Nothing gained	the PSK identity is carried in plaintext (s0561)
S07	所选 external PSK identity	value	Nothing gained	selected_identity is a plaintext index
S08	key_share 组	value	Nothing gained	the key_share group is a public negotiation parameter

ID	Party or value	Kind	Attacker	Why none
S09	cipher suite	value	Nothing gained	the cipher suite is a public negotiation parameter
S10	transcript	value	Nothing gained	the transcript is public handshake data; its integrity is the target
S11	binder	value	`psk_leak`	
S12	Finished	value	`psk_leak`	
S13	binder_key	value	`psk_leak`, `longterm_after_session`	
S14	finished_key	value	`psk_leak`	
S15	握手流量密钥	value	`dh_broken`, `longterm_after_session`	
S16	证书私钥	value	`sigkey_leak`, `longterm_after_session`	
S17	external PSK	value	`psk_leak`, `longterm_after_session`	
S18	tls_cert_with_extern_psk 扩展	value	Nothing gained	the extension is an empty plaintext flag
S19	psk_key_exchange_modes	value	Nothing gained	the mode field is plaintext
S20	selected_identity	value	Nothing gained	selected_identity is plaintext
S21	key_share	value	Nothing gained	the key shares are public; the shared secret is dh_broken
S22	Early Secret	value	`psk_leak`, `longterm_after_session`	
S23	Handshake Secret	value	`dh_broken`	
S24	Main Secret	value	`dh_broken`, `psk_leak`	
S25	握手与应用流量密钥	value	`psk_leak`, `dh_broken`, `longterm_after_session`	
S26	已完成会话的流量密钥与载荷	value	`longterm_after_session`	
S27	临时 (EC)DH 私钥	value	`dh_broken`	
S28	会话秘密	value	`dh_broken`, `psk_leak`	
S29	(EC)DH 共享秘密	value	`dh_broken`	
S30	会话流量密钥与载荷	value	`psk_leak`	
S31	会话流量密钥	value	`dh_broken`, `psk_leak`, `longterm_after_session`	
S32	resumption PSK	value	Outside the scope	resumption PSKs are excluded by the task and absent from the model
S33	密钥交换模式	value	Nothing gained	the mode field is plaintext; R05 asks that an accepted run used psk_dhe_ke
S34	PSK 类型	value	Nothing gained	the PSK type is a label, not a value an attacker holds
S35	early_data 标志	value	Nothing gained	the early_data flag is plaintext and must be refused (s0004); R05 asks that no accepted run carried it

ID	Party or value	Kind	Attacker	Why none
S36	证书认证	value	`sigkey_leak`	
S37	证书签名私钥	value	`sigkey_leak`, `longterm_after_session`	
S38	会话密钥	value	`dh_broken`, `psk_leak`, `longterm_after_session`	
S39	PSK identity	value	Nothing gained	the PSK identity is plaintext; the unlinkability goal is what its exposure breaks

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		
`psk_leak`	the external PSK of the pair, before or during the session; nothing else	The specification does not require that the PSK is known only by client and server, and discusses group knowledge and theft (s0058, s0059).	s0058, s0059
`sigkey_leak`	the certificate signing private key of the server (and separately of the client), before or during the session; not the PSK	Authentication depends on a signature validated with the certificate public key, so that private key protects authentication (s0007, s0034, s0052).	s0007, s0034, s0052
`longterm_after_session`	the certificate private keys and the external PSK, obtained after a session completed	The forward-secrecy discussion assumes the ephemeral key is gone and asks what a later compromise of long-term material gives (s0053, s0054, s0055).	s0053, s0054, s0055
`dh_broken`	the (EC)DH shared secret of a session, computable from the public key shares; not the external PSK	The documents assume the (EC)DH key agreement is broken while the PSK stays unknown (s0053, s0054).	s0053, s0054

ID	`network`	`psk_leak`	`sigkey_leak`	`longterm_after_session`	`dh_broken`
A02	Holds	Holds			Holds
A03	Holds				
A04	Holds				
A06	No answer	Holds			
A07		Holds			
A08	Holds				
K01	Holds				
C01				Holds	
C08					Holds
C03		Holds			

ID	network	psk_leak	sigkey_leak	longterm_after_session	dh_broken
K04	Holds				
R05	Holds				
U01		Holds	Holds		
U02	Holds	Holds			
U03	Holds				
X01		Fails			Holds
X02	Fails				

Source and targets

Source: 555 units, 150113 bytes, frozen in source-delivery-01.json. Targets: 15 approved (A02, A03, A04, A06, A07, A08, K01, C01, C08, C03, K04, R05, U01, U02, U03); 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: 批准源文包 protocol-2ae552d7044c652af21e3e484e2a

This step as it reports itself: 1 requests (1 failed), 0.000 USD, 68847 tokens in and 6161 out, 40 s.

Notes on the prover and the model

- A02: answered in ProVerif, and the analyst names dh_algebra for this target
- A03: answered in ProVerif, and the analyst names dh_algebra for this target
- A04: answered in ProVerif, and the analyst names dh_algebra for this target
- A06: answered in ProVerif, and the analyst names dh_algebra for this target
- A07: answered in ProVerif, and the analyst names dh_algebra for this target
- A08: answered in ProVerif, and the analyst names dh_algebra for this target
- K01: answered in ProVerif, and the analyst names dh_algebra for this target
- C01: 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
- C03: answered in ProVerif, and the analyst names dh_algebra for this target
- K04: answered in ProVerif, and the analyst names dh_algebra for this target
- R05: answered in ProVerif, and the analyst names dh_algebra for this target
- U01: answered in ProVerif, and the analyst names dh_algebra for this target
- U02: answered in ProVerif, and the analyst names dh_algebra for this target
- U03: 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 algebra (xor, dh_algebra, other_algebra) gives the attacker less than the real operation: a proof there is a proof against a weaker attacker, an attack there is an attack. A ProVerif model that leaves out a check on a value that changes (changing_state) lets parties accept more: a proof there holds, an attack there may be nothing but the missing check. Tamarin+ Agent models these as they are.

What the analyst named for targets:

step	targets	features	reason
40	A02, A03, A04, A06, A07, A08, K01, C01, C08, C03, K04, R05, U01, U02, U03, X01, X02	dh_algebra, many_sessions, plain_primitives	The key schedule uses one constructor per labelled derivation over an idealised HMAC/HKDF and AEAD, and the (EC)DH shared secret is modelled with the single law g^xy = g^yx only, so an attacker has less algebra than in the real group. Every property is asked for any number of sessions, and the reveals model the leaked long-term values of the respective attackers.

The analyst's closing note

The normal run exists and seventeen targets were answered on model.pv. Proved: mutual certificate authentication (also with the external PSK known or the (EC)DH shared secret computable), transcript agreement in both directions, single use of Finished in both directions, key confirmation by Finished, the identity being bound to the certificate key, enforcement of psk_dhe_ke, secrecy of the PSK and of the session keys against the network, forward secrecy when the certificate keys and the PSK leak after the session, survival of the session keys when the (EC)DH secret is computable, no session-key reuse, and that separate leaks of the signing key and of the PSK each give neither impersonation nor the session keys. Controls that must fail were falsified where a variant exists (a02ctl, a02rctl, keysec_ctl, c01ctl, x01). Two properties fail, and both are weaknesses the documents themselves state, so they are recorded and no finding is new: a PSK holder who can also compute the (EC)DH shared secret decrypts recorded traffic (X01, s0059), and the plaintext PSK identity links connections (X02, s0555/s0561). What blocks the rest: the direct claim that an accepted binder was computed by a client for the same partial ClientHello did not terminate in ProVerif (recorded as undecided; transcript agreement carries it indirectly), and the (EC)DH algebra is reduced to g^xy = g^yx plus an idealised shared-secret function, so Tamarin+ Agent, which models the algebra, would settle that last part. All results are symbolic: HMAC, HKDF and the AEAD are ideal keyed functions.