Security analysis of AF inference server composed flow

This report in another language: 中文.

19.2.0 · ProVerif+ Agent report

Documents	5G System; Application Function Artificial Intelligence/Machine Learning (AI/ML) Services; Stage 3 (protocol definition); 5G System; Network Data Analytics Services; Stage 3 (supporting)
Properties examined	5: 3 hold, 2 fail (2 attacks)
Findings	2; known results reproduced in Appendix B: 1

Contents

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

1 Protocol and scope

The Naf_Inference service lets an NF service consumer subscribe at an Application Function that acts as a VFL server, and be notified of inference results; the Nnwdaf_VFLInference service lets that AF, as a VFL server, subscribe at a Network Data Analytics Function acting as VFL client and receive its VFL inference results. In the composed flow the NWDAF consumer creates an inference subscription at the trusted AF, the AF creates a VFL inference subscription at the NWDAF VFL client for the UEs the consumer named, and after one round of client results the AF reports the inference result back to the consumer. Each request is an HTTP request on the service-based interface, authenticated and encrypted by the TLS session between the two NFs, and authorised by an OAuth2 access token that the NRF issues to the consumer for the service scope.

The parties:

- **AF**: trusted Application Function acting as VFL server. Holds: inferAnaSubs from the NWDAF; trained VFL model and its vflCorrId; OAuth2 token for naf-inference and nnwdaf-vflinference.
- **NWDAF (consumer)**: NF service consumer of Naf_Inference. Holds: the analytics subscription (inferAnaSubs, notifCorreId, notifUri); OAuth2 token for naf-inference.
- **NWDAF (VFL client)**: NWDAF acting as VFL client. Holds: its local training data; the VFL model of the vflCorrId; OAuth2 token for nnwdaf-vflinference.

Figure 1 shows the exchange as the documents specify it.

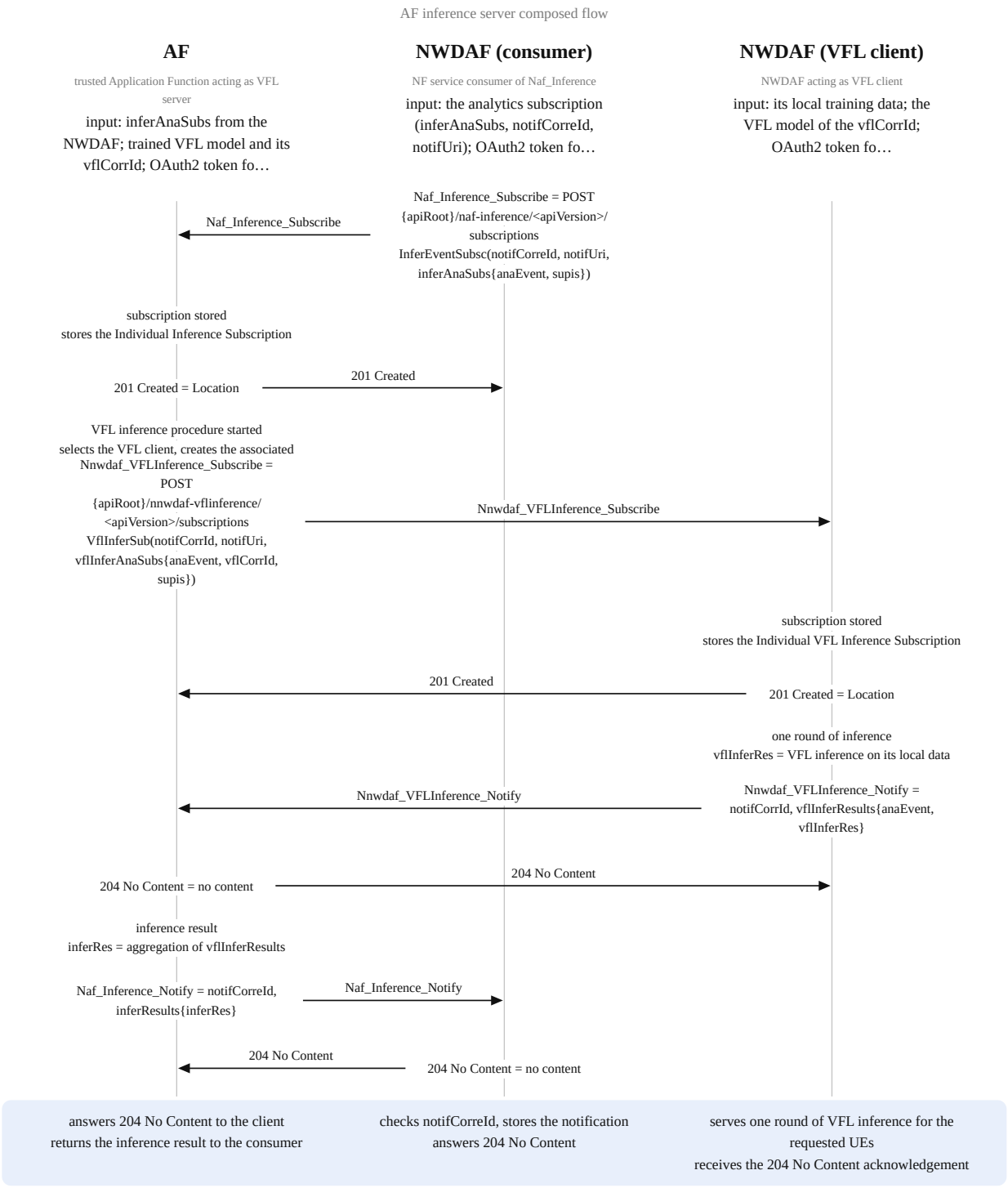


Figure 1. The protocol as specified.

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

The analysis has 2 findings: what follows is not stated in the documents. For 1 of them it is unclear whether the documents state it. 1 result that the documents already state is not counted and is in Appendix B.

The peer-to-peer authentication of the exchange and the service-scope token check hold in the model, but the composed flow does not bind an inference result to the subscription's current target, and the VFL client does not bind a VFL subscription to a participant of the VFL process. The first is a weakness the documents do not state: an inference round computed for the UE the consumer asked for before an update can be reported as the result of the updated subscription. The second is shown for a legitimate NF that holds a valid token for the service-wide scope. Of the five targets, two are answered by proofs and one is split between proofs and an attack; two are answered by an attack, one of them with a proof for its remaining claim. All proofs carry a control, and the proofs rest on the TLS session and on the access token being idealised, so they do not cover token expiry, revocation or a compromised endpoint.

The attack figures use the columns of Figure 1. Red is what the attacker does. Yellow marks a message that a party sends although the specified exchange does not have it at that point, and the outcome that the attack breaks.

2.1 An in-flight VFL round is reported as the result of the updated inference subscription

Kind	Weakness of the protocol as specified
New	Yes: the documents do not state it
Value	high
Properties	A08 (section 3)
Source text	s0055, s0059, s0086, s0593, s0599
Shown by	ProVerif attack query `a08`; Figure 2

What happens. The NWDAF consumer creates an inference subscription for one UE, then updates the same Individual Inference Subscription to another UE. The AF passes the update to the NWDAF VFL client under the same notifCorrId, but a round the client had already computed for the first UE set is still in flight. The notification carries only the notifCorrId and the VFL inference results, and the result carries no UE identity and no subscription version, so the AF cannot tell the two rounds apart and the consumer accepts the old result as the answer to its updated subscription.

What the attacker needs. A subscription that the consumer updates while a round of VFL inference for the previous target is still being computed or delivered; no credential and no action of the network attacker is needed.

What the attacker gains. The NWDAF consumer, and anything downstream of its analytics, attributes a VFL inference result about one UE set to the UE set it asked for after the update. A VFL correlation ID shared by several subscriptions of the same analytics event makes the confusion reachable without any attacker at all.

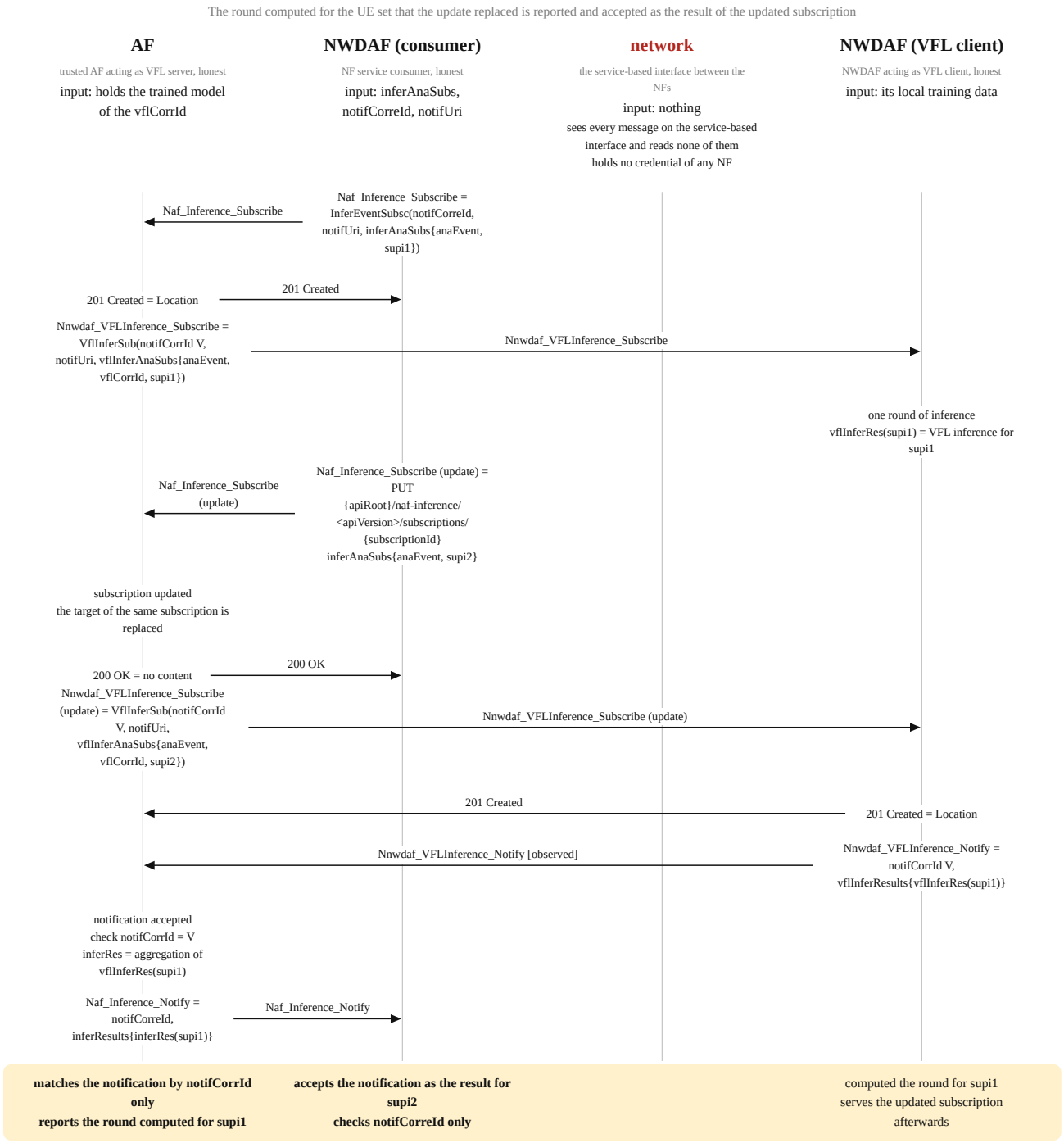


Figure 2. The round computed for the UE set that the update replaced is reported and accepted as the result of the updated subscription. Property A08; attack query a08.

Proposed change. Carry the round in the notification: add to VflInferNotif, and to InferNotif, an identifier of the request round (for example the vflCorrId together with the UE set the round was computed for, or a round counter kept with the subscription), and require the AF to report and the consumer to accept a notification only if that round matches the subscription state after the update. The same field in both notifications is what lets an editor check the rule.

Check of the change with ProVerif. Not checked: The change adds a field to two data structures and a comparison in two parties; it could not be built and run on a model within the time of this analysis.

2.2 A legitimate NF with only the service-wide scope joins another VFL process at the NWDAF client

Kind	Weakness of the protocol as specified
New	Unclear: the documents state it in part, or could not be searched
Value	medium
Properties	Z03 (section 3)
Source text	s0355, s0359, s0590, s0631, s0419
Shown by	ProVerif attack query `z03b`; Figure 3

What happens. A second NWDAF holds a valid access token for the naf-inference and nnwdaf-vflinference services, as any legitimate NF does. It sends Nnwdaf_VFLInference_Subscribe for the VFL correlation ID of the AF's VFL process and for the same UE, with its own notification URI. The NWDAF VFL client accepts the subscription and sends it the VFL inference results, because the service defines one scope for the whole API and the client is given no rule that ties a subscription to the participants of that VFL correlation.

What the attacker needs. A legitimate NF of the same trust domain with its own TLS credentials and its own access token for the nnwdaf-vflinference scope, and the VFL correlation ID of the VFL process, which is carried in the subscription it observes or guesses.

What the attacker gains. The NWDAF VFL client runs the VFL inference for that VFL correlation ID and UE set and returns the intermediate VFL inference results to it, so an NF that is not a participant of the VFL process obtains inference output about UEs of another function's subscription.

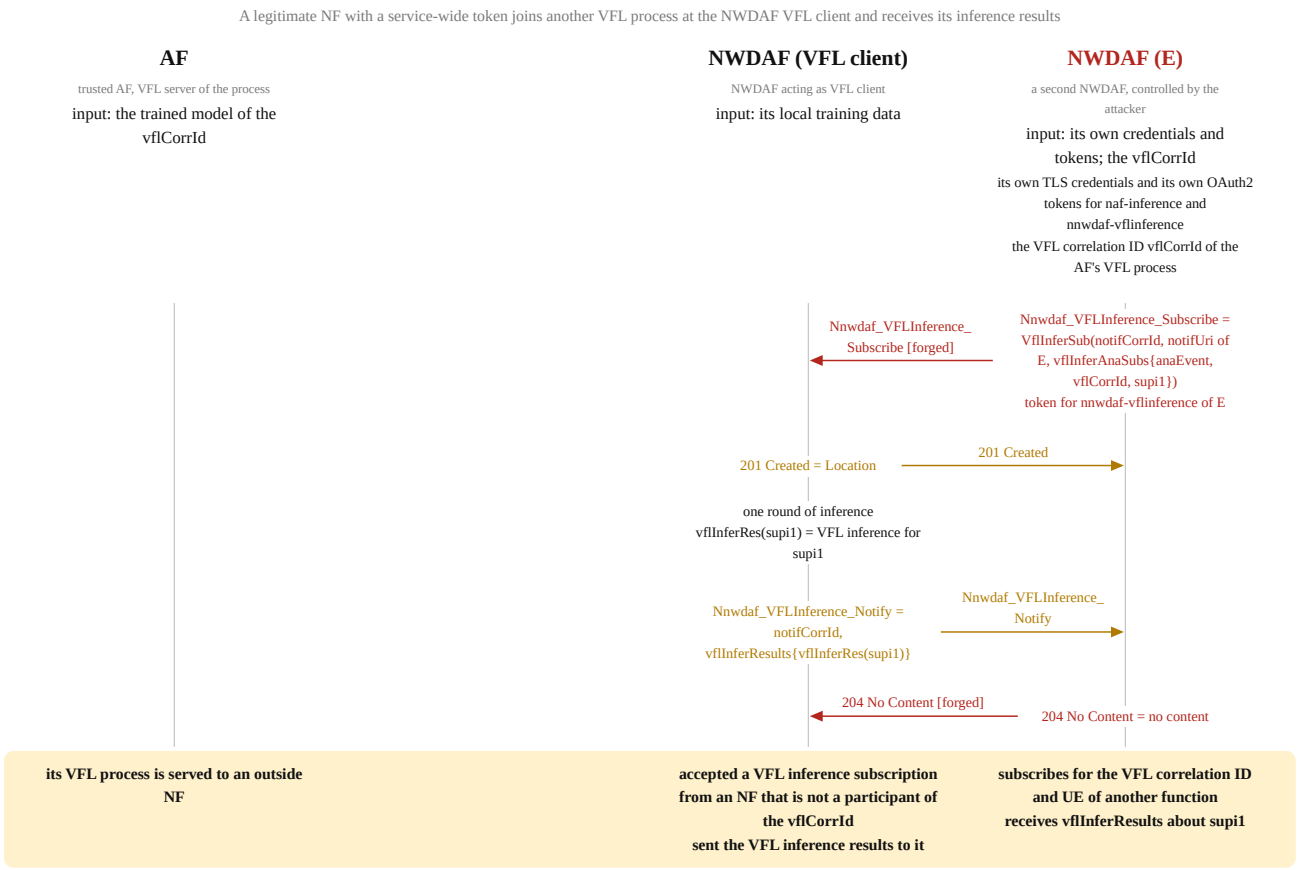


Figure 3. A legitimate NF with a service-wide token joins another VFL process at the NWDAF VFL client and receives its inference results. Property Z03; attack query z03b.

Proposed change. Require the NWDAF VFL client to check, before accepting an Nnwdaf_VFLInference_Subscribe, that the requesting consumer is one of the VFL participants of the vflCorrId and that the requested UE set is within what that VFL process was set up for, and reject with 403 otherwise.

Check of the change with ProVerif. Not checked: The model would need the participant list of the VFL process as an input the client compares against, and that list is defined in TS 29.552, outside the source units.

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	**Peer authentication of the exchange.** Whenever the AF accepts an inference subscription request from an NWDAF (or the NWDAF accepts the AF's VFL inference subscription request), the peer has previously run the corresponding flow with this party as its communication target.	no compromise	Holds	medium	yes
A03	**Correlation and field agreement.** The AF and the NWDAF agree on the notifCorreId, vflCorrId, anaEvent and notification target URI of the same subscription: every acceptance has a matching peer run, and these fields agree.	no compromise	Holds	medium	yes
A08	**Notification matches the current subscription.** The notifCorreId, vflCorrId and notification target URI carried by a notification sent by the AF correspond to the NWDAF's currently valid subscription (including an updated subscription), not to an old subscription that has been replaced or deleted.	no compromise	Fails (1 attack)	high	no
Z01	**Protected operations need a valid token.** Every subscription creation/update/deletion and every notification submission corresponds to a valid token issued by the authorization server that covers the scope of that service; a request without a valid token produces no protected operation.	no compromise	Holds	low	yes
Z03	**Token bound to its consumer and context.** A token can only be used for the consumer identity and subscription context for which it was granted; a token of one consumer cannot be used for another consumer's subscription or notification target.	no compromise	Fails (1 attack)	medium	unclear

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: Correlation between the outer inference subscription and the inner VFL result	The inner VFL notification carries only the notifCorrId and the VFL inference results, and the result carries the analytics event but neither the UE set nor a round identifier, so the AF relates a round to the outer subscription through the notifCorrId alone; after an update of the outer subscription the same notifCorrId admits the round computed for the old UE set, which is then reported as the answer to the updated subscription.	A08, A03
Q2: Binding of the three roles: client/AF/final consumer	The consumer and the AF authenticate each other, and so do the AF and the VFL client, so a party cannot act under another NF's identity; but the VFL client does not bind the requester to the VFL process, so a third NF holding a valid token can subscribe for the same VFL correlation ID and receive the results.	A02, Z03
Q3: Old inner results are not wrongly bound to an updated outer subscription	No: the client's results are not marked with the round or the UE set they belong to, so a result produced for the target that the update replaced is accepted and reported under the updated subscription; the consumer cannot tell it from a fresh result.	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 TLS session between two NFs is modelled as a pairwise authenticated-encryption envelope with a fixed key; record-level replay protection, connection setup and certificate handling of TLS 1.3 are not modelled, so a message replayed inside a session is possible in the model where TLS would refuse it.
- The OAuth2 access token is modelled as an unforgeable value bound to subject, audience and scope; its signature, expiry, revocation and the NRF token request are not modelled, so Z01 says nothing about an expired or revoked token.
- One session per role is modelled (one consumer, one AF, one VFL client, one further NWDAF); behaviour with many concurrent subscriptions of the same consumer, or with a consumer that is also a VFL client, is not covered.
- The untrusted AF and the NEF path of clause 5.5.2.2.2 are left out, as the task asks; the VFL training branch and the AF without a selected client (s0043, s0048) are left out as well.
- Deletion of a subscription is not modelled, because the tables of the prover only grow; the in-flight round after an update stands for it, and an old notification after a deletion is not asked.
- The content of vflInferRes is vendor-specific and not standardised (s0599); the model treats a result as an opaque value of the client, so it cannot say anything about the correctness of the inference itself.
- Availability, timing, the maximum response time indicated by vflNotOnTimeInd, and the strength of the cryptographic algorithms are outside what the prover can be asked.
- The injective reading of the agreement properties (one acceptance per peer run) was not asked separately; repeated subscription requests are not distinguished.

What the host records:

- The answers of A08 rest on an operation the model simplifies (Appendix D).

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

Appendix A: Reading of each property

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

A02: Peer authentication of the exchange

Whenever the AF accepts an inference subscription request from an NWDAF (or the NWDAF accepts the AF's VFL inference subscription request), the peer has previously run the corresponding flow with this party as its communication target.

The AF acts on a subscription request only if the NWDAF it names actually sent that request to it, and the NWDAF VFL client does the same for the VFL subscription. The consumer accepts a notification only from the AF. The guarantee is the one the SBI baseline asks for and rests entirely on the TLS session being in place.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: A proof of the mutual authentication the baseline requires, reproduced on the composed flow rather than a new result.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2871, s0040, s0086, s0419
- Rests on: The SBI hop is a pairwise authenticated-encryption envelope with a fixed key; TLS connection setup, certificates and record replay protection are not modelled.; One peer per role: the model has one consumer, one AF and one VFL client.; The control given for each proof shows the acceptance event occurs with a conclusion that does not hold, so the implication is not empty.
- Next step: Model a hop without the SBI envelope, to show that the agreement is exactly what the TLS session provides.

A03: Correlation and field agreement

The AF and the NWDAF agree on the notifCorreId, vflCorrId, anaEvent and notification target URI of the same subscription: every acceptance has a matching peer run, and these fields agree.

The notifCorreId of the inner VFL subscription, the resource the update applies to and the analytics target, and the notifCorreId of the outer subscription are all carried and compared, so each acceptance belongs to a request the peer sent with those fields. The fields that are not carried, the UE set of the round and any version of the subscription, are what the notification binding lacks.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: Confirms that the correlation identifiers of both APIs are bound as the data structures intend, and delimits what the notification cannot bind.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2875, s0237, s0243, s0584, s0590, s0593
- Rests on: Field agreement is proved for the fields the data structures carry; no check of fields the messages do not carry can be proved.; One session per role, so agreement is shown for one subscription lifecycle.
- Next step: Extend the data model with a round identifier and check whether the notification binding A08 asks for can then be enforced.

A08: Notification matches the current subscription

The notifCorreId, vflCorrId and notification target URI carried by a notification sent by the AF correspond to the NWDAF's currently valid subscription (including an updated subscription), not to an old subscription that has been replaced or deleted.

The notification the consumer accepts may report a round computed for the UE set that its update replaced, because neither notification carries the UE set or a round identifier; the AF has no field to check and the consumer has none to check either. No credential is needed: the round is a genuine message of the VFL client that was in flight when the update was made.

- Result: Fails (1 attack); answers what is asked: yes; usable
- Value high: A protection the composed flow is meant to give is missing, and the documents do not state the weakness.
- Stated in the documents: no. Source text it rests on: s0055, s0059, s0086, s0593, s0599
- Rests on: The consumer updates its subscription while a round is in flight; this ordering is what the model allows and what the source does not exclude.; Deletion is not modelled; the update stands for the change of the subscription state.; The result value is an opaque term, so the model says nothing about its content beyond the fact that it was computed for a UE set.
- Open: Whether the same confusion arises after a deletion, or between two subscriptions that share a VFL correlation ID, is not modelled.
- Next step: Model the unsubscription as well, and check whether an old notification is accepted after the consumer deletes the subscription.

Z01: Protected operations need a valid token

Every subscription creation/update/deletion and every notification submission corresponds to a valid token issued by the authorization server that covers the scope of that service; a request without a valid token produces no protected operation.

Neither the AF nor the NWDAF VFL client performs a subscription operation for a consumer that does not hold a token issued to it for the corresponding service scope; a request without such a token does not reach the protected operation at all. This is the OAuth2 authorisation the baseline prescribes, and the control shows that the property is the token check and not something else.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: The property follows from the OAuth2 authorisation the documents require; the analysis confirms the check is effective and gives it a control.
- Stated in the documents: yes. Where the documents say it: rfc6749:2870, ts-33-501:2867, s0272, s0273, s0628, s0629
- Rests on: The token is an unforgeable value bound to subject, audience and scope; its signature, expiry and revocation are not modelled.; A request of an NF without any token is modelled by the unchecked copy of the AF, which is the control; the real AF rejects it.
- Open: Expiry, revocation and the token request are not modelled, so the answer covers an unforgeable, unexpired token only.
- Next step: Model the NRF token request and token expiry, to ask the same property for an expired token.

Z03: Token bound to its consumer and context

A token can only be used for the consumer identity and subscription context for which it was granted; a token of one consumer cannot be used for another consumer's subscription or notification target.

A token cannot be presented for another consumer, and the network attacker cannot read the inference results. What fails is the subscription context: a legitimate NF with its own valid token subscribes at the NWDAF VFL client for the VFL correlation ID and UE of another function's VFL process and receives the results, because one scope covers the whole service and the client is given no participant check.

- Result: Fails (1 attack); answers what is asked: partly; usable with caveats
- Value medium: A legitimate NF obtains VFL inference results about UEs of another subscription; the documents state the service-wide scope but not this consequence.
- Stated in the documents: unclear. Source text it rests on: s0275, s0631, s0590, s0355, s0419
- Rests on: The service-wide scope of s0631 is modelled as a single scope value; per-resource authorisation, if an operator configured it, is not in the model.; The VFL correlation ID is assumed known or guessable by another NF; the model gives it to the second NWDAF directly.; The participant rules of the VFL signalling flows of TS 29.552 are not modelled.
- Open: Whether the VFL client is required to check VFL participation is defined in TS 29.552, which is not in the source units.; Token expiry and revocation are outside the model.
- Next step: Read the VFL participant rules of TS 29.552 and decide whether the client is already required to check them; if not, the client needs a participant check.

Appendix B: Known results reproduced

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

B.1 Peer authentication and the token check hold for every hop of the composed flow

Kind	Guarantee confirmed
Stated in the documents	ts-33-501:2871, main:10528
Value	medium
Properties	A02, A03, Z01 (section 3)
Source text	s0040, s0086, s0419

What happens. With the SBI TLS session and a token bound to its subject, audience and scope in place, the AF and the NWDAF VFL client act only on requests of the peer they name, the AF takes only notifications of the client it subscribed at, and the consumer takes only notifications of the AF for its own notifCorreId.

What the attacker gains. An implementer or an editor knows which properties hold on the transport the SBI baseline prescribes and which of them do not depend on it.

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	checked		2 proved
A03	proved		a03a	checked		3 proved
A08	none				a08 (holds_in_source)	1 attack
Z01	proved		z01a	checked		6 proved
Z03	proved		z01a	checked	z03b (depends_on_ambiguity)	3 proved, 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	The AF accepts a Naf_Inference subscription request (with that notifCorreId and notifUri) only if the NWDAF named as consumer ran the subscription flow and sent that request to the AF; the attacker is the one on the network, it holds no NF credential.	`network`	proved	a02	checked
A02	The NWDAF VFL client accepts an Nnwdaf_VFLInference subscription request only if the NF it names as consumer sent that very request (same notifCorrId, notifUri and UE target); the attacker is the one on the network.	`network`	proved	a02b	checked
A03	The AF accepts an Nnwdaf_VFLInference notification only for the notifCorrId it registered at the VFL client and only with the result payload that client sent; the attacker is the one on the network.	`network`	proved	a03a	checked
A03	The AF accepts an update of the inference subscription only for the Individual Inference Subscription resource the consumer created and only with the analytics target (UE) the consumer asked for, so the notifCorreId and the target agree between consumer and AF; the attacker is the one on the network.	`network`	proved	a03b	checked
A03	The consumer accepts an inference notification only when it carries its own notifCorreId and was produced by the AF for that subscription; the attacker is the one on the network.	`network`	proved	a03c	checked
A08	The notification the consumer accepts reports a result computed for the subscription's current analytics target (the UE after the update), not the round in flight for the target the update replaced; the attacker is the one on the network.	`network`	attack (holds_in_source)	a08	
Z01	The AF creates, updates or answers an inference subscription only for a consumer that holds an access token issued to it for the naf-inference service scope; the attacker is the one on the network, it holds no token and no NF credential.	`network`	proved	z01a	checked
Z01	The NWDAF VFL client accepts a VFL inference subscription only from a consumer that holds an access token issued to it for the nnwdaf-vflinference service scope; the attacker is the one on the network.	`network`	proved	z01b	checked

target	claim	attacker	answer	query	control
Z01	The AF creates, updates or answers an inference subscription only for a consumer holding a token issued to that very consumer for the naf-inference scope, so a legitimate NF with its own valid token cannot get the AF to act for it under another identity; the attacker is a legitimate NF with its own credentials and tokens.	`dishonest_nf`	proved	z01a	checked
Z01	The NWDAF VFL client accepts a VFL inference subscription only for a consumer holding a token issued to that very consumer for the nnwdaf-vflinference scope; the attacker is a legitimate NF with its own credentials and tokens.	`dishonest_nf`	proved	z01b	checked
Z01	A token that was issued to another consumer cannot be used at the AF, because the AF requires the token to be the one issued to the identity that presented it; the attacker has read a token of another NF from storage.	`stolen_token`	proved	z01a	checked
Z01	A token that was issued to another consumer cannot be used at the NWDAF VFL client, because the client requires the token to be the one issued to the identity that presented it; the attacker has read a token of another NF from storage.	`stolen_token`	proved	z01b	checked
Z03	A token is usable only by the consumer it was issued to: the AF and the NWDAF VFL client accept an operation only for the identity the token was issued to, so a legitimate NF cannot present its own token as another consumer's; the attacker is a legitimate NF with its own credentials and tokens.	`dishonest_nf`	proved	z01a	checked
Z03	A token read from storage cannot be presented for another consumer: the receiver requires the token issued to the identity that presented it; the attacker has read a token of another NF.	`stolen_token`	proved	z01a	checked
Z03	A consumer may use its token only for the subscription context it owns: another NF holding a valid token for the same service-wide scope must not obtain VFL inference results for a VFL correlation ID and UE it is not a participant of; the attacker is a legitimate NF with its own credentials and tokens.	`dishonest_nf`	attack (depends_on_ambiguity)	z03b	
Z03	The VFL inference result about a UE is not obtainable by the attacker on the network from the service-based interface messages; it reaches only the consumer over its TLS session with the producer.	`network`	proved	z03net	checked

Appendix D: How the analysis was run

An LLM analyst builds the model of the protocol from the documents and puts each property to ProVerif; every result in section 3 is an answer of the prover.

ProVerif+ Agent session session-20261114T224019, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 60, prover 3 s, wall 14 min, cost 0.132 USD.

ProVerif: 69 proofs run, 67 decided, 2 without an answer, 2 s, 23 answers recorded.

What was analysed

Protocol	AF inference server composed flow
Version	19.2.0
Analysed	The trusted AF acts as VFL server; the NWDAF consumer plus one NWDAF VFL client are modelled with a pre-trained model; after receiving a subscription, the AF may send Nnwdaf_VFLInference_Subscribe and reports once it has received one round of client results. Both the inner and outer sets of ID mappings are fully retained; NEF and the branch that triggers training when the model is missing are excluded.
Not analysed	The symbolic model does not prove the computational security or the post-quantum security of the algorithms; attack traces hold only under explicit assumptions and are not directly equivalent to specification vulnerabilities. Every property requires a reachability/positive control. Fixed at Rel-19. The external network attacker and the malicious legitimate NF holding its own credentials are distinguished separately; TLS authenticated encryption is modelled per endpoint, and a legitimate NF cannot read another NF's TLS session. OAuth2 subject/audience/scope/expiry and TLS identity binding follow the attached baseline; a service scope does not automatically equal resource ownership. Monotonic tables cannot model real deletion, and the finite-state lifecycle must be explicitly delimited. The target is the specified API sub-flow, not full 5G security.
Source text	677 numbered passages (377484 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	5.5	Naf_Inference Service
main	5.5.1	Service Description
main	5.5.1.1	Overview
main	5.5.1.2	Service Architecture
main	5.5.1.3	Network Functions
main	5.5.1.3.1	Application Function (AF)
main	5.5.1.3.2	NF Service Consumers
main	5.5.2	Service Operations
main	5.5.2.1	Introduction
main	5.5.2.2	Naf_Inference_Subscribe
main	5.5.2.2.1	General
main	5.5.2.2.2	Inference Subscription Creation

Document	Section	Title
main	5.5.2.2.3	Inference Subscription Update
main	5.5.2.3	Naf_Inference_Unsubscribe
main	5.5.2.3.1	General
main	5.5.2.3.2	Inference Subscription Deletion
main	5.5.2.4	Naf_Inference_Notify
main	5.5.2.4.1	General
main	5.5.2.4.2	Inference Notification
main	6.4	Naf_Inference Service API
main	6.4.1	Introduction
main	6.4.2	Usage of HTTP
main	6.4.2.1	General
main	6.4.2.2	HTTP standard headers
main	6.4.2.2.1	General
main	6.4.2.2.2	Content type
main	6.4.2.3	HTTP custom headers
main	6.4.3	Resources
main	6.4.3.1	Overview
main	6.4.3.2	Resource: Inference Subscriptions
main	6.4.3.2.1	Description
main	6.4.3.2.2	Resource Definition
main	6.4.3.2.3	Resource Standard Methods
main	6.4.3.2.3.1	POST
main	6.4.3.2.4	Resource Custom Operations
main	6.4.3.3	Resource: Individual Inference Subscription
main	6.4.3.3.1	Description
main	6.4.3.3.2	Resource Definition
main	6.4.3.3.3	Resource Standard Methods
main	6.4.3.3.3.1	PUT
main	6.4.3.3.3.2	PATCH
main	6.4.3.3.3.3	DELETE
main	6.4.3.2.4_2	Resource Custom Operations
main	6.4.4	Custom Operations without associated resources
main	6.4.5	Notifications
main	6.4.5.1	General
main	6.4.5.5	Inference notification
main	6.4.5.5.1	Description

Document	Section	Title
main	6.4.5.5.2	Target URI
main	6.4.5.5.3	Standard Methods
main	6.4.5.5.3.1	POST
main	6.4.6	Data Model
main	6.4.6.1	General
main	6.4.6.2	Structured data types
main	6.4.6.2.1	Introduction
main	6.4.6.2.2	Type InferEventSubsc
main	6.4.6.2.3	Type InferEventSubscPatch
main	6.4.6.2.4	Type InferAnaSub
main	6.4.6.2.5	Type InferNotif
main	6.4.6.2.6	Type InferResult
main	6.4.6.3	Simple data types and enumerations
main	6.4.6.3.1	Introduction
main	6.4.6.3.2	Simple data types
main	6.4.7	Error Handling
main	6.4.7.1	General
main	6.4.7.2	Protocol Errors
main	6.4.7.3	Application Errors
main	6.4.8	Feature negotiation
main	6.4.9	Security
main	6.4.10	HTTP redirection
ts-29-520	4.11	Nnwdaf_VFLInference Service
ts-29-520	4.11.1	Service Description
ts-29-520	4.11.1.1	Overview
ts-29-520	4.11.1.2	Service Architecture
ts-29-520	4.11.1.3	Network Functions
ts-29-520	4.11.1.3.1	Network Data Analytics Function (NWDAF)
ts-29-520	4.11.1.3.2	NF Service Consumers
ts-29-520	4.11.2	Service Operations
ts-29-520	4.11.2.1	Introduction
ts-29-520	4.11.2.2	Nnwdaf_VFLInference_Subscribe service operation
ts-29-520	4.11.2.2.1	General
ts-29-520	4.11.2.2.2	Subscription for VFL inference event notifications
ts-29-520	4.11.2.2.3	Update subscription for event notifications
ts-29-520	4.11.2.2.4	Partial update subscription for event notifications

Document	Section	Title
ts-29-520	4.11.2.3	Nnwdaf_VFLInference_Unsubscribe service operation
ts-29-520	4.11.2.3.1	General
ts-29-520	4.11.2.3.2	Unsubscribe from VFL inference notifications
ts-29-520	4.11.2.4	Nnwdaf_VFLInference_Notify service operation
ts-29-520	4.11.2.4.1	General
ts-29-520	4.11.2.4.2	Notification about subscribed event
ts-29-520	5.10	Nnwdaf_VFLInference Service API
ts-29-520	5.10.1	Introduction
ts-29-520	5.10.2	Usage of HTTP
ts-29-520	5.10.2.1	General
ts-29-520	5.10.2.2	HTTP standard headers
ts-29-520	5.10.2.2.1	General
ts-29-520	5.10.2.2.2	Content type
ts-29-520	5.10.2.3	HTTP custom headers
ts-29-520	5.10.3	Resources
ts-29-520	5.10.3.1	Resource Structure
ts-29-520	5.10.3.2	Resource: NWDAF VFL Inference Subscriptions
ts-29-520	5.10.3.2.1	Description
ts-29-520	5.10.3.2.2	Resource definition
ts-29-520	5.10.3.2.3	Resource Standard Methods
ts-29-520	5.10.3.2.3.1	POST
ts-29-520	5.10.3.2.4	Resource Custom Operations
ts-29-520	5.10.3.3	Resource: Individual VFL Inference Subscription
ts-29-520	5.10.3.3.1	Description
ts-29-520	5.10.3.3.2	Resource definition
ts-29-520	5.10.3.3.3	Resource Standard Methods
ts-29-520	5.10.3.3.3.1	PUT
ts-29-520	5.10.3.3.3.2	PATCH
ts-29-520	5.10.3.3.3.3	DELETE
ts-29-520	5.10.3.3.3.4	GET
ts-29-520	5.10.3.3.4	Resource Custom Operations
ts-29-520	5.10.4	Custom Operations without associated resources
ts-29-520	5.10.5	Notifications
ts-29-520	5.10.5.1	General
ts-29-520	5.10.5.2	VFL Inference Event Notification
ts-29-520	5.10.5.2.1	Description

Document	Section	Title
ts-29-520	5.10.5.2.2	Operation Definition
ts-29-520	5.10.6	Data Model
ts-29-520	5.10.6.1	General
ts-29-520	5.10.6.2	Structured data types
ts-29-520	5.10.6.2.1	Introduction
ts-29-520	5.10.6.2.2	Type VflInferSub
ts-29-520	5.10.6.2.3	Type VflInferSubPatch
ts-29-520	5.10.6.2.4	Type VflInferAnaSub
ts-29-520	5.10.6.2.5	Type VflInferNotif
ts-29-520	5.10.6.2.6	Type VflInferReq
ts-29-520	5.10.6.2.7	Type VflInferResult
ts-29-520	5.10.6.3	Simple data types and enumerations
ts-29-520	5.10.6.3.1	Introduction
ts-29-520	5.10.6.3.2	Simple data types
ts-29-520	5.10.6.3.3	Enumeration: VflTermCause
ts-29-520	5.10.7	Error handling
ts-29-520	5.10.7.1	General
ts-29-520	5.10.7.2	Protocol Errors
ts-29-520	5.10.7.3	Application Errors
ts-29-520	5.10.8	Feature negotiation
ts-29-520	5.10.9	Security

The purpose of the protocol as the source step located it in the documents: the trusted AF is the VFL server; it adds an NWDAF VFL client to the NWDAF consumer; the model is pre-trained, and once the AF receives the subscription it can send Nnwdaf_VFLInference_Subscribe and report after receiving one round of client results. The full mapping of both the inner and outer ID sets is preserved; NEF is excluded, as is the branch that triggers training when the model is missing.

The properties against the documents

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

ID	Name	About	What the documents say	Where
A02	Peer authentication of the exchange	The AF is assured that a Naf_Inference subscription request comes from the NWDAF that it names as consumer; the NWDAF VFL client is assured that an Nnwdaf_VFLInference subscription request comes from the AF that runs the composed flow; the consumer is assured that the notification comes from the AF.	The documents define the operations as HTTP requests between an NF service consumer and the AF as NF service producer (s0004, s0032, s0350, s0411) and leave authentication to the SBI baseline, which requires mutual authentication between consumer and producer (TS 33.501 5.9.2.1) and validation of every incoming message (TS 33.501 5.9.2.1).	ts-33-501:2871, s0004, s0021, s0079, s0086, s0322, s0350, s0411, s0419
A03	Correlation and field agreement	For one subscription the AF and the NWDAF agree on notifCorreId, vflCorrId, the analytics event and the notification target URI; every acceptance corresponds to a peer run that sent exactly these fields.	The data structures make notifCorreId, notifUri, vflInferAnaSubs (with vflCorrId and anaEvent) mandatory in the request and require the same notifCorrId in the notification (s0237, s0243, s0584, s0590, s0593, s0419); TS 33.501 requires each NF to validate incoming messages against the protocol specification and network state.	ts-33-501:2875, s0237, s0243, s0246, s0584, s0590, s0593, s0364, s0419
A08	Notification matches the current subscription	The notification the AF sends to the consumer carries the notifCorreId, vflCorrId and target URI of the NWDAF's currently valid subscription, and reports a result computed for that subscription's current analytics target, not for a subscription that an update has replaced.	The AF shall update the associated Nnwdaf_VFLInference subscriptions when the consumer updates its subscription (s0059) and shall notify through the notifUri of the corresponding Individual Inference Subscription resource (s0086); the notification itself carries only notifCorreId and results (s0246, s0593), so the binding has to come from the subscription the AF holds.	main:10677, s0032, s0055, s0059, s0061, s0086, s0243, s0419, s0590, s0593
Z01	Protected operations need a valid token	The AF performs no inference subscription creation, update or notification for a consumer, and the NWDAF VFL client accepts no VFL inference subscription, unless the request carries an OAuth2 access token that the authorization server issued to that consumer for the corresponding service scope.	Access to both APIs may be authorised by OAuth2 with the client credentials grant, the NRF being the authorization server, and the consumer shall obtain a token before consuming the service (s0272, s0273, s0628, s0629); TS 33.501 requires service requests to be authorized (5.9.2.1) and tokens to be unforgeable (RFC 6749 10.3).	rfc6749:2870, ts-33-501:2867, s0272, s0273, s0275, s0628, s0629, s0631
Z03	Token bound to its consumer and context	A token is usable only by the consumer it was issued to, and only for the subscription and notification target that consumer owns; another NF holding a token for the same service-wide scope cannot act for it or receive its inference results.	Both APIs define a single scope for the entire service and no scope at resource or operation level (s0275, s0631), so the token carries no subscription-level right; RFC 6749 notes that OAuth2 itself gives the resource server no way to check that a token was issued to the presenting client (10.3).	rfc6749:2879, main:10528, s0275, s0631, s0237, s0246, s0584, s0593

The analyst added no property: the drafted list covers what the documents state.

The security statements of the documents, paragraph by paragraph

The host listed the 398 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: 24. Not taken: 374. Which paragraphs are listed follows from the layout of the standard; where each went is the analyst's judgement.

Paragraphs	Where	Taken up as	Why not
P001 to P002	rfc6749:2778-2786	No goal stated	Introduction to the OAuth2 security considerations; states no requirement about this protocol.
P003 to P006	rfc6749:2790-2823	Z01, `stolen_token`, `dishonest_nf`	
P007 to P010	rfc6749:2827-2850	Z03, Z01, `stolen_token`, `dishonest_nf`	
P011 to P013	rfc6749:2859-2872	Z01, `stolen_token`	
P014	rfc6749:2874-2877	Z03	
P015	rfc6749:2879-2881	Z03	
P016 to P021	rfc6749:2885-2917	Outside the scope	Refresh tokens; the client credentials grant used by the 5GC has none.
P022 to P025	rfc6749:2921-2943	Outside the scope	Authorization codes; not used with the client credentials grant.
P026 to P029	rfc6749:2947-2980	Outside the scope	Redirection URI manipulation; no redirection in the client credentials grant or in these APIs.
P030 to P033	rfc6749:2984-3003	Outside the scope	Resource owner password credentials; not used.
P034 to P036	rfc6749:3013-3028	A02, `network`	
P037 to P039	rfc6749:3032-3041	The prover cannot be asked	Guessing of client credentials; NFs hold machine credentials and TLS keys, not user-chosen secrets, and whether such a credential is guessable is a question about the strength of a secret, which the prover cannot be asked.
P040 to P042	rfc6749:3045-3066	Outside the scope	Phishing of end users; no end user takes part.
P043 to P048	rfc6749:3070-3110	Outside the scope	Cross-site request forgery; no browser-based flow.
P049 to P050	rfc6749:3119-3138	Outside the scope	Clickjacking; no browser-based flow.
P051	rfc6749:3142-3146	A03	
P052	rfc6749:3148-3150	No goal stated	General advice on input validation; states no property of this protocol beyond A03.
P053 to P054	rfc6749:3154-3167	Outside the scope	Open redirectors; no redirection in this flow.
P055 to P062	rfc6749:3170-3210	Outside the scope	Implicit grant; not used by the 5GC profile.
P063	ts-33-501:2491-2491	Outside the scope	Bidding down of algorithms; the two APIs define no feature or algorithm negotiation in the selected clauses (empty feature tables, s0270, s0626).
P064 to P072	ts-33-501:2495-2511	Outside the scope	UE authentication, serving network authentication and UE authorization; about the access network, not an AF inference subscription.
P073 to P074	ts-33-501:2515-2517	Outside the scope	Keys of the access network.

Paragraphs	Where	Taken up as	Why not
P075 to P213	ts-33-501:2523-2853	Outside the scope	User data and signalling protection at the radio and access layers, secure storage of subscription credentials and the requirements on gNB, AMF, SEAF, AUSF, UDM and UDR; none of these parties takes part in the modelled AF flow.
P214	ts-33-501:2859-2859	`dishonest_nf`	
P215 to P219	ts-33-501:2865-2873	A02, `network`	
P220	ts-33-501:2875-2875	A03	
P221 to P227	ts-33-501:2879-2891	Outside the scope	NRF discovery and registration; the NRF is the token issuer, but its own procedures are not modelled or selected.
P228 to P234	ts-33-501:2895-2907	Outside the scope	NEF requirements; the task excludes the NEF path for untrusted AFs.
P235 to P241	ts-33-501:2911-2923	Outside the scope	SCP and indirect communication; the flow is modelled as direct SBI hops.
P242 to P254	ts-33-501:2929-2953	Outside the scope	Inter-PLMN interconnection and SEPP; no roaming is part of this flow.
P255 to P314	ts-33-501:2957-3081	Outside the scope	SEPP, roaming intermediaries and IPUPS; not part of the modelled flow.
P315 to P318	ts-33-501:3085-3091	Outside the scope	Originating PLMN ID header; roaming metadata outside the modelled flow.
P319 to P337	ts-33-501:3097-3141	No goal stated	Signalling monitoring, security visibility and configurability; operational measures, not properties of the protocol messages.
P338 to P369	ts-33-501:3149-3215	The prover cannot be asked	Ciphering and integrity algorithm identifiers and algorithm selection; the strength of the primitives is outside what a symbolic prover can be asked.
P370 to P398	ts-33-501:3219-11901	Outside the scope	Requirements on 5G-RG, NSSAAF, N3, EAP-TLS access, IAB, MBSF and NEF; none of these parties or interfaces takes part in the modelled flow.

The catalogue of properties, entry by entry

The properties were drafted from a catalogue of security and privacy properties. The draft left out 37 of its entries, and the analyst went through each of them: examined here as a property, or left out with the reason. Examined: 18. 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
K01	Key/credential confidentiality	Z01	
K02	Payload and end-to-end confidentiality	Z03	
K03	Strong confidentiality/indistinguishability of secret values	Z03	
K04	Key usage, direction and session separation	A02	
D01	Message integrity and data origin authenticity	A02	
A01	Peer aliveness (Aliveness)	A02	
A04	Injective agreement authentication	A02	

Entry	Property of the catalogue	Examined as	Why not
A05	Implicit key authentication	Not applicable	No key establishment is defined in these two APIs; the SBI TLS baseline is modelled as given.
A06	Explicit key confirmation	Not applicable	No key confirmation takes place in this flow.
A07	Identity–credential–key binding	Z03	
R01	Authentication/challenge freshness	A08	
R02	Message-level replay resistance	A08	
R03	At-most-once business operation	A08	
R04	Message acceptance order/window security	A08	
R05	Downgrade resistance/security policy preservation	Outside the scope	No version, suite or security level can be negotiated in the selected clauses; both feature tables are empty.
C01	Forward secrecy (FS)	Not applicable	No session key is established by these APIs.
C02	Key-compromise impersonation resistance (KCI)	Not applicable	No key agreement whose long-term key could be compromised.
C03	Known-key security/compromise isolation	Not applicable	No key hierarchy belongs to this flow.
C04	Ephemeral secret/partial state compromise resilience	Outside the scope	Reading a party's memory is outside the modelled attacker; only the party exposures S01-S03 were considered.
C05	Post-compromise security (PCS)	Not applicable	No ratchet or rekeying in this flow.
C06	Augmented authentication after registration credential compromise	Not applicable	No registration record or PAKE.
C07	Offline dictionary attack resistance	Not applicable	No password or other guessable secret is used by the NFs.
C08	Goal preservation under failure of a designated primitive	The prover cannot be asked	Resistance to a broken primitive is outside what the symbolic model supports.
Z02	Delegated authority does not expand	A08	
Z04	Revocation and expiry take effect	Z01	
Z05	Cross-tenant/cross-service security isolation	Z03	
S01	Encryption nonce uniqueness	Not applicable	No nonce-based encryption is part of this flow.
S02	Retransmission/key reinstallation security	Not applicable	No key installation or retransmission of key material.
S03	Rollback resistance and state monotonicity	A08	
S04	Cross-phase security context consistency	A08	
S05	Termination and invalidation of old state	A08	
P01	Identity anonymity	Not applicable	NF identities are not hidden on the SBI; neither API asks for anonymity.
P02	Session unlinkability	Not applicable	No unlinkability requirement for NF subscriptions.
P03	User/credential existence hiding	Not applicable	No user or credential existence hiding in these APIs.
V01	Conditional progress/error recovery	The prover cannot be asked	Conditional progress and recovery are liveness properties, outside what ProVerif can be asked.
V02	No unexpected deadlock	The prover cannot be asked	Absence of deadlock is a liveness property, outside what ProVerif can be asked.

Entry	Property of the catalogue	Examined as	Why not
V03	Bounded resource abuse/amplification resistance	The prover cannot be asked	Resource exhaustion and amplification are outside what ProVerif can be asked.

The parties and the protected values, one by one

The host listed the 16 parties and protected values that the drafted properties name, and the analyst said of each which attacker controls the party or holds the value, or why no such attacker was set. With an attacker: 5. Without: 11. 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	AF	party	Everything lost by definition	The AF is the party whose honesty the whole composed flow rests on; if it is controlled by the attacker it fabricates or redirects every inference result, and the documents ask none of these properties under a compromised trusted AF.
S02	NWDAF	party	`dishonest_nf`	
S03	NRF	party	Everything lost by definition	The NRF issues the access tokens; an attacker holding the authorization server can mint a token for any consumer, so every authorization property asked here is lost by definition and the documents ask none of them under a compromised authorization server.
S04	Consumer identity in the subscription request	value	Nothing gained	The consumer identity is carried in the request and known to the receiver; under TLS it cannot be used to speak for that consumer.
S05	Producer identity in the notification	value	Nothing gained	The producer identity is the identity of the TLS peer; knowing it gives nothing the network attacker lacks.
S06	notifCorreId	value	Nothing gained	The notification correlation identifier is chosen by the consumer; holding it does not let the network attacker send a notification over the AF's TLS session.
S07	vflCorrId	value	`dishonest_nf`	
S08	anaEvent	value	Nothing gained	The analytics event type is a public enumeration value.
S09	notifUri	value	Nothing gained	The notification URI is public and is carried in the request.
S10	Subscription context	value	Nothing gained	The subscription context is held by the AF and the consumer; a network attacker cannot use it because the notification is bound to the AF's TLS session.
S11	Notification target	value	Nothing gained	The notification target URI is public; delivering to it requires the producer's TLS session.
S12	OAuth2 access token	value	`stolen_token`	
S13	naf-inference / nnwdaf-vflinference scope	value	`stolen_token`	
S14	Access token	value	`stolen_token`	
S15	Subscription identifier notifCorreId	value	Nothing gained	Same as S06: holding the correlation identifier gives no way to answer for the producer.
S16	Notification target notifUri	value	Nothing gained	Same as S11: the target URI is public.

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		
`dishonest_nf`	A legitimate NF with its own TLS credentials and its own OAuth2 access tokens for the naf-inference and nnwdaf-vflinference scopes; it runs the consumer role of either API towards the AF or the NWDAF VFL client and chooses the values in its own requests.	TS 33.501 5.9.1 divides the network into trust zones and 5.9.2.1 requires each NF to authorize service requests; a legitimate NF of the same trust domain holds valid credentials, so it is the attacker the composed flow must withstand when it asks for inference about UEs it has no relation to. The task also requires separating the external attacker from a malicious legitimate NF.	ts-33-501:2859, ts-33-501:2867
`stolen_token`	An OAuth2 access token that the authorization server issued to another NF for the service scope, read from storage or from a transport it could observe.	RFC 6749 10.3 requires access token credentials to be kept confidential in transit and storage and notes that a malicious client can impersonate another client if the token is not kept confidential (10.2).	rfc6749:2859, rfc6749:2827

ID	`network`	`dishonest_nf`	`stolen_token`
A02	Holds		
A03	Holds		
A08	Fails		
Z01	Holds	Holds	Holds
Z03	Holds	Fails	Holds

Source and targets

Source: 677 units, 377484 bytes, frozen in source-delivery-01.json. Targets: 5 approved (A02, A03, A08, Z01, Z03); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

Decisions the host took by its documented default (task `decisions: auto`), for you to check:

- approve_package: approved the source package protocol-e60fcf52f8ed63a9b30641206c35

This step as it reports itself: 2 requests (2 failed), 0.000 USD, 356920 tokens in and 3708 out, 262 s.

Notes on the prover and the model

- A08: answered in ProVerif, and the analyst names changing_state for this target

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

State review by the analyst

This part is the analyst's own reading of the source, not an answer of the prover. "yes" under shown by ProVerif means that ProVerif found a trace with these steps in the unrolled model, so the steps are possible; that they break a target is said only by an attack in the table of answers, and what else they cost is the analyst's reading. "nothing found" says which patterns were looked at and proves nothing.

value	kept by	changes when	checked when	refs
notifCorreId, notifUri and the analytics target (UE) of the outer subscription, together with the notifCorrId of the inner VFL subscription	AF (VFL server), for the outer Naf_Inference subscription and for the associated Nnwdaf_VFLInference subscription	On the POST that creates the Individual Inference Subscription (s0040) and on the PUT/PATCH that updates it (s0055); the AF then updates the associated Nnwdaf_VFLInference subscriptions with the same notifCorrId (s0059).	When a VflInferNotif arrives: the AF matches only notifCorrId (s0419, s0593); the analytics target that was in force when the round was requested is not compared with anything in the notification.	s0040, s0055, s0059, s0419, s0593
notifCorreId and the current analytics target of its subscription	NWDAF consumer	On creation and on each PUT/PATCH it sends (s0055).	On each InferNotif: only notifCorreId is compared (s0086, s0246).	s0055, s0086, s0246, s0088
notifCorrId, notifUri, the UE target and the VFL correlation ID of a VFL inference subscription	NWDAF VFL client	On every Nnwdaf_VFLInference_Subscribe it receives, creation or update of the same subscription (s0355); the VFL correlation ID is the only handle on the VFL process (s0590).	When it sends a VflInferNotif it fills only notifCorrId and vflInferResults (s0419, s0593).	s0355, s0419, s0590, s0593, s0631

pattern	result	target	shown by ProVerif	steps
replay	candidate	A08	yes: ProVerif falsifies `a08` on bounded.pv	NWDAF creates an Individual Inference Subscription for supi1 with notifCorreId and notifUri (s0040). / AF creates the associated Nnwdaf_VFLInference subscription with its own notifCorrId for supi1 (s0355). / NWDAF updates the same subscription to supi2 (s0055); the AF updates the inner subscription, keeping the same notifCorrId (s0059). / The VFL client's round computed for supi1 is still in flight and reaches the AF. / AF matches it by notifCorrId, reports it to the NWDAF, and the NWDAF accepts it as the result for supi2 (s0086).
update_before_check	candidate	A08	yes: ProVerif falsifies `a08` on bounded.pv	AF holds a VFL subscription for supi1 under notifCorrId V. / NWDAF updates the outer subscription to supi2 (s0055). / AF immediately replaces the target of the VFL subscription with supi2 and sends the update to the client (s0059). / AF then accepts the next VflInferNotif carrying notifCorrId V without any check that it belongs to the new request (s0419).
race_between_sessions	candidate	A08	yes: ProVerif falsifies `a08` on bounded.pv	NWDAF's create and its update are two requests on the same Individual Inference Subscription (s0055). / The AF's two Nnwdaf_VFLInference_Subscribe requests (create and update) reach the VFL client in the same order, but the client's round for the first request is already on the way. (s0059) / The AF takes the in-flight round as the answer to the updated subscription and reports it (s0086).
repeat_one_time_action	nothing found			A refused update returns an error and leaves the subscription as it was (s0064); no key, counter or one-time resource is installed by these operations.

pattern	result	target	shown by ProVerif	steps
desynchronisation	nothing found			The documents do not say that the AF and the consumer keep different targets: both end on supi2; the divergence is in the result content, not in the stored subscription.
no_rollback_on_failure	nothing found			A failed create or update returns an HTTP error (s0050, s0064) and does not change the stored subscription; no state is left behind that a later message can use.
distinguishable_failure	nothing found			The only failure answers in these operations are HTTP status codes with a ProblemDetails body (s0050, s0088, s0367); nothing in the modelled flow branches on them, so no failure answer distinguishes a party or a stored value.
wrap_or_window	nothing found			The subscription state has no counter, sequence number or acceptance window; the only identifier is the notifCorrId, which never wraps.

What the analyst named for targets:

step	targets	features	reason
47	A08	changing_state	A08 rests on the inference subscription as a value that changes: the consumer replaces the analytics target of the same Individual Inference Subscription, the AF replaces the target of the associated Nnwdaf_VFLInference subscription with the same notifCorrId, and the property is about which round of results that context accepts. The model unrolls that lifecycle to a fixed three steps (create, update, notify) in bounded.pv.
47	A02, A03, Z01, Z03	plain_primitives, other	A02, A03, Z01 and Z03 rest only on the SBI transport modelled as a pairwise authenticated-encryption envelope (the TLS session) and on the OAuth2 access token modelled as an unforgeable value bound to subject, audience and scope; no algebraic law is needed. The 'other' feature is the access token, which the model idealises as an unforgeable term instead of a signed JWT with expiry, so token expiry and revocation are not modelled.

The analyst's closing note

Model of the composed flow (Naf_Inference subscription create/update/notify at a trusted AF with the Nnwdaf_VFLInference subscription and notification to a NWDAF VFL client) runs a complete session (model.pv, EndRun reached), and bounded.pv states the same lifecycle as a fixed three-step unrolling.

Answers, all re-run by the host: A02 peer authentication of both subscribe hops is proved against the network attacker (2 claims, with falsified controls). A03 field and correlation agreement (inner notifCorrId, update of the resource and the analytics target, the consumer's own notifCorreId) is proved (3 claims, with controls). Z01, the token check at the AF and at the VFL client, is proved for the network attacker, for a legitimate NF with its own tokens and for a stolen token, with a control in which the AF accepts a request whose token is for another scope. Z03 is split: the token cannot be presented for another consumer and the network attacker cannot read the inference results (proved), but a legitimate NF holding only the service-wide scope subscribes at the NWDAF VFL client for another function's VFL correlation ID and UE and receives the results (falsified; verdict depends_on_ambiguity because the participant rules are in TS 29.552, outside the source units). A08 is falsified on model.pv and on bounded.pv: after the consumer updates its subscription from one UE to another, the VFL round computed for the first UE that is still in flight is reported by the AF and accepted by the consumer as the answer to the updated subscription, because neither notification carries the UE set or a round identifier and the only comparison is the unchanged notifCorrId. That is the main finding (novelty: not recorded in the documents).

The state review records the three values a party keeps and changes and marks replay, update_before_check and race_between_sessions as confirmed candidates (all shown by the a08 query) and the other five patterns as nothing found.

What blocks more: the TLS session and the OAuth2 token are idealised (no expiry, revocation, record replay protection or NRF), deletion and multi-session state are not modelled, and neither recommendation (round identifier in the two notifications; participant check in the VFL client) was checked with check_fix within the budget. Token expiry and the TS 29.552 participant rules need Tamarin+Agent or the missing specification to settle.