DCCF 分析可用性与获取的安全性分析

本报告的其他语言版本：English。

19.7.0 · ProVerif+ Agent 报告

文稿	5G System; Data Collection Coordination 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	1

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

TS 29.520 的 Ndccf_DataManagement API 允许 NF 服务消费者在 DCCF 处订阅聚合分析数据，这些数据由 DCCF 从 NWDAF 收集。该流程涉及三方：NF 服务消费者持有其 OAuth2 访问令牌和 TLS 凭据；DCCF 持有已存储的订阅和已收集的分析数据；NWDAF 产出分析数据。

消费者发起订阅，并指定通知地址 anaNotifUri 与通知关联标识 anaNotifCorrId。DCCF 回应订阅，向上游发起订阅；当分析数据就绪时，DCCF 通知消费者，但不直接给出数据，而是给出由 fetchUri 和 fetchCorrIds 组成的获取指令。消费者确认该通知，随后向获取 URI 提交获取关联标识，以收集分析数据。

交换结束后，消费者持有分析数据；这些数据是 DCCF 在其自身订阅的通知路径上从 NWDAF 收到的，并与其自身的关联标识绑定。DCCF 持有一份订阅，其内容由经过认证的消费者提议；DCCF 只对其自身签发的标识回应 Fetch，并且只回应给出示这些标识的消费者。

涉及的各方：

- **NF 服务消费者**：订阅分析数据并收集这些数据。持有：OAuth2 访问令牌、TLS 凭据。
- **DCCF**：协调收集并通知可用性。持有：已存储的订阅、已签发的 fetchCorrId、已存储的分析数据。
- **NWDAF**：产出分析数据。持有：分析数据来源。

图 1 展示了文稿所规定的交换流程。

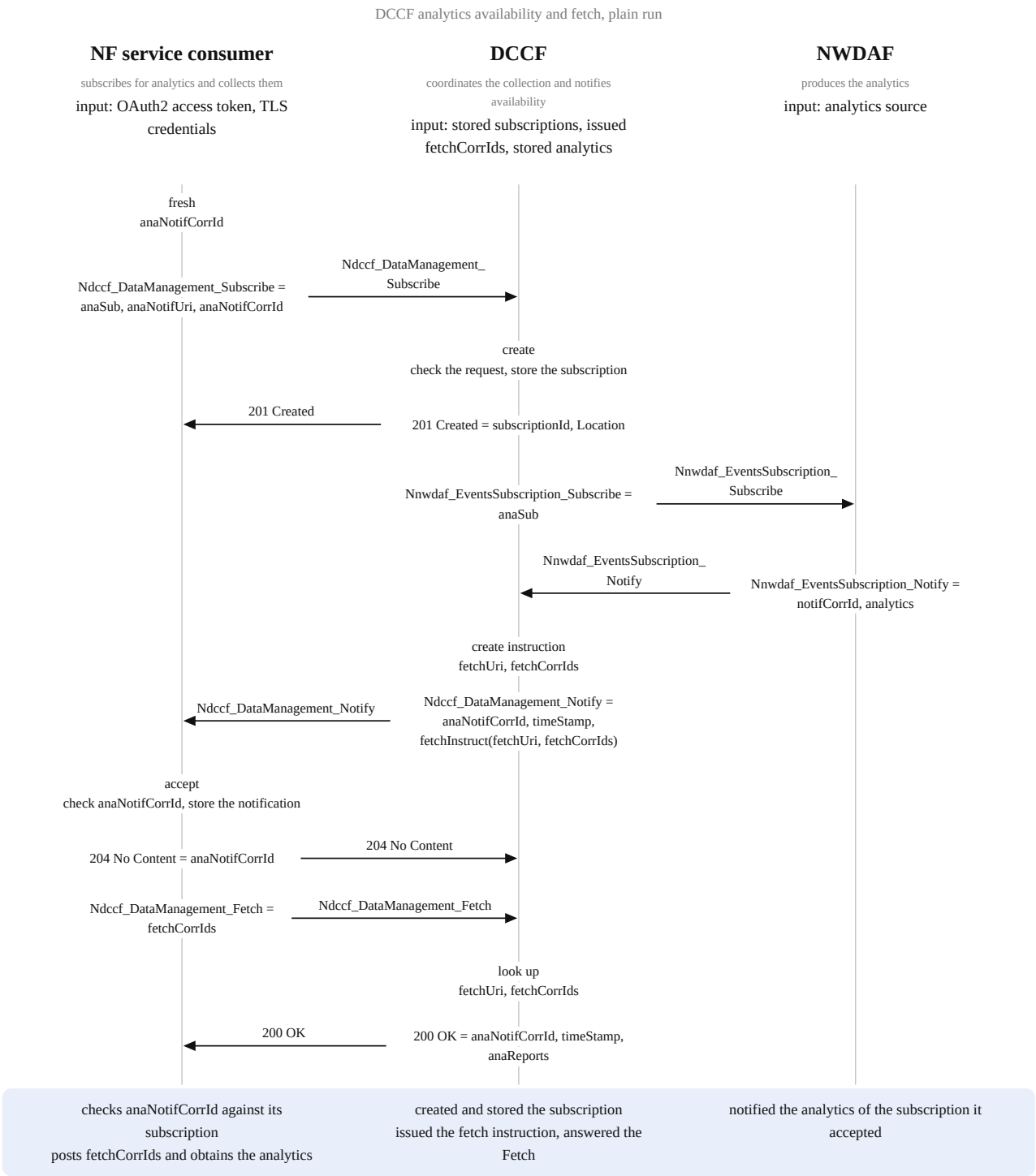


图 1. 文稿所规定的协议。

分析范围见附录 D：分析了什么、遗漏了什么，以及所依据的文稿章节。

2 发现

本次分析有 1 项发现：以下内容在文稿中没有写明。其中 1 项，文稿是否写明并不明确。

在已分析的范围内，该流程提供了它应当提供的保证：DCCF 仅为经过认证的 NF 实例行事；消费者的通知与 Fetch 都绑定到其自身的订阅；没有消费者能获得他人的分析数据。有一项发现：DCCF 仍会回应在订阅更新之前签发的获取指令，因为文稿中没有任何内容撤回该指令。消费者之间的隔离依赖于获取关联标识

的保密性与不可预测性，而文稿将这些标识定义为普通字符串，从未将其绑定到发起请求的 NF。全部 11 个目标都有答案，其中 10 个在任意数量会话的模型上得到证明，1 个在展开模型上被展示为攻击。这些证明依赖于所建模的 TLS 与 OAuth2；对于会变化的值，检查仅在一个订阅、一次更新的情形下完成。

攻击图使用图 1 的列。红色表示攻击者的行为。黄色标记的是某一方发送的消息，而规定的交换在该时点并没有这条消息；黄色还标记攻击所破坏的结果。

2.1 获取指令在签发时所对应的订阅状态失效后仍然有效

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	低
涉及的性质	U03 （见第 3 节）
原文	s0077, s0079, s0080, s0149, s0162
依据	ProVerif attack query `u03_old`; 图 2

发生了什么。 某条 Fetch 的 fetchUri 与 fetchCorrIds 是在订阅被修改之前签发的，而修改后，已存储的指令被替换为新指令，DCCF 仍会回应这条 Fetch。第 4.2.2.5.2 条要求 DCCF 根据请求查找分析数据，并且没有为不再有效的指令规定任何错误。该轨迹是在展开模型上展示的；在 PUT 时撤回这些标识的修改可以消除它。

攻击者需要什么。 消费者保留其在修改订阅之前收到的获取指令，并再次出示该指令。不需要攻击者：DCCF 会用已存储的分析数据回应它。

攻击者能得到什么。 消费者已不再持有的订阅状态所对应的分析数据，仍可通过文稿从未撤回的指令取回；任何之后获得该指令的一方都能取回它，而消费者永远不会被告知其指令已过期。

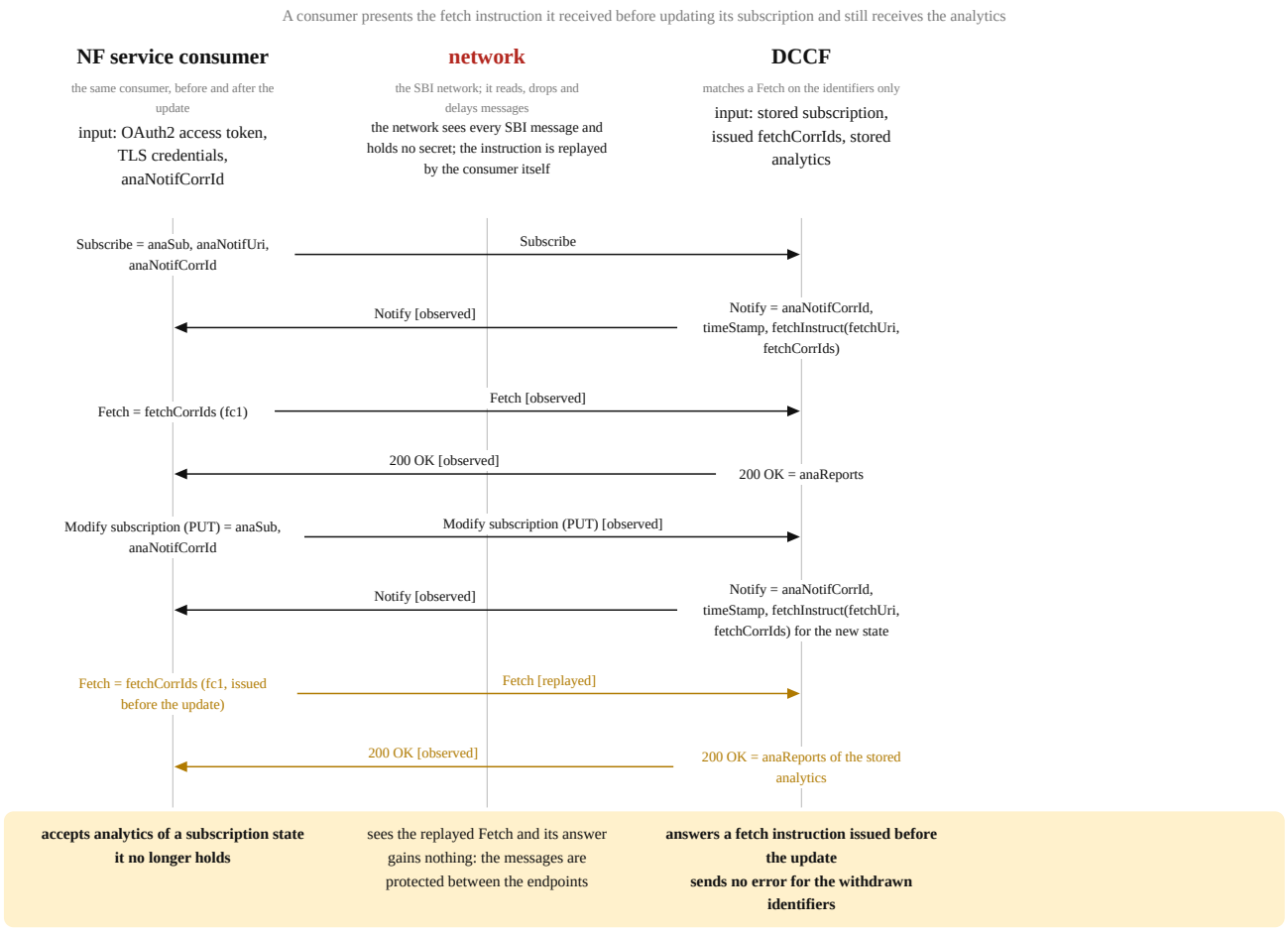


图 2. 消费者出示其在更新订阅之前收到的获取指令，仍然收到分析数据。性质 U03；攻击查询 u03_old。

拟议修改。 在订阅修改（PUT）和删除（DELETE）时，DCCF 撤回在变更之前签发的获取关联标识；携带已撤回标识的 Fetch 会收到 HTTP 错误响应，而不是已存储的分析数据。

用 ProVerif 复核。 该修改被加入模型的副本中：在订阅更新时，DCCF 撤回在更新之前签发的获取关联标识；订阅修改（PUT）会替换已存储的指令；携带较早的 fetchUri 或 fetchCorrIds 的 Fetch 不会得到回应。在修改后的模型上，ProVerif 证明了 `u03_old`：那里不再可能发生该攻击。这是对模型的初步检查：它表明该修改在模型中消除了这一攻击，并不表明它不会引入其他问题。修改后的模型：`records/fixes/0053-bounded-fix.pv`。

3 检查过的性质

每一行都是一条提交给 ProVerif 的性质。结果是在分析者所建模型上 ProVerif 给出的答案；价值以及文稿是否已经写明该结果，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	**对端身份认证。** 每当 DCCF 接受一个订阅或 Fetch 请求时，声称的消费者 NF 实例必须确实以该 DCCF 为对象运行过该流程；每当消费者接受一条通知时，DCCF 必须确实以该消费者为对象运行过该流程。只要求身份匹配，不要求具体参数一致。	无	成立	低	是
A03	**关联标识符协商。** 消费者接受的每条通知，都存在 DCCF 的一次真实运行，其订阅标识、anaNotifCorrId 与消费者本地值一致；DCCF 处理的每个 Fetch 请求，都存在消费者的一次真实运行，其 fetchCorrIds 与 DCCF 先前在通知中给出的值一致。	无	成立	低	不确定

ID	性质	假定的泄露	结果	价值	文稿是否已写
A04	**消息不被重复接受。** 消费者接受的每条通知、DCCF 处理的每个 Fetch 请求，都对应不同的对端真实运行，不存在同一逻辑消息被重复接受的情况。	无	成立	低	不确定
A08	**订阅参数绑定。** DCCF 接受的订阅参数对应消费者真实提议的值；通知中的 anaNotifUri、anaNotifCorrId 与订阅时一致；Fetch 响应中的 analytics 对应请求中的 fetchCorrIds 所指向的订阅。	无	成立	低	不确定
K01	**访问令牌与TLS密钥的保密性。** 攻击者不能推导出消费者的 OAuth2 token、DCCF 与消费者之间的 TLS 会话密钥，也不能把 token 用于未授权的服务或操作。	无	成立	低	是
Z01	**受保护操作前的授权。** DCCF 提交的每个受保护操作（创建订阅、投递通知、回应 Fetch）都必须有 NRF 签发的、覆盖该操作与资源的有效授权；没有有效授权的请求不能产生这些操作。	无	成立	中	不确定
Z03	**授权绑定到资源与操作。** 每个被授权的操作，其授权主体、受众、资源（订阅标识）、操作（订阅/通知/Fetch）与使用上下文必须与授权时一致；token 或关联标识不能被移到其他消费者、其他订阅或其他操作上。	无	成立	中	不确定
Z05	**消费者域之间的隔离。** 允许控制一个消费者域（其订阅、通知、Fetch 请求），对另一个消费者域重新检查：该消费者不能获得不属于自己的订阅的 analytics，也不能用自己的 fetchCorrIds 取到其他订阅的数据。	无	成立	中	不确定
U01	**所获取分析数据的上游来源。** 消费者 C 的每次 Fetch 接受，都对应 DCCF 从 C 指定的上游 NWDAF 收到 analytics 并向 C 发出含相同 fetchCorrIds 的通知。	无	成立	低	否
U02	**获取属于发起请求的消费者。** 消费者 B 的每次 Fetch 接受，其 fetchCorrIds 与订阅都属于 B；B 不能取回 A 的 analytics。	无	成立	中	不确定
U03	**通知与获取绑定到当前订阅。** 消费者 C 的每次通知或 Fetch 接受，都对应 C 当前有效的订阅，且关联标识一致；旧订阅或已更新订阅的旧通知/指令不能被接受。	无	不成立（1 个攻击）	低	不确定

所提出的问题，以及分析者的回答：

问题	回答	依据
Q1: 上游 analytics、下游 fetchCorrId 与消费者身份绑定	上游路径是健全的：消费者获取的分析数据，是 DCCF 在该消费者的通知路径上从 NWDAF 收到的；消费者据以行动的指令，也是 DCCF 为该消费者签发的。获取关联标识与消费者身份之间的绑定并未强制执行：DCCF 只匹配标识，因此这一保证依赖于标识不可猜测以及 TLS，尽管文稿将这些标识定义为字符串。	U01, U02, Z03
Q2: 其他合法 NF 的 fetch 隔离	消费者 B 无法取回消费者 A 的分析数据：每次 Fetch 都需要一个对发起请求的 NF 有效的令牌，以及发给 A 的标识，而 B 从未收到这些标识；用网络提供的值进行探测不会返回任何内容。隔离依靠的是标识的保密性，而不是检查请求者是否拥有该订阅。	Z05, U02, Z03
Q3: 旧通知/指令不会关联到不同或已更新订阅	通知不能被附加到另一个订阅，因为消费者会将 anaNotifCorrId 与自己的值进行比较；但在订阅更新之前签发的指令在更新之后仍会得到回应，因此与当前订阅的绑定只实现了一半。	U03, A03

4 本次分析的局限

这里的结果是协议模型上的结果：它说明规定的消息和检查在密码学被视为理想的情况下提供或未能提供什么，而不涉及任何实现。

本次分析不覆盖的内容，按分析者的陈述如下：

- TLS 被建模为认证加密，密钥由两个端点身份派生；TS 29.500 不属于被建模的文本，TLS 重放保护、重协商和证书均未被建模。

- OAuth2 被建模为 NRF 签发的令牌，其主体必须等于发起请求的 NF；令牌过期、受众、scope 字符串和客户端凭据交换均未被建模。
- DCCF 在每个会话的每个进程实例中处理一个订阅；其订阅表、并发访问以及第 5.1.6.2.15 条中的存储处理均未被建模。
- 展开模型覆盖一个订阅带一次更新；删除、重复更新以及一个订阅的多条通知不在其范围内。
- 建模了一个 NWDAF 与一个消费者对：MFAF、多上游复用和 ADRF 被排除在范围之外，targetNfId 或 targetNfSetId 中指定的 NWDAF 不发生变化。
- 用户同意分支（Nudm_SDM、403 USER_CONSENT_NOT_GRANTED）和 400 SUBSCRIPTION_CANNOT_BE_SERVED 路径未被建模。
- 重定向响应（307、308）未被建模。
- 分析数据的真实性未被建模：受攻击者控制的 NWDAF 可以提供任意值，且没有目标要求其正确性。
- 原语的强度未被建模，关联标识符被当作不可猜测的名称处理，尽管文本将其类型定为字符串。

宿主记录的内容：

- A04、U03 的答案依赖于模型简化的一项操作（附录 D）。
- 没带对照查询的证明（附录 C）：A02: a02_notif, A02: a02_fetch, A04: a04_fetch, K01: k01_tok, U03: u03_notif。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、什么能解决它们，见附录 A。

附录 A：逐条性质的解读

这是分析者本人对证明器答案的解读，而非证明器的答案：这些答案对协议意味着什么、解读依据什么、能否按原样使用，以及它的价值如何。

A02: 对端身份认证

每当 DCCF 接受一个订阅或 Fetch 请求时，声称的消费者 NF 实例必须确实以该 DCCF 为对象运行过该流程；每当消费者接受一条通知时，DCCF 必须确实以该消费者为对象运行过该流程。只要求身份匹配，不要求具体参数一致。

DCCF 只处理那些经由消息所声称身份的 NF 实例的 TLS 会话传来的订阅和 Fetch 请求，消费者只接受来自其订阅所属 DCCF 的通知。仅校验身份；交换的参数没有任何一致性约定。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值低：TS 33.501 要求 NF 服务消费者与生产者之间进行双向认证；证明确认消息设计没有丢失这一绑定。
- 文稿中已陈述：是。文稿中的位置：ts-33-501:2871
- 依据：TLS 被建模为以两个端点身份的私有函数为密钥的认证加密；OAuth2 令牌被建模为其他方无法产生的 NRF 值
- 下一步：对令牌主体校验和 TS 29.500 的 TLS 绑定建模，以确认 DCCF 使用的身份正是其对端认证时所用的身份。

A03: 关联标识符协商

消费者接受的每条通知，都存在 DCCF 的一次真实运行，其订阅标识、anaNotifCorrId 与消费者本地值一致；DCCF 处理的每个 Fetch 请求，都存在消费者的一次真实运行，其 fetchCorrIds 与 DCCF 先前在通知中给出的值一致。

消费者接受的通知携带的是 DCCF 确实为其创建的订阅的通知关联标识符，Fetch 只针对 DCCF 确实在通知中签发的 fetch 关联标识符来服务。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值低：文稿陈述了关联要求，但从未将其作为安全目标；一致性来自对新鲜值的匹配。
- 文稿中已陈述：不明确
- 依据：关联标识符被建模为新鲜且不可猜测的值；每个订阅一个 DCCF 实例；anaNotifCorrId 在 201 响应中被回显，模型正是以此将应答绑定到请求
- 下一步：关联本身无需下一步；标识符的时效性在 U03 下回答。

A04: 消息不被重复接受

消费者接受的每条通知、DCCF 处理的每个 Fetch 请求，都对应不同的对端真实运行，不存在同一逻辑消息被重复接受的情况。

在建模的运行中，每条通知只被接受一次，每条指令只被服务一次，因此没有消息会被处理两次。重复被模型的形状排除，而非被文本的规则排除；指令的重复使用在 U03 下展示。

- 结果：成立；回答了所问：部分；可在有保留的情况下使用

- 价值低：对每个请求只应答一次的会话做证明，对重复几乎说明不了什么，而且文本没有陈述新鲜性规则。
- 文稿中已陈述：不明确
- 依据：TLS 重放保护未被建模；流程是每个订阅一条线性会话，因此重复消息必须通过展开来建模
- 下一步：TLS 记录模型能解决受保护的消息是否可被重放；展开模型能展示指令是否可能被有意使用多次。

A08: 订阅参数绑定

DCCF 接受的订阅参数对应消费者真实提议的值；通知中的 anaNotifUri、anaNotifCorrId 与订阅时一致；Fetch 响应中的 analytics 对应请求中的 fetchCorrIds 所指向的订阅。

消费者接受的通知携带其提议的通知地址和关联标识符，DCCF 存储的订阅就是被请求的那个订阅。Fetch 上交付的数据就是该订阅存储的分析数据。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值低：通过新鲜值进行绑定是结构性的；答案表明消息携带了正确的参数。
- 文稿中已陈述：不明确
- 依据：每个订阅一个 DCCF 实例；订阅内容按第 4.2.2.2.2 条的要求取自请求体
- 下一步：无。

K01: 访问令牌与TLS密钥的保密性

攻击者不能推导出消费者的 OAuth2 token、DCCF 与消费者之间的 TLS 会话密钥，也不能把 token 用于未授权的服务或操作。

消费者的访问令牌和 TLS 会话密钥不会落入攻击者之手，一个 NF 的令牌不会被接受用于另一个 NF 的请求。

- 结果：成立；回答了所问：是；可在有保留的情况下使用
- 价值低：结果来自所建模的原语；文稿将授权令牌列为需要保密保护的值。
- 文稿中已陈述：是。文稿中的位置：ts-33-501:3071
- 依据：令牌在模型中是私有函数值；TLS 被建模为认证加密
- 下一步：无。

Z01: 受保护操作前的授权

DCCF 提交的每个受保护操作（创建订阅、投递通知、回应 Fetch）都必须有 NRF 签发的、覆盖该操作与资源的有效授权；没有有效授权的请求不能产生这些操作。

每个订阅和每个 Fetch 都需要请求 NF 的有效令牌。由于该 API 为整个服务定义了一个 scope，授权无法区分订阅与 Fetch，也无法区分一个消费者的订阅与另一个消费者的订阅。

- 结果：成立；回答了所问：部分；可在有保留的情况下使用
- 价值中：它机械地表明第 5.1.9 条的单一 scope 使操作层面得不到保护。
- 文稿中已陈述：不明确
- 依据：建模的校验是令牌主体等于请求身份；令牌过期、受众和 scope 字符串未被建模
- 下一步：来自 TS 29.501 的 scope 与受众模型能解决是否有可能按资源进行任何绑定。

Z03: 授权绑定到资源与操作

每个被授权的操作，其授权主体、受众、资源（订阅标识）、操作（订阅/通知/Fetch）与使用上下文必须与授权时一致；token 或关联标识不能被移到其他消费者、其他订阅或其他操作上。

令牌绑定到其携带身份所对应的 NF，因此一个消费者无法使用另一个消费者的令牌。fetch 关联标识符与消费者或操作的绑定未被强制执行：DCCF 只匹配标识符，且它们的类型是字符串，所以保密就是全部保护。

- 结果：成立；回答了所问：部分；可在有保留的情况下使用
- 价值中：它揭示出资源绑定依赖于一个未陈述的假设，编辑应就此作出决定。
- 文稿中已陈述：不明确
- 依据：标识符被建模为不可猜测的名称，尽管文本将其类型定为字符串；整个 API 只有一个 scope
- 下一步：让攻击者能够猜测关联标识符的模型能解决若它们可预测还剩下什么。

Z05: 消费者域之间的隔离

允许控制一个消费者域（其订阅、通知、Fetch 请求），对另一个消费者域重新检查：该消费者不能获得不属于自己的订阅的 analytics，也不能用自己的 fetchCorrIds 取到其他订阅的数据。

在模型中，一个消费者域无法获得另一个消费者的分析数据，进行探测的合法 NF 在 fetch 端点上什么也得不到。这一点成立是因为标识符从未离开签发给的消费者，而不是因为 DCCF 检查了订阅的归属。

- 结果：成立；是否回答了所问：部分；可在有保留的情况下使用
- 价值 中：模型中的隔离是真实的，但它依赖文本未声明的假设，而该 API 的读者需要知道这一点。
- 文稿中是否声明：不明确
- 依赖：fetch 关联标识是不可猜测的新鲜名称；TLS 将指令保持在消费者的会话内；DCCF 不比较请求者与订阅者，因为文本不要求这样做
- 下一步：将关联标识建模为可猜测，看看恶意 NF 届时能获得什么。

U01: 所获取分析数据的上游来源

消费者 C 的每次 Fetch 接受，都对应 DCCF 从 C 指定的上游 NWDAF 收到 analytics 并向 C 发出含相同 fetchCorrIds 的通知。

消费者获取的 analytics 是由 DCCF 从该消费者通知路径上的 NWDAF 收到的，它据以行动的指令正是为该通知发出的那条。模型中只有一个 NWDAF，因此上游 NWDAF 的选择本身未被检验。

- 结果：成立；是否回答了所问：部分；可在有保留的情况下使用
- 价值 低：在文稿中查找关于 DCCF 必须使用订阅中所指定 NWDAF 的声明，未找到。
- 文稿中是否声明：否
- 依赖：单一 NWDAF 身份；通知路径是线性会话
- 下一步：对第二个 NWDAF 和 targetNfId 属性建模，以询问 DCCF 是否必须使用消费者指定的 NWDAF。

U02: 获取属于发起请求的消费者

消费者 B 的每次 Fetch 接受，其 fetchCorrIds 与订阅都属于 B；B 不能取回 A 的 analytics。

消费者获得的 analytics 属于该消费者创建的订阅，因此在建模流程中一个消费者无法取回另一个消费者的 analytics。DCCF 从不比较请求者与订阅者；答案来自标识从不共享。

- 结果：成立；是否回答了所问：部分；可在有保留的情况下使用
- 价值 中：它明确说明订阅的所有权不是 Fetch 操作所检查的性质。
- 文稿中是否声明：不明确
- 依赖：标识不可猜测；TLS 将指令保持在消费者的会话内
- 下一步：将标识建模为可猜测，或建模一个读取另一消费者通知的消费者，看看缺失的所有者检查是否可被利用。

U03: 通知与获取绑定到当前订阅

消费者 C 的每次通知或 Fetch 接受，都对应 C 当前有效的订阅，且关联标识一致；旧订阅或已更新订阅的旧通知/指令不能被接受。

通知部分成立：消费者只对其创建的订阅的通知采取行动，并比较自己的关联标识。时效性部分不成立：订阅更新后，DCCF 仍会回答更新前发出的指令，且消费者不会被告知该指令已过期。

- 结果：不成立（1 个攻击）；是否回答了所问：部分；可在有保留的情况下使用
- 价值 低：缺口真实但不大：它需要一个保留旧指令的消费者，而文本中没有任何内容禁止返回该指令。
- 文稿中是否声明：不明确
- 依赖：展开模型只运行一次更新，而非删除；原文未声明失效规则；第二阶段规范描述了 fetch 指令中的截止时间，而本文本未包含该内容
- 下一步：对订阅的删除和第二次更新建模，看看同样的结论是否适用于 DELETE 和重复更新。

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、是否检查了对照，以及每个攻击及分析者的判定。

目标	级别	界	查询	对照	攻击	主张
A02	proved		a02_sub	5 中的 3		5 个 proved
A03	proved		a03_notif	checked		3 个 proved
A04	proved		a04_notif	4 中的 3		4 个 proved
A08	proved		a08_notif	checked		2 个 proved
K01	proved		k01_tok	4 中的 2		4 个 proved
Z01	proved		z01_sub	checked		3 个 proved
Z03	proved		z03_tok	checked		3 个 proved
Z05	proved		z05_data	checked		2 个 proved
U01	proved		u01_path	checked		3 个 proved
U02	proved		u02_data	checked		1 个 proved
U03	proved		u03_notif	2 中的 1	u03_old (depends_on_ambiguity)	2 个 proved，1 个 attack

逐条主张，按分析者对每条主张的措辞；每条主张的答案是证明器给出的：

目标	主张	攻击者	回答	查询	对照
A02	DCCF 接受的每个订阅都是由那个向它发起 Subscribe 流程的 NF 实例提议的（只做身份匹配，不要求参数一致）；攻击者只拥有网络。	`network`	proved	a02_sub	checked
A02	消费者接受的每条通知，都是 DCCF 针对该关联标识符发送给该消费者的，因此消费者的对端就是其订阅所属的 DCCF；攻击者只拥有网络。	`network`	proved	a02_notif	none
A02	DCCF 服务的每个 Fetch 都是由其自身 Fetch 请求所属的那个消费者请求的，因此持有自身凭据的合法 NF 无法让 DCCF 为它提供并非它发起的请求。	`malicious_nf`	proved	a02_fetch	none
A02	消费者接受的每条通知，都是 DCCF 针对该关联标识符发送给该消费者的；攻击者只拥有网络。	`network`	proved	a02_notif	checked
A02	DCCF 服务的每个 Fetch 都是由其自身 Fetch 请求所属的那个消费者请求的；攻击者是持有自身凭据的合法 NF。	`malicious_nf`	proved	a02_fetch	checked
A03	每条被接受的通知都携带 DCCF 确实为该消费者创建的某个订阅的 anaNotifCorrId。	`network`	proved	a03_notif	checked
A03	DCCF 服务的每个 Fetch 都携带一个 fetch 关联标识符，该标识符是 DCCF 此前确实在某条通知中签发的。	`network`	proved	a03_fetch	checked
A03	出示 fetch 关联标识符的合法 NF 无法让 DCCF 为一个并非签发给它的标识符提供 Fetch。	`malicious_nf`	proved	a03_fetch	checked
A04	对每个已签发的 fetch 关联标识符，Fetch 至多被服务一次，即使合法 NF 出示了先前发给它的标识符也是如此；攻击者只拥有自己的凭据。	`malicious_nf`	proved	a04_fetch	none

目标	主张	攻击者	回答	查询	对照
A04	对 DCCF 签发的每条通知，消费者至多接受一次。	network	proved	a04_notif	checked
A04	对 DCCF 签发的每个 fetch 关联标识符，Fetch 至多被服务一次。	network	proved	a04_fetch	checked
A04	对每个已签发的 fetch 关联标识符，Fetch 至多被服务一次，即使合法 NF 出示了先前发给它的标识符也是如此。	malicious_nf	proved	a04_fetch	checked
A08	消费者接受的通知携带消费者自己在订阅中提议的 anaNotifUri 和 anaNotifCorrId 值。	network	proved	a08_notif	checked
A08	合法 NF 无法让 DCCF 向自己的端点投递携带另一消费者关联标识符的通知。	malicious_nf	proved	a08_notif	checked
K01	NF 服务消费者的 OAuth2 访问令牌对网络上的攻击者保持机密。	network	proved	k01_tok	none
K01	持有自身凭据的合法 NF 无法获知另一消费者的访问令牌。	malicious_nf	proved	k01_tok	none
K01	消费者与 DCCF 之间的 TLS 会话密钥无法被网络上的攻击者推导出来。	network	proved	k01_key	checked
K01	消费者的 OAuth2 访问令牌对网络上的攻击者保持机密。	network	proved	k01_tok	checked
Z01	合法 NF 只能让 DCCF 针对自己的令牌行事：令牌主体必须等于发起请求的 NF 身份。	malicious_nf	proved	z03_tok	checked
Z01	DCCF 只有在请求身份获得了 NRF 签发的有效授权之后才创建订阅。	network	proved	z01_sub	checked
Z01	DCCF 只有在请求身份获得了 NRF 签发的有效授权之后才服务 Fetch。	network	proved	z01_fetch	checked
Z03	一个消费者的令牌不会被另一消费者的请求所接受。	network	proved	z03_tok	checked
Z03	fetch 关联标识符只会被服务给其被签发时所指向的消费者。	network	proved	z03_corr	checked
Z03	合法 NF 无法让 DCCF 为它服务一个签发给另一消费者的 fetch 关联标识符。	malicious_nf	proved	z03_corr	checked
Z05	DCCF 只向它签发了 fetch 指令的消费者发送已通知的分析结果，因此一个消费者域无法获得另一消费者的分析结果。	malicious_nf	proved	z05_data	checked
Z05	用网络提供的值探测 Fetch 端点的合法 NF 永远无法获得任何已获取的分析结果。	malicious_nf	proved	z05_never	checked
U01	合法 NF 无法获得来自另一消费者通知路径上 NWDAF 的分析结果。	malicious_nf	proved	u01_path	checked
U01	消费者获得的分析结果，是 DCCF 在该消费者的通知路径上从上游 NWDAF 收到的。	network	proved	u01_path	checked
U01	消费者获取的分析结果是其订阅所属的 NWDAF 产生的，而不是网络上编造的值。	network	proved	u01_data	checked
U02	消费者通过 Fetch 获得的分析结果属于该消费者创建的订阅，因此一个消费者无法取回另一消费者的分析结果。	malicious_nf	proved	u02_data	checked
U03	消费者据以采取行动的通知属于 DCCF 为它创建的订阅。	network	proved	u03_notif	none

目标	主张	攻击者	回答	查询	对照
U03	消费者据以采取行动的 fetch 指令属于当时仍然有效的订阅：在订阅被更新之前签发的指令不会被接受。	`malicious_nf`	attack (depends_on_ambiguity)	u03_old	
U03	消费者据以采取行动的通知是 DCCF 针对该订阅发送给它的，而不是由另一 NF 提供的。	`malicious_nf`	proved	u03_notif	checked

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，并把每个性质交给 ProVerif 求解；第 3 节中的每个结果都是证明器给出的答案。

ProVerif+ Agent 会话 session-20261115T094233，2026-11-15。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：69，证明器耗时 5 秒，实际耗时 13 分钟，成本 0.112 美元。

ProVerif：共运行 118 次证明，118 次给出结论，0 次无答案，耗时 4 秒，记录 53 个答案。

分析了什么

协议	DCCF analytics availability and fetch
版本	19.7.0
分析范围	NF consumer、DCCF、NWDAF；固定非 UE 聚合 analytics，一次订阅、上游通知、下游 availability/fetch instructions、Fetch 与数据回应；保留 notify correlation 及 fetch correlation ID，排除 MFAF 和多个上游复用。
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/正例控制。 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	1271 numbered passages (928151 bytes) from the sections listed below. A reference such as s0001 in this report is one of these passages

分析所依据的文稿章节：

文稿	章节	标题
main	4.2.2.2.2	订阅分析通知
main	4.2.2.4.2	关于已订阅分析的通知
main	4.2.2.5	Ndccf_DataManagement_Fetch 服务操作
main	4.2.2.5.1	概述
main	4.2.2.5.2	取回已通知的分析和数据
main	5.1	Ndccf_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
main	5.1.3	资源
main	5.1.3.1	概览

文稿	章节	标题
main	5.1.3.2	资源：DCCF 分析订阅
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	5.1.3.2.4	资源自定义操作
main	5.1.3.3	资源：单个 DCCF 分析订阅
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	PUT
main	5.1.3.3.3.2	DELETE
main	5.1.3.3.4	资源自定义操作
main	5.1.3.4	资源：DCCF 数据订阅
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 DCCF 数据订阅
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	PUT
main	5.1.3.5.3.2	DELETE
main	5.1.3.5.4	资源自定义操作
main	5.1.3.6	空
main	5.1.3.7	空
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：transfer-data-sub
main	5.1.4.2.1	描述
main	5.1.4.2.2	操作定义
main	5.1.5	通知
main	5.1.5.1	概述
main	5.1.5.2	分析通知

文稿	章节	标题
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	数据通知
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	5.1.5.4	Fetch 通知
main	5.1.5.4.1	描述
main	5.1.5.4.2	目标 URI
main	5.1.5.4.3	标准方法
main	5.1.5.4.3.1	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 NdccfAnalyticsSubscription
main	5.1.6.2.3	类型 NdccfDataSubscription
main	5.1.6.2.4	类型 NdccfAnalyticsSubscriptionNotification
main	5.1.6.2.5	类型 NdccfDataSubscriptionNotification
main	5.1.6.2.6	类型 FormattingInstruction
main	5.1.6.2.7	类型 ProcessingInstruction
main	5.1.6.2.8	类型 ParameterProcessingInstruction
main	5.1.6.2.9	类型 NotifSummaryReport
main	5.1.6.2.10	类型 EventParamReport
main	5.1.6.2.11	类型 ReportingOptions
main	5.1.6.2.12	空
main	5.1.6.2.13	类型 DccfEvent
main	5.1.6.2.14	类型 NotifyEndpoint
main	5.1.6.2.15	类型：StorageHandlingInformation
main	5.1.6.2.16	类型：DeletionAlert
main	5.1.6.2.17	类型：NotifResponse
main	5.1.6.2.18	空
main	5.1.6.2.19	类型：DataTransferResp

文稿	章节	标题
main	5.1.6.3	简单数据类型和枚举
main	5.1.6.3.1	引言
main	5.1.6.3.2	简单数据类型
main	5.1.6.3.3	枚举：SummarizationAttribute
main	5.1.6.3.4	枚举：AggregationLevel
main	5.1.6.3.5	枚举：DataCollectionPurpose
main	5.1.6.3.6	枚举：TermCause
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	5.1.6.5	二进制数据
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
main	5.1.9	安全
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	取消事件通知订阅
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

文稿	章节	标题
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
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

文稿	章节	标题
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

源文包中的源文步骤在文稿中定位到的协议用途是：NF 消费者、DCCF、NWDAF；固定非 UE 聚合分析，一次订阅、上游通知、下游可用性/取回指令、Fetch 与数据回应；保留通知关联及取回关联 ID，排除 MFAF 和多个上游复用。

对照文稿确定性质

在编写查询之前，分析者将起草的性质列表与文稿本身对协议用途的说明作了对照。名称、某个性质所涉内容以及文稿的说明均属分析者的解读；所提及的位置确实存在于文稿中。

ID	简称	对象	文稿怎么说	出处
A02	对端身份认证	DCCF 确信它接受 Subscribe 或 Fetch 的消费者，就是真正向它发起该流程的 NF 实例；消费者也确信通知来自其订阅所在的 DCCF	4.2.2.2.2 要求提供订阅信息；这一绑定来自任务范围内的 TS 29.500 TLS 与 TS 33.501 NF 双向认证	s0005, s0021, s0050, s0076, s0417
A03	关联标识符协商	每个被接受的通知都携带 DCCF 确实创建的某个订阅的 anaNotifCorrId，每个被处理的 Fetch 都携带 DCCF 确实签发的 fetchCorrIds	s0051, s0337 规定 anaNotifCorrId；s0077, s0310 要求通知携带 fetch 关联标识符	s0051, s0337, s0077, s0310
A04	消息不被重复接受	消费者接受的每个通知、DCCF 服务的每个 Fetch，都对应一次独立的对端运行；没有任何内容被处理两次	未说明重放保护；timeStamp 只标记准备时间（s0052, s0337）	s0050, s0052, s0077
A08	订阅参数绑定	DCCF 按消费者提出的值（anaSub、anaNotifUri、anaNotifCorrId）行事，Fetch 返回其 fetchCorrIds 所指向订阅的分析数据	s0021, s0027, s0029 关于使用和存储请求；s0051 关于通知；s0310, s0080 关于 fetch 响应	s0005, s0021, s0051, s0080
K01	访问令牌与TLS密钥的保密性	消费者的 OAuth2 令牌与 TLS 会话密钥保持机密，一个 NF 不能拿另一个 NF 的令牌去执行并非签发给它的操作	s0417, s0418：以 NRF 为授权服务器的客户端凭据	s0417, s0418
Z01	受保护操作前的授权	每个受保护的 DCCF 操作，只对持有覆盖该操作的、由 NRF 签发的有效授权的请求方执行	s0417, s0418 规定访问须经 OAuth2；原文未说明 DCCF 执行哪项检查	s0417, s0418
Z03	授权绑定到资源与操作	令牌或 fetch 关联标识符始终与其主体、资源和操作绑定，不被用于另一个消费者、订阅或操作	s0420 为整个服务定义一个 scope，因此资源绑定必须来自消息标识符（s0051, s0077）	s0420, s0051, s0077
Z05	消费者域之间的隔离	控制自己订阅和 Fetch 请求的消费者，无法获得另一个消费者的分析数据	s0310, s0080 将响应与提交的标识符绑定；未陈述隔离目标，它来自任务范围	s0077, s0080
U01	所获取分析数据的上游来源	获取的分析数据来自该消费者订阅的上游 NWDAF，且指令携带该通知的标识符	s0014 允许消费者指定目标 NWDAF，s0021, s0026 允许 DCCF 向其订阅；未陈述来源目标	s0014, s0021, s0026, s0077
U02	获取属于发起请求的消费者	消费者获取的分析数据属于该消费者的某个订阅	s0310, s0080 未提及请求方身份，因此该目标读自任务范围	s0077, s0080

ID	简称	对象	文稿怎么说	出处
U03	通知与获取绑定到当前订阅	消费者据以行动的通知或 fetch 指令，属于当时当前的某个订阅，且不被接受两次	存在 PUT 和 DELETE（s0149, s0162），而 fetchCorrIds 只被说成是先前提供的（s0077）；未陈述失效规则	s0077, s0149, s0162

分析者没有补充性质：起草的清单已覆盖文稿写明的目标。

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 32 段，没选 366 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	intro
P003 至 P006	rfc6749:2790-2823	K01, A02, `network`	
P007 至 P010	rfc6749:2827-2850	A02, Z01, `malicious_nf`	
P011 至 P015	rfc6749:2859-2881	K01, Z01, Z03, `network`	
P016 至 P033	rfc6749:2885-3003	不在范围内	refresh tokens, authorization codes, redirect URIs and ROPC belong to other grants
P034 至 P035	rfc6749:3013-3019	K01, `network`	
P036	rfc6749:3023-3028	A02, `network`, `malicious_nf`	
P037 至 P039	rfc6749:3032-3041	K01, `network`	
P040 至 P050	rfc6749:3045-3138	不在范围内	phishing, CSRF and clickjacking need an end user or browser; the modelled NFs have none
P051 至 P052	rfc6749:3142-3150	没有陈述目标	advice to validate input; not askable
P053 至 P062	rfc6749:3154-3210	不在范围内	authorization-server redirect endpoints and the implicit grant
P063	ts-33-501:2491-2491	不在范围内	UE/gNB bidding down
P064 至 P213	ts-33-501:2495-2853	不在范围内	UE authentication, keys, RAN, UDM, AUSF, UDR and subscriber privacy are outside this API sub-flow
P214	ts-33-501:2859-2859	没有陈述目标	trust-zone assumption
P215 至 P220	ts-33-501:2865-2875	A02, Z01, A04, `network`, `malicious_nf`	
P221 至 P227	ts-33-501:2879-2891	Z01, K01, `malicious_nf`	

段落	出处	定为	不选的原因
P228 至 P241	ts-33-501:2895-2923	不在范围内	NEF and SCP requirements; neither is a party of the modelled flow
P242 至 P312	ts-33-501:2929-3075	不在范围内	interconnect, SEPP, N32-f and roaming intermediary requirements; no inter-PLMN interface here
P313 至 P314	ts-33-501:3079-3081	不在范围内	IPUPS packet filtering is a transport function
P315 至 P318	ts-33-501:3085-3091	没有陈述目标	originating network header is deployment information
P319 至 P337	ts-33-501:3097-3141	没有陈述目标	operator requirements on monitoring configuration and security visibility/configurability
P338 至 P369	ts-33-501:3149-3215	证明器问不了	algorithm identifiers and negotiation: the strength of a primitive is not askable
P370 至 P398	ts-33-501:3219-11901	不在范围内	5G-RG, NSSAAF, N3, EAP-TLS, IAB, MBSF, MBSTF and NEF annex requirements

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 29 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 3 项，没设的 26 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 服务消费者	参与方	`malicious_nf`	
S02	DCCF	参与方	按定义全部失守	a controlled DCCF produces every notification and fetch answer, so no target is asked under it
S03	NWDAF	参与方	不在范围内	truthfulness of analytics content is outside this API sub-flow
S04	NRF	参与方	按定义全部失守	a compromised authorization server voids all authorization
S05	NF 服务消费者 A	参与方	`malicious_nf`	
S06	NF 服务消费者 B	参与方	`malicious_nf`	
S07	订阅资源	值	按定义全部失守	reading it yields the correlation identifiers, whose secrecy the fetch properties rest on
S08	通知目标地址 anaNotifUri 与通知关联标识 anaNotifCorrId	值	按定义全部失守	same: the notification identifiers are what ties a notification to a subscription
S09	Fetch 关联标识 fetchCorrIds	值	按定义全部失守	the fetch correlation identifiers are the bearer credential of the Fetch
S10	subscriptionId	值	攻击者得不到更多	the subscription identifier is not used as an authenticator
S11	anaNotifCorrId	值	攻击者得不到更多	a correlator, not a credential: producing a notification still needs the DCCF TLS session
S12	fetchCorrIds	值	按定义全部失守	whoever holds them may fetch, as s0077 and s0080 require
S13	fetchUri	值	攻击者得不到更多	the URI alone is not enough; the body must carry the correlation identifiers

ID	参与方或值	类别	攻击者条件	不设的原因
S14	通知消息	值	按定义全部失守	a holder of the notification holds its fetch correlation identifiers
S15	Fetch 请求	值	按定义全部失守	a holder of the request holds its fetch correlation identifiers
S16	anaSub	值	攻击者得不到更多	subscription content is configuration, not an authenticator
S17	anaNotifUri	值	攻击者得不到更多	the callback URI is an address, not a credential
S18	targetNfId/targetNfSetId	值	攻击者得不到更多	NF identifiers are discoverable via the NRF
S19	OAuth2 access token	值	按定义全部失守	a holder of the token is authorized as that consumer for the single scope; K01 already asks that it stay secret
S20	TLS 会话密钥	值	按定义全部失守	with the session keys the attacker reads and writes that session
S21	scope ndccf-datamanagement	值	攻击者得不到更多	a scope name grants nothing without a token
S22	OAuth2 token 中的 scope 与受众	值	攻击者得不到更多	token metadata grants nothing by itself
S23	Fetch 响应中的 analytics	值	按定义全部失守	the analytics is the protected value itself
S24	OAuth2 token	值	按定义全部失守	same as S19
S25	各消费者的订阅资源	值	按定义全部失守	same as S07
S26	通知内容	值	按定义全部失守	same as S14
S27	上游 analytics	值	按定义全部失守	the upstream analytics is the value whose delivery is analysed
S28	analytics	值	按定义全部失守	same as S23
S29	订阅更新状态	值	攻击者得不到更多	knowing whether a subscription is current gives no credential

每条性质在各攻击者条件下的答案

攻击者由分析者设定，在编写查询之前就已确定；每个答案都记录在其中一个攻击者之下。单元格是证明器在该攻击者下对该性质给出的答案；“无答案”表示该攻击者下证明器让某条主张处于未决状态，或者没有记录任何内容；空单元格表示分析者没有为该性质设定该攻击者。

攻击者	持有什么	依据	出处
`network`	只有网络：它能读取、丢弃、更改并发送消息，不持有任何一方的秘密		
`malicious_nf`	一个合法的 NF 服务消费者，拥有自己的身份和自己由 NRF 签发的、范围为 ndccf-datamanagement 的令牌，拥有自己的 TLS 会话，并完全控制自己的订阅、通知端点和 Fetch 请求	5.1.9 允许任何持有令牌的 NF 使用该 API，并为整个服务定义一个范围（s0417、s0420）	s0417、s0418、s0420

ID	`network`	`malicious_nf`
A02	成立	成立
A03	成立	成立
A04	成立	成立
A08	成立	成立

ID	network	malicious_nf
K01	成立	成立
Z01	成立	成立
Z03	成立	成立
Z05		成立
U01	成立	成立
U02		成立
U03	成立	不成立

原文和目标

出处：1271 个原文单元，928151 字节，冻结于 source-delivery-01.json。 目标：11 个已批准（A02、A03、A04、A08、K01、Z01、Z03、Z05、U01、U02、U03）；提出的问题：3 个。 目标清单事先按草稿获得批准，使用 --approve-targets。

宿主按其文档化默认值做出的决定（任务 `decisions: auto`），供你核查：

- approve_package：批准源文包 protocol-10e4fd6d8a752af4dad43e70f59f

本步骤的自我报告：1 个请求（1 个失败），0.000 USD，输入 388996 个令牌，输出 5362 个令牌，耗时 277 秒。

关于证明器和模型的说明

- A04：在 ProVerif 中作答，分析者为该目标标注 changing_state
- U03：在 ProVerif 中作答，分析者为该目标标注 changing_state

对代数（xor、dh_algebra、other_algebra）做简化的 ProVerif 模型给攻击者的能力弱于真实操作：其中的证明是针对更弱攻击者的证明，其中的攻击则就是攻击。省略对变化值检查（changing_state）的 ProVerif 模型让各方接受得更多：其中的证明成立，其中的攻击可能只是缺少了检查。Tamarin+ Agent 按原样对这些建模。

分析者的状态审视

这部分是分析者自己对原文的解读，不是证明器的答案。ProVerif 显示“是”表示 ProVerif 在展开模型中找到了包含这些步骤的迹，因此这些步骤是可能的；它们是否破坏某个目标，只由答案表中的攻击来说明，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，并不证明什么。

值	由谁保持	何时变化	何时检查	出处
anaNotifCorrId（通知关联标识符）	DCCF 保存在已存储的订阅中，以及提出该订阅的 NF 服务消费者（s0008、s0030）	创建订阅时（POST，s0030），以及通过 PUT 修改订阅时（s0149、s0155）	DCCF 在每条通知中复制该值（s0051、s0337），消费者在接受通知时将其与自己的值进行比较	s0008、s0051、s0149、s0337
fetchUri 和 fetchCorrIds（获取指令）	DCCF 与已存储的分析结果一起保存，消费者则保存在其存储的通知中（s0061、s0077、s0310）	发出新通知时，以及在订阅更新后 DCCF 为同一订阅发出新指令时	每次 Fetch 时，DCCF 都将提交的值与其发出的值进行比较（s0077、s0079）	s0076、s0077、s0079、s0080、s0310

值	由谁保持	何时变化	何时检查	出处
已存储订阅的有效性（是否存在以及哪些内容是当前的）	DCCF（s0029）	由 POST 创建（s0026..s0029），由 PUT 修改（s0149），由 DELETE 删除（s0162）	DCCF 在创建、修改或删除订阅时查找该订阅；在使用已发出的指令时，并不与它们进行比较	s0029、s0149、s0162、s0155

模式	结果	目标	ProVerif 显示	步骤
重放	候选	U03	是：ProVerif 在 bounded.pv 上否证 `u03_old`	1. NF 服务消费者发起订阅；DCCF 创建订阅并存储（s0026..s0029）。 / 2. DCCF 以 fetchInstruct（fetchUri、fetchCorrIds）发出通知（s0056、s0076、s0077）；消费者执行获取并得到应答（s0078..s0080）。 / 3. 消费者通过 PUT 修改订阅（s0149..s0161）；DCCF 替换已存储的值并发出新指令（s0155、s0030）。 / 4. 消费者再次提交先前的 fetchUri 和 fetchCorrIds；DCCF 按请求找到已存储的分析结果（s0079、s0080）并返回 200 OK。
失步	候选		是：ProVerif 在 bounded.pv 上否证 `u03_old`	1. 带有指令 fc1 的订阅在双方都是当前的。 / 2. PUT 替换订阅内容；DCCF 发出 fc2（s0149、s0155）。 / 3. 原文中没有任何内容告知消费者 fc1 已被撤回，而 DCCF 继续应答 fc1，因为它只匹配请求中的标识符（s0077、s0079）。
失败时不回滚	候选		否：未用 ProVerif 尝试	1. 消费者通过获取指令取得分析结果。 / 2. 订阅被 DELETE（s0162）删除或被放弃。 / 3. 原文没有说明已存储的分析结果及其签发时所用的关联标识符被撤回，因此该指令仍指向可检索的数据。
重复一次性动作	未发现			流程中没有任何步骤被设定为只发生一次：对同一个订阅，通知和 Fetch 可以合法地重复。
先更新后检查	未发现			DCCF 在替换已存储的订阅值之前先校验 PUT 的令牌；原文中没有任何内容颠倒这一顺序。
会话间的竞态	未发现			每个订阅都有自己的关联标识符，获取操作与通知实例的值相匹配，因此两次运行不会读取并更改同一个值。
可区分的失败	未发现			5.1.7 条的错误应答区分了原因（SUBSCRIPTION_CANNOT_BE_SERVED s0021、USER_CONSENT_NOT_GRANTED s0023），但它们是把请求者自己的请求相关情况返回给请求者，而所建模的流程中没有拒绝应答会告诉第三方另一个消费者保留了什么。
回绕或窗口	未发现			关联标识符是字符串而非计数器，流程中没有接受窗口：timePeriod 涉及分析结果的目标时段，与指令的接受无关。

分析者为各目标标注的内容：

步骤	目标	特征	理由
46	U03、A04	changing_state	U03 和 A04 依赖于 DCCF 保持并更改的每订阅值：当前订阅的关联标识符、为其签发的获取指令以及已存储的分析结果；旧指令在订阅更新或删除后是否仍被应答取决于该状态。任意会话数的模型无法表达该状态的替换，因此这些答案在展开模型中并结合状态审视完成。

分析者的结束语

对 TS 29.520 第 4.2.2.2.2、4.2.2.4.2 和 4.2.2.5.2 条的流程进行了建模（消费者、DCCF、NWDAF），将 TLS 视为端点之间的认证加密，OAuth2 令牌绑定到请求的 NF，通知/获取关联标识符视为新鲜值。全部 11 个目标都有了答案：10 个在任意会话数的模型上得到证明，并且每个证明都有对照查询，U03 则通过针对通知部分的证明和针对时效部分的展开模型上的攻击得到作答。已确立：DCCF 只为已认证的 NF 实例行动，通知和 Fetch 与 DCCF 自己的标识符一致，一个消费者无法获得另一个消费者的分析结果，分析结果经由消费

者自己的通知路径来自 NWDAF。一项发现：在订阅更新之前签发的指令在更新之后仍被应答，因为原文中没有任何内容将其撤回；检查了一项在 PUT 时撤回标识符的拟议修改，它消除了该迹。阻碍更强答案的因素：消费者的隔离和获取关联标识符的资源绑定依赖于这些标识符不可猜测，而原文将它们类型化为普通字符串，且该 API 的单一 OAuth2 范围覆盖所有操作；将标识符建模为可猜测，并按 TS 29.501 对范围和受众建模，即可解决这一点。对变化值的检查仅针对一个订阅和一次更新完成，删除、重复更新以及第 2 阶段规范中描述的获取指令期限则留给 Tamarin+ Agent 或进一步的展开。