MFAF 配置上下文传输的安全分析

本报告的其他语言版本：English。

19.7.0 · ProVerif+ Agent 报告

文稿	5G System; Messaging Framework Adaptor Services; Stage 3 （协议定义）
检查的性质	11 条：7 条成立，4 条不成立（5 个攻击）
发现	3

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

Messaging Framework Adaptor Function 的 Nmfaf_ContextManagement 服务把现有的 MFAF 配置从持有它的 MFAF 迁移到另一个 MFAF，使消费者的数据采集或分析采集能在新的 MFAF 上继续。这里涉及三方：配置者，即 DCCF 或 NWDAF，它创建了 Individual MFAF Configuration 并持有其资源 URI；源 MFAF，它持有该配置及其事务引用；目标 MFAF，它发起传输并成为新的持有者。两个 MFAF 使用的访问令牌都由 NRF 签发。

目标 MFAF 发送 ContextTransferReq，携带它想要的配置的资源 URI；源 MFAF 以 200 OK 应答，返回这些配置、传输前到传输后的 URI 映射以及目标 MFAF 的通知信息，并删除自己的副本；目标 MFAF 在其新的事务引用 id 下创建每个配置。交换结束后，目标 MFAF 确信响应来自它所寻址的源 MFAF，并且恰好携带了该源所持有的配置和通知绑定；但整个流程中没有任何环节让源 MFAF 确信请求者有权获得该配置。

各方：

- **配置者（DCCF）**：创建配置，之后更新或删除配置。持有：NF 实例 id、用于 nmfaf-3dadatamanagement 的访问令牌。
- **源 MFAF**：持有配置及其事务引用。持有：配置存储、TLS 凭据。
- **目标 MFAF**：发起传输并成为持有者。持有：NF 实例 id、用于 nmfaf-contextmanagement 的访问令牌、TLS 凭据。
- **NRF**：授权服务器，签发访问令牌。持有：签名密钥。

图 1 展示了文稿所规定的交换过程。

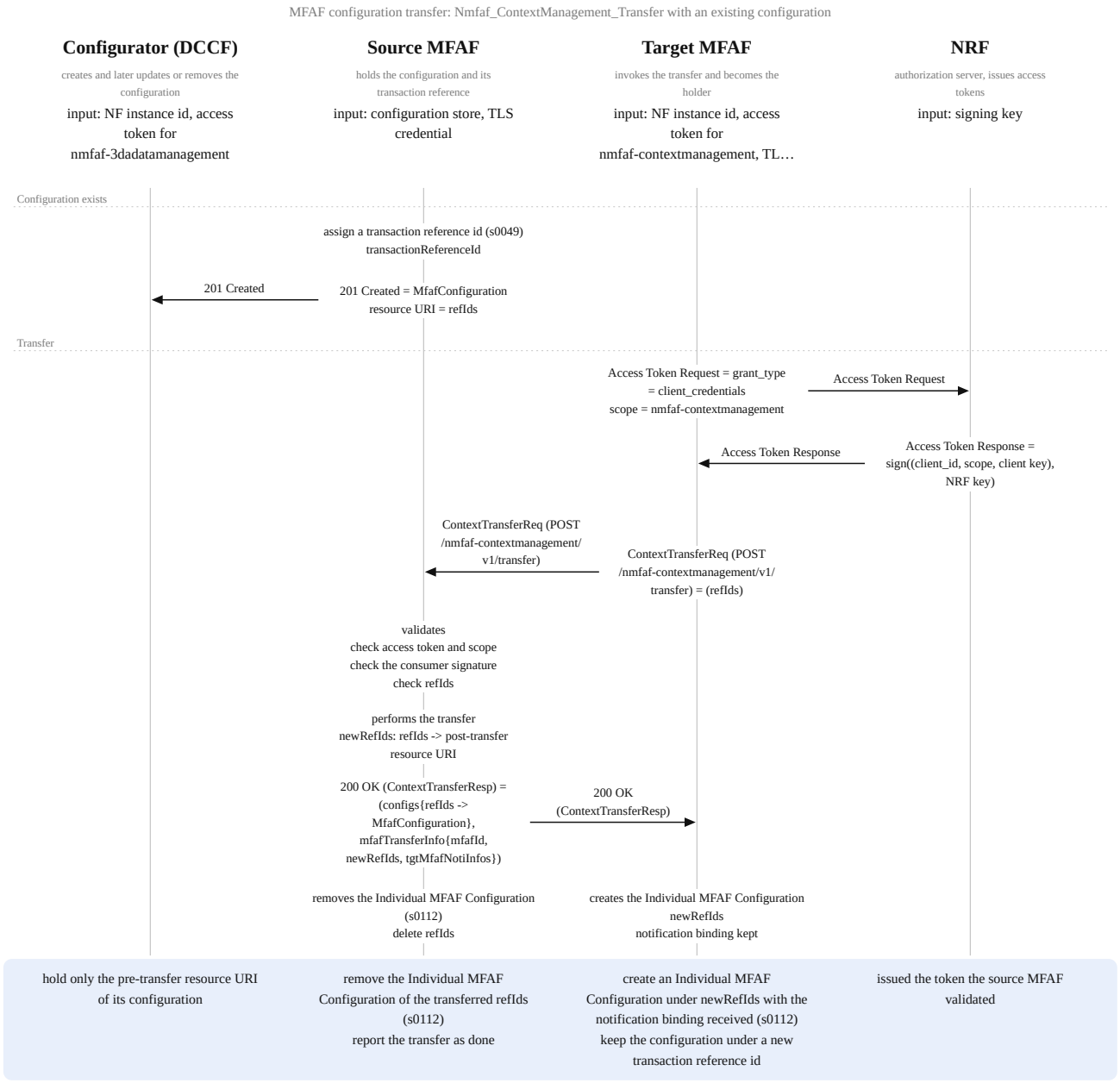


图 1. 协议的规定形式。

分析范围见附录 D：分析了什么、遗漏了什么，以及分析所依据的文稿章节。

2 发现

本次分析有 3 项发现：以下内容文稿中均未写明。其中 1 项是否在文稿中有写明不明确。

该流程兑现了它关于交换的承诺：目标 MFAF 只接受它所寻址的源 MFAF 确实发送的、针对相同 refIds 的响应，并且迁移后的配置保留其通知绑定。但它没有把配置保留在配置者放置的位置：任何持有服务范围且知道传输前资源 URI 的合法 NF 都能获得该配置，源在应答的同一步骤中删除自己的副本，而配置者持有的 URI 在迁移后不再能解析。三项发现，均未在文稿中写明；把授权留给"谁能命名资源"这一层面的规则，就是文稿确实写明的单一服务范围。这些答案依赖于理想化的 TLS 以及事务引用 id 不可猜测的假设，重放和失败问题只对状态模型展开的两个步骤给出了答案。

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：某一方发送了规定交换在该时点本没有的消息，以及该攻击所破坏的结果。

2.1 持有服务范围的合法 NF 取走另一消费者的配置

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	高
涉及的性质	Z01, S04 （见第 3 节）
原文	s0110, s0112, s0332, s0362, s0365
依据	ProVerif attack query `z01_attack`; Figures 2 and 3

发生了什么。 合法 NF 为另一消费者的配置调用 Nmfaf_ContextManagement_Transfer 时，会得到正常应答：源 MFAF 校验访问令牌和范围，找到请求所命名的配置，在 configs 中返回它并删除自己的副本，而从不检查请求者就是创建该配置的消费者，或是配置者指定的 MFAF。唯一的保护是传输前资源 URI 必须在 refIds 中被命名。

攻击者需要什么。 一个拥有自身凭据的合法 NF 实例，以及一个用于 nmfaf-contextmanagement 范围的有效 NRF 令牌，外加该配置的传输前资源 URI（即 refIds 值，模型将其交给攻击者）。它不需要密钥、不需要任何一方被攻陷，也不需要协议错误。

攻击者能得到什么。 它成为该配置的持有者：它读取配置内容和通知绑定（notificationURI、correId、mfafNotiInfo），而源 MFAF 删除自己的副本，于是配置在配置者从未指定的某一方继续存在。

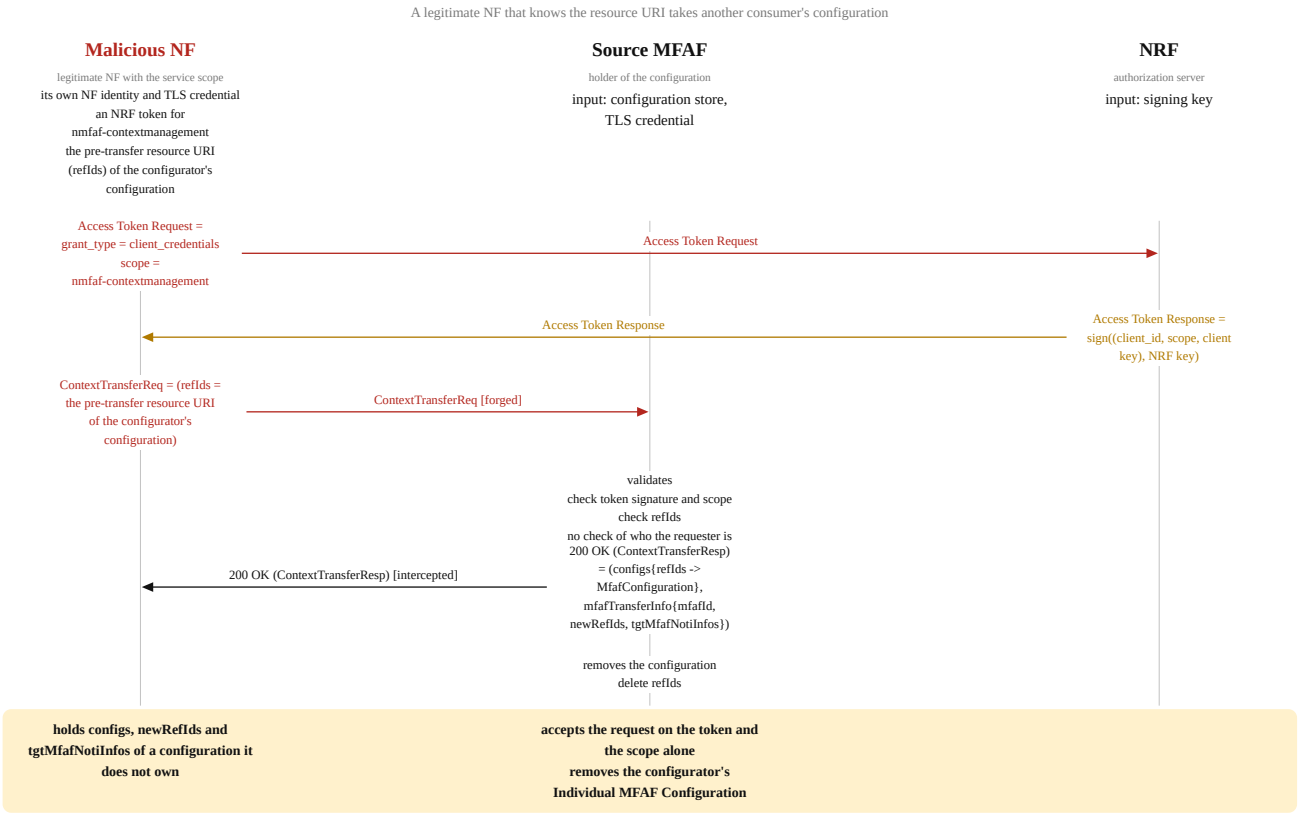


图 2. 知道资源 URI 的合法 NF 取走另一消费者的配置。性质 Z01；攻击查询 z01_attack。

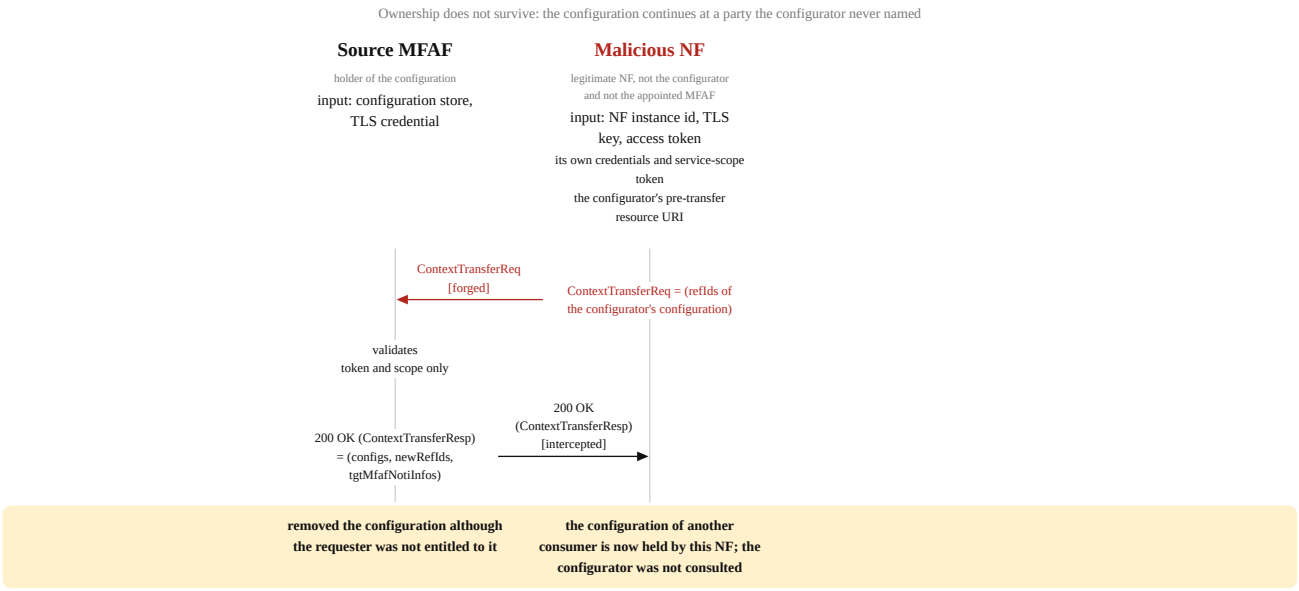


图 3. 所有权未能延续：配置在配置者从未指定的某一方继续存在。性质 S04；攻击查询 z01_attack。

拟议修改。 把该操作绑定到被指定的 MFAF，而不是服务范围：让配置记录配置者为其指定的目标 MFAF（字段 mfafTransferInfo 已经命名了持有者 MFAF 并触发传输，TS 29.572 s0037, s0051），并让源 MFAF 在令牌及其范围校验之外，再把已认证的消费者与该值比较，否则应答 403 Forbidden。在模型副本上检查：攻击查询变为已验证，正常流程保持不变，其他已记录的答案无一改变。

用 ProVerif 复核。 该修改被放入模型的一个副本：配置携带配置者为其指定的目标 MFAF（在创建配置时设置，正如 mfafTransferInfo 命名持有者和传输触发器，s0037, s0051）。收到 ContextTransferReq 时，源 MFAF 在令牌及其范围校验之外，把已认证的消费者与该指定的 MFAF 比较，否则应答 403 Forbidden。在修改后的模型上，ProVerif 证明了 `z01_attack`：该攻击在该模型中不再可能。协议的正常流程仍能被找到。其他 13 条已记录的答案保持不变。这是对模型的首次检查：它表明该修改在模型中消除了这一攻击，而不是表明它不会引入其他问题。修改后的模型：`records/fixes/0045-fix-owner.pv`。

2.2 传输响应被丢弃会在两个 MFAF 上销毁配置

类型	协议规范本身的弱点
是否新发现	是：文稿里没有写到
价值	高
涉及的性质	U01 （见第 3 节）
原文	s0112, s0113
依据	ProVerif attack query `u01`; Figure 4

发生了什么。 源 MFAF 以 200 OK 应答并携带这些配置，并在同一步骤中删除自己的副本（s0112），而原文单元对响应从未到达目标的情形没有任何规定：没有定时器、没有重试、没有补偿，s0113 只覆盖处理请求时引发的错误。因此，丢弃响应的攻击者会让配置在源处被删除、在目标处从未创建。

攻击者需要什么。 只需能在网络上丢弃一条消息：源 MFAF 发给目标 MFAF 的 200 OK。不需要凭据、不需要密钥、不需要任何一方被攻陷。

攻击者能得到什么。 配置在任何地方都不存在：源已将其删除，目标从未创建它，配置者的 URI 不再能解析，目标的重新尝试在源处找不到任何配置。在该配置下交付的数据会丢失，直到配置者重新配置一个 MFAF。

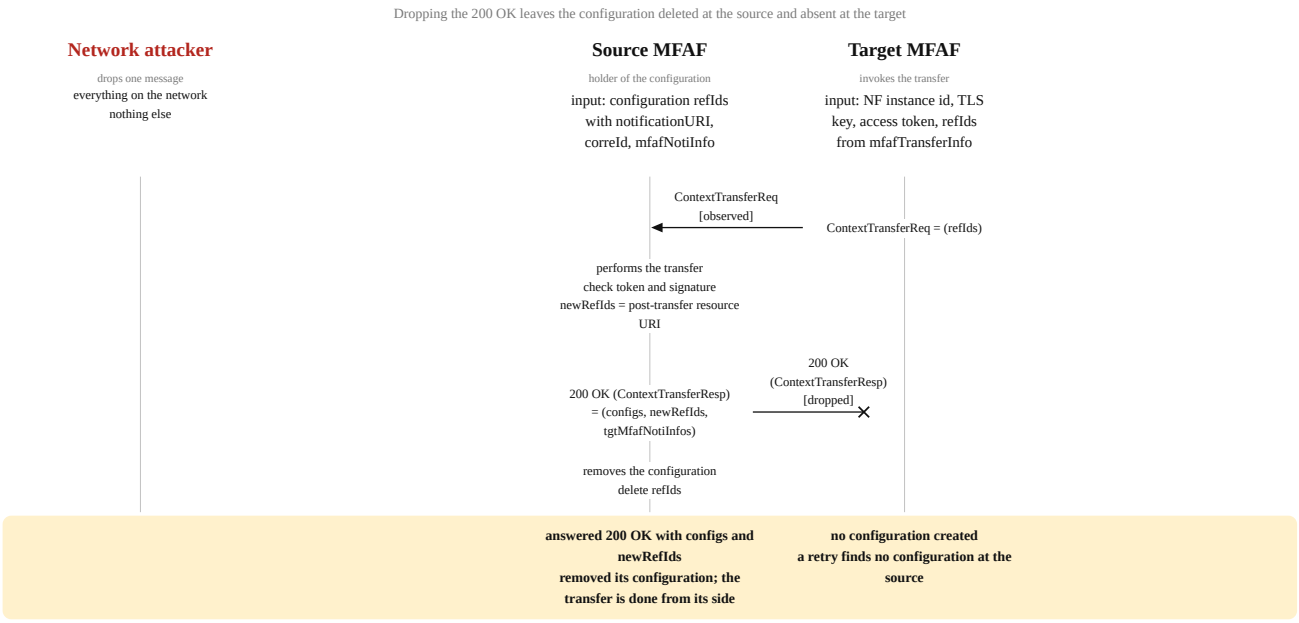


图 4. 丢弃 200 OK 会使配置在源处被删除、在目标处缺失。性质 U01；攻击查询 u01。

拟议修改。 保留源侧配置并标记为已传输，直到迁移的消费者确认创建，并对相同 refIds 的重复传输请求以相同的 newRefIds 应答而不是报错，从而使丢失的响应可以得到修复；只有在该确认之后才报告删除完成。

用 ProVerif 复核。 未检查：消除该弱点意味着改变交换过程，而不仅仅是做一次检查：要么源 MFAF 保留配置直到消费者确认创建，而这是该流程所没有的消息；要么目标的重新尝试必须从一个源为某传输标识保留的副本得到应答。无论哪种方式，修复后的文件都不再是对所建模流程的单处修改副本，因此未做检查。

2.3 配置者在迁移后丢失其配置的句柄

类型	协议规范本身的弱点
是否新发现	是：文稿里没有写到
价值	中
涉及的性质	X02, S04（见第 3 节）
原文	s0055, s0071, s0112, s0234
依据	ProVerif attack query `x02`; 图 5 和图 6

发生了什么。 迁移之后，源 MFAF 已删除该配置，目标 MFAF 将其保存在一个新的 transaction reference id 之下，但 newRefIds 只出现在调用该迁移的 consumer 所收到的响应里（s0234）。创建该配置的 configurator 仍保留旧的 resource URI，它的 Update 和 Deconfigure 请求都发往该 URI（s0055, s0071），而该 URI 在任何地方都已无法解析。

攻击者需要什么。 不需要任何超出正常成功迁移的东西：该配置只需被迁移一次即可。

攻击者能得到什么。 configurator 再也无法 Update 或 Deconfigure 该配置，因此它既不能改变 consumer 数据的 delivery，也不能删除该配置；只有持有该配置的 MFAF 及其新引用仍然有效。

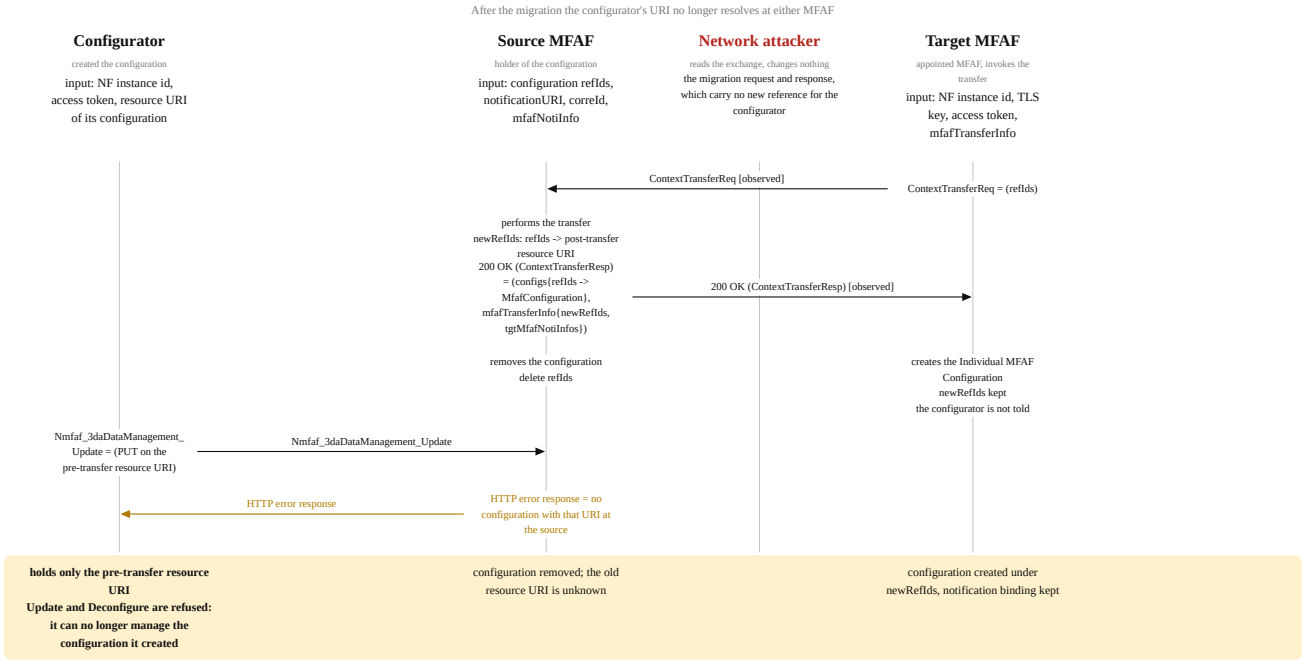


图 5. 迁移之后，configurator 的 URI 在两个 MFAF 上都无法解析。性质 X02；attack query x02。

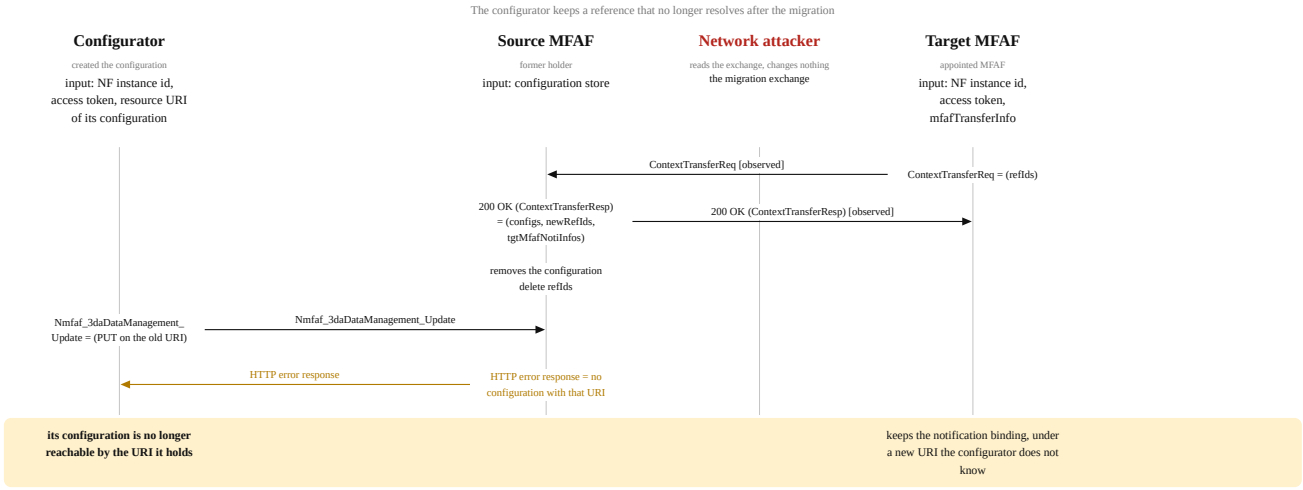


图 6. configurator 保留的引用在迁移之后已无法解析。性质 S04；attack query x02。

拟议修改。 将被迁移配置的 newRefIds 映射返回给创建它的 consumer，例如放入目标 MFAF 已经收到的 TargetMfafNotiInfo 中；或者让源 MFAF 在迁移期间保留一条从旧 resource URI 到新 URI 的转发记录。

用 ProVerif 复核。 未检查：该补救措施把新引用交给响应目前并不寻址的一方（configurator），这属于新增一条消息或在通知中新增一个字段，而不是对已有消息的检查；模型必须加入这次交换，因此未做检查。

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给出的答案；价值和文稿是否已写明该结果由分析者判断，理由见附录 A。

ID	性质	假定的泄露	结果	价值	文稿是否已写
A03	**接受的响应与真实转移匹配。** 每当目标 MFAF 接受一次迁移响应（200 OK 且含 configs），都存在源 MFAF 的一次真实迁移运行，二者在源 MFAF 身份、目标 MFAF 身份、refIds 集合与对应配置内容上一致。	无	成立	中	否
A08	**迁移保留引用和回调映射。** 目标 MFAF 接受的每个 newRefIds 条目都对应请求中同一 refIds 条目，且该配置的 notificationURI、correId、notifEndpoints 与 mfafNotiInfo 与源 MFAF 持有的原配置一致；tgtMfafNotiInfos 的键在请求中取自 refIds、在响应中取自 newRefIds，内层键为 correId。	无	成立	中	否
D01	**请求与响应具有真实性。** 目标 MFAF 接受的每个 ContextTransferResp 都对应源 MFAF 的真实发送，且 refIds、configs、newRefIds 与源 MFAF 实际持有的配置一致；源 MFAF 接受的每个 ContextTransferReq 都对应目标 MFAF 的真实发送。	无	成立	中	否
R03	**转移副作用至多发生一次。** 对同一逻辑迁移操作（同一 refIds 集合、同一源/目标 MFAF 对），源 MFAF 的本地删除副作用与目标 MFAF 的资源创建副作用各至多提交一次；请求可重传，但不得导致重复删除或重复创建。	无	在界内成立（变化的值展开到 2 步）	低	否
S04	**迁移不改变配置所有权。** 目标 MFAF 上迁移后的每个配置，其来源配置（refIds 指向的源端资源）、创建该配置的 NF service consumer 身份、以及 notificationURI/correId 绑定，与迁移前源 MFAF 上的对应配置一致；迁移不改变配置的归属主体。	无	不成立（2 个攻击）	高	否
Z01	**转移由有效令牌授权。** 每次被源 MFAF 执行的迁移（删除本地配置并返回 configs）都有目标 MFAF 持有的、由 NRF 签发的有效访问令牌，且该令牌的 scope 覆盖 nmfaf-contextmanagement；仅持有该 scope 的合法 NF 不能迁移其无权拥有的配置。	无	不成立（1 个攻击）	高	不明确
A03.7	**响应的身份与refIds绑定。** 目标 MFAF 接受的迁移响应与源 MFAF 的真实运行在源/目标 MFAF 身份与 refIds 上一致。	无	成立	中	否
A08.8	**引用与端点映射一致性。** 目标 MFAF 接受的每个 newRefIds 条目与请求中同一 refIds 条目对应，且 notificationURI、correId、notifEndpoints、mfafNotiInfo 与源端原配置一致。	无	成立	中	否
U01	**无半提交成功。** 迁移的成功判定与两端副作用（源端删除、目标端创建）在完整成功与部分成功/失败路径上保持一致，不出现单端已提交而另一端未提交却被视为完整成功。	无	不成立（1 个攻击）	高	否
X01	**配置内容在传输中保持机密。** 外部网络攻击者无法得知迁移响应中的 configs（notificationURI、correId、mfafNotiInfo）与 bufferedNotifs，也无法得知请求中的 refIds：任何由网络攻击者驱动的取值都不依赖这些内容。	无	成立	中	否
X02	**迁移后配置器保留其句柄。** 迁移成功后，创建该配置的 NF service consumer 仍能对它寻址以便 Update 或 Deconfigure：源文本或者让源/目标 MFAF 在迁移后仍以旧 URI 服务该配置者，或者把新的 resource URI 送给该配置者。	无	不成立（1 个攻击）	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, X02。

所提问题及分析者的回答：

问题	回答	依据
Q1: 迁移上下文与源/目标 MFAF 身份绑定	是，针对网络攻击者：被接受的响应绑定到目标所寻址的源 MFAF、绑定到目标自身，也绑定到源实际持有的 refIds 和内容；只有在攻击者已经持有迁移前 resource URI 时才会失效，而那只是一个名字，不是身份绑定。	A03, A03.7, D01

问题	回答	依据
Q2: 新旧引用及 notification endpoint 映射的一致性	refIds 到 newRefIds 的映射和通知绑定在每次迁移中都得以保留，但新 URI 只到达调用迁移的 MFAF，因此创建该配置的 configurator 保留了一个已无法解析的引用。	A08, A08.8, X02
Q3: 迁移失败与部分成功不能被当作完整成功	否：响应被丢弃的迁移不会在任何地方被报告为成功，但源已经删除了该配置，因此结果是配置丢失，而不是可恢复的失败；单个响应内部的部分成功超出了建模的界。	U01, X02

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，其中密码学被视为理想的，与实现无关。

按分析者的陈述，本分析不覆盖的内容：

- 两个 NF 端点之间的 TLS 被理想化为认证加密加上各端点长期密钥的签名，身份绑定由 NRF 签名的令牌承载；密钥协商、证书处理和记录层重放保护未建模。
- OAuth2 令牌是客户端身份、scope 和客户端密钥的签名声明；audience、过期、claim 匹配以及跨请求的令牌重用未建模，NRF 是单一可信的令牌签发者。
- 每次迁移一个配置：多条目情形（configs、refIds、bufferedNotifs 含 1..N 个条目）以及单个响应内部的部分成功未建模。
- 拒绝路径被建模为拒绝，而不是 HTTP 状态码和 ProblemDetails，因此失败预言问题（错误码是否会告诉请求方某个 resource URI 存在）在状态审查中记为候选，未予证明。
- 迁移操作的 307/308 重定向响应（含 Location 和由 SCP 提供的可选 3gpp-Sbi-Target-Nf-Id）未建模；按 TS 33.501 5.9.2.1，间接通信没有双向认证，因此被重定向的迁移请求超出了所分析的范围。
- 配置内容是不透明值；MessageConfiguration、notifEndpoints 和 MfafNotiInfo 仅通过 notificationURI/correId 绑定出现，通知投递流程（MFAF 到 consumer）不在所建模的子流程之内。
- 在 bounded.pv 中源 MFAF 展开到两步（配置存在，然后被删除），目标展开到一步，因此更长的运行、在同一 URI 下重新创建的配置以及多个 consumer 同时迁移都超出该模型。
- 分析覆盖单个 PLMN 和两个 MFAF 之间的直接通信；漫游、SEPP 和信任域边界未建模。

宿主记录的内容：

- A08、R03、S04、A08.8、U01、X02 的答案依赖于模型所做的一处简化（附录 D）。
- 没带对照查询的证明（附录 C）：A08: a08。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。

每条性质还有哪些悬而未决之处、什么能解决它，见附录 A。

附录 A：逐条性质的解读

这是分析者对证明器答案的解读，不是证明器的答案：这些答案对协议意味着什么、解读依据什么、能否照原样使用、以及它有多大价值。

A03: 接受的响应与真实转移匹配

每当目标 MFAF 接受一次迁移响应（200 OK 且含 configs），都存在源 MFAF 的一次真实迁移运行，二者在源 MFAF 身份、目标 MFAF 身份、refIds 集合与对应配置内容上一致。

目标 MFAF 只有在它所寻址的源 MFAF 确实发送了该配置、且 refIds 相同、内容相同时才接受该配置，这一点针对网络上的攻击者成立。该保证依托于直接 NF 到 NF 交换的双向 TLS，源文本将其交由 TS 33.501 和 TS 29.500 规定。

- 结果：成立；回答了所问的问题：是；可带保留地使用
- 价值中：这是该操作的核心保证，在未增加攻击者能力的情况下得到证明，但其强度来自模型所理想化的基线 TLS。
- 文稿是否已写：否。所依据的原文：s0110, s0112, s0332, s0335
- 依据：TLS 被建模为认证加密加上每个端点的签名，端点身份由 NRF 签名的令牌绑定。；目标带外（discovery 或证书）知道源 MFAF 的公钥，建模为参数。；每次迁移一个配置。

A08: 迁移保留引用和回调映射

目标 MFAF 接受的每个 newRefIds 条目都对应请求中同一 refIds 条目，且该配置的 notificationURI、correId、notifEndpoints 与 mfafNotiInfo 与源 MFAF 持有的原配置一致；tgtMfafNotiInfos 的键在请求中取自 refIds、在响应中取自 newRefIds，内层键为 correId。

目标 MFAF 创建的配置保留了源 MFAF 持有的通知绑定和内容，新 URI 与所请求的 refIds 条目相对应。协议未检查的是，请求方是否本就有权持有该配置。

- 结果：成立；是否回答了所问：是；可用但有保留
- 价值 中：对所声明的一致性要求（s0234、s0237）的证明，并指出了键控简化。
- 文稿中是否声明：否。所依据的原文：s0231、s0234、s0237、s0335
- 依据：源 MFAF 被视为其返回映射的权威；由它选择新的事务参考 id，而这一点原文并未确定（s0112 称由消费者分配，s0234 称由生产者包含）。；目标 MFAF 的通知信息表示为以新 URI 为键的单个条目，其内层键为 correId；映射结构与 notifEndpoints 未单独建模。

D01: 请求与响应具有真实性

目标 MFAF 接受的每个 ContextTransferResp 都对应源 MFAF 的真实发送，且 refIds、configs、newRefIds 与源 MFAF 实际持有的配置一致；源 MFAF 接受的每个 ContextTransferReq 都对应目标 MFAF 的真实发送。

源 MFAF 接受的请求确实来自其所携带身份的消费者，目标 MFAF 接受的响应确实来自其所寻址的源。此处真实性不等于授权：源 MFAF 只知道调用者是谁，不知道该调用者是否有权取走配置。

- 结果：成立；是否回答了所问：是；可用但有保留
- 价值 中：它准确显示了真实性止于何处、授权缺失于何处，而这正是主要发现的根源。
- 文稿中是否声明：否。所依据的原文：s0110、s0112、s0362

- 依据：TLS 端点按上述方式理想化；消费者身份由 NRF 签名的令牌携带，并与 TLS 客户端签名匹配。；拒绝被建模为参与方停止，而非 HTTP 状态码。

R03: 转移副作用至多发生一次

对同一逻辑迁移操作（同一 refIds 集合、同一源/目标 MFAF 对），源 MFAF 的本地删除副作用与目标 MFAF 的资源创建副作用各至多提交一次；请求可重传，但不得导致重复删除或重复创建。

在展开的运行中，无论请求被重放多少次，源 MFAF 至多删除一次配置；第二次请求因配置已不存在而被拒绝。若源保留了配置，就会删除两次，对照变体展示了这一点。

- 结果：在界内成立（变化值展开为 2 步）；是否回答了所问：是；可用但有保留
- 价值 低：此处的至多一次源于删除本身而非重放计数器，且仅在展开的步数上验证。
- 文稿中是否声明：否。所依据的原文：s0112
- 依据：bounded.pv 将源展开为两步、目标展开为一步；重放的响应侧未展开。；源 MFAF 未复制，因此两个请求方先后被服务。
- 未决：重放的 200 OK 是否会使目标 MFAF 重复创建同一资源，未得到回答：目标仅展开为一个响应，而重放的消息命名的是同一个新 URI。

S04: 迁移不改变配置所有权

目标 MFAF 上迁移后的每个配置，其来源配置（refIds 指向的源端资源）、创建该配置的 NF service consumer 身份、以及 notificationURI/correId 绑定，与迁移前源 MFAF 上的对应配置一致；迁移不改变配置的归属主体。

迁移后的配置与其 notificationURI 和 correId 的绑定得以保留，因此配置仍是配置者的配置。作为控制的所有权未能保留：知道迁移前 URI 的合法 NF 会收到该配置，且在任何迁移之后配置者都无法再寻址它。

- 结果：不成立（2 个攻击）；是否回答了所问：部分；可用
- 价值 高：它区分了协议确实保留的（通知绑定）与未保留的（对谁持有配置以及以哪个引用持有的控制）。
- 文稿中是否声明：否。所依据的原文：s0035、s0055、s0071、s0112、s0234
- 依据：两个失败在不同模型上得到回答：第三方接管在 model.pv 中将 refIds 交给攻击者，句柄丢失在 bounded.pv 中。；Update 与 Deconfigure 被建模为向旧 URI 发送一条 Update 消息，复用转移的令牌机制。
- 未决：不诚实的 MFAF 或不诚实的配置者能否将配置指向别处：能，凭其自身授权，而文稿对此并未限制。；共享回调端点的第二个消费者是否改变绑定：模型中每次转移一个配置。

Z01: 转移由有效令牌授权

每次被源 MFAF 执行的迁移（删除本地配置并返回 configs）都有目标 MFAF 持有的、由 NRF 签发的有效访问令牌，且该令牌的 scope 覆盖 nmfaf-contextmanagement；仅持有该 scope 的合法 NF 不能迁移其无权拥有的配置。

源执行的每次转移都由带有服务 scope 的有效令牌支撑，不知道迁移前 URI 的合法 NF 会被阻止。被阻止只意味着它无法命名该配置：资源名就是授权，知道资源名的 NF 无需任何所有权检查即可获得该配置。

- 结果：不成立（1 个攻击）；是否回答了所问：是；可用但有保留
- 价值 高：证明表明保护来自不可猜测的资源名而非授权，这正是编辑者需要的事实；原文声明了 scope 规则但未声明这一后果。

- 文稿中是否声明：不明确。所依据的原文：s0362、s0363、s0365
- 依据：refIds 被建模为不可猜测的名字；知道它们的攻击者是单独的 refid_known 攻击者，这是一个假设，而非原文所述。；令牌校验了签名、主体与 scope；未建模受众、过期与重用。
- 未决：部署是否在 API 之外强制执行所有权检查，原文既未要求也未禁止。

A03.7: 响应的身份与refIds绑定

目标 MFAF 接受的迁移响应与源 MFAF 的真实运行在源/目标 MFAF 身份与 refIds 上一致。

被接受的响应与真实的源端运行在两个 MFAF 身份和 refIds 上均一致。交互中没有任何东西能让网络攻击者使目标接受另一次转移或另一个源的响应。

- 结果：成立；是否回答了所问：是；可用但有保留
- 价值 中：与 A03 相同的证明，针对用户所问的身份与 refIds 绑定陈述。
- 文稿中是否声明：否。所依据的原文：s0110、s0112、s0332、s0335
- 依据：与 A03 相同的 TLS 理想化和身份绑定。；每次转移一个配置。

A08.8: 引用与端点映射一致性

目标 MFAF 接受的每个 newRefIds 条目与请求中同一 refIds 条目对应，且 notificationURI、correId、notifEndpoints、mfafNotiInfo 与源端原配置一致。

每个被接受的新 URI 都与所请求的 refIds 条目对应，并携带相同的通知数据。这一对应由源 MFAF 保证，其返回的映射受到信任。

- 结果：成立；是否回答了所问：是；可用但有保留
- 价值 中：对用户所问映射要求的证明，映射结构作了简化。
- 文稿中是否声明：否。所依据的原文：s0234、s0237、s0335
- 依据：目标 MFAF 通知信息以 newRefIds 为键、其内层映射以 correId 为键，这通过随条目传递 correId 来表示，而非通过映射结构。；notifEndpoints 与 MessageConfiguration 未单独建模。

U01: 无半提交成功

迁移的成功判定与两端副作用（源端删除、目标端创建）在完整成功与部分成功/失败路径上保持一致，不出现单端已提交而另一端未提交却被视为完整成功。

源 MFAF 在应答的同一步中删除其配置，因此响应丢失会使配置在源端被删除、在目标端从未创建。没有重试、定时器或补偿，源端的第二次尝试什么也找不到，因此该失败在流程内不可恢复。

- 结果：不成立（1 个攻击）；是否回答了所问：是；可用
- 价值 高：仅凭网络攻击者即可造成配置丢失，而这条路径原文根本没有描述。
- 文稿中是否声明：否。所依据的原文：s0112、s0113、s0349
- 依据：该失败通过攻击者丢弃响应来建模；源不会被告知，原文也未给出确认步骤。；每次转移一个配置，因此部分成功归结为此情形。
- 未决：单个响应内的部分成功，即多条目请求中部分配置被转移，超出建模的界。

X01: 配置内容在传输中保持机密

外部网络攻击者无法得知迁移响应中的 configs（notificationURI、correId、mfafNotiInfo）与 bufferedNotifs，也无法得知请求中的 refIds：任何由网络攻击者驱动的取值都不依赖这些内容。

仅中继、更改或丢弃消息的网络攻击者既无法得知配置内容及其通知绑定，也无法得知迁移前的资源 URI。这成立是因为每次交互都加密至所寻址的端点并由发送方签名。

- 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 中：它说明了基线对本操作所携带的取值所要求的传输保护（TS 33.501 5.9.2.1），并表明该保护仅与资源名一样可靠。
- 是否在文稿中说明：否。所依据的原文单元：s0335、s0231
- 依据：TLS 被建模为到端点密钥的认证加密；端点密钥通过带外方式获得。；内容取值被建模为新鲜名；猜测不在符号模型范围内。
- 待解决：保护的强度恰好取决于事务参考 id 的不可猜测性，而原文并未要求这一点；如果它可枚举，内容就会暴露。

X02: 迁移后配置器保留其句柄

迁移成功后，创建该配置的 NF service consumer 仍能对它寻址，以便执行 Update 或 Deconfigure：原文或者让源/目标 MFAF 在迁移后仍以旧 URI 为该配置器服务，或者把新的 resource URI 发送给该配置器。

迁移之后，配置器为其创建的配置发出的 Update 会被源 MFAF 拒绝，因为该配置已不在那里，也没有任何东西告诉配置器新的 URI。迁移之前，同一请求会被接受，这就是对照查询。

- 结果：不成立（1 个攻击）；是否回答了所问：是；可用
- 价值 中：它指出了仅向迁移的 consumer 返回 newRefIds 的确切后果，而原文并未讨论这一点。
- 是否在文稿中说明：否。所依据的原文单元：s0055、s0071、s0112、s0234
- 依据：bounded.pv 将原文展开为两步，因此只有在同一次运行中移除之后才会到达拒绝。；Update 被建模为一条消息，使用与传输相同的令牌机制；第 4.2 节的数据管理 API 在其他方面不在模型范围内。

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，以及各攻击及分析者对每个攻击的判定。

目标	级别	界	查询	对照	攻击	主张
A03	proved		a03	checked		2 proved
A08	proved		a08	3 of 5		5 proved
D01	proved		d01	checked		3 proved
R03	bounded	变化的值展开到 2 步	r03	checked		2 bounded
S04	proved		a08	checked	z01_attack (holds_in_source)；x02 (holds_in_source)	2 proved, 2 attack
Z01	proved		z01_tok	checked	z01_attack (holds_in_source)	4 proved, 1 attack
A03.7	proved		a03	checked		2 proved
A08.8	proved		a08	checked		3 proved
U01	none				u01 (holds_in_source)	1 attack
X01	proved		x01_bind	checked		2 proved
X02	none				x02 (holds_in_source)	1 attack

逐条主张，按分析者对每条主张的措辞；每条主张的答案来自证明器：

目标	主张	攻击者	回答	查询	对照
A03	目标 MFAF 接受的每个携带配置的 200 OK，都对应源 MFAF 上次真实的传输运行，源 MFAF 身份、目标 MFAF 身份、refIds 和配置内容均相同。	`network`	proved	a03	checked
A03	目标 MFAF 接受的每个携带配置的 200 OK，都对应源 MFAF 上次真实的传输运行，身份、refIds 和内容均相同，即使网络中有一个持有自身凭据和服务范围令牌的合法 NF 也是如此。	`malicious_nf`	proved	a03	checked
A08	目标 MFAF 接受的每个 newRefIds 条目，都对应其请求的同一 refIds 条目，且迁移后的配置保留源 MFAF 所持配置的 notificationURI、correId、notifEndpoints 和 mfafNotiInfo，即使网络中有一个持有自身凭据的合法 NF 也是如此。	`malicious_nf`	proved	a08	none
A08	目标 MFAF 执行的每次迁移都保留映射和通知绑定，对于持有另一消费者配置的传输前资源 URI 的攻击者也是如此。	`refid_known`	proved	a08	none
A08	目标 MFAF 接受的每个 newRefIds 条目，都对应其请求的同一 refIds 条目，且迁移后的配置保留源 MFAF 所持配置的 notificationURI、correId、notifEndpoints 和 mfafNotiInfo。	`network`	proved	a08	checked
A08	目标 MFAF 接受的每个 newRefIds 条目，都对应其请求的同一 refIds 条目，且迁移后的配置保留源 MFAF 所持配置的通知绑定，即使网络中有一个持有自身凭据的合法 NF 也是如此。	`malicious_nf`	proved	a08	checked

目标	主张	攻击者	回答	查询	对照
A08	每次迁移都保留映射和通知绑定，对于持有另一消费者配置的传输前资源 URI 的攻击者也是如此。	`refid_known`	proved	a08	checked
D01	目标 MFAF 接受的每个 ContextTransferResp，都对应源 MFAF 向该目标真实发送的一次。	`network`	proved	a03	checked
D01	源 MFAF 接受的每个 ContextTransferReq，都对应该消费者 MFAF 向该源真实发送的一次。	`network`	proved	d01	checked
D01	源 MFAF 接受的每个 ContextTransferReq，都对应该消费者 MFAF 的真实发送，即使一个持有自身凭据的合法 NF 发送请求也是如此。	`malicious_nf`	proved	d01	checked
R03	对于一次逻辑迁移，源 MFAF 的本地删除至多发生一次，无论请求被重放多少次，且目标 MFAF 至多为它创建一个资源。	`network`	bounded（变化的值展开到 2 步）	r03	checked
R03	对于一次逻辑迁移，源 MFAF 的本地删除至多发生一次，即使一个持有自身凭据的合法 NF 重放传输请求也是如此。	`malicious_nf`	bounded（变化的值展开到 2 步）	r03	checked
S04	配置不会落到持有自身凭据和服务范围、但不知道传输前资源 URI 的第三方，且保留的绑定属于配置者。	`malicious_nf`	proved	z01_noleak	checked
S04	配置只交给配置者指定的 MFAF：知道传输前资源 URI 的第三方无法获得它，对于在网络之外持有传输前资源 URI 的攻击者也是如此。	`refid_known`	attack (holds_in_source)	z01_attack (z01_noleak 的对照)	
S04	迁移之后，配置者仍能寻址其配置：其针对该配置的 Update 仍被持有该配置的 MFAF 接受。	`network`	attack (holds_in_source)	x02	
S04	迁移后的配置保留源 MFAF 所持的 notificationURI/correId 绑定和内容，因此迁移之后它仍是配置者的配置，而不是一个新配置。	`network`	proved	a08	checked
Z01	只持有服务范围的合法 NF 在不知道传输前资源 URI 时无法获得不属于它的配置：阻止它的是不可猜测的资源名，而不是所有权检查。	`network`	proved	z01_noleak	checked
Z01	只持有服务范围的合法 NF 如果不知道传输前资源 URI，就无法获得不属于它的配置；它自己的令牌不会给它任何基于所有权的限制。	`malicious_nf`	proved	z01_noleak	checked
Z01	只持有服务范围但知道另一消费者配置的传输前资源 URI 的合法 NF 能获得该配置，对于在网络之外持有传输前资源 URI (refIds) 的攻击者也是如此；源 MFAF 不检查请求者是否拥有或是否有权访问该配置。	`refid_known`	attack (holds_in_source)	z01_attack (z01_noleak 的对照)	
Z01	源 MFAF 执行的每次传输，都由该消费者持有的、NRF 签发的具有 nmfaf-contextmanagement 范围的有效访问令牌作支撑。	`network`	proved	z01_tok	checked
Z01	源 MFAF 执行的每次传输，都由该消费者持有的、NRF 签发的具有 nmfaf-contextmanagement 范围的有效访问令牌作支撑，对于持有自身令牌的合法 NF 也是如此。	`malicious_nf`	proved	z01_tok	checked
A03.7	被接受的迁移响应与真实的源端运行在源 MFAF 身份、目标 MFAF 身份和 refIds 上一致。	`network`	proved	a03	checked
A03.7	被接受的迁移响应与真实的源端运行在两个 MFAF 身份和 refIds 上一致，即使网络中有一个持有自身凭据的合法 NF 也是如此。	`malicious_nf`	proved	a03	checked

目标	主张	攻击者	回答	查询	对照
A08.8	每个被接受的 newRefIds 条目都对应请求中同一 refIds 条目，且 notificationURI、correId、notifEndpoints 和 mfafNotiInfo 与源端原始配置一致。	`network`	proved	a08	checked
A08.8	每个被接受的 newRefIds 条目都对应请求中同一 refIds 条目，且通知数据与源端原始配置一致，即使网络中有一个持有自身凭据的合法 NF 也是如此。	`malicious_nf`	proved	a08	checked
A08.8	每个被接受的 newRefIds 条目都对应请求中同一 refIds 条目，且通知数据与源端原始配置一致，即使攻击者知道传输前资源 URI 也是如此。	`refid_known`	proved	a08	checked
U01	源 MFAF 只有在目标 MFAF 已创建配置之后才删除该配置：响应丢失不得导致配置在源端被删除、在目标端不存在。	`network`	attack (holds_in_source)	u01	
X01	配置内容及其通知绑定（configs、notificationURI、correId、mfafNotiInfo）对迁移响应上的网络攻击者保持未知。	`network`	proved	x01_bind	checked
X01	传输前资源 URI (refIds) 对迁移请求上的网络攻击者保持未知。	`network`	proved	x01_ref	checked
X02	迁移之后，配置者仍能用其持有的资源 URI 寻址其配置：其 Update 被接受，或者新 URI 被告知给它。	`network`	attack (holds_in_source)	x02	

附录 D：分析是怎么跑的

LLM 分析者根据文稿构建协议模型，并把每个性质交给 ProVerif；第 3 节中的每个结果都是证明器的回答。

ProVerif+ Agent 会话 session-20261115T083807，2026-11-15。状态：已完成。模型：deepseek-v4-1-flash-261017。步骤：63，证明器 5 s，实际耗时 19 min，成本 0.139 USD。

ProVerif：运行了 89 个证明，89 个已判定，0 个无答案，3 s，记录了 39 个回答。

分析了什么

协议	MFAF configuration context transfer
版本	19.7.0
分析范围	固定一个已存在 MFAF configuration 从旧 MFAF 到目标 MFAF 的有界迁移请求和响应；一个合法配置者，保留 transaction reference、源/目标 NF 身份、回调端点映射；仅在规范适用条件成立时建模，不包含 UE mobility 全流程。
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	406 个带编号的原文单元（160566 字节），来自下面列出的章节。本报告中诸如 s0001 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	4.2	Nmfaf_3daDataManagement Service
main	4.2.1	服务描述
main	4.2.1.1	概述
main	4.2.1.2	服务架构
main	4.2.1.3	网络功能
main	4.2.1.3.1	消息框架适配功能（MFAF）
main	4.2.1.3.2	NF 服务消费者
main	4.2.2	服务操作
main	4.2.2.1	引言
main	4.2.2.2	Nmfaf_3daDataManagement_Configure 服务操作
main	4.2.2.2.1	概述
main	4.2.2.2.2	MFAF 的初始配置
main	4.2.2.2.3	更新 MFAF 的配置
main	4.2.2.3	Nmfaf_3daDataManagement_Deconfigure 服务操作
main	4.2.2.3.1	概述

文稿	章节	标题
main	4.2.2.3.2	移除 MFAF 配置
main	4.4	Nmfaf_ContextManagement Service
main	4.4.1	服务描述
main	4.4.1.1	概述
main	4.4.1.2	服务架构
main	4.4.1.3	网络功能
main	4.4.1.3.1	消息框架适配功能（MFAF）
main	4.4.1.3.2	NF 服务消费者
main	4.4.2	服务操作
main	4.4.2.1	引言
main	4.4.2.2	Nmfaf_ContextManagement_Transfer 服务操作
main	4.4.2.2.1	概述
main	4.4.2.2.2	传输 MFAF 配置
main	5.1	Nmfaf_3daDataManagement Service 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
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
main	5.1.3	资源
main	5.1.3.1	概述
main	5.1.3.2	资源：MFAF 配置
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 MFAF 配置
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	5.1.4	无关联资源的自定义操作

文稿	章节	标题
main	5.1.5	通知
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：MfafConfiguration
main	5.1.6.2.3	类型：MessageConfiguration
main	5.1.6.2.4	类型：MfafNotiInfo
main	5.1.6.2.5	类型：MfafTransferInfo
main	5.1.6.2.6	类型：TargetMfafNotiInfo
main	5.1.6.3	简单数据类型与枚举
main	5.1.6.3.1	引言
main	5.1.6.3.2	简单数据类型
main	5.1.6.4	描述替代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
main	5.3	Nmfaf_ContextManagement Service API
main	5.3.1	引言
main	5.3.2	HTTP 的使用
main	5.3.2.1	概述
main	5.3.2.2	HTTP 标准头
main	5.3.2.2.1	概述
main	5.3.2.2.2	内容类型
main	5.3.2.3	HTTP 自定义头
main	5.3.3	资源
main	5.3.3.1	概述
main	5.3.4	无关联资源的自定义操作
main	5.3.4.2	操作：transfer
main	5.3.4.2.1	描述
main	5.3.4.2.2	操作定义
main	5.3.5	通知

文稿	章节	标题
main	5.3.6	数据模型
main	5.3.6.1	概述
main	5.3.6.2	结构化数据类型
main	5.3.6.2.1	引言
main	5.3.6.2.2	类型：ContextTransferReq
main	5.3.6.2.3	类型：ContextTransferResp
main	5.3.6.3	简单数据类型与枚举
main	5.3.6.3.1	引言
main	5.3.6.3.2	简单数据类型
main	5.3.6.4	描述替代数据类型或数据类型组合的数据类型
main	5.3.6.5	二进制数据
main	5.3.7	错误处理
main	5.3.7.1	概述
main	5.3.7.2	协议错误
main	5.3.7.3	应用错误
main	5.3.8	特性协商
main	5.3.9	安全

根据源步骤在文稿中定位到的结果，该协议的用途是：固定一个已存在 MFAF configuration 从旧 MFAF 到目标 MFAF 的有界迁移请求和响应；一个合法配置者，保留 transaction reference、源/目标 NF 身份、回调端点映射；仅在规范适用条件成立时建模，不包含 UE mobility 全流程。

对照文稿定性质

在撰写查询之前，分析者将草拟的性质清单与文稿本身对协议用途的说明进行了对照。名称、性质所涉及的对象以及文稿的表述均出自分析者的解读；所引用的位置确实存在于文稿中。

ID	简称	对象	文稿怎么说	出处
A03	接受的响应与真实转移匹配	目标 MFAF（NF service consumer，一个 MFAF，s0088/s0101）在接受 200 OK 时所获得的保证：响应来自它寻址的源 MFAF、对应源端一次真实迁移运行，并且在源 MFAF 身份、目标 MFAF 身份、refIds 与配置内容上一致。	s0112 规定源 MFAF 执行迁移并回 200 OK 与 ContextTransferResp，目标 MFAF 收到响应后建立配置；s0335 定义 configs 的键必须取自请求的 refIds。认证如何取得见基线：33.501 5.9.2.1 要求直接 NF-to-NF 通信双向认证，29.500 13.3.1 要求校验访问令牌与 scope。源文本本身未规定 MFAF 间认证（s0362 只说 OAuth2 可用于授权）。	ts-33-501:2871, s0083, s0095, s0101, s0104, s0110, s0112, s0332, s0335

ID	简称	对象	文稿怎么说	出处
A08	迁移保留引用和回调映射	目标 MFAF 建立的新 Individual MFAF Configuration 与其源端原配置的一致性：newRefIds 每项对应请求中同一 refIds 项，notificationURI、correId、notifEndpoints、mfafNotiInfo（mfafNotifUri/mfafCorreId）不变，tgtMfafNotiInfos 在响应中按 newRefIds 为键、内层按 correId 为键。	s0234 规定 newRefIds 由生产者放入响应；s0237 规定 tgtMfafNotiInfos 的键在请求中取 refIds、响应中取 newRefIds，内层键为 correId；s0231 定义 mfafNotifUri/mfafCorreId；s0112 要求目标 MFAF 为每个成功迁移的配置建立资源。	s0110, s0112, s0231, s0234, s0237, s0332, s0335
D01	请求与响应具有真实性	双方的消息真实性：源 MFAF 接受的每个 ContextTransferReq 都对应目标 MFAF 的真实发送，目标 MFAF 接受的每个 ContextTransferResp 都对应源 MFAF 的真实发送；外部网络攻击者不能伪造或篡改。	s0110/s0112 规定请求与响应；33.501 5.9.2.1 要求直接 NF-to-NF 服务请求与响应双向认证，并要求每个 NF 校验所有收到的消息；TS 29.572 s0362 只说访问可用 OAuth2 授权，消息完整性依赖 29.500 的 SBI 安全。	ts-33-501:2871, ts-33-501:2875, s0110, s0112, s0302, s0309, s0311, s0362
R03	转移副作用至多发生一次	同一逻辑迁移操作（同一 refIds 集合、同一源/目标 MFAF 对）在源端的本地删除（s0112）与在目标端的新资源创建（s0112）各至多提交一次；请求可重传，重传不得导致重复删除或重复创建。	s0112 描述成功路径上的两个副作用（源端本地删除、目标端建立资源并分配新 transaction reference id）；源文本没有规定迁移请求的重试、幂等键或重复响应的处理。	s0112, s0173
S04	迁移不改变配置所有权	迁移不改变配置的归属：目标端迁移后的配置仍绑定迁移前创建它的 NF service consumer（配置者）与同一 notificationURI/correId，且配置落在配置者指定的目标 MFAF，而不是任何能命名该资源的第三方。	s0035/s0048-s0053 规定配置由 NF service consumer 创建、MFAF 分配 transaction reference id 并回 201 Created；s0112 规定迁移后由发起迁移的 consumer 建立资源；s0234 规定 newRefIds 只回给发起迁移的 consumer；s0105 说明配置者（DCCF）通过配置触发迁移以完成 UE 的 MFAF 重定位。	s0035, s0049, s0053, s0105, s0112, s0234, s0335
Z01	转移由有效令牌授权	源 MFAF 的资源保护：每次被执行的迁移都有 NRF 签发的、覆盖 nmfaf-contextmanagement scope 的有效访问令牌；且仅持有该服务级 scope 的合法 NF 不能迁移它无权拥有的（其他消费者创建的）配置。	s0362/s0363：访问可用 OAuth2 授权，consumer 须先向 NRF 取令牌；s0365：本 API 只定义单一服务级 scope，不定义资源级或操作级 scope。29.500 13.3.1 要求服务端校验令牌、有效期与 scope，并说明只有服务级 scope 时授权只能覆盖“不需要专门授权的操作”。33.501 5.9.2.1 还要求 NRF 能判定注册/发现/授权请求。	ts-29-500:4994, ts-29-500:4996, s0264, s0265, s0267, s0362, s0363, s0365
A03.7	响应的身份与refIds绑定	目标 MFAF 接受的迁移响应与其真实运行在源/目标 MFAF 身份与 refIds 上一致（Q1 的表述）。	同 A03：s0110/s0112/s0332/s0335；源文本未规定 MFAF 间认证机制（s0362）。	s0110, s0112, s0332, s0335
A08.8	引用与端点映射一致性	目标 MFAF 接受的 newRefIds 条目与请求中同一 refIds 条目对应，且 notificationURI、correId、notifEndpoints、mfafNotiInfo 与源端原配置一致（Q2 的表述）。	同 A08：s0112、s0234、s0237、s0335。	s0110, s0112, s0234, s0237, s0335

ID	简称	对象	文稿怎么说	出处
U01	无半提交成功	完整成功与部分成功/失败路径上两端副作用一致：不出现源端已删除、目标端未建立，却被当作成功推进的状态（Q3 的表述）。	s0112 只描述成功路径（源端删除 + 目标端建立）；s0113 只说错误时返回 HTTP 错误响应并引用 5.3.7（s0349 引用 29.500 的通用错误码），未规定部分成功或响应丢失时的补偿。	s0112, s0113, s0335, s0349
X01	配置内容在传输中保持机密	源 MFAF 与目标 MFAF 之间传输的配置内容与缓冲通知；被保护于仅具网络能力的攻击者。	33.501 5.9.2.1 要求 SBA 的注册/发现支持机密性、完整性与重放保护，且直接 NF-to-NF 通信须双向认证（P215-P220）；33.501 13.1 规定 PLMN 内 SBI 使用 TLS；TS 29.572 s0335 的 bufferedNotifs 与 s0231/s0234 的 notification 信息属于该响应体。	ts-33-501:2871, ts-33-501:2865, s0335, s0231
X02	迁移后配置器保留其句柄	创建配置的 NF service consumer 对自己配置的管理可达性，跨迁移前后。	s0055 的 Update 与 s0071 的 Deconfigure 都以 Individual MFAF Configuration 的资源 URI 寻址；s0112 规定迁移后源端删除该资源、目标端以新 transaction reference id 建立资源；s0234 规定 newRefIds 只出现在给发起迁移的 consumer 的响应中。	s0055, s0071, s0112, s0234, s0105

分析者依据文稿补充的性质：X01, X02.

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 30 段，没选 368 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	OAuth 框架的安全考虑概述与威胁模型引用，未对本 API 陈述必须成立或可被攻击的性质。
P003 至 P006	rfc6749:2790-2823	Z01	
P007 至 P010	rfc6749:2827-2850	Z01, S04, `malicious_nf`	
P011 至 P015	rfc6749:2859-2881	Z01, A03, `malicious_nf`	
P016 至 P033	rfc6749:2885-3003	不在范围内	refresh token、authorization code、redirection URI 与 resource owner password 授权；本 API 只用 client credentials 授权（s0362），无这些流程。
P034 至 P035	rfc6749:3013-3019	X01	
P036	rfc6749:3023-3028	A03, D01, X01	

段落	出处	定为	不选的原因
P037 至 P039	rfc6749:3032-3041	证明器问不了	令牌可猜测性的概率界（2^-128 等）是计算性论断，符号模型只把原语视为完美，不能作答。
P040 至 P062	rfc6749:3045-3210	不在范围内	phishing、CSRF、clickjacking、code injection、open redirector、implicit grant 的令牌误用；本 API 为直接 NF-to-NF 的 client credentials 流程（s0362，33.501:2871），不使用 user-agent 流程。
P063	ts-33-501:2491-2491	没有陈述目标	5G 安全特性的 bidding down（UE 与网络之间）与本 API 无关：本 API 的特征协商只涉及可选 API 特征（s0358），不协商安全算法。
P064 至 P072	ts-33-501:2495-2511	不在范围内	UE 与网络的签约认证、服务网络鉴权与紧急业务；不在本 API 子流程内。
P073 至 P213	ts-33-501:2515-2853	不在范围内	5G 密钥层次、RAN/NAS/AS 的机密性与完整性、签约者隐私、UDM/AUSF/UDR 要求；均与本 SBI 子流程无关。
P214	ts-33-501:2859-2859	X01, A03	
P215 至 P220	ts-33-501:2865-2875	A03, D01, Z01, X01, `malicious_nf`	
P221 至 P227	ts-33-501:2879-2891	Z01	
P228 至 P234	ts-33-501:2895-2907	不在范围内	NEF 安全要求；NEF 不是本流程的参与方。
P235 至 P241	ts-33-501:2911-2923	不在范围内	SCP（间接通信）安全要求；本流程为直接 NF-to-NF。TS 29.572 的迁移响应可携带 SCP 的 307/308 重定向（s0311-s0315），属本任务范围外的部署选项，列入 limits。
P242 至 P303	ts-33-501:2929-3055	不在范围内	SEPP 与漫游、漫游中介；两个 MFAF 在同一 PLMN 内（s0086）。
P304 至 P312	ts-33-501:3059-3075	不在范围内	SEPP 的属性保护（漫游）。
P313 至 P314	ts-33-501:3079-3081	不在范围内	IPUPS 功能，未建模。
P315 至 P318	ts-33-501:3085-3091	不在范围内	NF 在 3gpp-Sbi-Originating-Network-Id 头中填写 PLMN/SNPN ID 的要求；本模型不带 PLMN 边界，故不产生本流程的目标。
P319 至 P337	ts-33-501:3097-3141	不在范围内	信令监控数据的配置与流式传输、安全可见性/可配置性；属运营流程，不是本协议消息的性质。
P338 至 P369	ts-33-501:3149-3215	证明器问不了	加密与完整性算法标识及算法选择的安全性；符号模型把原语视为完美，不作答。
P370 至 P398	ts-33-501:3219-11901	不在范围内	5G-RG、NSSAAF、EAP-TLS、IAB、MBSF/MBSTF、NEF 等其他功能的要求。

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 28 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 7 项，没设的 21 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	源 MFAF	参与方	按定义全部失守	控制源 MFAF 就是控制持有配置、发送响应的那一方；本流程的性质都陈述为它的行为，因而按定义丧失，文档也没有在源 MFAF 被攻破的前提下提出这些性质。
S02	目标 MFAF	参与方	按定义全部失守	目标 MFAF 是被保证的一方；控制它则本流程所有性质按定义丧失。
S03	外部网络攻击者	参与方	攻击者得不到更多	该参与方就是 network 这一攻击者本身（只读、注入、丢弃、重放），没有额外能力可命名为另一个攻击者。
S04	持有自身凭据的恶意合法 NF	参与方	`malicious_nf`	
S05	配置者	参与方	按定义全部失守	配置者按其自身授权触发向它指定目标的迁移（s0037、s0051），因此“它可以把其配置迁到哪里”按定义不受约束；其他参与方（目标 MFAF 对源 MFAF 的保证）已在 network 下作答，不受影响。
S06	NRF	参与方	按定义全部失守	NRF 是令牌的信任锚（s0362、s0363）；控制它就能给任何 NF 签发任意 scope 的令牌，Z01 按定义丧失，文档未在 NRF 被攻破的前提下提出任何要求。
S07	refIds	值	`refid_known`	
S08	configs	值	按定义全部失守	持有配置内容就是知道被迁移的配置；机密性目标按定义丧失，其余目标不问其机密性。
S09	源/目标 MFAF 的 NF 实例身份	值	攻击者得不到更多	NF 实例标识是公开的、可经 NRF 发现（s0086、s0087），不构成额外能力。
S10	refIds / newRefIds 映射	值	`refid_known`	
S11	notificationURI	值	攻击者得不到更多	notificationURI、correId、notifEndpoints、mfafNotiInfo 用于标识消费者一方，不是本子流程中的认证依据；用它们伪造通知属 4.2 的通知流程，本子流程的目标不涉及。
S12	correId	值	攻击者得不到更多	notificationURI、correId、notifEndpoints、mfafNotiInfo 用于标识消费者一方，不是本子流程中的认证依据；用它们伪造通知属 4.2 的通知流程，本子流程的目标不涉及。
S13	notifEndpoints	值	攻击者得不到更多	notificationURI、correId、notifEndpoints、mfafNotiInfo 用于标识消费者一方，不是本子流程中的认证依据；用它们伪造通知属 4.2 的通知流程，本子流程的目标不涉及。
S14	mfafNotiInfo	值	攻击者得不到更多	notificationURI、correId、notifEndpoints、mfafNotiInfo 用于标识消费者一方，不是本子流程中的认证依据；用它们伪造通知属 4.2 的通知流程，本子流程的目标不涉及。
S15	tgtMfafNotiInfos	值	攻击者得不到更多	tgtMfafNotiInfos 的键值只是通知信息的重新关联，本子流程没有以它为认证依据的目标。
S16	ContextTransferReq 中的 refIds	值	`refid_known`	
S17	ContextTransferResp 中的 configs 与 newRefIds	值	`refid_known`	

ID	参与方或值	类别	攻击者条件	不设的原因
S18	源 MFAF 上的 Individual MFAF Configuration 资源	值	按定义全部失守	持有源端/目标端的 Individual MFAF Configuration 资源就是控制持有该资源的 MFAF 的存储（S01/S02），相关性质按定义丧失。
S19	目标 MFAF 上新建的 Individual MFAF Configuration 资源及新 transaction reference id	值	按定义全部失守	持有源端/目标端的 Individual MFAF Configuration 资源就是控制持有该资源的 MFAF 的存储（S01/S02），相关性质按定义丧失。
S20	Individual MFAF Configuration 资源及其 transaction reference id	值	按定义全部失守	持有源端/目标端的 Individual MFAF Configuration 资源就是控制持有该资源的 MFAF 的存储（S01/S02），相关性质按定义丧失。
S21	配置者身份	值	攻击者得不到更多	配置者身份是公开的 NF 实例标识（s0053 的 201 Created 体），不构成额外能力。
S22	notificationURI 与 correId	值	攻击者得不到更多	同 S11-S14：notificationURI 与 correId 不作为本子流程的认证依据。
S23	OAuth2 访问令牌及其 scope	值	`malicious_nf`	
S24	源/目标 MFAF 身份	值	攻击者得不到更多	源/目标 MFAF 的 NF 实例身份可经 NRF 发现，是公开信息。
S25	refIds/newRefIds	值	`refid_known`	
S26	源端 Individual MFAF Configuration 资源	值	按定义全部失守	同 S18：持有源端资源即控制源 MFAF 的存储。
S27	目标端新建资源	值	按定义全部失守	同 S19：持有目标端资源即控制目标 MFAF 的存储。
S28	ContextTransferResp 中的 configs	值	按定义全部失守	同 S08：持有响应中的 configs 即知道被迁移的配置，机密性目标按定义丧失。

每条性质在各攻击者条件下的答案

攻击者由分析者设定，在写下任何查询之前就已确定；每个答案都记录在其中一个攻击者之下。单元格是证明器对该性质在该攻击者下的答案；“无答案”表示该攻击者下证明器让某条主张悬而未决，或没有记录任何内容；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：它读取、丢弃、改动并发送消息，不持有任何一方的秘密		
`malicious_nf`	一个合法 NF 实例自己的身份、mTLS 证书与私钥，以及 NRF 签发给它的、覆盖 nmfaf-contextmanagement 服务级 scope 的有效访问令牌；能作为 NF service consumer 调用本 API 的任何操作。	s0365 规定本 API 只有单一服务级 scope、不定义资源级或操作级 scope，因此持有该 scope 的合法 NF 对该 API 资源拥有同一授权；29.500 13.3.1 与 33.501 5.9.2.1 只要求对服务级 scope 做校验；RFC 6749 10.2 指出恶意客户端问题，10.3 指出资源服务器无法判定令牌是否签发给出示它的客户端。	ts-29-500:4994, ts-29-500:4996, rfc6749:2827, s0365, s0362
`refid_known`	另一消费者配置的迁移前资源 URI（refIds，含 transaction reference id）；除此之外只有网络能力与自己的合法身份。	测试 A08/S04/Z01 都把 refIds 列为受保护资产；s0049/s0053 说明 MFAF 建立配置时分配 transaction reference id 并把 201 Created 回给配置者，s0273 说明资源 URI 结构由 29.501 4.4.1 定义，源文本没有规定该标识不可猜测或不可通过管理接口枚举。这是一个能力假设，不是文档陈述。	s0049, s0053, s0273, s0332

ID	network	malicious_nf	refid_known
A03	成立	成立	
A08	成立	成立	成立
D01	成立	成立	
R03	成立	成立	
S04	不成立	成立	不成立
Z01	成立	成立	不成立
A03.7	成立	成立	
A08.8	成立	成立	成立
U01	不成立		
X01	成立		
X02	不成立		

原文和目标

Source: 406 units, 160566 bytes, frozen in source-delivery-01.json. Targets: 7 approved (A03, A08, D01, R03, S04, Z01, U01); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

宿主按文档默认值（任务 `decisions: auto`）所做的决定，请你核对：

- approve_package: 批准源文包 protocol-0bef38e35135794d2771dc7b9c8b

这一步自身的报告：2 个请求（2 个失败），0.000 USD，输入 177564 tokens，输出 9523 tokens，224 s。

关于证明器和模型的说明

- A08：在 ProVerif 中作答，分析者为该目标指明 changing_state
- R03：在 ProVerif 中作答，分析者为该目标指明 changing_state
- S04：在 ProVerif 中作答，分析者为该目标指明 changing_state
- A08.8：在 ProVerif 中作答，分析者为该目标指明 changing_state
- U01：在 ProVerif 中作答，分析者为该目标指明 changing_state
- X02：在 ProVerif 中作答，分析者为该目标指明 changing_state

对代数作了化简（xor、dh_algebra、other_algebra）的 ProVerif 模型给攻击者的能力弱于真实运算：在那里得到的证明是针对更弱攻击者的证明，在那里找到的攻击仍是攻击。漏掉对会变化的值（changing_state）的检查的 ProVerif 模型让参与方接受得更多：在那里得到的证明成立，在那里找到的攻击可能只是缺少该检查。Tamarin+ Agent 按原样对这些建模。

以步数计量的界是 bounded.pv 上的答案：这是分析者的第二个 ProVerif 模型，其中持有会变化的值的参与方被展开到那么多步，并且检查在位。该模型做的事少于协议本身，因此性质只在这些步数内成立，再往后不成立；在那里找到的攻击仍是攻击。步数是分析者的陈述。

分析者的状态审查

这部分是分析者本人对原文的解读，而不是证明器的回答。“shown by ProVerif”下的“yes”表示 ProVerif 在展开的模型中找到了一条包含这些步骤的轨迹，因此这些步骤是可能的；至于它们是否攻破某个目标，只有回答表中的一次攻击才能说明，而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，并不证明任何事情。

值	kept by	changes when	checked when	refs
消费者在源 MFAF 处的个人 MFAF 配置及其事务引用 id（迁移前的资源 URI, refIds）	源 MFAF（当前持有该配置的 MFAF, s0234）	当指名它的 ContextTransferReq 被接受时：源 MFAF 回答 200 OK, 并在同一步本地删除该资源（s0112）。	当 ContextTransferReq 到达时，将请求的 refIds（ContextTransferReq.refIds，即包含事务引用 id 的资源 URI，s0332）与它所持有的资源 URI 进行比较。	s0049, s0053, s0112, s0173, s0332
目标 MFAF 处迁移后配置的新事务引用 id（newRefIds，迁移后的资源 URI）	目标 MFAF（调用迁移的 NF 服务消费者）	当 200 OK 被接受时：目标 MFAF 创建一个个人 MFAF 配置资源，并由响应交给它新的事务引用 id（newRefIds）（s0112, s0234）。	响应会对照目标所请求的 refId 和它所寻址的源身份被接受；configs 映射的 refIds 必须是所请求的 refIds 之一（s0335）。	s0112, s0234, s0335
配置者为它所创建的配置持有的资源 URI（它用来更新或取消配置的句柄）	配置者（创建该配置的 NF 服务消费者，s0035）	仅当配置者自己更改它时；迁移过程不会给它新值。	每当配置者发送 Update 或 Deconfigure 时，这些操作由它所持有的个人 MFAF 配置资源 URI 来寻址（s0055, s0071）。	s0035, s0055, s0071, s0112, s0234

pattern	result	target	shown by ProVerif	steps
replay	nothing found			在 bounded.pv 中，对同一配置的第二次迁移请求会被拒绝（ReqRefused, Src_gone），并且 r03 得到验证：没有第二次删除。重放的响应会让目标在同一个 newRefId 下再次创建同一资源，而该 id 命名一个资源；目标侧没有展开到两个响应，因此这没有被查询确认。
repeat_one_time_action	nothing found			删除只能发生一次：源忘记该配置（Src_have 变为 Src_gone），并且 r03 得到验证；在保留它的有缺陷变体中，r03_ctl 被证伪，同一配置被删除两次。
desynchronisation	candidate	X02	yes: ProVerif falsifies `x02` on bounded.pv	1. 配置者在源 MFAF 处创建了该配置，并持有其资源 URI（refIds），s0035, s0049, s0053。/ 2. 目标 MFAF 调用迁移，源以带有 newRefIds 的 200 OK 作答并删除其配置，s0110, s0112。/ 3. 目标在新的事务引用 id 下创建该配置，s0112。/ 4. 原文只把 newRefIds 交给迁移的消费者（s0234），没有定义配置者借以得知新 URI 的任何操作。/ 5. 配置者之后发往它所持有 URI 的 Update 或 Deconfigure（s0055, s0071）在源处找不到配置；各方得到不同的值，且在迁移过程内无法回退。
update_before_check	nothing found			Src_have 在作答之前、在 ConfRemoved 之前执行所有检查（令牌签名和范围、消费者签名、refIds 匹配）；在被验证之前，没有任何消息会改变所保留的值。
no_rollback_on_failure	candidate	U01	yes: ProVerif falsifies `u01` on bounded.pv	1. 目标 MFAF 发送迁移请求（s0110）。/ 2. 源 MFAF 接受它，以带有 configs 和 newRefIds 的 200 OK 作答，并在同一步删除其配置（s0112）。/ 3. 网络上的攻击者丢弃该响应。/ 4. 目标什么也不创建；原文没有定义重试、定时器或补偿（s0113 只涵盖处理请求时引发的错误）。/ 5. 该配置现在不由任何一个 MFAF 持有，尽管源 MFAF 已把迁移当作完成。
race_between_sessions	nothing found			在 bounded.pv 中，一个配置被迁移目标请求，同时被一个合法的 NF 以攻击者的名义请求；展开的源服务第一个并拒绝第二个，因此两个请求者的任何交错都不会产生两次删除（r03 已验证）。在两个请求之间重新创建该配置的运行不在展开的三步之内。

pattern	result	target	shown by ProVerif	steps
distinguishable_failure	candidate		no: not tried with ProVerif	1. 一个持有服务范围的合法 NF 请求一个猜测的资源 URI。/ 2. 如果配置存在，源 MFAF 以带有 configs 的 200 OK 作答（s0112），否则以错误响应作答（s0113）。/ 3. 两个回答不同，因此请求者得知该猜测是否命名了一个配置，并能枚举迁移赖以提供保护的名称空间。
wrap_or_window	nothing found			迁移中不存在计数器、序列号或接受窗口；事务引用 id 是资源名称，而不是单调递增的值。

分析者为目标命名的内容：

step	targets	features	reason
39	R03, S04, X02, U01, A08, A08.8	changing_state	这些目标依赖于迁移所替换的事务引用 id（refIds 变为 newRefIds，s0112, s0234）以及源对配置的删除，这是一个在单次运行的各步骤之间发生变化的值。ProVerif 在 bounded.pv 中针对展开的源回答它们，而不是针对任意数量的会话。
39	A03, A03.7, D01, Z01, X01	plain_primitives	模型只使用无哈希的原语：用于 SBI 各端点 TLS 的非对称加密和签名，以及作为 NRF 签名的 OAuth2 令牌。为 TLS 假定的 Diffie-Hellman 密钥协商被理想化为普通的密钥协商，且任何回答都不依赖原语的代数定律。

分析者的结束说明

回答了十一个目标，每个都配有对照。已证明：目标 MFAF 接受的响应在身份、refIds 和内容上与真实的源侧运行相匹配（A03, A03.7, D01），并对跳过签名和令牌检查的目标和源设有对照；迁移后的配置保持其 notificationURI/correId 绑定以及 refIds 到 newRefIds 的对应关系（A08, A08.8）；配置和 refIds 针对网络攻击者的传输机密性（X01）；以及每次迁移都由带有服务范围的令牌作支撑（Z01，第一个主张）。在两个展开步骤上验证：无论请求被重放多少次，每个配置只删除一次（R03，对照：保留配置的源会删除两次）。

已攻击：一个只持有服务范围且知道迁移前资源 URI 的合法 NF 会得到另一个消费者的配置，并且源删除其副本，因为源检查了令牌和范围，却从不检查请求者是否有权拥有该配置（Z01, S04；建议在模型副本上检查，攻击消失，正常流程保持，且没有其他回答改变）；被丢弃的 200 OK 使配置在源处被删除，却从未在目标处创建（U01）；并且在任何迁移之后，配置者自己的 URI 不再解析，因此它既不能更新也不能取消配置它所创建的配置（X02, S04）。S04 的两次攻击与 Z01 和 X02 的是同一些运行，并被归在它们之下。

阻碍更多结论的因素：证明依赖于理想化的端点 TLS 以及事务引用 id 不可猜测——知道它的攻击者是单独的假设 refid_known；拒绝被建模为各方停止，而不是 HTTP 状态码，因此状态审查中看到的枚举预言机仍是一个候选，未被证明；通过 SCP 的 307/308 重定向和间接通信不在范围之内；每次迁移只建模一个配置，因此多条目响应中的部分成功没有得到回答；并且状态问题只对 bounded.pv 展开的两个步骤作出回答。三条建议中有两条改变的是交换过程而不是某项检查，因此没有被检查。