DCCF 采集配置文件生命周期的安全分析

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）
检查的性质	10 条：3 条成立，6 条不成立（7 个攻击），1 条没有答案
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

Ndccf_ContextManagement 服务允许 NF 服务消费者在 Data Collection Coordination Function 上保存其数据采集兴趣：NWDAF 或 ADRF 注册一个数据采集配置文件、更新它并注销它，以便之后的数据采集可以从该配置文件进行协调。配置文件在 anaSub 或 dataSub 中携带采集信息，并附带一个消费者标识符；DCCF 分配 profileId，消费者、DCCF 和 NRF 通过服务化接口和 NRF 链路上的 HTTP/2 互相认证，NRF 签发 API 所依赖的访问令牌。

交换结束时，消费者确信 DCCF 接受了它发送的请求，DCCF 确信请求是经由一个持有 ndccf-contextmanagement 服务有效令牌的 NF 的已认证连接到达的。DCCF 并不确信请求体中的消费者标识符就是发送该请求的 NF 的标识符，子流程中也没有任何东西将 profileId 绑定到创建它的消费者。

各方：

- **NWDAF**：NF 服务消费者，注册、更新并注销其配置文件。持有：nwdafId、客户端凭据、访问令牌。
- **DCCF**：提供 Ndccf_ContextManagement，保存配置文件。持有：配置文件（profileId、消费者标识符、采集信息）、授权服务器公钥。
- **NRF**：授权服务器，签发访问令牌。持有：各 NF 的客户端凭据、签名密钥。

图 1 展示了文稿所规定的交换过程。

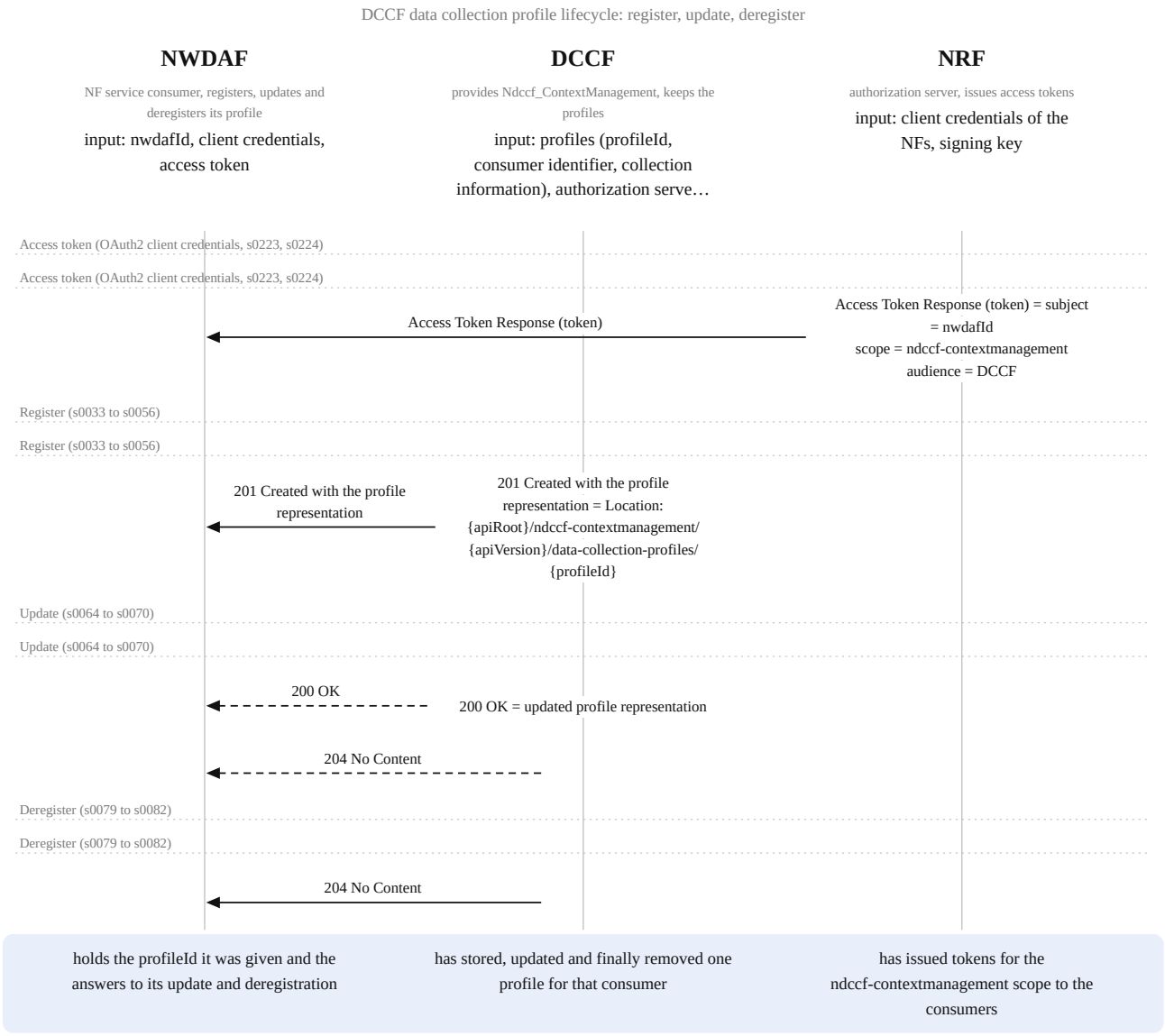


图 1. 按规范规定的协议。

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 1 个发现：以下内容并未在文稿中写明。对其中 1 个，文稿是否写明尚不明确。

令牌检查成立：DCCF 提交的每一次注册、更新和注销都有 NRF 为该 NF 签发的、面向 ndccf-contextmanagement 范围的令牌作支撑，采集信息只到达 DCCF。不成立的是请求与其所指配置文件之间的绑定：DCCF 原样存储请求体中的消费者标识符，不与已认证的消费者比较；只要 profileId 已知或可预测，它就接受对另一个 NF 的配置文件的更新或注销。这是文稿未写明的发现，因为文稿只记录了一个服务级范围，没有所有权规则，因此 profileId 在暗中承担了授权。在检查的十条性质中，全部在两种攻击者下都有答案，六条只有部分答案，失败的一半以攻击形式展示，两条依赖于 ProVerif 无法完全证明的展开生命周期；Tamarin+ Agent 可以解决这些状态问题。

攻击图使用图 1 的列。红色表示攻击者的行为。黄色标记的是某方在规范交换中本不应在该位置发送的消息，以及攻击所破坏的结果。

2.1 任何持有服务范围令牌的 NF 都能写入另一个 NF 的配置文件

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	A02, A03, A08, A08.7, Z03, Z03.8（见第 3 节）
原文	s0033, s0047, s0052, s0064, s0226
依据	ProVerif 攻击查询 `A03_bind`、`A08_owner`、`A08_owner_b`；图 2、3、4、5 和 6

发生了什么。 DCCF 检查访问令牌的签名、范围（scope）和受众（audience），然后存储或替换配置文件，却不将请求体中的消费者标识符与令牌的主体进行比较。因此，第二个合法 NF 可以注册一个指名另一个 NF（例如 NWDAF 的 nwdafId）的配置文件。在 profileId 已知或可预测的情况下，它还可以更新或注销以另一消费者标识符记录的配置文件，而 DCCF 会接受。

攻击者需要什么。 只需一个真实 NF 服务消费者实例的客户端凭据，以及 NRF 为该服务范围签发的访问令牌。要更改另一个 NF 的配置文件，它还必须知道或预测 profileId，而原文对此完全没有规定：只说 DCCF 会分配一个，丝毫未提如何分配。

攻击者能得到什么。 它可以让另一个消费者的已注册采集请求表达它想要的内容，或删除另一个消费者的注册；它还可以将任意消费者标识符写入配置文件，使所存储的配置文件错误地表明是谁注册了该采集。

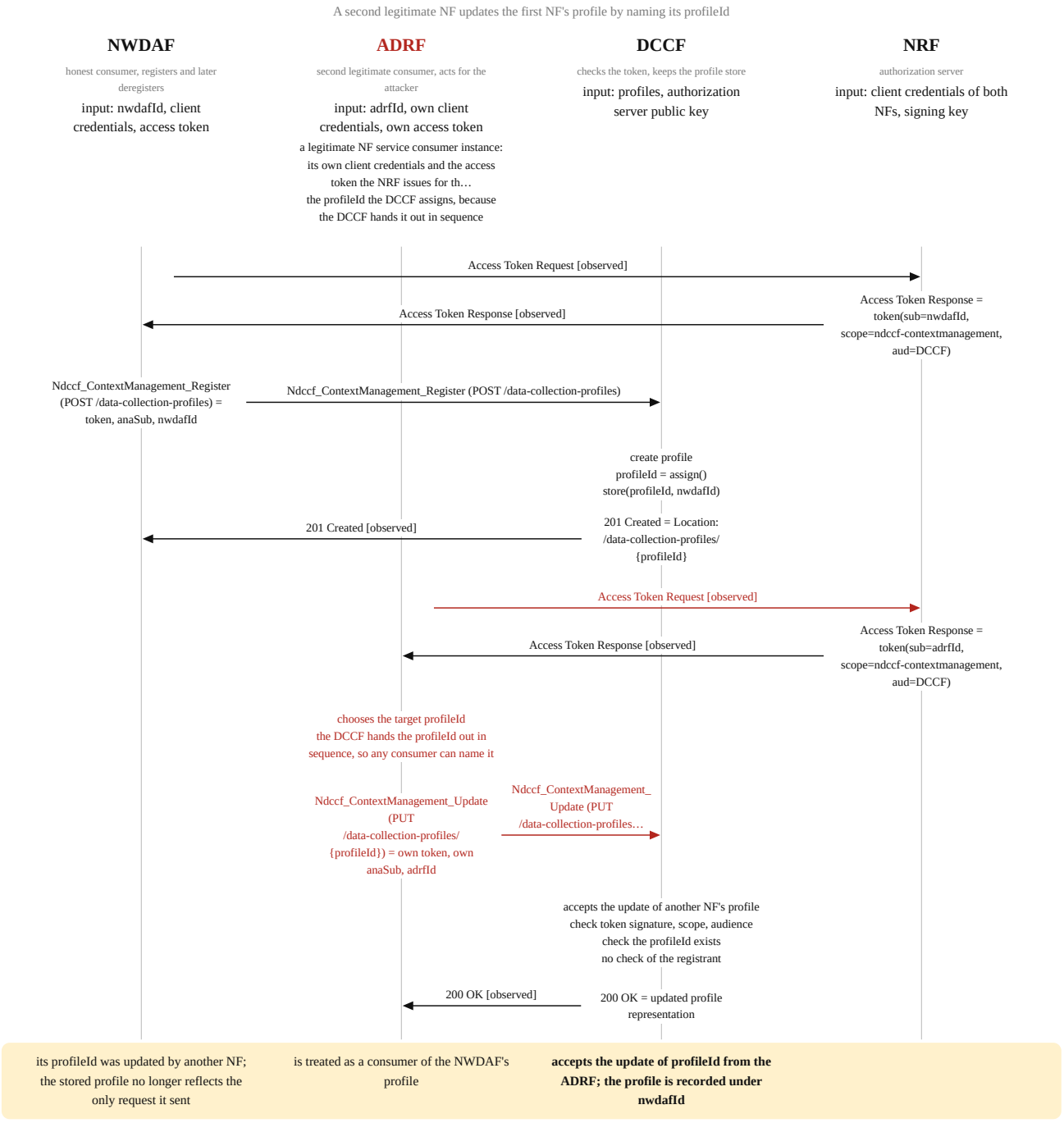


图 2. 第二个合法 NF 通过指名其 profileId 更新第一个 NF 的配置文件。性质 Z03.8；攻击查询 A08_owner、A08_owner_b。

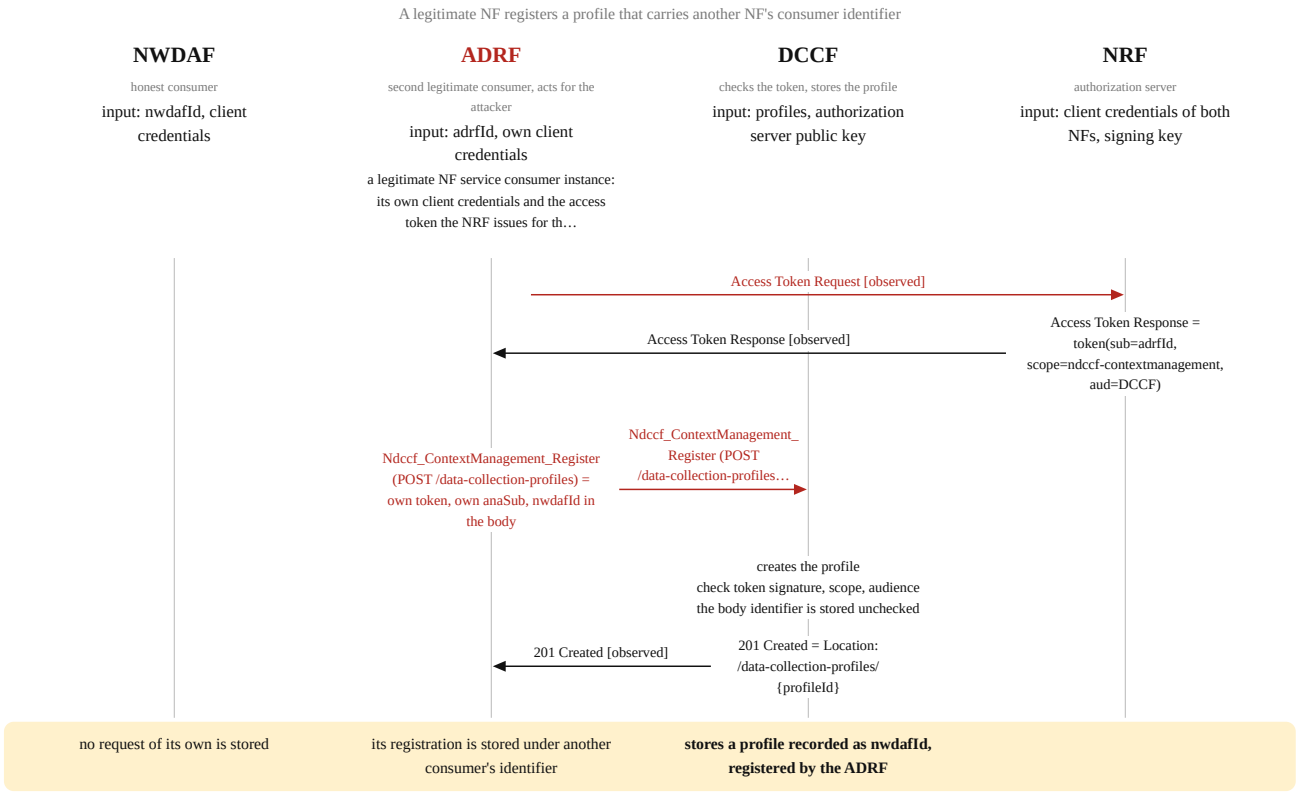


图 3. 一个合法 NF 注册的配置文件携带了另一个 NF 的消费者标识符。性质 A03；攻击查询 A03_bind。

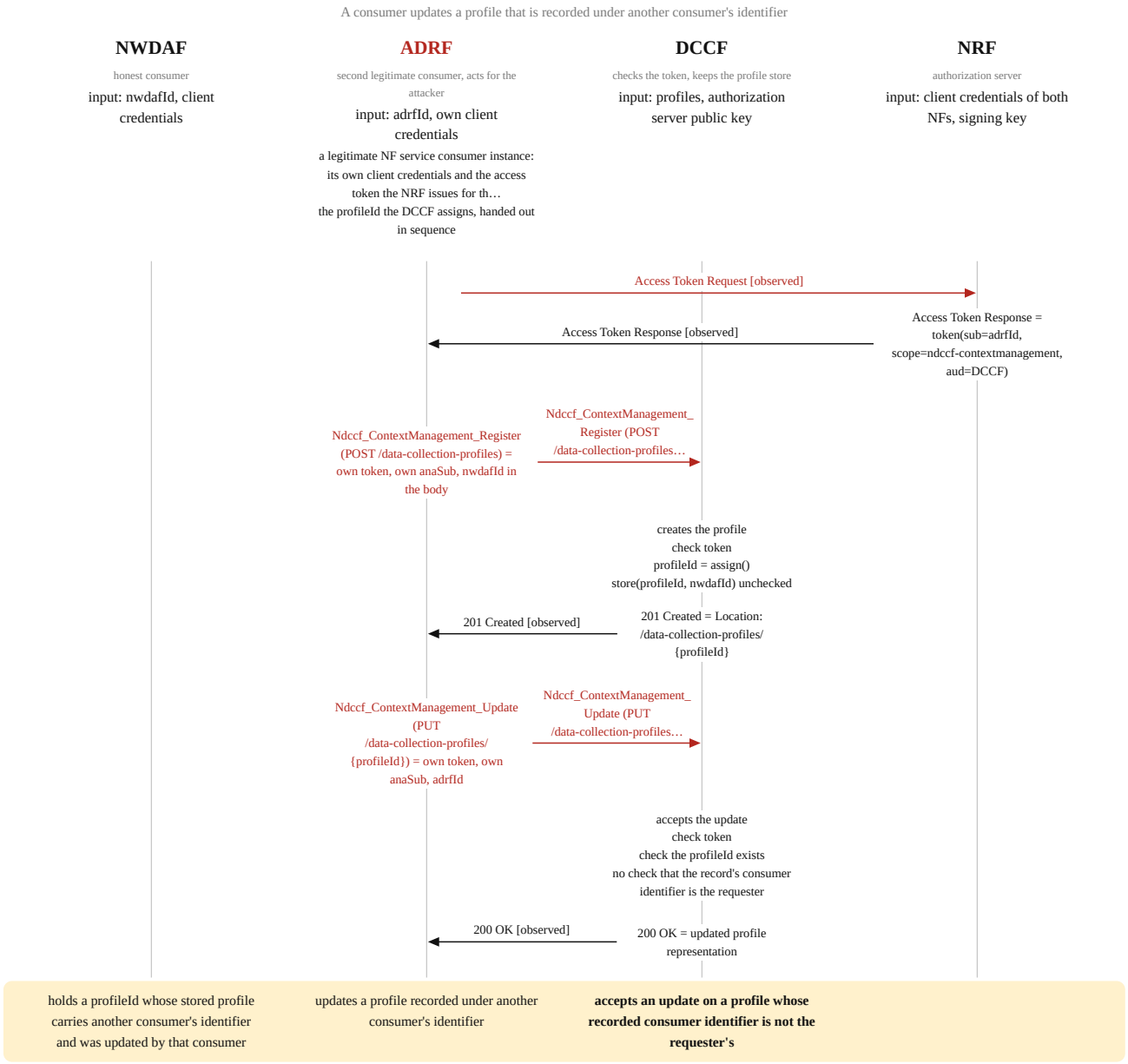


图 4. 一个消费者更新了以另一消费者标识符记录的配置文件。性质 A08；攻击查询 A08_owner。

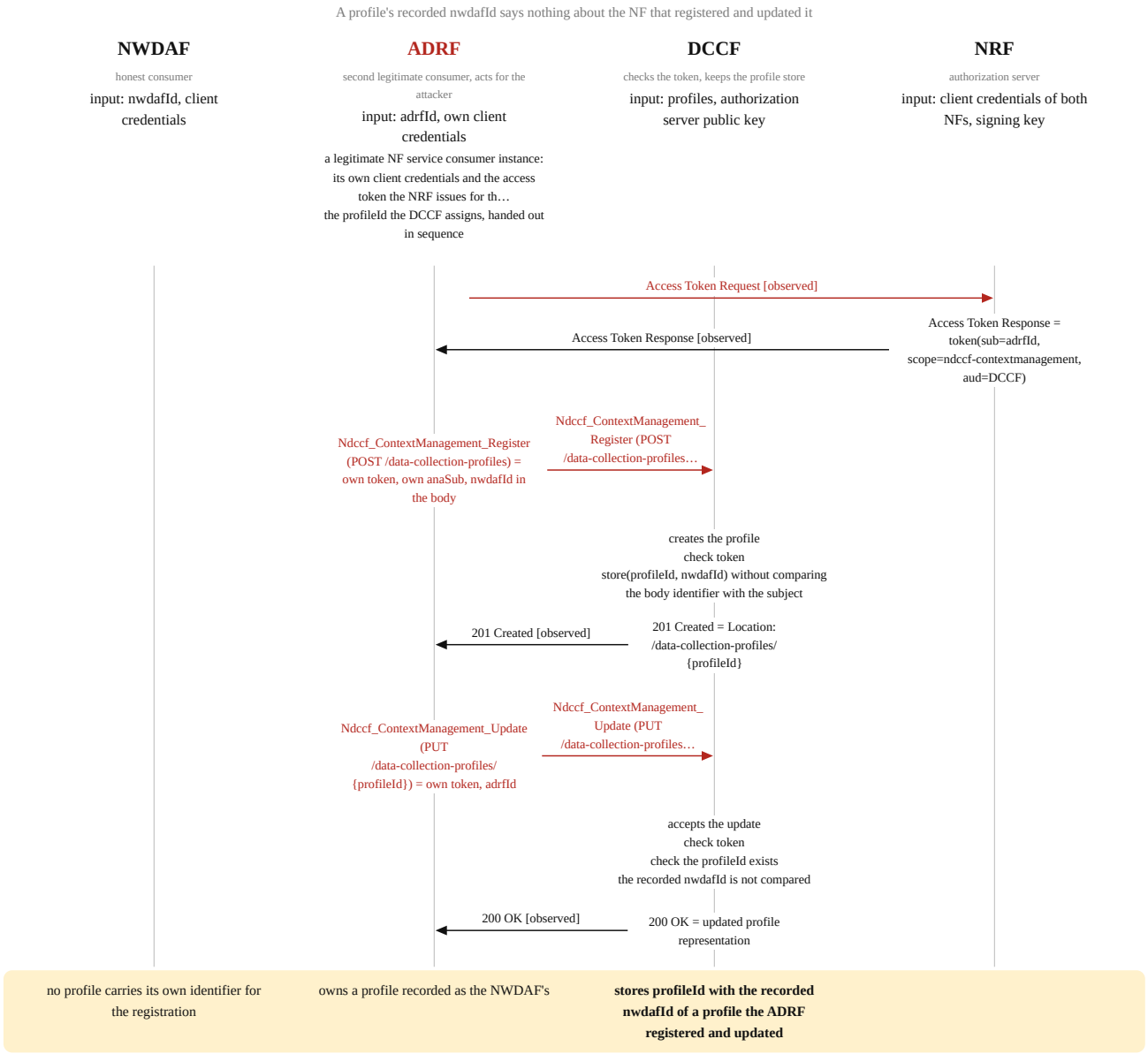


图 5. 配置文件中记录的 nwdafId 并不能说明是哪个 NF 注册并更新了它。性质 A08.7；攻击查询 A08_owner。

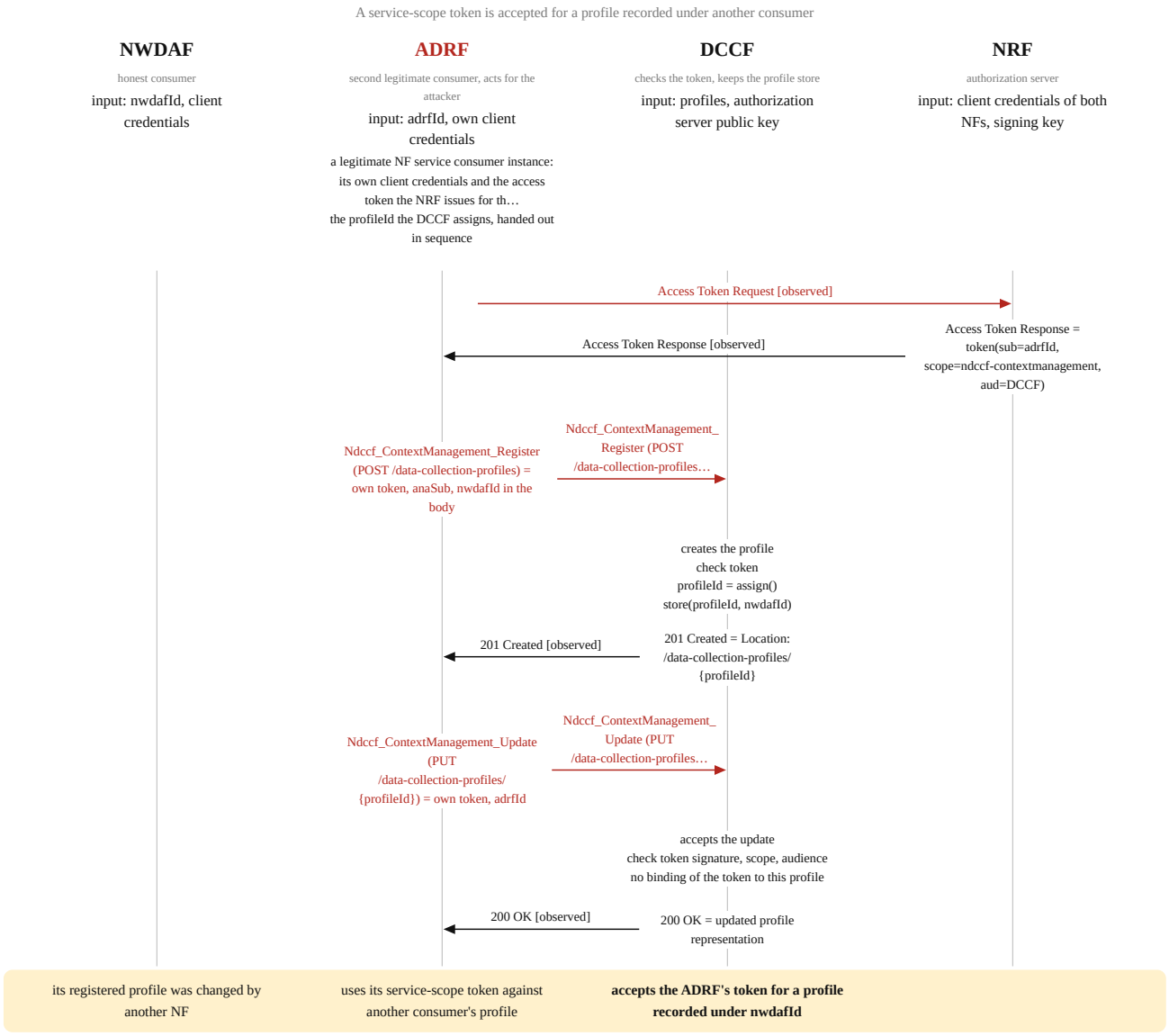


图 6. 服务范围令牌被接受用于以另一消费者记录的配置文件。性质 Z03；攻击查询 A08_owner。

拟议修改。 让 DCCF 将 NdccfDataCollectionProfile 请求体中的消费者标识符（nwdafId、adrfId、nwdafSetId 或 adrfSetId）与所出示访问令牌的主体进行比较，不匹配则拒绝请求。将每个创建的配置文件绑定到该主体，并在令牌主体与注册时记录的主体不一致时，在触碰配置文件之前以 403 拒绝 PUT 或 DELETE。

用 ProVerif 复核。 未检查：该修改是在 DCCF 内部新增一项检查，并不改变消息，而原文将无效请求的拒绝留给条款 5.2.7；带有该检查的模型将不得不自行创设这一拒绝行为，从而不再是本文本的模型。对规范文本的任何模型都无法消除该弱点。

以下该发现的攻击未提供图：A02: A03_bind、A02: A08_owner、A02: A08_owner_b、A03: A08_owner、A03: A08_owner_b 、 A08.7: A03_bind 、 A08.7: A08_owner_b 、 A08: A03_bind 、 A08: A08_owner_b 、 Z03.8: A03_bind、Z03: A03_bind、Z03: A08_owner_b。

3 检查过的性质

每一行都是一条提交给 ProVerif 的性质。结果是 ProVerif 对分析者所建协议模型的答案；价值以及文稿是否已写明该结果属于分析者的判断，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	**每个请求的发起者经过认证。** 每当 DCCF 接受一个注册、更新或注销请求，声称发起方（NWDAF 或 ADRF 实例）必须确实以该身份运行过本服务。	无	不成立（1 个攻击）	中	不确定
A03	**注册的消费者标识与发送方匹配。** 每当 DCCF 接受一次注册，都存在一个匹配的消费者运行，双方对采集信息（anaSub/dataSub）与消费者标识（nwdafId/adrfId/nwdafSetId/adrfSetId）一致。	无	不成立（1 个攻击）	中	不确定
A08	**profileId 绑定到其创建消费者。** DCCF 接受的每个 profileId 都对应注册时由该消费者创建的那一个 profile，其采集信息与消费者标识与创建时一致。	无	不成立（1 个攻击）	中	不确定
Z01	**被接受的操作携带有效的服务令牌。** 每次 DCCF 提交注册、更新或注销操作，都必须有授权服务器签发的、对 ndccf-contextmanagement 服务有效的令牌。	无	成立	低	不确定
Z03	**令牌绑定到主体、资源和操作。** 一个令牌只能用于其授权主体所声明的消费者身份、其所属的 profileId 以及被授权的操作类型；不能用于其他 NF 的 profile 或其他操作。	无	不成立（1 个攻击）	中	不确定
S05	**已注销的 profileId 不再有效。** 注销生效点之后，被删除的 profileId 不能再触发对该 profile 的更新或被视为有效注册。	无	只确认可达	低	不确定
A08.7	**profileId 绑定到创建者和 nwdafId。** DCCF 接受的每个 profileId 都对应注册时由该消费者创建的那一个 profile，其 nwdafId/adrfId 与创建时一致。	无	不成立（1 个攻击）	中	不确定
Z03.8	**对 profile 的更新和注销权限。** 一个令牌只能用于其授权主体所声明的消费者身份、其所属的 profileId 以及被授权的操作类型；不能用于其他 NF 的 profile 或其他操作。	无	不成立（2 个攻击）	高	不确定
U01	**响应与请求匹配且删除是最终的。** 注册/更新响应与对应请求匹配；注销后旧 profileId 失效。	无	成立	低	不确定
X01	**消费者认证 DCCF 端点。** 任何 NF 服务消费者都不会将其访问令牌或其数据采集配置文件（包括 anaSub/dataSub 中的采集信息）发送给它所寻址的 DCCF 实例以外的任何一方：它所通信的端点已认证为 DCCF，因此非 DCCF 的一方永远不会收到本服务的请求。	网络上在 DCCF 地址上应答的一方	成立	低	是

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。

提出的问题及分析者的回答：

问题	回答	依据
Q1: profileId 与创建者身份和 nwdafId/adrfId 的绑定	否。DCCF 记录请求体携带的任何消费者标识，从不将其与已认证的消费者进行比对，因此 ADRF 注册的 profile 可以声称 nwdafId = NWDAF；探测目标是 A08.7，其攻击在多会话模型和展开的生命周期上都成立。	A08.7, A08, A03
Q2: 其他合法 NF 的更新/删除权限是否来自规范明确规则	否。原文中唯一的授权规则是单一的服务级范围 ndccf-contextmanagement，更新和注销操作都不检查注册者，因此任何知道 profileId 的 NF 都能触及另一个 NF 的 profile；这就是 Z03.8、Z03 和 A08 报告的弱点。	Z03.8, Z03, A08
Q3: 注册/更新响应关联与注销后的有限状态行为	响应与请求相关联：只有当 201 响应中收到的 profileId 在更新和注销响应中返回时，消费者才完成一次生命周期，这一点成立。对于注销后的状态，原文只说 profile 被删除；在展开的生命周期上，存储不再持有该 profileId，后续对该 profileId 的请求会被拒绝，但 ProVerif 无法证明该拒绝是可到达的，因此删除问题仍部分开放（目标 U01 和 S05）。	U01, S05

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给出了什么或未能给出什么，密码学被视为理想的，与实现无关。

本次分析未覆盖的内容，如分析者所述：

- 服务化接口上和 NRF 链路上的 TLS 被建模为每对端点之间的一条认证机密通道，因此网络攻击者既不能读取、修改也不能注入该流量；只有真正的 NF 才能威胁这些消息。
- 访问令牌是对主体、范围和受众的签名；过期、缓存和撤销均未建模，模型的授权服务器只签发全服务范围的令牌，因此服务之间的范围混淆仅表现为针对另一服务的令牌。
- 在 model.pv 中，DCCF 的 profile 存储是一张 ProVerif 表，只增不减，因此注销所执行的删除无法表示，且 profileId 是一个新的不可猜的名字；因此生命周期也在 bounded.pv 中建模，展开为四步，带有贯穿状态并按序发放 profileId。
- 在 bounded.pv 中，ProVerif 无法为链条最后几步重构迹，因此展开模型后续步骤的供给未获证明；Tamarin+ Agent 可以解决这些问题。
- 请求体被简化为一个消费者标识和一个采集信息值；结构化的 NdccfDataCollectionProfile 类型、其强制属性以及 5.2.7 条的错误响应均未建模。
- 本分析仅覆盖此子过程：从源采集数据或分析、通知、检索以及 DCCF 之间的传输不在范围内。
- 网络攻击者和持有自身凭据的合法 NF 之外的其他攻击者未建模：从另一端点出示的被窃令牌、签发令牌的 NRF 以及不诚实的 DCCF 均无任何查询能够回答。
- 仅建立了迹的性质：关于可用性、吞吐量或签名、令牌或 TLS 的计算强度均未涉及。

宿主记录的内容：

- 无答案：S05。
- A08、Z03、S05、A08.7、Z03.8、U01 的回答依赖于模型简化的一个操作（附录 D）。
- 分析者为性质设定了攻击者条件，但既没有证明也没有攻击的（附录 D）：S05 在 `network` 下; S05 在 `legit_nf` 下。
- 没带对照查询的证明（附录 C）：A02: A02_auth_upd, A02: A02_auth, A08: A08_created_b, Z01: Z01_upd, Z01: Z01_reg, U01: A03_body, X01: X01_secret。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。

每条性质还有什么悬而未决，以及什么能解决它，见附录 A。

附录 A：逐条性质的解读

这是分析者自己对证明器回答的解读，不是证明器的回答：这些回答对协议意味着什么，解读基于什么，能否照原样使用，以及它有什么价值。

A02: 每个请求的发起者经过认证

每当 DCCF 接受一个注册、更新或注销请求，声称发起方（NWDAF 或 ADRF 实例）必须确实以该身份运行过本服务。

每个被接受的注册、更新和注销都由 NRF 签发给出示该令牌的 NF 的令牌作为支撑，因此任何外部方和任何 NF 都不能以它不持有的身份行事。不能保证的是 profile 内部写入的身份：NF 可以注册一个声称是另一个 NF 的 nwdafId 或 adrfId 的 profile，因为 DCCF 从不比对两者。

- 结果：不成立（1 个攻击）；回答了所问：部分；有保留地可用
- 价值 中：证明覆盖了原文所要求的那一半目标；另一半上的攻击是消费者标识的偏移，原文并未明文禁止。
- 文稿中是否陈述：不明确。所依据的原文：ts-33-501:2871, s0026, s0033, s0064, s0079
- 依据：TLS 被建模为每对端点之间的一条认证机密通道。；令牌是对主体、范围和受众的签名；过期未建模。
- 开放：原文没有任何内容说明 DCCF 是否比对请求体标识与已认证身份；声称它会比对的主张是被攻击的，而非被证明的。
- 下一步：在双方之间有 SCP 的模型中，对注销提出同样的主张，正如 TS 33.501 第 5.9.2.1 条的注释所允许的。

A03: 注册的消费者标识与发送方匹配

每当 DCCF 接受一次注册，都存在一个匹配的消费者运行，双方对采集信息（anaSub/dataSub）与消费者标识（nwdafId/adrfId/nwdafSetId/adrfSetId）一致。

存储的采集信息正是注册 NF 所发送的内容，因此无人能在途中提供或篡改它。与其一同存储的消费者标识并未与发送方绑定：ADRF 可以注册一个读起来像是 NWDAF 请求的 profile。

- 结果：不成立（1 个攻击）；回答了所问：部分；有保留地可用
- 价值 中：存储 profile 的读者会对谁注册了该采集产生误解；原文的 shall 条款覆盖请求体的内容，但不覆盖是谁写的。
- 文稿中是否陈述：不明确。所依据的原文：s0033, s0047, s0052, s0056, s0196
- 依据：TLS 是端点通道；SBI 传输未被攻击。；请求体被建模为一个消费者标识加一个采集信息值。
- 开放：DCCF 是否被要求比对请求体标识与已认证消费者。
- 下一步：让编辑决定请求体标识是否必须等于令牌主体，然后要么添加检查，要么明确声明该自由。

A08: profileId 绑定到其创建消费者

DCCF 接受的每个 profileId 都对应注册时由该消费者创建的那一个 profile，其采集信息与消费者标识与创建时一致。

在多会话模型上，只有当 DCCF 已持有某个 profile 时才会更新它，因此更新不能凭空创造 profile；profileId 是新的，NF 无法得知另一个 NF 的值，因此在那里跨 NF 写入不可到达。在展开的生命周期上，profileId 按序发放，ADRF 更新了 NWDAF 的 profileId，而 DCCF 接受了它。

- 结果：不成立（1 个攻击）；回答了所问：部分；有保留地可用
- 价值 中：它隔离了缺口：原文既没有将 profileId 绑定到其创建者，也没有将请求绑定到注册者，而回答取决于原文留待开放的一个假设。
- 文稿中是否陈述：不明确。所依据的原文：s0054, s0056, s0064, s0148
- 依据：在 model.pv 上，profileId 是一个新的不可猜的名字，原文并未要求这一点；在 bounded.pv 上它是可预测的，原文并未禁止这一点。；在 bounded.pv 上链条有四步，ProVerif 无法证明其后续步骤是可到达的。；更新替换存储的内容，但模型没有将内容贯穿更新传递。
- 开放：profileId 是否可以被猜到，这决定了另一个 NF 究竟能否寻址某个 profile；原文对其生成方式没有任何说明。
- 下一步：跨更新对存储内容建模，使覆盖另一个 NF 的 profile 表现为 DCCF 将据以行动的内容发生改变。

S05: 已注销的 profileId 不再有效

注销生效点之后，被删除的 profileId 不能再触发对该 profile 的更新或被视为有效注册。

在展开的生命周期上，注销清空了 DCCF 的存储，因此后续指名该 profileId 的请求找不到 profile，会被拒绝而不是重新注册它。多会话模型只能表明注销至少会被接受，因为它的表无法遗忘 profile；它报告的否证是该表的产物。

- 结果：仅可到达；是否回答了所问：部分；可在有保留条件的情况下使用
- 价值低：该答案涉及 ProVerif 只能以展开链表达的状态，结论性部分无法得到证实。
- 文稿中是否说明：不明确。所依据的原文单元：s0079, s0080, s0081, s0082
- 依据：生命周期只有四步被展开，ProVerif 无法为最后一步重建轨迹。；原文没有说明 profileId 在删除后是否可以被重用。
- 待解决：后续请求指名某个已注销的 profileId 时，它是被拒绝还是被静默重新创建；展开模型的拒绝无法被证实为可到达。
- 下一步：重建展开链，使 ProVerif 能到达注销和后续更新，例如让每个请求使用自己的信道，然后再次询问删除后的主张。

A08.7: profileId 绑定到创建者和 nwdafId

DCCF 接受的每个 profileId 都对应注册时由该消费者创建的那一个 profile，其 nwdafId/adrfId 与创建时一致。

profile 携带由发送方选择的标识符，因此 profile 的读取者不能依赖它来判断是谁注册了该集合。profileId 无法弥补这一点：未注册该 profile 的消费者可以更新它。

- 结果：不成立（1 个攻击）；是否回答了所问：部分；可在有保留条件的情况下使用
- 价值中：它回答了 profile 的读取者会问的问题，即 nwdafId 是否说明了是谁注册的，答案是否。
- 文稿中是否说明：不明确。所依据的原文单元：s0047, s0054, s0056, s0064, s0148, s0196
- 依据：profile 主体被建模为消费者标识符加集合信息。；两个模型对 profileId 可预测性的处理不同。
- 待解决：标识符改变后 DCCF 收集的内容是否也随之改变，不在所分析的子过程范围内。
- 下一步：在存储保留 profile 内容的模型上询问同样的主张，以便看到 NWDAF 已注册集合信息被覆盖的情况。

Z01: 被接受的操作携带有效的服务令牌

每次 DCCF 提交注册、更新或注销操作，都必须有授权服务器签发的、对 ndccf-contextmanagement 服务有效的令牌。

没有经 NRF 为 ndccf-contextmanagement 范围以及为该 DCCF 签发的令牌，任何操作都不会被提交；同一 NRF 为其他服务签发的令牌会被拒绝。这就是检查的全部：令牌不绑定到 profile，因此它能打开 DCCF 持有的每一个 profile。

- 结果：成立；是否回答了所问：部分；可在有保留条件的情况下使用
- 价值低：它确认了规范和 RFC 6749 对授权步骤的要求；该结果并无悬念。
- 文稿中是否说明：不明确。所依据的原文单元：rfc6749:2870, ts-33-501:2883, s0223, s0224, s0226
- 依据：令牌是对主体、范围和受众的签名；过期、缓存和吊销未被建模。；模型中 NRF 只签发整服务范围的令牌。
- 待解决：操作类型绑定：该 API 为注册、更新和注销定义了同一个范围，因此令牌也不绑定到操作；原文说明了这一点，且没有查询区分这些操作。
- 下一步：加入令牌过期和第二个授权服务器（客户端凭据授予允许这样做），以观察来自另一个 NRF 的令牌是否会被接受。

Z03: 令牌绑定到主体、资源和操作

一个令牌只能用于其授权主体所声明的消费者身份、其所属的 profileId 以及被授权的操作类型；不能用于其他 NF 的 profile 或其他操作。

令牌不绑定到 profile：DCCF 会接受 ADRF 的令牌去操作一个记录的消费者标识符属于 NWDAF 的 profile，并且在展开的生命周期上接受它去操作 NWDAF 注册的 profileId。单一的服务级范围加上没有任何所有权检查，使令牌成为打开 DCCF 持有的每个 profile 的钥匙；而仅处于网络上的主体无法提交任何操作。

- 结果：不成立（1 个攻击）；是否回答了所问：部分；可在有保留条件的情况下使用
- 价值中：它以机械方式展示了单一范围对该 API 意味着什么，文稿只把它作为范围决策来陈述，而没有说明它对 profile 的后果。
- 文稿中是否说明：不明确。所依据的原文单元：s0064, s0079, s0226
- 依据：跨 NF 部分需要 profileId 已知或可预测。；来自另一个 DCCF 实例的受众混淆未被建模。
- 待解决：跨 NF 部分需要 profileId 已知或可预测，原文对此既不要求也不排除。
- 下一步：在存储保留内容的模型上询问该主张，以便从轨迹中读出另一个 NF 的集合信息被覆盖的情况。

Z03.8: 对 profile 的更新和注销权限

一个令牌只能用于其授权主体所声明的消费者身份、其所属的 profileId 以及被授权的操作类型；不能用于其他 NF 的 profile 或其他操作。

一旦知道 profileId，该子过程中没有任何东西能阻止一个 NF 修改或删除另一个 NF 的注册；在展开的生命周期上，ADRF 正是这样对 NWDAF 的 profile 操作，而 DCCF 接受了。只有 profileId 的不可预测性构成障碍，而原文并不要求它不可预测。

- 结果：不成立（2 个攻击）；是否回答了所问：部分；可在有保留条件的情况下使用
- 价值高：一个允许任何已授权消费者覆盖或删除另一个消费者注册的服务，对其定义的 profile 没有授权模型，而文稿从未说过 profileId 是承载该授权的秘密。

- 文稿中是否说明：不明确。所依据的原文单元：s0064, s0079, s0226
- 依据：在 bounded.pv 上，profileId 在构造上就是可预测的；在 model.pv 上，攻击者无法获得另一个 NF 的 profileId。；修改被接受，但模型没有展示结果内容。
- 待解决：部署是否依赖 profileId 不可猜测；原文没有说明。
- 下一步：对存储的内容建模，使另一个 NF 的注册被修改能表现为 DCCF 将据以行动的内容发生了变化。

U01: 响应与请求匹配且删除是最终的

注册/更新响应与对应请求匹配；注销后旧 profileId 失效。

消费者接受 201、200 或 204 作为对自身请求的应答：只有当此前获得的 profileId 在后续响应中返回时，它才完成自己的生命周期。注销后 profileId 是否失效，此处记录的答案并未确定。

- 结果：成立；是否回答了所问：部分；可在有保留条件的情况下使用
- 价值低：相关的那一半来自请求与响应的配对方式；状态那一半仍未解决。
- 文稿中是否说明：不明确。所依据的原文单元：s0056, s0068, s0069, s0070, s0080, s0081, s0082
- 依据：响应相关性由模型的端点信道承载，这些信道代表基于 TLS 的 HTTP/2 的请求与响应对。
- 待解决：在被接受的注销之后旧 profileId 是否会被拒绝；S05 覆盖了同一问题。
- 下一步：构建展开运行，使 ProVerif 能证实删除后的拒绝，然后在那里回答该目标的后一半。

X01: 消费者认证 DCCF 端点

No NF service consumer ever sends its access token or its data collection profile, including the collection information in anaSub/dataSub, to a party other than the DCCF instance it is addressing: the endpoint it talks to is authenticated as the DCCF, so a party that is not the DCCF never receives a request of this service.

消费者的令牌及其集合信息只会到达它所寻址的 DCCF 实例；模型中的任何其他主体，包括另一个合法 NF，都得不到 NWDAF 注册的集合信息。这成立是因为两个端点之间的传输经过认证，接口规范正是这样规定的。

- 结果：成立；是否回答了所问：是；可在有保留条件的情况下使用
- 价值低：它确认了整个分析所依赖的传输假设，而一旦插入中间方，同一目标就会失效。
- 文稿中是否说明：是。文稿中的位置：s0099, s0100, s0104
- 依据：TLS 被建模为两个端点之间经过认证的机密信道，该结果所证明的内容不超过这一假设。
- 待解决：仅限直接通信：TS 33.501 第 5.9.2.1 条指出，当双方之间存在 SCP 或 SEPP 时，没有双向认证。
- 下一步：在 SCP 等中间方与消费者共享端点的模型中询问同一主张。

附录 C：主张和证明器的答案

证明器为每个目标记录的：达到的级别、达到该级别的查询、是否检查了对照查询，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_auth_upd	4 项中的 2 项	A02_claimed (depends_on_ambiguity)	4 项 proved，1 项 attack
A03	proved		A03_body	checked	A03_bind (depends_on_ambiguity)	1 项 proved，1 项 attack
A08	bounded	变化值展开至 4 步	A08_created_b	2 项中的 1 项	A08_owner (holds_in_source)	2 项 bounded，1 项 attack
Z01	proved		Z01_upd	4 项中的 2 项		4 项 proved
Z03	proved		Z01_upd	checked	A08_owner (holds_in_source)	1 项 proved，1 项 attack
S05	reachable		run		S05_del (model_artifact)	2 项 reachable，1 项 attack
A08.7	bounded	变化值展开至 4 步	A08_created_b	checked	A08_owner (holds_in_source)	1 项 bounded，1 项 attack
Z03.8	proved		Z01_upd	checked	A08_owner (holds_in_source)；A08_owner_b (depends_on_ambiguity)	1 项 proved，2 项 attack
U01	proved		A03_body	2 项中的 1 项		2 项 proved
X01	proved		X01_secret			2 项 proved

逐条主张，以分析者对每条主张的措辞给出；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	被接受的更新由所出示访问令牌所指明身份的 NF 发出：请求在 DCCF 的已认证主体确实以该 NF 的身份运行了服务。（同一主张也适用于注册和注销。）	`legit_nf`	proved	A02_auth_upd	none
A02	被接受的注册由所出示访问令牌所指明身份的 NF 发出：请求在 DCCF 的已认证主体确实以该 NF 的身份运行了服务。	`network`	proved	A02_auth	none
A02	被接受配置所声称的发起者（请求体中的 nwdafId/adrfId）确实运行了本服务：DCCF 会检查请求体中的标识就是已认证的消费者。	`legit_nf`	attack (depends_on_ambiguity)	A02_claimed (Z01_upd 的对照查询)	
A02	被接受的更新由所出示访问令牌所指明身份的 NF 发出：请求在 DCCF 的已认证主体确实以该 NF 的身份运行了服务。	`legit_nf`	proved	A02_auth_upd	checked
A02	被接受的注册由所出示访问令牌所指明身份的 NF 发出。	`network`	proved	A02_auth	checked
A03	被接受配置中记录的消费者标识等于注册该配置的 NF，因此配置的读者可以确信是谁注册了它。	`legit_nf`	attack (depends_on_ambiguity)	A03_bind (A03_body 的对照查询)	

目标	主张	攻击者	回答	查询	对照
A03	被接受配置中存储的采集信息就是已认证消费者在其请求中发送的那份。	`network`	proved	A03_body	checked
A08	被接受的更新所指名的 profileId 就是发出该更新的消费者的配置：一个 NF 只能更新自己注册的配置。	`legit_nf`	attack (holds_in_source)	A08_owner	
A08	被接受的更新所指向的 profileId 是由其之前的某次注册创建的：DCCF 只更新已存在的配置，绝不把未知的 profileId 当作配置处理。	`legit_nf`	bounded (变化值展开至 4 步)	A08_created_b	none
A08	被接受的更新所指向的 profileId 是由其之前的某次注册创建的。	`network`	bounded (变化值展开至 4 步)	A08_created_b	checked
Z01	被接受的更新由 NRF 在对请求 NF 完成认证后签发给它的访问令牌支撑；没有这样的令牌，任何操作都不会被接受。	`legit_nf`	proved	Z01_upd	none
Z01	被接受的注册由 NRF 签发给请求 NF 的访问令牌支撑，并且 DCCF 会拒绝作用域和受众不属于 ndccf-contextmanagement 服务的令牌。	`network`	proved	Z01_reg	none
Z01	被接受的更新由 NRF 在对请求 NF 完成认证后签发给它的访问令牌支撑。	`legit_nf`	proved	Z01_upd	checked
Z01	被接受的注册由 NRF 签发给该 NF 的令牌支撑，并且作用域和受众属于其他服务的令牌会被拒绝。	`network`	proved	Z01_reg	checked
Z03	令牌与其签发给的消费者的配置相绑定：它不能被用来更改带有另一个消费者标识的配置。	`legit_nf`	attack (holds_in_source)	A08_owner	
Z03	没有 NRF 为 ndccf-contextmanagement 服务签发的令牌，任何注册、更新或注销操作都不会被提交，因此只有网络访问权限的一方无法使其中任何一项被提交。	`network`	proved	Z01_upd	checked
S05	DCCF 接受注销，配置从其存储中移除，消费者收到 204 No Content 响应。		reachable	run	
S05	DCCF 接受注销，配置从其存储中移除。	`legit_nf`	reachable	run	
S05	只有 DCCF 已持有的 profileId 才会被接受注销：被移除的配置就是先前注册的那个。	`network`	attack (model_artifact)	S05_del	
A08.7	被接受的更新所指名的 profileId 是注册它的消费者的配置，且记录的 nwdafId/adrfId 就是该消费者的。	`legit_nf`	attack (holds_in_source)	A08_owner	
A08.7	被接受的更新所指向的 profileId 是由其之前的某次注册创建的，且其记录的 nwdafId/adrfId 就是注册时存储的那个。	`network`	bounded (变化值展开至 4 步)	A08_created_b	checked
Z03.8	持有该服务令牌的合法 NF 只能修改或删除自己注册的配置。	`legit_nf`	attack (holds_in_source)	A08_owner	
Z03.8	更新中被接受的 profileId 属于注册它的消费者：DCCF 只接受创建该配置的 NF 发来的更新。	`legit_nf`	attack (depends_on_ambiguity)	A08_owner_b (A08_created_b 的对照查询)	
Z03.8	配置只能由持有该服务令牌的一方修改或删除；仅在网络中的一方做不到。	`network`	proved	Z01_upd	checked
U01	消费者收到的 201 响应携带为其自身请求所创建配置的 profileId，其更新和注销响应携带它所请求的 profileId。	`legit_nf`	proved	A03_body	none

目标	主张	攻击者	回答	查询	对照
U01	消费者收到的 201 响应携带为其自身请求所创建配置的 profileId，并且只有更新和注销响应携带同一个 profileId 时生命周期才会完成。	`network`	proved	A03_body	checked
X01	一个 NF 注册的采集信息只到达它所寻址的 DCCF，不会到达其他任何一方。	`network`	proved	X01_secret	none
X01	一个 NF 注册的采集信息只到达它所寻址的 DCCF，不会到达其他任何一方，包括另一个合法 NF。	`legit_nf`	proved	X01_secret	none

附录 D：分析是怎么跑的

LLM 分析者根据这些文稿构建协议模型，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-20261114T211021，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：54，证明器 5 秒，实际耗时 16 分钟，费用 0.135 USD。

ProVerif：运行了 89 个证明，87 个有结论，2 个无答案，耗时 3 秒，记录了 31 个答案。

分析了什么

协议	DCCF collection profile lifecycle
版本	19.7.0
分析范围	两个 NF 与一个 DCCF；选择 analytics anaSub 的不涉及单个 UE 的聚合信息，注册、更新、注销独立 data-collection-profile。排除真正上游数据采集及跨 DCCF 转移。
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	来自下列章节的 226 个编号原文单元（31762 字节）。本报告中的 s0001 之类的引用就是这些原文单元之一

分析所依据的文稿章节：

文稿	章节	标题
main	4.3	Ndccf_ContextManagement 服务
main	4.3.1	服务描述
main	4.3.1.1	概述
main	4.3.1.2	服务架构
main	4.3.1.3	网络功能
main	4.3.1.3.1	数据收集协调功能（DCCF）
main	4.3.1.3.2	NF 服务消费者
main	4.3.2	服务操作
main	4.3.2.1	引言
main	4.3.2.2	Ndccf_ContextManagement_Register 服务操作
main	4.3.2.2.1	概述
main	4.3.2.2.2	向 DCCF 注册数据收集配置
main	4.3.2.3	Ndccf_ContextManagement_Update 服务操作
main	4.3.2.3.1	概述
main	4.3.2.3.2	更新已注册的数据收集配置
main	4.3.2.4	Ndccf_ContextManagement_Deregister 服务操作

文稿	章节	标题
main	4.3.2.4.1	概述
main	4.3.2.4.2	注销数据收集配置
main	5.2	Ndccf_ContextManagement 服务 API
main	5.2.1	引言
main	5.2.2	HTTP 的使用
main	5.2.2.1	概述
main	5.2.2.2	HTTP 标准头
main	5.2.2.2.1	概述
main	5.2.2.2.2	内容类型
main	5.2.2.3	HTTP 自定义头
main	5.2.3	资源
main	5.2.3.1	概述
main	5.2.3.2	资源：DCCF 数据收集配置
main	5.2.3.2.1	描述
main	5.2.3.2.2	资源定义
main	5.2.3.2.3	资源的标准方法
main	5.2.3.2.3.1	POST
main	5.2.3.2.4	资源的自定义操作
main	5.2.3.3	资源：单个 DCCF 数据收集配置
main	5.2.3.3.1	描述
main	5.2.3.3.2	资源定义
main	5.2.3.3.3	资源的标准方法
main	5.2.3.3.3.1	PUT
main	5.2.3.3.3.2	DELETE
main	5.2.3.3.4	资源的自定义操作
main	5.2.4	无关联资源的自定义操作
main	5.2.5	通知
main	5.2.6	数据模型
main	5.2.6.1	概述
main	5.2.6.2	结构化数据类型
main	5.2.6.2.1	引言
main	5.2.6.2.2	类型：NdccfDataCollectionProfile
main	5.2.6.3	简单数据类型与枚举
main	5.2.6.3.1	引言
main	5.2.6.3.2	简单数据类型
main	5.2.6.4	描述备选数据类型或数据类型组合的数据类型

文稿	章节	标题
main	5.2.6.5	二进制数据
main	5.2.7	错误处理
main	5.2.7.1	概述
main	5.2.7.2	协议错误
main	5.2.7.3	应用错误
main	5.2.8	特性协商
main	5.2.9	安全

协议的目的，按源文步骤在文稿中的定位：两个 NF 与一个 DCCF；选择 analytics anaSub 的不涉及单个 UE 的聚合信息，注册、更新、注销独立 data-collection-profile。排除真正的上游数据采集及跨 DCCF 转移。

对照文稿定性质

在编写任何查询之前，分析者先把草拟的性质清单与文稿本身对协议目的的表述逐条对照。名称、性质所涉及的对象以及文稿的说法都出自分析者的解读；所引用的位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A02	每个请求的发起者经过认证	DCCF 确信：被接受的注册、更新或注销请求，是由该请求所声称、且令牌所指名的那个 NF 实例（NWDAF 或 ADRF）发出的。	这些操作由 NF 服务消费者发起（s0026、s0033、s0064、s0079）；访问可使用来自 NRF 的 OAuth2 客户端凭据进行授权（s0223、s0224）；NF 服务请求与响应应支持消费者与生产者之间的双向认证。	ts-33-501:2871, s0026, s0033, s0064, s0079, s0223, s0224
A03	注册的消费者标识与发送方匹配	配置文件中存储的消费者标识（nwdafId、adrfId、nwdafSetId 或 adrfSetId）连同采集信息，就是注册该配置的 NF 的身份，因此读取该配置文件的人能确信是谁注册了它。	请求体应包含与 NF 服务消费者相关的标识之一（s0033、s0047 至 s0051）以及 anaSub/dataSub 中的采集信息；DCCF 创建并存储该配置文件，然后将其返回（s0052 至 s0056）。原文没有说明 DCCF 会将请求体中的标识与已认证身份进行比对。	s0033, s0047, s0052, s0056, s0196
A08	profileId 绑定到其创建消费者	profileId：DCCF 与注册该配置的 NF 都确信，更新或注销所寻址的 profileId 就是该消费者创建的配置文件，并且它仍然携带注册时给出的采集信息和消费者标识。	DCCF 分配 profileId，并在 201 响应和 Location 头中返回它（s0054、s0056）；更新和注销通过 {profileId} 寻址该配置文件（s0064、s0079、s0148）。原文没有说明 profileId 如何生成，也没有说明它是否可被猜出。	s0054, s0056, s0064, s0079, s0148
Z01	被接受的操作携带有效的服务令牌	OAuth2 访问令牌：只有当 NF 出示 NRF 签发给该 NF、用于 ndccf-contextmanagement 服务且针对此 DCCF 的令牌时，DCCF 才接受注册、更新或注销操作。	访问可使用 OAuth2 客户端凭据进行授权，NRF 作为授权服务器，消费者在消费该服务前获取令牌（s0223、s0224）；API 定义了单一范围 ndccf-contextmanagement（s0226）；令牌不得由未授权方生成或猜测。	rfc6749:2870, ts-33-501:2883, s0223, s0224, s0226

ID	简称	对象	文稿怎么说	出处
Z03	令牌绑定到主体、资源和操作	profileId 与 OAuth2 令牌：令牌只能用于签发给它的消费者所拥有的配置文件，并且只能用于它所授权的操作，不能用于另一个 NF 的配置文件。	API 为整个服务定义了一个范围，没有资源级或操作级范围（s0226）；RFC 6749 记载，资源服务器没有方法来检查出示的令牌是否签发给出示它的客户端，并且当凭据未被保密时，恶意客户端可以冒充另一个客户端行事（rfc6749 10.2、10.3）。任务边界指出，服务范围本身并不意味着对某个资源的所有权。	rfc6749:2879, rfc6749:2827, s0064, s0079, s0226
S05	已注销的 profileId 不再有效	DCCF 的配置文件存储：一旦某个 profileId 的注销被接受，该 profileId 就不再标识任何配置文件，不能再被更新，也不能再被视为有效注册。	在 DELETE 被接受后，DCCF 应删除相应的已注册配置文件并返回 204 No Content（s0080、s0081、s0082）；单个配置文件资源由 {profileId} 标识（s0145、s0148）。原文没有说明 profileId 在删除后是否可被重用。	s0079, s0080, s0081, s0082
A08.7	profileId 绑定到创建者和 nwdafId	profileId 连同该配置文件的 nwdafId/adrfId：被接受的 profileId 属于该消费者注册的配置文件，其记录的消费者标识就是该消费者的标识。	DCCF 分配 profileId（s0054）并返回它（s0056）；更新和注销通过 {profileId} 寻址该配置文件（s0064、s0079、s0148）；配置文件主体携带 nwdafId/adrfId（s0047 至 s0051、s0196）。原文没有说明 profileId 的生成方式。	s0054, s0056, s0064, s0079, s0148, s0196
Z03.8	对 profile 的更新和注销权限	对另一个 NF 的配置文件的更新和注销操作：持有该服务令牌的合法 NF 只能修改或删除自己注册的配置文件。	更新和注销仅通过 {profileId} 寻址（s0064、s0079）；API 定义了单一的服务级范围，没有资源级范围（s0226）；任务边界指出，服务范围不等于资源所有权。	rfc6749:2879, s0064, s0079, s0223, s0226
U01	响应与请求匹配且删除是最终的	NF 服务消费者：它收到的 201/200 响应属于它自己的请求，并且在注销被接受后，旧的 profileId 即告失效。	DCCF 返回 201 以及所创建的配置文件和携带该配置文件 URI 的 Location 头（s0056）；PUT 返回 200 及表示，或返回 204（s0068 至 s0070）；DELETE 删除该配置文件并返回 204（s0080 至 s0082）。	s0056, s0068, s0069, s0070, s0080, s0081, s0082
X01	消费者认证 DCCF 端点	NF 服务消费者（NWDAF 或 ADRF），持有令牌和配置文件内容，面对一个冒充 DCCF 的对方。	NF 服务请求与响应流程应支持 NF 服务消费者与 NF 服务生产者之间直接通信的双向认证。	ts-33-501:2871

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

宿主把文稿中说明协议保护什么、防范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，或者为什么两者都不选。选了 28 段，没选 370 段。列出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	文中说 OAuth 的安全考虑取决于客户端类型，并指向 OAuth 威胁模型；与本 API 的性质无关。

段落	出处	定为	不选的原因
P003 至 P006	rfc6749:2790-2823	Z01, `legit_nf`	
P007 至 P010	rfc6749:2827-2850	Z01, Z03, `legit_nf`	
P011 至 P015	rfc6749:2859-2881	Z01, Z03, `network`, `legit_nf`	
P016 至 P021	rfc6749:2885-2917	没有陈述目标	本 API 使用的客户端凭据授予中不存在刷新令牌（s0223）。
P022 至 P025	rfc6749:2921-2943	没有陈述目标	未使用授权码授予及其重定向端点：消费者使用客户端凭据授予（s0223）。
P026 至 P029	rfc6749:2947-2980	没有陈述目标	客户端凭据授予中不注册也不使用重定向 URI。
P030 至 P033	rfc6749:2984-3003	没有陈述目标	未使用资源所有者口令授予；本 API 中没有资源所有者。
P034 至 P035	rfc6749:3013-3019	Z01, `network`	
P036	rfc6749:3023-3028	X01, Z01, `network`	
P037 至 P039	rfc6749:3032-3041	证明器问不了	猜中生成令牌的概率（2^-128）是令牌生成与熵的性质，无法向符号证明器提问。
P040 至 P042	rfc6749:3045-3066	没有陈述目标	针对终端用户的钓鱼；本 API 没有终端用户。
P043 至 P048	rfc6749:3070-3110	没有陈述目标	CSRF 防护涉及授权端点和用户代理；客户端凭据授予两者都没有。
P049 至 P050	rfc6749:3119-3138	没有陈述目标	点击劫持涉及向终端用户展示的授权页面；本 API 中没有这样的页面。
P051 至 P052	rfc6749:3142-3150	没有陈述目标	所提到的值（state、redirect_uri）在客户端凭据授予中不存在；本 API 的请求体校验由 A03 和 A08 覆盖。
P053 至 P054	rfc6749:3154-3167	没有陈述目标	开放重定向涉及重定向端点，本授予不使用重定向端点。
P055 至 P062	rfc6749:3170-3210	没有陈述目标	未使用隐式流程及其令牌替换和受众限制：消费者通过后端通道以客户端凭据获取令牌（s0224）。
P063	ts-33-501:2491-2491	不在范围内	涉及 UE 安全特性的降级，而非此处建模的 SBI 子过程。
P064 至 P072	ts-33-501:2495-2511	不在范围内	涉及 UE 签约和服务网络认证的要求；任务不包含逐 UE 的过程。
P073 至 P074	ts-33-501:2515-2517	不在范围内	涉及 5GC 和 NG-RAN 用于 AS 与 NAS 保护的密钥的要求。
P075 至 P078	ts-33-501:2523-2529	不在范围内	涉及 UE 与 ng-eNB 之间经 Uu 的加密和完整性。
P079 至 P089	ts-33-501:2533-2553	不在范围内	涉及无线接口和 N3 接口上的用户数据与信令数据机密性。
P090 至 P103	ts-33-501:2557-2583	不在范围内	涉及无线接口和 N3 接口上的用户数据与信令数据完整性。
P104 至 P110	ts-33-501:2587-2599	不在范围内	涉及 UE、USIM 和网络功能中签约凭据的安全存储与处理，而非 DCCF 的 profile。

段落	出处	定为	不选的原因
P111 至 P125	ts-33-501:2603-2631	不在范围内	涉及签约用户隐私；所分析的子过程选择的是聚合分析，不涉及单个 UE。
P126 至 P213	ts-33-501:2637-2853	不在范围内	涉及 UE、NG-RAN、gNB、AUSF、UDM 和 UDR 对接入网和签约用户过程的要求；任务只覆盖 SBI 上的 DCCF profile 生命周期。
P214	ts-33-501:2859-2859	Z01, X01, `network`	
P215	ts-33-501:2865-2865	没有陈述目标	服务发现与注册的重放防护是 SBI 传输的性质，由端点通道模型承载，而非 profile 生命周期的消息级性质；生命周期自身的重放问题由 S05 和 U01 提出。
P216	ts-33-501:2867-2867	Z01	
P217	ts-33-501:2869-2869	不在范围内	管理域之间的拓扑隐藏属于 SEPP 和漫游，任务不包含这部分。
P218	ts-33-501:2871-2871	A02, X01, `network`, `legit_nf`	
P219	ts-33-501:2873-2873	A02, `legit_nf`	
P220	ts-33-501:2875-2875	A03, Z01, A08	
P221 至 P227	ts-33-501:2879-2891	Z01, A02, `network`, `legit_nf`	
P228 至 P234	ts-33-501:2895-2907	不在范围内	涉及 NEF 和 AF 接口的要求；所分析的子过程没有对外暴露。
P235 至 P241	ts-33-501:2911-2923	不在范围内	涉及 SCP 用于间接通信的要求，任务不包含这部分。
P242 至 P312	ts-33-501:2929-3075	不在范围内	涉及与 SEPP、漫游中间节点、属性保护和 IPUPS 的端到端互联安全：漫游和跨 PLMN 通信不在所建模的子过程之内。
P313 至 P314	ts-33-501:3079-3081	不在范围内	涉及互联的 IPUPS 要求，不在所建模的子过程之内。
P315 至 P318	ts-33-501:3085-3091	不在范围内	涉及 3gpp-Sbi-Originating-Network-Id 头和 PLMN ID 处理，属于跨 PLMN 信令。
P319 至 P337	ts-33-501:3097-3141	不在范围内	涉及运营商网络的信令监控、安全可视性和可配置性要求，而非本 API 的消息。
P338 至 P369	ts-33-501:3149-3215	不在范围内	涉及接入网的加密与完整性算法标识及算法选择；密码原语的强度不在符号模型的范围之内。
P370 至 P380	ts-33-501:3219-6548	不在范围内	涉及 5G-RG、NSSAAF 和 N3 的要求，不属于 DCCF profile 生命周期。
P381 至 P398	ts-33-501:9656-11901	不在范围内	附录涉及 EAP-TLS、IAB、MBSF/MBSTF 和 NEF 的要求，不在所建模的子过程之内。

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 13 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 2 项，没设的 11 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	DCCF	参与方	按定义全部失守	DCCF 是执行所有检查并保存存储的一方；受控的 DCCF 会使全部九个目标按定义失效，且文稿没有要求在 DCCF 不诚实的情况下考察其中任何一个。
S02	NWDAF 消费者	参与方	`legit_nf`	
S03	ADRF 消费者	参与方	`legit_nf`	
S04	NRF 授权服务器	参与方	按定义全部失守	授权服务器受控后，可以为任意主体铸造令牌，因此所有基于令牌的性质（A02、Z01、Z03）按定义失效；profile 存储相关的性质（A08、S05）不依赖 NRF，改为在网络攻击者和合法 NF 攻击者条件下考察。
S05	profile 归属身份	值	攻击者得不到更多	profile 的归属身份是 DCCF 从请求体中记录的内容；读取它的参与方除了发送请求之外得不到任何额外信息，而发送请求正是已定义的合法 NF 攻击者。
S06	采集信息	值	攻击者得不到更多	采集信息经 NF 自己的端点通道传给 DCCF；读取它的参与方持有该 NF 的凭据，即合法 NF 攻击者。
S07	消费者标识	值	攻击者得不到更多	消费者标识 nwdafId/adrfId 是 NF 实例的公开身份。
S08	profileId	值	不在范围内	profileId 能否被猜中取决于 DCCF 如何生成它，而原文没有规定；分析通过在带新标识的模型上以及在展开模型上考察 A08 覆盖了两种解读，而猜测本身是熵的问题。
S09	OAuth2 访问令牌	值	攻击者得不到更多	令牌只能从 NF 自己的已认证端点通道使用，因此只持有令牌而没有 NF 凭据的参与方得不到任何东西；同时持有两者的参与方即合法 NF 攻击者。
S10	profile 资源	值	攻击者得不到更多	profile 资源位于 DCCF，只能经由所建模的检查到达；到达它的途径就是成为合法 NF，而这正是已定义的攻击者。
S11	profile 内容	值	攻击者得不到更多	profile 内容是消费者自己选择并发送的采集信息；持有它的参与方持有的正是注册 NF 所发送的内容。
S12	nwdafId/adrfId	值	攻击者得不到更多	nwdafId/adrfId 是用于发现的公开 NF 实例和集合标识。
S13	响应	值	攻击者得不到更多	响应把 profile 表示和 profileId 送回发出请求的 NF；读取它的参与方位于该 NF 的端点通道上。

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”是指该攻击者条件下证明器对某条主张未能判定或没有记录任何结果；空单元格是指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`	仅网络：可读取、丢弃、修改并发送消息，不持有任何一方的秘密		
`legit_nf`	一个真实 NF 服务消费者实例（同一网络中的 ADRF 或 NWDAF）的客户端凭据，以及由 NRF 签发的、用于 ndccf-contextmanagement 范围的有效访问令牌；它通过自己经过认证的 TLS 连接与 DCCF 通信，使用 HTTP/2，自行选择放入请求体中的消费者标识和集合信息，并可指定任何它能构造或猜到的 profileId。	任务边界将外部网络攻击者与持有自身凭据的恶意合法 NF 区分开来；该服务的消费者是 NWDAF 和 ADRF（s0010, s0011），该 API 有一个服务级范围（s0226）。	rfc6749:2827, s0010, s0011, s0223, s0226

ID	`network`	`legit_nf`
A02	成立	不成立

ID	network	legit_nf
A03	成立	不成立
A08	成立	不成立
Z01	成立	成立
Z03	成立	不成立
S05	无答案	无答案
A08.7	成立	不成立
Z03.8	成立	不成立
U01	成立	成立
X01	成立	成立

原文和目标

Source: 226 units, 31762 bytes, frozen in source-delivery-01.json. Targets: 7 approved (A02, A03, A08, Z01, Z03, S05, U01); the questions asked: 3. 目标列表已按草稿事先批准，批准时使用了 --approve-targets。

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

- approve_package: 批准源文包 protocol-ddaaa5507937a0ef8f8ff266ee1e

本步骤自报的情况：2 次请求（2 次失败），0.000 USD，输入 68074 个 token，输出 7953 个 token，用时 222 秒。

关于证明器和模型的说明

- A08：在 ProVerif 中得到了答案，分析者为该目标标注了 changing_state
- Z03：在 ProVerif 中得到了答案，分析者为该目标标注了 changing_state
- S05：在 ProVerif 中得到了答案，分析者为该目标标注了 changing_state
- A08.7：在 ProVerif 中得到了答案，分析者为该目标标注了 changing_state
- Z03.8：在 ProVerif 中得到了答案，分析者为该目标标注了 changing_state
- U01：在 ProVerif 中得到了答案，分析者为该目标标注了 changing_state

对代数做了简化（xor、dh_algebra、other_algebra）的 ProVerif 模型给出的攻击者能力弱于真实运算：在其中得到的证明是针对更弱攻击者的证明，在其中发现的攻击仍是攻击。省略了对会变化的值（changing_state）的检查的 ProVerif 模型会让各方接受更多：在其中得到的证明成立，在其中发现的攻击可能只是缺少了那项检查所致。Tamarin+ Agent 按原样对这些进行建模。

以步数表示的界是在 bounded.pv 上得到的答案：这是分析者的第二个 ProVerif 模型，其中持有会变化的值的一方被展开到该步数，并保留其检查。该模型做的事情少于协议实际所做的，因此该性质只对这些步成立，不能推得更远；在其中发现的攻击仍是攻击。步数由分析者给出。

分析者的状态审查

这一部分是分析者本人对原文的解读，不是证明器的答案。ProVerif 显示一列中的“是”表示 ProVerif 在展开后的模型中找到了包含这些步骤的迹，因此这些步骤是可能的；至于它们是否破坏了某个目标，只有答案表中的攻击才能说明，它们还付出了什么代价则是分析者的解读。“nothing found”说明查看了哪些模式，并不能证明任何事。

值	由谁持有	何时变化	何时检查	出处
profileId 以及所存储的数据收集配置文件（消费者标识、集合信息）	DCCF	注册时创建，更新时替换其内容，注销时删除	更新或注销会指名它：DCCF 必须找到该 profileId 对应的已存储配置	s0052, s0053, s0054, s0055, s0064, s0065, s0066, s0067, s0079, s0080, s0081, s0082, s0209, s0214
用于 ndccf-contextmanagement 范围的 OAuth2 访问令牌	NF 服务消费者（NWDAF、ADRF）	NF 在消费该服务前从 NRF 获取新令牌	在每个注册、更新和注销请求上，DCCF 校验令牌的签名、范围和受众	s0223, s0224, s0226

模式	结果	目标	ProVerif 是否显示	步骤
replay	candidate		否：未用 ProVerif 尝试	NF 服务消费者携带有效令牌发送 POST /data-collection-profiles（s0033）。/ DCCF 创建新配置，分配 profileId 并存储（s0052 至 s0055）。/ 同一个 POST 携带同一个令牌再次发送；DCCF 没有识别重复的规则，于是又创建了第二个配置，并分配了第二个 profileId（s0052 至 s0055）。/ 于是一次注册意图对应了两个配置；原文和模型都没有排除这种情况。
repeat_one_time_action	candidate		否：未用 ProVerif 尝试	消费者因为 201 响应丢失而重复其注册请求。/ DCCF 为同一集合信息又创建了一个配置（s0053 至 s0055）。/ 因此注册并不是每个消费者和集合信息只执行一次的动作。
update_before_check	candidate		否：未用 ProVerif 尝试	消费者发送 PUT /data-collection-profiles/{profileId}，指定一个不存在的 profileId（例如已注销的，或消费者猜到的）。/ DCCF 被告知更新相应 profileId 的配置并存储该配置（s0065 至 s0067）；原文没有提到存在性检查，也没有提到这种情况下的 404 响应。/ DCCF 要么以错误拒绝（s0071），要么在该 profileId 下存储配置；原文没有说明是哪一种。
race_between_sessions	candidate	Z03.8	是：ProVerif 在 bounded.pv 上否证了 `A08_owner_b`	NWDAF 注册一个配置并收到其 profileId（s0054, s0056）。/ ADRF 持有自己用于同一服务范围的有效令牌，对该 profileId 发送 PUT（s0064）。/ DCCF 检查了令牌，但没有检查是谁注册了该配置，于是替换了配置内容，因此 NWDAF 的注册现在表达的是 ADRF 发送的内容（s0065 至 s0067）。/ 原文中没有任何内容规定这两个写入者的先后顺序，也没有把 profileId 与其注册者绑定。
distinguishable_failure	candidate		否：未用 ProVerif 尝试	消费者对它并不拥有的 profileId 发送 PUT 或 DELETE。/ DCCF 以 HTTP 错误应答；原文把错误的选择留给第 5.2.7 条（s0071, s0083），并单独列出了应用层错误（s0214 至 s0217）。/ 如果“无此 profileId”的错误与“不是你的配置”的错误不同，消费者就能从差异中得知某个 profileId 是否存在，即其他 NF 的配置是如何编号的。
no_rollback_on_failure	nothing found			原文指出出错时 DCCF 发送错误响应（s0071, s0083），没有提到部分应用的更新，因此不会留下被改动的值；展开后的模型只在令牌检查通过后才应用更新。
desynchronisation	nothing found			消费者除了收到的响应之外不持有配置的副本；注销被接受后，双方都认为该 profileId 已不存在，原文也没有给出会让二者产生分歧的恢复步骤。
wrap_or_window	nothing found			配置生命周期不维护计数器或接受窗口；profileId 由 DCCF 分配，而不是由消费者递增，原文也没有设定任何窗口。

分析者为各目标标注的内容：

步骤	目标	特征	理由
36	A08, A08.7, S05, U01, Z03, Z03.8	changing_state, many_sessions	DCCF 存储的配置表及其分配的 profileId 在整个生命周期中会变化：注册创建配置，更新替换其内容，注销删除它；profileId 是 DCCF 发放的值，其他方可能尝试指名它。ProVerif 的表只会增长，因此这些性质在 model.pv 上针对任意数量的会话提问，并在 bounded.pv 上提问，其中存储被串入一条展开的四步链，包含真实的删除。依赖 profileId 可预测性的答案来自展开后的模型，其中 DCCF 按顺序发放 profileId。

分析者的结束说明

模型和答案已记录。普通流程存在于 model.pv 中（在模拟 TLS 的端点通道上，使用 NRF 签发的令牌，执行注册、更新、注销），并且每个目标在两种攻击者（仅网络，以及持有自身凭据的合法 NF）下都有答案。

成立的内容：每一次已提交的注册、更新或注销都有 NRF 为 ndccf-contextmanagement 范围和此 DCCF 签发的令牌作为依据（Z01，已证明，带对照查询），经过认证的主体确实运行了该服务（A02，已证明），所存储的集合信息就是真实消费者发送的内容（A03，已证明），只有当 DCCF 持有其 profileId 时才会更新配置（A08，在 4 个展开步骤上有界，带对照查询），并且集合信息只到达 DCCF（X01，已证明）。

不成立的是什么，发现又是什么：DCCF 从不比较请求体中的消费者标识符与令牌的主体，也从不检查是谁注册了某个 profileId。因此，第二个合法 NF 可以注册一个声明 nwdafId = NWDAF 的 profile（针对 A03、A02、A08.7 的攻击），并且在 profileId 已知或可预测的情况下——原文对此完全没有限定——更新或注销另一个 NF 的 profile（针对 A08、Z03、Z03.8 的攻击，以及在状态审查中确认的竞态）。新颖性不明确：文稿声明了一个服务级的范围，没有所有权规则，但从未说明这会破坏绑定。

阻碍其余工作的因素：ProVerif 的单调 profile 表无法建模删除，并且会对表上的对应关系产生证伪；它也无法为展开链的后续步骤重建迹，因此 S05 的删除后主张和 U01 的状态部分被记录为 reachable/model-artifact，并保持开放。Tamarin+ Agent 按原样建模已存储的 profile 及其删除，可以解决这些问题。建议（将请求体标识符与令牌主体比较；将 profile 绑定到其注册者，并对来自其他主体的 PUT/DELETE 以 403 拒绝）未经过机器检查，因为修复是缺少一项检查，而不是更改消息。