ADRF 订阅与存储分析的安全性分析

本报告的其他语言版本：English。

19.7.0 · ProVerif+ 智能体报告

文稿	5G System; Analytics Data Repository Services; Stage 3 （协议定义）; 5G System; Network Data Analytics Services; Stage 3 （辅助）
检查的性质	11 条：10 条成立，1 条不成立（1 个攻击）
发现	2

目录

1 协议和范围
2 发现
3 检查过的性质
4 本次分析的局限
附录 A：逐条性质的解读
附录 C：主张和证明器的答案
附录 D：分析是怎么跑的

1 协议和范围

ADRF 数据管理服务让 NF 服务消费者能够让其数据或分析被收集并存储：消费者发送 StorageSubscriptionRequest，ADRF 订阅名为 targetNfId 的 NWDAF，存储 NWDAF 发送到其给出的通知 URI 的通知，并在消费者发送 StorageSubscriptionRemoval 时移除上游订阅。各方为 NF 服务消费者、ADRF、上游 NWDAF 以及作为授权服务器的 NRF。消费者持有其 NF 凭据以及具有存储范围的 OAuth2 令牌，ADRF 持有自己的凭据以及其分配的 transRefId 与上游订阅之间的映射，NWDAF 持有订阅标识符及其收到的通知 URI。

交换结束时，ADRF 对每个已订阅响应和通知的对端以及对其记录的订阅参数确信无疑，消费者确信其自己的请求创建了该事务并持有 transRefId，NWDAF 确信 ADRF 在具有事件订阅范围的令牌下请求了其持有的订阅。每条消息都在端点身份之间通过 TLS 传输，每个操作只有在令牌主体为发送它的端点时才被接受。

各方：

- **NF 服务消费者**：收集和存储的请求方。持有：NF 凭据；OAuth2 令牌：主体为消费者，受众为 ADRF，范围为 nadrf-datamanagement:storage-read-delete-subs。
- **ADRF**：服务生产者，存储分析。持有：其 NF 凭据；其分配的 transRefId 及其上游订阅。
- **上游 NWDAF**：分析生产者。持有：其 NF 凭据；其创建的 subscriptionId。
- **NRF**：授权服务器。持有：NF 凭据以及授权服务器角色。

图 1 展示了文稿所规定的交换过程。

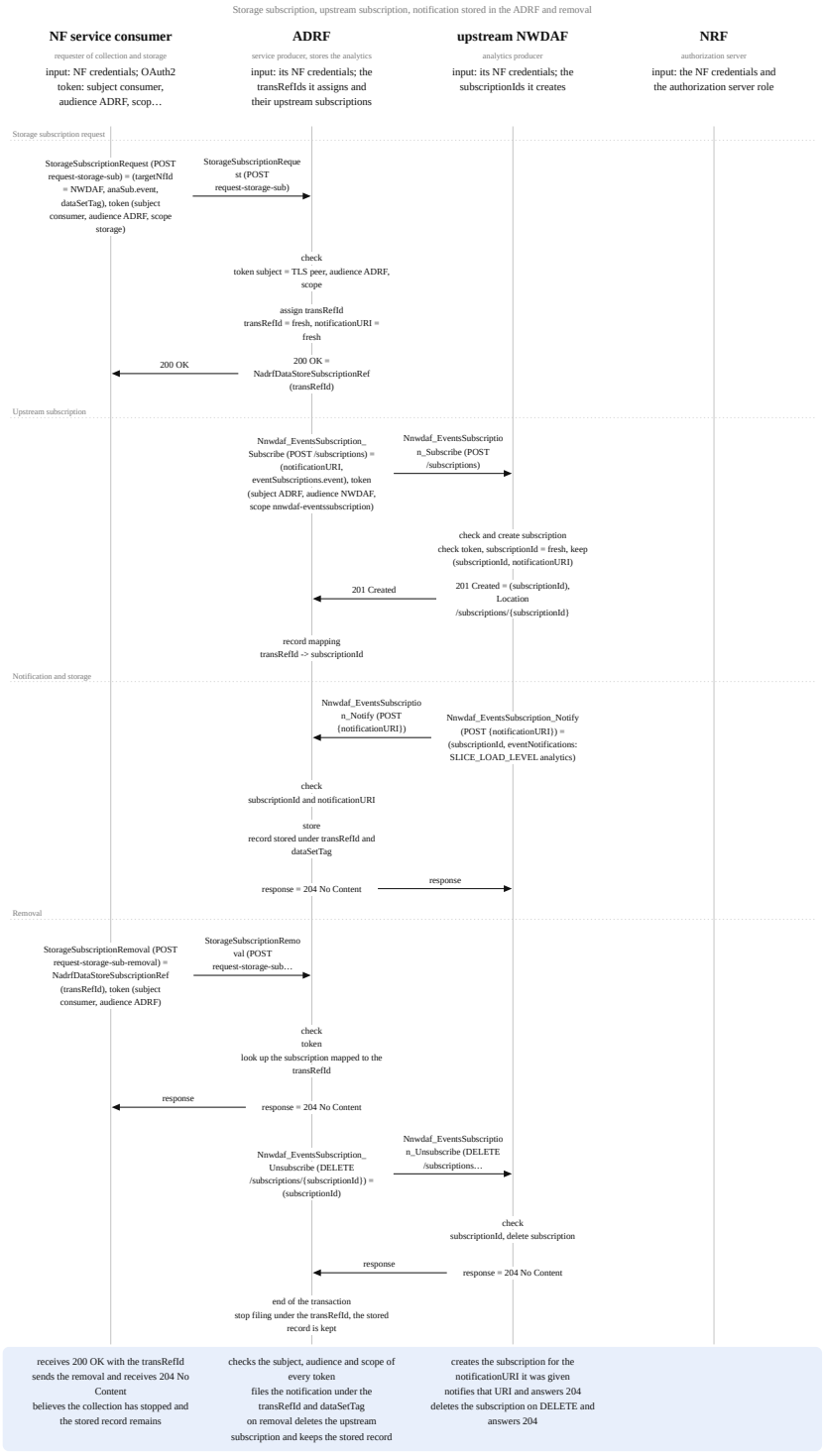


图 1. 文稿所规定的协议。

分析的范围见附录 D：分析了什么、省略了什么，以及所依据的文稿章节。

2 发现

分析有 2 项发现：以下内容并未在文稿中陈述。其中 2 项是否由文稿陈述尚不明确。

API 子流程在面对网络上的攻击者时完成了其预期功能，但有一个弱点：一旦移除请求指名了 transRefId，ADRF 就会在 API 的任何有效令牌下移除上游订阅，因此知道另一个消费者 transRefId 的消费者可以停止该消费者的收集。其余所询问的一切在模型中均成立，包括上游 NWDAF 的对端认证、订阅参数的一致、存

储记录与消费者请求的绑定、令牌的主体、受众和范围绑定，以及移除后的生命周期。第二个弱点是从原文中读出的，而非由查询展示的：存储订阅在上游订阅存在之前就被确认，且没有回滚。这些答案依赖于一个模型，其中 TLS 是每对端点之间的理想化密钥，OAuth2 没有过期，ADRF 生命周期展开为三步，且多个 transRefId 共享一个上游订阅未被建模。

攻击图使用图 1 的列。红色是攻击者的行为。黄色标记某方发送的消息，尽管规定的交换在该点并没有该消息，以及攻击所破坏的结果。

2.1 已知的 transRefId 让任何已授权 NF 停止另一个消费者的收集

类型	协议规范本身的弱点
是否新发现	不确定：文稿只写了一部分，或没能检索
价值	中
涉及的性质	Z03 （见第 3 节）
原文	s0034, s0035, s0293
依据	ProVerif 攻击查询 `z03g`; 图 2

发生了什么。 移除请求指名一个 transRefId，ADRF 移除映射到它的上游订阅，如 s0035 所要求，但没有任何东西将该 transRefId 所记录的消费者与请求令牌的主体进行比较。因此，知道该标识符的第二个 NF 服务消费者可以使第一个消费者的上游订阅被删除，其收集被停止。

攻击者需要什么。 一个在其自身有效凭据下行动的已授权 NF 服务消费者，以及另一个消费者存储订阅的 transRefId。该标识符只在 ADRF 与其所有者之间传输，因此攻击者需要从日志、从所有者处，或从对猜测的错误应答中获取它。

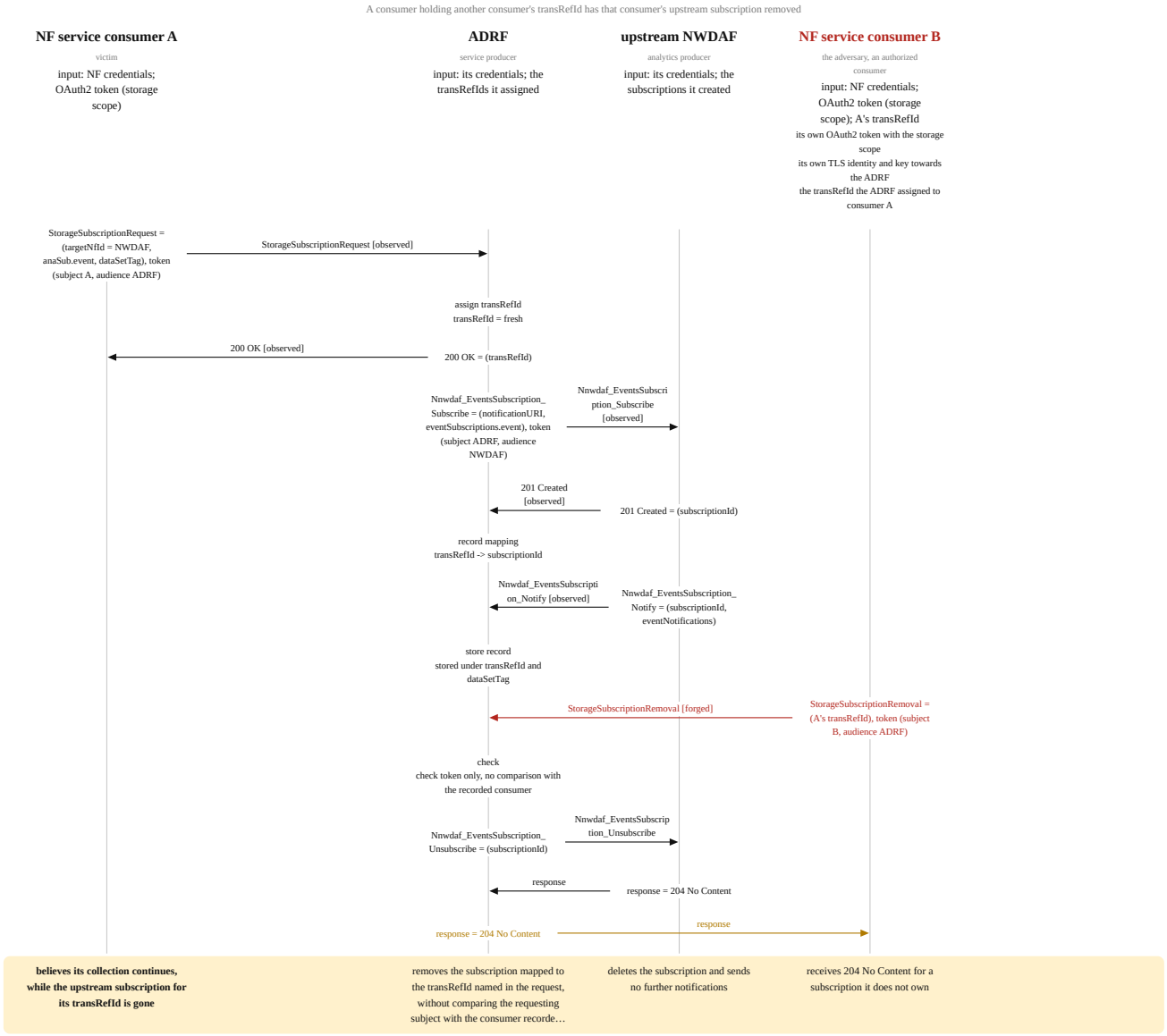


图 2. 持有另一个消费者 transRefId 的消费者使该消费者的上游订阅被移除。性质 Z03；攻击查询 z03g。

拟议修改。 ADRF 应记录创建每个 transRefId 的消费者的 NF 实例身份，并在收到 StorageSubscriptionRemoval 请求时，在移除映射的上游订阅之前验证令牌主体和 TLS 对端身份就是该消费者；指名另一个消费者 transRefId 的移除以 403 Forbidden 应答。在展开模型的副本上检查：该攻击查询在那里通过验证，且其他记录的答案仍然成立。

用 ProVerif 复核。 该修改被放入模型的副本中：ADRF 记录创建每个 transRefId 的消费者的 NF 实例身份，并在收到 StorageSubscriptionRemoval 请求时验证令牌主体（以及 TLS 对端身份）就是所记录的消费者；指名另一个消费者 transRefId 的移除以 403 Forbidden 应答，且不移除任何东西。在修改后的模型上，ProVerif 证明了 `z03g`：该攻击在那里不再可能。其他 3 个记录的答案不变。这是对模型的首次检查：它表明该修改在那里移除了此攻击，而非表明它不引入其他问题。修改后的模型：`records/fixes/0048-fix-owner.pv`。

2.2 存储订阅在其上游订阅存在之前就被确认，且没有回滚

类型	协议规范本身的弱点
是否新发现	不确定：文稿只写了一部分，或没能检索
价值	中
涉及的性质	A08.8 （见第 3 节）
原文	s0021, s0023, s0026

发生了什么。 ADRF 在创建上游订阅之前就分配 transRefId 并应答 200 OK，而原文对随后失败的创建没有给出任何规则。于是消费者持有一个其订阅从未存在过的 transRefId，而 ADRF 持有一个指向空无的映射。

攻击者需要什么。 除了网络和一个被 NWDAF 拒绝或丢失的上游创建之外，别无他物，两者都是原文所允许的（400 Bad Request、500 Internal Server Error）。

拟议修改。 ADRF 应仅在上游订阅存在之后才以 200 OK 和 transRefId 应答，或者通过以 notifCorrId 关联的通知向消费者报告失败的创建。

用 ProVerif 复核。 未检查：该性质是失败的创建被回滚并报告；没有记录的查询展示它，因为原文根本没有陈述回滚规则，而查询只能重述模型所做的一个选择。

3 检查过的性质

每一行是一条提交给 ProVerif 的性质。结果是 ProVerif 在分析者协议模型上的答案；价值和文稿是否已陈述该结果是分析者的判断，理由见附录 A。

ID	性质	假定的泄露	结果	价值	文稿是否已写
A02	**ADRF 处上游对端认证。** 每当 ADRF 接受一个上游 NWDAF 的订阅响应或事件通知时，都存在一个真实运行的上游 NWDAF，其意图对端身份就是该 ADRF。	无	成立	中	否
A03	**上游订阅参数协商一致。** 每当 ADRF 认为某个上游订阅已按某组参数建立时，都存在一个上游 NWDAF 的真实运行，其角色、身份与订阅参数（anaSub/dataSub、targetNfId、formatInstruct/procInstruct、storeHandl、dataSetTag）与 ADRF 所记录的一致。	无	成立	低	否
A08	**存储记录绑定到请求者请求。** ADRF 接受并存储的通知所携带的 notifCorrId、dataSetTag/dataSetId 与 targetNfId 必须对应请求者在该次 StorageSubscriptionRequest 中给出的值，且与当前安全上下文（TLS 端点身份、OAuth2 令牌的 subject/audience/scope）一致。	无	成立	中	否
K01	**OAuth2 令牌保密性。** 攻击者不能推导出 requester、ADRF 或上游 NWDAF 持有的 OAuth2 访问令牌或 NF 凭据。	无	成立	低	否
Z01	**API 操作授权。** 每次 ADRF 提交存储订阅创建、存储订阅删除或上游订阅创建操作时，都必须存在由 NRF 签发的、subject 为该请求者、audience 为 ADRF、scope 覆盖该操作的未过期令牌。	无	成立	中	否
Z03	**删除绑定到拥有主体。** 每次 ADRF 接受 StorageSubscriptionRemoval 并删除某个 transRefId 或 dataSetId 对应的上游订阅时，该 transRefId/dataSetId 必须属于提交该请求的令牌 subject。	无	不成立（1 个攻击）	中	不确定
S05	**删除后生命周期。** 在 ADRF 成功处理 StorageSubscriptionRemoval 之后，被删除的 transRefId 不能再用于创建新的上游订阅或接受新的通知存储；仍被其他活跃 transRefId 引用的上游订阅不被删除。	无	在一个界内成立（变化的值展开到 3 步）	中	否

ID	性质	假定的泄露	结果	价值	文稿是否已写
A08.8	**记录身份对应性。** ADRF 存入的记录所携带的 transRefId、dataSetTag/dataSetId 与 notifCorrId 必须对应请求者在该次 StorageSubscriptionRequest 中给出的值，并对应 ADRF 为该请求创建的上游订阅。	无	成立	低	否
U01	**无跨消费者参数变更。** 对任意两个不同请求者 A、B，ADRF 为 A 建立或修改的上游订阅参数与存储处理参数只能由 A 的真实请求引起。	无	成立	中	否
S05.10	**删除不删除存储记录。** StorageSubscriptionRemoval 成功后，被删除的 transRefId 不能再创建上游订阅或接受新通知存储；已存记录不被该操作删除，只有 remove-stored-data-analytics 才删除已存记录。	无	在一个界内成立（变化的值展开到 3 步）	中	否
X01	**传输中 API 消息的机密性与完整性。** 上游 NWDAF 发给 ADRF 的通知中携带的分析数据，以及存储订阅请求和删除请求的参数，都不能被网络上的攻击者读取或篡改：没有任何存储通知的事件使用攻击者能获得的分析数据，也没有任何操作会以攻击者改动过的参数被接受。	无	成立	中	否

文稿写明而起草的清单里没有、由分析者补充的目标（见附录 D）：X01.

提出的问题，以及分析者的回答：

问题	回答	依据
Q1: 请求者、上游订阅与生成的 ADRF record 绑定	ADRF 只把通知归入 NWDAF 所报告其上游订阅的那个 transRefId，并归入消费者在该请求中给出的 data set tag，因此消费者、transRefId 与记录由 ADRF 自己的映射绑定在一起。缺失的是上游订阅从未创建时的回滚，以及删除一侧未绑定到拥有者，这就是关于 transRefId 的发现。	A08.8, A08, Z03
Q2: 恶意消费者不能改变别人的采集/存储参数	否：消费者只以自己的令牌和 TLS 身份行事，因此不能用另一个消费者的 target、event 或 data set tag 创建订阅，也不能改动为另一个消费者记录的参数。如果它知道 transRefId，就可以删除另一个消费者的上游订阅，这就是所报告的发现。	U01, Z01, Z03
Q3: 删除存储订阅与已有存储记录的生命周期不能混为一谈	对已建模的步骤而言是：删除会终止上游订阅，该路径上不删除任何已存记录，在删除被接受之后到达的通知会被拒绝。生命周期只针对每个上游订阅的一个 transRefId 和三个展开步骤作答，因此关于更多活跃 transRefId 的共享规则以及被删除标识符此后的任何复用，仍是未决的。	S05, S05.10

4 本次分析的局限

这里的结果是协议模型上的结果：它说明所规定的消息与检查给出了什么、未能给出什么；其中密码学被视为理想的，不涉及任何实现。

据分析者所述，本次分析不覆盖以下内容：

- TLS 被建模为一种经认证的加密，其密钥由两个 NF 身份派生、按对使用；握手、证书及其吊销未作分析。
- OAuth2 被建模为一个携带 subject、audience 与 scope 的令牌；过期、吊销以及与 NRF 的客户端凭据交换未建模，且该令牌可在任意多次运行中重复使用。
- ADRF 的订阅状态展开到三步，因此生命周期相关的回答只对这些步骤成立；s0035 中“仍被更多活跃 transRefId 映射的上游订阅不被删除”这一规则未建模。
- 已存记录、其检索以及由 remove-stored-data-analytics 执行的删除不在范围之内，未建模。
- 如范围所述，DCCF 上游、MFAF 与 UDM 同意分支被排除在外。

- 建模了三个消费者、一个 ADRF 和一个 NWDAF 实例；NF set 标识符、重定向与服务通信代理未建模。
- 被攻陷的消费者与被攻陷的生产者未建模；攻击者只掌握网络和一个泄露的 transRefId。
- MAC 与加密原语的强度，以及抗量子攻击能力，超出 ProVerif 所能回答的范围。

宿主记录的内容：

- 在以下解读写成之后，回答发生了变化：Z03。
- Z03、S05、S05.10 的回答依赖于模型所简化的一项操作（附录 D）。
- 没带对照查询的证明（附录 C）：K01: k01, X01: x01。对照查询是这条性质的一个必然不成立的变体；没有它，就没有东西表明这条性质照现在的写法在模型里有可能不成立。

每条性质还有哪些未决之处，以及什么能解决它，见附录 A。

附录 A：逐条性质的解读

以下为分析者本人对证明器回答的解读，而非证明器的回答：这些回答对协议意味着什么、解读依据是什么、能否照原样使用，以及它的价值如何。

A02: ADRF 处上游对端认证

每当 ADRF 接受一个上游 NWDAF 的订阅响应或事件通知时，都存在一个真实运行的上游 NWDAF，其意图对端身份就是该 ADRF。

ADRF 接受的每一条通知和每一个订阅响应都来自它所订阅的那个 NWDAF，带有该 NWDAF 自己接受的订阅标识符、事件和通知 URI。该保证依赖于端点 TLS 和令牌 audience，而不依赖于载荷内部的任何内容。

- 结果：成立；回答了所问：是；可带保留地使用
- 价值中：它以机械化的方式确认了 API 文本交由 TLS 与 OAuth2 承担的对端认证；证明简化了 TLS，因此它是对模型的证明，而非对已部署接口的证明。
- 文稿是否已写：否。所依据的原文：s0026, s0913, s0961
- 依据：TLS 被建模为经认证的加密，其密钥由两个 NF 身份派生。；令牌携带 subject、audience 与 scope，但不携带过期时间。
- 未决：一致性不是单射的：同一 NWDAF 运行的重放消息与新消息无法区分。；NF set 标识符与重定向未建模。
- 下一步：建模一个在 anaSub 中放入自己 notificationURI 的消费者，以确定由谁选择回调 URI。

A03: 上游订阅参数协商一致

每当 ADRF 认为某个上游订阅已按某组参数建立时，都存在一个上游 NWDAF 的真实运行，其角色、身份与订阅参数（anaSub/dataSub、targetNfId、formatInstruct/procInstruct、storeHandl、dataSetTag）与 ADRF 所记录的一致。

ADRF 为上游订阅记录的参数、所订阅的事件和通知 URI，就是它发给 NWDAF 的那些，因此服务于该事务的订阅正是 ADRF 所请求的那一个。

- 结果：成立；回答了所问：是；可带保留地使用
- 价值低：它成立是因为响应经过了认证；真正有意思的问题——由谁选择回调 URI——留给了原文。
- 文稿是否已写：否。所依据的原文：s0018, s0026, s0855
- 依据：ADRF 被建模为设置自己的通知 URI。
- 未决：原文未说明当消费者在 anaSub 中提供通知 URI 时，ADRF 是否保留自己的通知 URI。
- 下一步：从原文确定 ADRF 是否会覆盖消费者放入 anaSub 的 notificationURI，然后对该情形建模。

A08: 存储记录绑定到请求者请求

ADRF 接受并存储的通知所携带的 notifCorrId、dataSetTag/dataSetId 与 targetNfId 必须对应请求者在该次 StorageSubscriptionRequest 中给出的值，且与当前安全上下文（TLS 端点身份、OAuth2 令牌的 subject/audience/scope）一致。

一条已存通知携带的是创建其 transRefId 的那个请求的 data set tag，并被归入 ADRF 映射到该 transRefId 的订阅标识符之下，且是经 ADRF 给出的通知 URI 到达的。网络无法把一条通知移入另一条记录。

- 结果：成立；回答了所问：是；可使用
- 价值中：它确定了已存数据的记录身份，而原文仅将其表述为一项映射义务；这正是存储实现者最可能出错的地方。
- 文稿是否已写：否。所依据的原文：s0020, s0021, s0026
- 依据：data set tag 是消费者选择的不透明值。；model.pv 中建模为每个订阅一条通知；展开后的模型发送两条。
- 未决：只有所订阅的事件和 data set tag 代表了订阅参数。
- 下一步：建模 Formatting/Processing 指令与 StorageHandlingInfo，以检查它们是否绑定到同一条记录。

K01: OAuth2 令牌保密性

攻击者不能推导出 requester、ADRF 或上游 NWDAF 持有的 OAuth2 访问令牌或 NF 凭据。

网络上的攻击者无法计算出消费者与 ADRF 一起使用的令牌，因为该令牌是在由消费者身份派生的密钥下计算的 MAC，而接口只在端点 TLS 会话内携带它。

- 结果：Holds；回答了所问：是；可使用但有保留
- 价值 低：该答案来自理想化的令牌构造；其价值在于没有任何消息泄露该凭据。
- 文稿中是否有陈述：否。所依据的原文：s0334、s0335、s0339
- 依据：令牌是在按身份私有的密钥下计算的 MAC；真实令牌格式以及 NRF 的作用被抽象为此。
- 未决：未建模过期与吊销。；该保密性查询没有记录对照变体，因为没有进程会把令牌交给攻击者。
- 下一步：建模一个令牌被泄露的消费者，以将令牌保密性与它所承载的权限分开。

Z01: API 操作授权

每次 ADRF 提交存储订阅创建、存储订阅删除或上游订阅创建操作时，都必须存在由 NRF 签发的、subject 为该请求者、audience 为 ADRF、scope 覆盖该操作的未过期令牌。

ADRF 只接受来自发送该请求的 NF 的存储订阅创建或删除，且令牌的 subject 为该 NF、audience 为 ADRF；NWDAF 只接受来自 ADRF 的上游订阅。不同的 NF 无法为另一个 subject 出示令牌，因为它无法铸造这样的令牌。

- 结果：Holds；回答了所问：是；可使用但有保留
- 价值 中：它展示了 subject、audience 与 scope 绑定承载了两个 API 的授权；对于网络攻击者，在此路径上未发现规范弱点。
- 文稿中是否有陈述：否。所依据的原文：s0334、s0339、s1143
- 依据：令牌是在由 subject 身份派生的密钥下计算的 MAC，验证器从其同时也与 TLS 对端比较的 subject 重新计算它。；scope 对每个 API 是一个常量。
- 未决：未建模消费者被攻陷的解读。；除一个存储 scope 外，未建模过期与按操作的 scope。
- 下一步：建模一个凭据被攻陷的消费者，使其执行另一个消费者的操作。

Z03: 删除绑定到拥有主体

每次 ADRF 接受 StorageSubscriptionRemoval 并删除某个 transRefId 或 dataSetId 对应的上游订阅时，该 transRefId/dataSetId 必须属于提交该请求的令牌 subject。

该目标的答案在此解读写成之后发生了变化。

仅针对网络而言，ADRF 只对携带创建该 transRefId 的消费者的有效令牌的删除请求删除上游订阅，因为该标识符无法被猜到。针对知道另一个消费者 transRefId 的消费者，该绑定会失败：ADRF 会对任何有效令牌采取行动，因此仅由标识符决定。

- 结果：Fails（1 个攻击）；回答了所问：是；可使用但有保留
- 价值 中：一种解读被证明，另一种被以精确步骤攻击；其价值在于表明保护来自一个原文从未要求保密的标识符的不可猜测性。
- 文稿中是否有陈述：不明确。所依据的原文：s0034、s0035、s0293
- 依据：transRefId 被建模为不可猜测的新鲜名称，这正是使网络情形成立的原因。；展开模型为每个上游订阅携带一个 transRefId。
- 未决：原文从未说明 transRefId 是秘密还是持有者能力，因此攻击的前提是对 s0021 和 s0293 的一种解读。
- 下一步：建模两个 transRefId 映射到一个上游订阅，以观察删除共享订阅会产生什么效果。

S05: 删除后生命周期

在 ADRF 成功处理 StorageSubscriptionRemoval 之后，被删除的 transRefId 不能再用于创建新的上游订阅或接受新的通知存储；仍被其他活跃 transRefId 引用的上游订阅不被删除。

在 ADRF 接受删除之后，该订阅的通知不会被存储，重复的删除不会被第二次执行。正是状态变化使这一点成立：不替换保留值的对照变体会存储迟到的通知。

- 结果：在某个界内 Holds（变化值展开到 3 步）；回答了所问：是；可使用但有保留
- 价值 中：一个生命周期保证与其对照一起展示，模型简化了删除与共享；这正是读者在面对一个删除后必须停止收集的订阅时所需要的。
- 文稿中是否有陈述：否。所依据的原文：s0035、s0961
- 依据：ADRF 展开到创建、第一个通知、删除和两条后续消息。；上游删除被建模为总是成功。
- 未决：映射到更多活跃 transRefId 的上游订阅不被删除这一条款未建模。；生命周期仅对三个展开步骤作答。
- 下一步：展开两个 transRefId 映射到一个上游订阅，以回答 s0035 的共享条款。

A08.8: 记录身份对应性

ADRF 存入的记录所携带的 transRefId、dataSetTag/dataSetId 与 notifCorrId 必须对应请求者在该次 StorageSubscriptionRequest 中给出的值，并对应 ADRF 为该请求创建的上游订阅。

ADRF 存储的记录归档在 transRefId 和 ADRF 自身映射到它的上游订阅标识符之下，因此消费者的标识符、ADRF 的映射和该记录通过 ADRF 自身的状态保持一致。

- 结果：Holds；回答了所问：是；可使用但有保留
- 价值 低：在此模型中该答案是结构性的：它表明映射被一致地使用，而不是实现会检查它。
- 文稿中是否有陈述：否。所依据的原文：s0021、s0026、s0035
- 依据：每个 ADRF 会话持有自己的 transRefId 和订阅标识符。；未建模数据集标识符及其在消费者之间的唯一性。
- 未决：未建模上游创建失败的回滚；它改为作为发现报告。
- 下一步：建模两个具有相同数据集标签的消费者，以观察他们的记录如何被区分。

U01: 无跨消费者参数变更

对任意两个不同请求者 A、B，ADRF 为 A 建立或修改的上游订阅参数与存储处理参数只能由 A 的真实请求引起。

ADRF 仅在消费者自己以其自己的令牌和 TLS 身份发送的请求上创建存储订阅。没有其他消费者引起该创建，因此订阅的目标、事件和数据集标签都是它自己的。

- 结果：Holds；回答了所问：是；可使用但有保留
- 价值 中：它区分了两个消费者，并表明一个不能移动另一个的参数；恶意消费者的解读仍未决。
- 文稿中是否有陈述：否。所依据的原文：s0008、s0020、s0026
- 依据：每个消费者有自己的令牌和 TLS 密钥。；参数仅由目标、事件和数据集标签表示。
- 未决：未建模消费者被攻陷的解读。；未建模对同一数据集的第二个请求，s0022 可能从现有订阅中服务该请求。
- 下一步：建模一个凭据被攻陷的消费者，并询问它是否能产生另一个消费者订阅的任何事件。

S05.10: 删除不删除存储记录

StorageSubscriptionRemoval 成功后，被删除的 transRefId 不能再创建上游订阅或接受新通知存储；已存记录不被该操作删除，只有 remove-stored-data-analytics 才删除已存记录。

存储订阅删除仅删除上游订阅，别的什么都不做：该路径上不删除任何已存记录，已存储的记录保留。删除已存数据是一个具有自己 scope 的独立操作。

- 结果：在某个界内 Holds（变化值展开到 3 步）；回答了所问：是；可使用但有保留
- 价值 中：它回答了停止收集与删除数据之间的混淆，这两个操作的实现者必须正确处理这一点。
- 文稿中是否有陈述：否。所依据的原文：s0035、s0186、s0215
- 依据：记录由一个删除路径从不逆转的事件表示。；展开模型覆盖三个步骤。
- 未决：未建模 remove-stored-data-analytics 操作，因此该陈述仅从删除侧展示。；已存记录未被建模为可检索对象。
- 下一步：建模 remove-stored-data-analytics，以表明这两个操作作用于不同的对象。

X01: 传输中 API 消息的机密性与完整性

从上游 NWDAF 到 ADRF 的通知中携带的分析数据，以及存储订阅和删除请求的参数，不能被网络上的攻击者读取或更改：没有任何存储通知的事件使用攻击者能够获得的分析数据，也没有任何操作以攻击者更改过的参数被接受。

通知中的分析数据和请求的参数永远不会被网络上的攻击者读取或更改：每条消息只在两个端点身份的密钥下传输，合法持有端点的方能看见它们。

- 结果：成立；所问是否得到回答：是；可用
- 取值中：这是本子流程中基于服务接口的机密性与完整性陈述；它成立靠的是 TLS 抽象，而不是 API 额外添加的任何东西。
- 文稿中是否有陈述：否。它所依据的原文：s0913、s0961、s0334
- 依据：TLS 是每一对有序 NF 身份各有一把私钥，无需握手。；分析数据是每条通知一个新鲜 nonce。
- 待解决：未对受损端点建模，因此只回答了网络攻击者。；此保密性查询没有记录对照变体。
- 下一步：对受损的 ADRF 或 NWDAF 建模，看看另一方仍保留什么。

附录 C：主张和证明器的答案

证明器对每个目标的记录：达到的级别、达到该级别的查询、是否检查过对照，以及各项攻击及分析者对每项攻击的判定。

目标	级别	界	查询	对照	攻击	主张
A02	proved		a02n	checked		2 proved
A03	proved		a03	checked		1 proved
A08	proved		a08	checked		2 proved
K01	proved		k01			1 proved
Z01	proved		z01	checked		2 proved
Z03	proved		z03	checked	z03g (holds_in_source)	1 proved, 1 attack
S05	bounded	变化值展开到 3 步	s05	checked		2 bounded
A08.8	proved		a08b	checked		1 proved
U01	proved		u01	checked		1 proved
S05.10	bounded	变化值展开到 3 步	s0510	checked		1 bounded
X01	proved		x01			1 proved

逐条主张，按分析者对每条主张的措辞列出；每条的答案是证明器给出的：

目标	主张	攻击者	回答	查询	对照
A02	每当 ADRF 为某个上游订阅存储一条通知时，都是某个真实的上游 NWDAF 为该订阅标识符发送了这条通知：写下该消息的对端就是 ADRF 所订阅的那个 NWDAF。	network	proved	a02n	checked
A02	每当 ADRF 将某个上游订阅记为已建立时，都是某个真实 NWDAF 以相同的已订阅事件和通知 URI 接受了该订阅，且其消费者是 ADRF。	network	proved	a02s	checked
A03	ADRF 记录该上游订阅时所依据的参数——已订阅事件和通知 URI——就是 ADRF 发送给 NWDAF 的那些参数。	network	proved	a03	checked
A08	ADRF 存储某条通知时所依据的数据集标签，就是请求方在创建该 transRefId 的存储订阅请求中给出的数据集标签。	network	proved	a08	checked
A08	一条通知只会存储在 ADRF 映射到该 transRefId 的订阅标识符之下，并且只在它到达 ADRF 所给出的通知 URI 时才被存储。	network	proved	a08b	checked
K01	网络上的攻击者无法计算出请求方在 ADRF 上用于存储范围的访问令牌凭据。	network	proved	k01	none
Z01	ADRF 仅在令牌的主体是以自身 TLS 身份发送该请求的 NF、受众是 ADRF、范围为存储范围时才接受存储订阅创建。	network	proved	z01	checked
Z01	NWDAF 仅在令牌的主体是发送该订阅的 ADRF、受众是 NWDAF、范围为事件订阅范围时才接受上游订阅。	network	proved	z01n	checked

目标	主张	攻击者	回答	查询	对照
Z03	当 ADRF 移除某个 transRefId 的上游订阅时，该移除请求携带的是拥有该 transRefId 的主体的有效令牌：没有其他 NF 能让 ADRF 移除另一个消费者的存储订阅。	`network`	proved	z03	checked
Z03	当 ADRF 移除某个 transRefId 的上游订阅时，该移除请求绑定到拥有该 transRefId 的主体，因此另一个知道该标识符的已授权 NF 无法让它被移除。	`leaked_transref`	attack (holds_in_source)	z03g	
S05	在 ADRF 已接受某个 transRefId 的移除之后，该订阅上的后续通知不会被存储：已移除的 transRefId 不接受任何新的已存储通知。	`network`	bounded（变化值展开到 3 步）	s05	checked
S05	其移除已被接受的 transRefId 之后不会再创建新的上游订阅，而且对同一 transRefId 的重复移除不会被第二次接受：这一一次性动作只发生一次。	`network`	bounded（变化值展开到 3 步）	s05b	checked
A08.8	ADRF 存储的记录归档在 transRefId 以及 ADRF 自身映射到该 transRefId 的上游订阅标识符之下，而不是归入另一个请求的订阅。	`network`	proved	a08b	checked
U01	ADRF 仅在由该消费者本身发送、且带有该消费者自己的目标、事件和数据集标签的请求上创建存储订阅；没有其他消费者能导致它创建。	`network`	proved	u01	checked
S05.10	存储订阅移除只移除上游订阅：该 transRefId 已存储的数据或分析数据不会因此被删除，且该路径上不会发生任何已存储记录的删除。	`network`	bounded（变化值展开到 3 步）	s0510	checked
X01	ADRF 所存储的通知中携带的分析数据，以及存储订阅请求和移除请求的参数，网络上的攻击者永远无法获取或篡改：每条消息都只在端点 TLS 密钥之下传输。	`network`	proved	x01	none

附录 D：分析是怎么跑的

一名 LLM 分析者根据文稿构建协议模型，并把每条性质交给 ProVerif；第 3 节中的每个结果都是证明器给出的答案。

ProVerif+ Agent 会话 session-20261115T092715，2026-11-15。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：65，证明器 4 秒，实际耗时 17 分钟，成本 0.120 USD。

ProVerif：运行 95 次证明，95 次判定，0 次无答案，3 秒，记录 25 个答案。

分析了什么

协议	ADRF subscribe and store analytics
版本	19.7.0
分析范围	一个请求方、ADRF、NWDAF 分析数据生产者；聚合非 UE 分析数据，StorageSubscriptionRequest 触发一个上游 NWDAF 订阅，通知存入 ADRF，随后是 StorageSubscriptionRemoval。包含真实上游 API 与 ID 映射；排除 DCCF/MFAF/UDM 同意分支。
不在范围内	符号模型不证明算法的计算安全或抗量子安全；攻击轨迹只在明确假设下成立，不直接等同规范漏洞。每个性质必须有可达性/正例对照。固定 Rel-19。分别区分外部网络攻击者与持有自身凭据的恶意合法 NF；TLS 认证加密按端点建模，合法 NF 不能读取其他 NF 的 TLS 会话。OAuth2 subject/audience/scope/expiry 及 TLS 身份绑定依照所附基线；服务 scope 不自动等于资源所有权。单调表不能模拟真实删除，有限状态生命周期必须明确界定。目标是指定 API 子流程，不是完整 5G 安全。
原文	1182 个编号原文单元（839894 字节），来自下文列出的各节。本报告中如 s0001 这样的引用就是这些原文单元之一

分析所依据的文稿各节：

文稿	章节	标题
main	4.2.2.3	Nadrf_DataManagement_StorageSubscriptionRequest 服务操作
main	4.2.2.3.1	概述
main	4.2.2.3.2	请求订阅并存储数据或分析
main	4.2.2.4	Nadrf_DataManagement_StorageSubscriptionRemoval 服务操作
main	4.2.2.4.1	概述
main	4.2.2.4.2	请求移除数据或分析的订阅
main	5.1	Nadrf_DataManagement 服务 API
main	5.1.1	引言
main	5.1.2	HTTP 的使用
main	5.1.2.1	概述
main	5.1.2.2	HTTP 标准头部
main	5.1.2.2.1	概述
main	5.1.2.2.2	内容类型
main	5.1.2.3	HTTP 自定义头部

文稿	章节	标题
main	5.1.3	资源
main	5.1.3.1	概览
main	5.1.3.2	资源：ADRF 数据存储记录
main	5.1.3.2.1	描述
main	5.1.3.2.2	资源定义
main	5.1.3.2.3	资源标准方法
main	5.1.3.2.3.1	POST
main	5.1.3.2.3.2	GET
main	5.1.3.2.4	资源自定义操作
main	5.1.3.3	资源：单个 ADRF 数据存储记录
main	5.1.3.3.1	描述
main	5.1.3.3.2	资源定义
main	5.1.3.3.3	资源标准方法
main	5.1.3.3.3.1	DELETE
main	5.1.3.3.4	资源自定义操作
main	5.1.3.4	资源：ADRF 数据检索订阅
main	5.1.3.4.1	描述
main	5.1.3.4.2	资源定义
main	5.1.3.4.3	资源标准方法
main	5.1.3.4.3.1	POST
main	5.1.3.4.4	资源自定义操作
main	5.1.3.5	资源：单个 ADRF 数据检索订阅
main	5.1.3.5.1	描述
main	5.1.3.5.2	资源定义
main	5.1.3.5.3	资源标准方法
main	5.1.3.5.3.1	DELETE
main	5.1.3.5.4	资源自定义操作
main	5.1.4	无关联资源的自定义操作
main	5.1.4.1	概览
main	5.1.4.2	操作：request-storage-sub
main	5.1.4.2.1	描述
main	5.1.4.2.2	操作定义
main	5.1.4.3	操作：request-storage-sub-removal
main	5.1.4.3.1	描述
main	5.1.4.3.2	操作定义
main	5.1.4.4	操作：remove-stored-data-analytics

文稿	章节	标题
main	5.1.4.4.1	描述
main	5.1.4.4.2	操作定义
main	5.1.5	通知
main	5.1.5.1	概述
main	5.1.5.2	检索通知
main	5.1.5.2.1	描述
main	5.1.5.2.2	目标 URI
main	5.1.5.2.3	标准方法
main	5.1.5.2.3.1	POST
main	5.1.5.3	ADRF 告警通知
main	5.1.5.3.1	描述
main	5.1.5.3.2	目标 URI
main	5.1.5.3.3	标准方法
main	5.1.5.3.3.1	POST
main	5.1.6	数据模型
main	5.1.6.1	概述
main	5.1.6.2	结构化数据类型
main	5.1.6.2.1	引言
main	5.1.6.2.2	类型：NadrfDataStoreRecord
main	5.1.6.2.3	类型：NadrfDataStoreSubscription
main	5.1.6.2.4	类型：NadrfDataRetrievalSubscription
main	5.1.6.2.5	类型：NadrfDataRetrievalNotification
main	5.1.6.2.6	类型：NadrfDataStoreSubscriptionRef
main	5.1.6.2.7	类型：NadrfStoredDataSpec
main	5.1.6.2.8	类型：DataSubscription
main	5.1.6.2.10	类型：StorageHandlingInfo
main	5.1.6.2.11	类型：NadrfAlertNotification
main	5.1.6.2.12	类型：NadrfAlertNotificationResponse
main	5.1.6.2.13	类型：DataSetTag
main	5.1.6.3	简单数据类型与枚举
main	5.1.6.4	描述备选数据类型或数据类型组合的数据类型
main	5.1.7	错误处理
main	5.1.7.1	概述
main	5.1.7.2	协议错误
main	5.1.7.3	应用错误
main	5.1.8	特性协商

文稿	章节	标题
main	5.1.9	安全
main	5.1.6.2.9	
ts-29-520	4.2.1	服务描述
ts-29-520	4.2.1.1	概览
ts-29-520	4.2.1.2	服务架构
ts-29-520	4.2.1.3	网络功能
ts-29-520	4.2.1.3.1	网络数据分析功能（NWDAF）
ts-29-520	4.2.1.3.2	NF 服务消费者
ts-29-520	4.2.2.2	Nnwdaf_EventsSubscription_Subscribe 服务操作
ts-29-520	4.2.2.2.1	概述
ts-29-520	4.2.2.2.2	订阅事件通知
ts-29-520	4.2.2.2.3	更新事件通知订阅
ts-29-520	4.2.2.3	Nnwdaf_EventsSubscription_Unsubscribe 服务操作
ts-29-520	4.2.2.3.1	概述
ts-29-520	4.2.2.3.2	取消订阅事件通知
ts-29-520	4.2.2.4	Nnwdaf_EventsSubscription_Notify 服务操作
ts-29-520	4.2.2.4.1	概述
ts-29-520	4.2.2.4.2	关于已订阅事件的通知
ts-29-520	5.1.1	引言
ts-29-520	5.1.2	HTTP 的使用
ts-29-520	5.1.2.1	概述
ts-29-520	5.1.2.2	HTTP 标准头部
ts-29-520	5.1.2.2.1	概述
ts-29-520	5.1.2.2.2	内容类型
ts-29-520	5.1.2.3	HTTP 自定义头部
ts-29-520	5.1.3.1	资源结构
ts-29-520	5.1.3.2	资源：NWDAF 事件订阅
ts-29-520	5.1.3.2.1	描述
ts-29-520	5.1.3.2.2	资源定义
ts-29-520	5.1.3.2.3	资源标准方法
ts-29-520	5.1.3.2.3.1	POST
ts-29-520	5.1.3.2.4	资源自定义操作
ts-29-520	5.1.3.3	资源：单个 NWDAF 事件订阅
ts-29-520	5.1.3.3.1	描述
ts-29-520	5.1.3.3.2	资源定义
ts-29-520	5.1.3.3.3	资源标准方法

文稿	章节	标题
ts-29-520	5.1.3.3.3.1	DELETE
ts-29-520	5.1.3.3.3.2	PUT
ts-29-520	5.1.3.3.4	资源自定义操作
ts-29-520	5.1.5	通知
ts-29-520	5.1.5.1	概述
ts-29-520	5.1.5.2	事件通知
ts-29-520	5.1.5.2.1	描述
ts-29-520	5.1.5.2.2	操作定义
ts-29-520	5.1.6.2.2	类型 NnwdafEventsSubscription
ts-29-520	5.1.6.2.3	类型 EventSubscription
ts-29-520	5.1.6.2.4	类型 NnwdafEventsSubscriptionNotification
ts-29-520	5.1.6.2.5	类型 EventNotification
ts-29-520	5.1.6.2.7	类型 EventReportingRequirement
ts-29-520	5.1.6.2.22	类型 NetworkPerfRequirement
ts-29-520	5.1.6.2.23	类型 NetworkPerfInfo
ts-29-520	5.1.6.2.30	类型 ThresholdLevel
ts-29-520	5.1.6.2.49	类型 ConsumerNfInformation
ts-29-520	5.1.6.3.3	枚举：NotificationMethod
ts-29-520	5.1.6.3.4	枚举：NwdafEvent
ts-29-520	5.1.6.3.10	枚举：NetworkPerfType
ts-29-520	5.1.6.3.26	枚举：TermCause
ts-29-520	5.1.7	错误处理
ts-29-520	5.1.7.1	概述
ts-29-520	5.1.7.2	协议错误
ts-29-520	5.1.7.3	应用错误
ts-29-520	5.1.8	特性协商
ts-29-520	5.1.9	安全

源文包中源文步骤在文稿中定位到的协议用途：一个 requester、ADRF、NWDAF analytics producer；聚合非 UE analytics，StorageSubscriptionRequest 触发一个上游 NWDAF subscription，通知存入 ADRF，再 StorageSubscriptionRemoval。包含真实上游 API 与 ID 映射；排除 DCCF/MFAF/UDM consent 分支。

对照文稿定性质

在编写查询之前，分析者将草拟的性质清单对照文稿本身对协议用途的说明进行了核对。名称、性质所涉及的内容以及文稿的说明均为分析者的解读；所提及的位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A02	ADRF 处上游对端认证	ADRF 确信它接受的订阅响应或事件通知来自真实的上游 NWDAF，且该 NWDAF 的预期对端是 ADRF。	ADRF 向 NWDAF 订阅（s0026），NWDAF 向它收到的 notificationURI 发送通知（s0913）；NF 服务流程应支持双向认证（ts-33-501 P219）；API 访问受 TLS 和 OAuth2 保护（s0334、s1138）。	s0026, s0334, s0913, s0961, s1138
A03	上游订阅参数协商一致	ADRF 与上游 NWDAF 就所创建的订阅达成一致：ADRF 的身份、所订阅的事件以及通知 URI。	ADRF 按消费者的参数创建上游订阅（s0026、s0018），NWDAF 以 201 Created 返回所创建的订阅和 Location（s0855）。	s0008, s0021, s0026, s0287, s0855
A08	存储记录绑定到请求者请求	ADRF 仅在其创建上游订阅所用的那个订阅（transRefId）下存储通知，并带上请求者给出的数据集标签，且仅在通知到达 ADRF 给出的通知 URI 时才存储。	ADRF 分配事务参考标识符（s0021），并将其映射到上游订阅（s0026）；数据集标签与订阅及所采集的数据相关联（s0020、s0313）。	s0020, s0021, s0026, s0313
K01	OAuth2 令牌保密性	请求者、ADRF 或 NWDAF 为 Nadrf_DataManagement 和 Nnwdaf_EventsSubscription API 持有的访问令牌不能被攻击者获知。	令牌从 NRF 获取，用于授权 API 访问（s0335、s0337、s1139）；访问令牌凭据必须保密（rfc6749 P011）。	s0334, s0335, s0337, s1139
Z01	API 操作授权	ADRF 仅在令牌的主体为请求 NF、受众为 ADRF、范围为存储范围时才接受存储订阅的创建或删除；NWDAF 仅在令牌为其自身时才接受上游订阅。	对 API 的访问可通过 OAuth2 客户端凭据及范围 nadrf-datamanagement:storage-read-delete-subs 和 nnwdaf-eventssubscription 授权（s0334、s0339、s1143）。	s0334, s0337, s0339, s1143
Z03	删除绑定到拥有主体	当 ADRF 为某个 transRefId 移除上游订阅时，该 transRefId 属于删除请求令牌的主体。	删除请求指明一个 transRefId 或 dataSetId（s0034），ADRF 移除映射到它的订阅（s0035）；文稿未规定将该标识符绑定到请求 NF 的规则。	s0034, s0035, s0293
S05	删除后生命周期	ADRF 接受删除后，该 transRefId 不再创建上游订阅，也不再接受已存储的通知；仍被另一个活跃 transRefId 引用的上游订阅不会被删除。	除非还有活跃的事务参考标识符映射到该订阅，否则删除操作会移除映射到该事务参考标识符的订阅（s0035）；已存储的数据由单独的操作移除（s0029、s0215）。	s0029, s0035, s0186, s0215
A08.8	记录身份对应性	ADRF 存储的记录带有请求者所创建订阅的 transRefId 和数据集标签，以及 ADRF 自身发给 NWDAF 的通知 URI。	ADRF 将分配的事务参考标识符映射到为该事务服务的订阅（s0026），并存储所通知的数据或分析（s0029、s0035）；消费者的参数用于上游订阅（s0018）。	s0018, s0021, s0026, s0035
U01	无跨消费者参数变更	恶意 NF 服务消费者 B 不能使 ADRF 创建或更改消费者 A 的订阅的采集或存储参数。	存储订阅携带消费者的订阅、目标和存储处理参数（s0008、s0017、s0019、s0020），ADRF 用它们创建上游订阅（s0026）；文稿未规定谁可以更改它们。	s0008, s0017, s0019, s0020, s0026
S05.10	删除不删除存储记录	StorageSubscriptionRemoval 停止采集并移除上游订阅，而已存储的数据或分析只能由 remove-stored-data-analytics 移除。	删除操作移除订阅（s0035）；单独的操作移除已存储的数据或分析（s0186、s0215）。	s0029, s0035, s0186, s0215
X01	传输中 API 消息的机密性与完整性	上游 NWDAF 交付给 ADRF 用于存储的分析，以及存储订阅和删除请求的参数	NWDAF 通过服务化接口向 notificationURI 发送通知（s0913）；NF 服务流程应支持机密性和完整性（ts-33-501 P217）。	s0913, s0961, s0334

分析者依据文稿补充的性质：X01.

文稿里的安全表述，逐段去向

宿主把文稿里说明协议保护什么、防什么的章节逐段列出，共 398 段；分析者对每一段说明了去向：定为性质、定为攻击者条件，或者为什么都不选。选了 42 段，没选 356 段。列哪些段落由标准的章节结构决定；每一段的去向是分析者的判断。

段落	出处	定为	不选的原因
P001 至 P002	rfc6749:2778-2786	没有陈述目标	引言；未对本协议提出任何要求。
P003 至 P006	rfc6749:2790-2823	`compromised_consumer`	
P007 至 P010	rfc6749:2827-2850	Z01, `compromised_consumer`	
P011 至 P015	rfc6749:2859-2881	K01, `compromised_consumer`	
P016 至 P021	rfc6749:2885-2917	不在范围内	刷新令牌属于其他授权类型；本协议使用客户端凭据授权。
P022 至 P029	rfc6749:2921-2980	不在范围内	授权码和重定向 URI 不与客户端凭据授权一起使用。
P030 至 P033	rfc6749:2984-3003	不在范围内	不使用资源所有者口令授权。
P034 至 P035	rfc6749:3013-3019	K01	
P036	rfc6749:3023-3028	A02, Z01	
P037 至 P039	rfc6749:3032-3041	证明器问不了	猜测客户端凭据：秘密的强度无法被询问；本协议使用 NF 凭据，而非用户口令。
P040 至 P050	rfc6749:3045-3138	不在范围内	网络钓鱼、CSRF 和点击劫持涉及用户代理；本协议没有用户代理。
P051 至 P052	rfc6749:3142-3150	没有陈述目标	对实现者的输入校验建议；不涉及此消息流的性质。
P053 至 P054	rfc6749:3154-3167	不在范围内	开放重定向；此处没有重定向端点。
P055 至 P062	rfc6749:3170-3210	不在范围内	不使用隐式授权。
P063	ts-33-501:2491-2491	证明器问不了	算法降级：算法的强度超出证明器所能被询问的范围。
P064 至 P072	ts-33-501:2495-2511	Z01, K01, `network`, `compromised_consumer`	
P073 至 P074	ts-33-501:2515-2517	不在范围内	5G 核心网和 NG-RAN 的密钥；本协议不使用任何密钥。
P075 至 P213	ts-33-501:2523-2853	不在范围内	接入和核心接口上的用户数据、信令数据、凭据和用户隐私；本任务仅覆盖 ADRF API 子流程。
P214	ts-33-501:2859-2859	没有陈述目标	它陈述了这些要求所依赖的信任区假设。

段落	出处	定为	不选的原因
P215 至 P220	ts-33-501:2865-2875	A02, Z01, `compromised_consumer`	
P221 至 P227	ts-33-501:2879-2891	Z01, K01, `compromised_consumer`	
P228 至 P241	ts-33-501:2895-2923	不在范围内	NEF 和 SCP 要求；NEF 被排除，且建模的是直接通信。
P242 至 P314	ts-33-501:2929-3081	不在范围内	漫游、SEPP、PRINS、IPUPS 以及网络间的属性保护；分析的是无漫游的单一网络。
P315 至 P318	ts-33-501:3085-3091	A02, `compromised_consumer`	
P319 至 P337	ts-33-501:3097-3141	没有陈述目标	信令监控、可见性和可配置性：属于运行要求，不涉及此消息流的性质。
P338 至 P375	ts-33-501:3149-3231	证明器问不了	算法标识符和选择：密码的强度超出证明器所能被询问的范围。
P376 至 P392	ts-33-501:6540-10909	不在范围内	N3、EAP-TLS 附着和 IAB 要求；不属于此 API 子流程。
P393 至 P398	ts-33-501:10913-11901	不在范围内	IAB 安全环境、MBSF 和 MBSTF 要求以及 NEF 安全；不属于此 API 子流程。

参与方和受保护的值，逐项是否设为攻击者条件

宿主把起草的性质里点到的参与方和受保护的值逐项列出，共 24 项；分析者对每一项说明了哪个攻击者条件控制这个参与方或持有这个值，或者为什么不设。设了攻击者条件的 17 项，没设的 7 项。清单来自性质本身；每一项怎么处理是分析者的判断。

ID	参与方或值	类别	攻击者条件	不设的原因
S01	NF service consumer	参与方	`compromised_consumer`	
S02	ADRF	参与方	`leaked_transref`	
S03	上游 NWDAF	参与方	`compromised_nwdaf`	
S04	NRF	参与方	按定义全部失守	本协议的所有令牌和 NF 凭据都来自授权服务器；NRF 被控制后，授权性质按定义即告丧失，文稿也未要求在 NRF 失守下满足其中任何一条。
S05	请求者 A	参与方	`compromised_consumer`	
S06	请求者 B	参与方	`compromised_consumer`	
S07	上游订阅关系	值	`compromised_nwdaf`	
S08	通知内容	值	`compromised_nwdaf`	
S09	订阅参数	值	`compromised_consumer`	
S10	transRefId 与上游订阅的映射	值	`leaked_transref`	
S11	targetNfId/targetNfSetId	值	攻击者得不到更多	targetNfId 是从 NRF 发现的公开 NF 实例标识符，不是秘密。
S12	dataSetTag/dataSetId	值	`leaked_transref`	
S13	notifCorrId	值	攻击者得不到更多	notifCorrId 是消费方自行选择的相关标识符；s0287 要求它按消费方唯一，而非保密。

ID	参与方或值	类别	攻击者条件	不设的原因
S14	transRefId	值	`leaked_transref`	
S15	OAuth2 访问令牌	值	按定义全部失守	令牌的持有者就是令牌签发给的一方；要求它不能使用自己的令牌等于没有要求。
S16	NF 凭据	值	`compromised_consumer`	
S17	存储订阅	值	`leaked_transref`	
S18	上游订阅	值	`compromised_nwdaf`	
S19	存储记录	值	不在范围内	读取存储记录属于检索子流程，任务未涵盖该流程。
S20	dataSetId	值	`leaked_transref`	
S21	已存记录	值	不在范围内	已存记录及其删除属于检索子流程与 remove-stored-data-analytics 子流程，任务未涵盖这些流程。
S22	上游订阅参数	值	`compromised_consumer`	
S23	存储处理参数	值	`compromised_consumer`	
S24	dataSetTag	值	攻击者得不到更多	dataSetTag 是消费方为其自身订阅选择的标签（s0020、s0313）；标签不是秘密，只有标识符绑定在 `leaked_transref` 下作答。

每条性质在各攻击者条件下的答案

攻击者由分析者设定，且在编写查询之前就已确定；每个答案都在其中某个攻击者下记录。单元格为证明器在该攻击者下对相应性质的答案；“无答案”表示分析者为该性质设定的某个攻击者下，证明器对某条主张未作判定或未记录任何内容；空单元格表示分析者未为该性质设定该攻击者。

攻击者	持有什么	依据	出处
`network`	仅网络：可读取、丢弃、更改和发送消息，不持有任何一方的秘密		
`compromised_consumer`	ADRF API 的一个已授权 NF service consumer 的凭据与存储状态	消费方持有 OAuth2 令牌和 TLS 凭据（s0334、s0335）；失陷消费方即范围中所称的恶意合法 NF	s0334, s0335
`compromised_nwdaf`	一个真实上游 NWDAF 分析生产者的凭据与订阅状态	NWDAF 通过 OAuth2 和 TLS 向 ADRF 认证自身（s1138、s1139）	s1138, s1139
`leaked_transref`	ADRF 分配给另一消费方存储订阅的事务引用标识符	s0293 将 transRefId 定义为普通字符串属性，且没有任何条款为其规定保密规则	s0021, s0034, s0293

ID	`network`	`compromised_consumer`	`compromised_nwdaf`	`leaked_transref`
A02	成立			
A03	成立			
A08	成立			
K01	成立			
Z01	成立			
Z03	成立			不成立
S05	成立			
A08.8	成立			

ID	network	compromised_consumer	compromised_nwdaf	leaked_transref
U01	成立			
S05.10	成立			
X01	成立			

原文和目标

原文：1182 个原文单元，839894 字节，已冻结于 source-delivery-01.json。 目标：8 个已批准（A02、A03、A08、K01、Z01、Z03、S05、U01）；所提问题：3 个。 目标列表事先按草稿获批，使用的是 --approve-targets。

宿主按其文档默认值作出的决定（任务 `decisions: auto`），供你核查：

- approve_package：批准源文包 protocol-97d3615022da9a42cbd490b5b948

本步骤自报：2 次请求（2 次失败），0.000 USD，输入 718123 个令牌，输出 9369 个，294 秒。

关于证明器和模型的说明

- Z03：在 ProVerif 中作答，分析者为该目标指定 changing_state
- S05：在 ProVerif 中作答，分析者为该目标指定 changing_state
- S05.10：在 ProVerif 中作答，分析者为该目标指定 changing_state

简化代数运算（xor、dh_algebra、other_algebra）的 ProVerif 模型赋予攻击者的能力少于真实运算：其中的证明是针对更弱攻击者的证明，其中的攻击仍是攻击。省略对会变化的值的检查（changing_state）的 ProVerif 模型使各参与方接受更多：其中的证明仍成立，其中的攻击可能不过是缺失了该检查。Tamarin+Agent 按原样对这些建模。

以步数表示的界是针对 bounded.pv 的答案：这是分析者的第二个 ProVerif 模型，其中持有会变化的值的一方展开到该步数，且其检查在位。该模型所做少于协议本身，因此性质仅在这些步数内成立，不能再多；其中发现的攻击仍是攻击。步数由分析者确定。

分析者的状态审查

本部分是分析者对原文自身的解读，而非证明器的答案。ProVerif 显示栏中的“是”表示 ProVerif 在展开模型中找到了带有这些步骤的轨迹，因此这些步骤是可能的；说这些步骤破坏某个目标，只由答案表中的“攻击”给出，它们还付出了什么代价则是分析者的解读。“未发现”只说明查看了哪些模式，并不证明任何事。

值	由谁持有	何时变化	何时检查	出处
transRefId 及其所映射到的上游订阅	ADRF	在受理 StorageSubscriptionRequest 时，ADRF 分配新的 transRefId，并记录其与所创建上游订阅的映射；受理的删除会清除该 transRefId 的作用（s0021、s0026、s0035）	当 StorageSubscriptionRemoval 指名它时，以及当通知必须归入其名下时（s0034、s0035）	s0021, s0026, s0034, s0035, s0036
上游订阅标识符 subscriptionId 及其通知 URI	上游 NWDAF	NWDAF 在 201 Created 时创建它，在受理的 HTTP DELETE 上删除它（s0855、s0898）	在每个指名该 subscriptionId 的 HTTP DELETE 上，以及该订阅的每条通知上（s0897、s0913、s0956）	s0855, s0897, s0898, s0901

值	由谁持有	何时变化	何时检查	出处
含数据集标签及其生命周期的已存记录	ADRF	在受理的删除后，ADRF 停止按该 transRefId 归档数据；已存记录只能由 remove-stored-data-analytics 删除（s0035、s0215）	当某条通知要为该订阅存储时，以及当存储数据被检索或删除时（s0961、s0962）	s0029, s0035, s0186, s0215
订阅的存储处理信息（storeHandl）与数据集标签	ADRF	ADRF 在受理时确定存储方式，并可在响应中更改（s0024）	当存储生命周期应用于所收集的数据时（s0024、s0025）	s0024, s0025

模式	结果	目标	ProVerif 是否显示	步骤
重放	未发现			按原文单元的规则，已接受移除之后的通知会被拒绝（s05 已验证），重复的移除会被消费而不会发生第二次移除（s05b 已验证）；没有替换值的对照查询会将其存储（ctl_s05 已反证）。
重复一次性操作	未发现			重复的移除会再次到达 ADRF，但不会被处理，也不会重复删除上游订阅；ADRF 对它不作应答，原文单元对此没有规定。
失步	候选		否：未用 ProVerif 尝试	消费者携带其 transRefId 发送 request-storage-sub-removal（s0034）。/ ADRF 接受该请求并立即应答 204 No Content（s0035）。/ 随后 ADRF 为上游订阅发送 HTTP DELETE（s0035、s0897）。/ 如果 DELETE 失败，NWDAF 应答 HTTP 错误并保留订阅（s0901），且继续就该订阅发送通知（s0913）。/ 消费者认为收集已经停止，而 NWDAF 仍保留订阅；ADRF 拒绝后续通知，因此不再存储任何内容，但没有任何条款要求它重试删除或告知消费者。
先更新后校验	候选		否：未用 ProVerif 尝试	消费者携带 targetNfId 和 anaSub 发送 request-storage-sub（s0008）。/ ADRF 分配 transRefId，并在创建上游订阅之前就携带它应答 200 OK（s0021）。/ 只有在此之后，ADRF 才在 NWDAF 处创建订阅（s0026）。/ 如果该创建失败，则 transRefId 在消费者处和 ADRF 处都存在，但没有上游订阅，原文单元对此没有规定任何规则。
失败时无回滚	候选		否：未用 ProVerif 尝试	ADRF 分配 transRefId 并应答 200 OK（s0021）。/ 上游订阅的创建失败或被放弃（s0026、s0023）；没有任何条款删除 transRefId 或告知消费者。/ 指定该 transRefId 的移除请求会被接受并应答 204（s0035），尽管没有任何内容映射到它。/ 消费者无法得知上游订阅从未存在过。
会话之间的竞争	候选		否：未用 ProVerif 尝试	消费者 A 为某个数据集发送 request-storage-sub；ADRF 创建上游订阅（s0026）。/ 消费者 B 为同一数据或分析发送 request-storage-sub；根据 s0022 和 s0026，ADRF 分配第二个 transRefId，并且可能不会创建第二个上游订阅，而是将两个 transRefId 都映射到已存在的那个订阅。/ 消费者 A 为其 transRefId 发送移除请求；由于 B 的 transRefId 仍映射到该上游订阅，s0035 会保留该订阅。/ 有界模型中每个上游订阅只有一个 transRefId，因此这一保留规则以及两次移除之间的竞争没有被覆盖。
可区分的失败	候选		否：未用 ProVerif 尝试	消费者 B 指定一个猜测的 transRefId 发送 request-storage-sub-removal（s0034）。/ 如果有订阅映射到该 transRefId，ADRF 应答 204 No Content，否则应答 HTTP 错误响应（s0035、s0036）。/ 这两种应答告诉 B 所猜测的 transRefId 是否存在，而且只要存在就会将其删除，不论其所有者是谁（Z03 攻击）。
回绕或窗口	未发现			transRefId 和 dataSetId 是字符串，在原文单元中没有范围、计数器或接受窗口。

分析者为目标命名的内容：

步骤	目标	特征	原因
40	S05、S05.10、Z03	changing_state	S05、S05.10 以及 Z03 的移除绑定主张都依赖于 ADRF 的订阅生命周期：transRefId 和上游订阅映射只创建一次，由移除请求删除，并且在移除之后到达的通知不得被存储。model.pv 不保留此类状态（表只增长，且 ProVerif 会忘记某一步运行了多少次），因此这些答案来自 bounded.pv，其中 ADRF 被展开为固定数量的步骤。
40	A02、A03、A08、A08.8、K01、Z01、U01、X01	plain_primitives、many_sessions	A02、A03、A08、A08.8、K01、Z01、U01 和 X01 只使用一个 MAC、一个用于 TLS 的认证加密抽象，以及每个端点身份一个私钥函数；这些性质针对两个消费者、ADRF 和 NWDAF 的任意数量的会话提出。不涉及算术定律，因此发现的攻击也会是对协议的攻击，而证明也是针对该演算的证明。

分析者的结尾说明

全部十一个目标都已作答。其中十个在网络攻击者单独作用下成立，且每个在存在变体时都配有对照查询：上游 NWDAF 的对端认证（A02）、上游订阅参数的一致性（A03）、存储的通知与消费者请求、数据集标签及所映射订阅的绑定（A08、A08.8）、令牌保密性（K01）、两个 API 的主体/受众/范围绑定（Z01）、已接受移除之后的生命周期以及移除与已存储记录的分离（S05、S05.10，展开为三步）、无跨消费者参数更改（U01），以及传输中消息的机密性和完整性（X01）。Z03 有两种解读：对网络攻击者成立，因为 transRefId 无法被猜测；对知道另一消费者 transRefId 的消费者不成立，因为 ADRF 会依据任何有效令牌删除上游订阅，而原文单元没有规定所有权规则。该攻击是主要发现，记录为 holds_in_source；建议（记录所属消费者，并在移除时将其与令牌主体比较）已在展开模型的副本上检查，其中该攻击查询变为已验证，且其他已记录的答案仍然成立。第二个发现是从原文单元读出的，而不是由查询显示的：存储订阅在上游订阅存在之前就被确认，且没有回滚。仍未建模的内容：被攻陷的消费者和被攻陷的生产者、一个上游订阅映射到多个 transRefId 时的 s0035 共享规则、令牌过期，以及被排除的 DCCF/MFAF/UDM 分支；TLS 被理想化为每对端点一个密钥。