← 述

OmniAUSF 置

配置从应用程序环境键 `:omniausf` 中读取，该键由 `config/runtime.exs` 填充。大多数运行时键由启动时读取的操作系统环境变量支持，如下所示；两个上下文回收键直接从 `:omniausf` 应用程序环境中读取，而不依赖环境变量。

```
config :omniausf,
  udm_uri:                 "http://127.0.0.12:7777",
  sbi_scheme:              "http",
  sbi_addr:                "127.0.0.19",
  sbi_port:                7777,
  nrf_uri:                 "http://127.0.0.1:7777",
  mcc:                     "999",
  mnc:                     "70",
  prometheus_metrics_port: 9568,
  heartbeat_interval:      10_000

config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all
```

□□□□

□□	□□	□□	□□	
`udm_uri`	□□□	□	`"http://127.0.0.12:7777"`	`UDM_U`
`sbi_scheme`	□□□	□	`"http"`	`SBI_S`
`sbi_addr`	□□□	□	`"127.0.0.19"`	`SBI_A`
`sbi_port`	□□	□	`7777`	`SBI_P`
`nrf_uri`	□□□	□	`"http://127.0.0.1:7777"`	`NRF_U`
`mcc`	□□□	□	`"999"`	`MCC`
`mnc`	□□□	□	`"70"`	`MNC`

参数	类型	必需	默认	
`prometheus_metrics_port`	整数	否	`9568`	`PROME`
`heartbeat_interval`	整数（毫秒）	否	`10000`	`HEART`
`context_ttl_seconds`	整数（秒）	否	`3600`	-
`context_reap_interval_ms`	整数（毫秒）	否	`60000`	-

> 关于 **`nrf_uri`** 默认值的说明：`config/runtime.exs` 将 `NRF_URI` 默认设置为 `http://127.0.0.1:7777`。在标准的回环部署中，NRF 位于 `http://127.0.0.10:7777`；请显式设置 `NRF_URI` 以匹配您的 NRF 地址。

日志记录

日志以 JSON 格式（`OmniLogger.JsonFormatter`）输出，包含所有元数据。认证路径上的日志行携带会话上下文（`supi`、`auth_ctx_id`、`procedure`），因此可以通过 `auth_ctx_id` 或 SUPI/SUCI 追踪单个认证交换的端到端。日志级别可以通过管理 API 在运行时更改（见下文）。

管理 API（端口 8443）

除了 SBI 监听器和指标端口外，OmniAUSF 在端口 `8443` 上公开 HTTPS 管理/OAM API（TLS，产品名称 `OmniAUSF`）。它用于操作检查和运行时控制，并不是 3GPP SBI 的一部分。关键端点：

方法	路径	目的
GET	`/api/status/api`	API 服务状态
GET	`/api/status/nrf`	NRF 注册状态
GET	`/api/status/nf`	NF 摘要（类型 `AUSF`，`nausf-auth` 服务，活动 UE 上下文计数）
GET	`/api/ausf_sessions`	列出活动认证会话（上下文 ID，SUCI/SUPI，服务网络名称，认证类型/结果）
DELETE	`/api/ausf_sessions/{id}`	通过上下文 ID 清除特定认证会话
POST	`/api/ausf_sessions`	清除所有活动认证会话
GET	`/api/statistics`	认证会话计数器（总计 / 成功 / 失败 / 待处理）
GET	`/api/oam/config`	查看有效的运行时配置（`sbi_addr`，`sbi_port`，`nrf_uri`，`udm_uri`，NF 容量/优先级，日志级别）
PATCH	`/api/oam/config/{id}`	更新运行时配置：`udm_uri`，`nrf_uri`，`nf_capacity`，`nf_priority`，`log_level`
POST	`/api/oam/log_level`	在运行时更改日志级别（`emergency`…`debug`）
GET	`/api/oam/resources`	BEAM VM 资源使用情况（内存，进程/端口计数，OTP/BEAM 版本）
GET	`/api/oam/diagnostics`	BEAM VM 诊断（运行队列，I/O，GC，连接节点）
POST	`/api/oam/nrf/reregister`	强制重新注册到 NRF

通过 `PATCH /api/oam/config` 和 `POST /api/oam/log_level` 进行的运行时更改仅适用于运行中的节点，并不会持久化；它们在重启时恢复为配置的值。

仅 OAM 设置

通过 `GET`/`PATCH /api/oam/config` 暴露了两个值，但在上表中不是启动参数 - 它们存在于 NRF 配置文件中，而不是核心 `:omniausf` 配置中：

字段	类型	意义
`nf_capacity`	整数 (0-65535)	在 AUSF 的 NRF 配置文件中宣传的相对容量，供消费者在多个 AUSF 实例注册时作为负载均衡权重。`PATCH` 重新向 NRF 宣传新值
`nf_priority`	整数 (0-65535)	在 NRF 配置文件中宣传的相对选择优先级；较低的优先级更受欢迎。`PATCH` 重新向 NRF 宣传新值

`PATCH /api/oam/config` 在应用之前验证这些值：`nf_capacity`/`nf_priority` 必须是整数（接受并解析十进制字符串），而 `log_level` 必须是已知级别。无效值将以 `400` 拒绝整个请求，而不应用任何内容。

检查认证会话

`GET /api/ausf_sessions` 返回每个内存中 UE 认证上下文 的一个条目，包含以下字段：

字段	意义
`ctx_id`	`authCtxId`（UUID）；传递给 `DELETE /api/ausf_sessions/{ctx_id}` 以清除一个会话
`suci`	AMF 在初始化时呈现的 `supiOrSuci`
`supi`	UDM 解析的普通 SUPI（在向量响应到达之前为 null）
`serving_network_name`	会话的密钥绑定的 SN 名称
`auth_type`	一旦存储了向量，则为 `5G_AKA`
`auth_result`	确定的结果，或在会话仍在进行中（待处理）时为 null
`has_kausf`	此会话是否持有 `K_AUSF` - 表示已获取向量并派生了密钥。密钥材料本身从不暴露

由于上下文在确认结算的瞬间被删除，因此此列表中的条目通常是短暂的；持久的会话是 AMF 发起但从未确认或注销的会话。

← 概述

OmniAUSF 指标

指标在 `http://{host}:{prometheus_metrics_port}/metrics` 处公开（默认端口 `9568`）。

AUSF 指标

指标	类型	标签	描述
`omni_ausf_auth_count`	计数器	`result`	UE 认证操作。`result` 是 `initiated`（获取向量，返回挑战）、`success`（RES* 匹配 XRES*）、`failure`（UDM 错误、格式错误的向量或 RES* 不匹配）、`resync`（检测到 AUTS/SQN 重新同步并转发给 UDM）或 `resync_invalid`（格式错误的重新同步数据被丢弃）之一
`omni_ausf_nrf_registration_status`	计量器	`nf_type`	NRF 注册状态（1 = 已注册，0 = 未注册）
`omni_ausf_active_contexts_count`	计量器	-	活跃的 UE 认证上下文数量。由遥测事件 `[:omniausf, :ue_contexts, :active]` 驱动，实时计量，事件在每次上下文创建和删除以及每次 ContextReaper 清理时发出，因此它跟踪当前的在飞上下文计数

> `omni_ausf_active_contexts_count` 是一个实时计量器。它由遥测事件 `[:omniausf, :ue_contexts, :active]` 驱动，该事件在活跃上下文[illegible]变化时由 AUSF 上下文存储发出：在上下文创建时、删除时（确认、注销或认证结果删除）以及在每次清理空闲上下文的 ContextReaper 清理时。因此，该计量器反映当前活跃的认证上下文计数，安全用于警报和仪表板。OAM API 仍可用于交叉检查：`GET /api/status/nf`（包括活跃的 UE 上下文计数）或 `GET /api/statistics`（总计 / 成功 / 失败 / 待处理会话计数器）。

> `omni_ausf_auth_count` 涵盖主要的认证流程以及重新同步路径。`resync` 和 `resync_invalid` 标签计算在初始化请求中检测到的 SQN 重新同步尝试（分别为验证并转发和格式错误并丢弃）。注销和认证结果删除路径发出自己的遥测事件，但目前未附加到任何导出的 Prometheus 指标。

> 内部遥测（不是 **Prometheus** 指标）。上下文清理器发出 `[:omniausf, :ausf, :context_reap]` 遥测事件，携带每次清理中清理的上下文计数。这是供操作员连接自己处理程序的内部遥测；它未注册为导出的 Prometheus 指标。

示例 **PromQL**

- 5 分钟内的认证成功率：`sum(rate(omni_ausf_auth_count{result="success"}[5m])) / sum(rate(omni_ausf_auth_count{result="initiated"}[5m]))`
- 每秒认证失败次数：`sum(rate(omni_ausf_auth_count{result="failure"}[5m]))`
- 当前注册的 AUSF 与 NRF：`omni_ausf_nrf_registration_status == 1`
- 每秒 SQN 重新同步尝试（有效与格式错误）：`sum(rate(omni_ausf_auth_count{result="resync"}[5m]))` 和 `sum(rate(omni_ausf_auth_count{result="resync_invalid"}[5m]))`
- 活跃的在飞认证上下文：`omni_ausf_active_contexts_count`

为了独立交叉检查在飞上下文计数，OAM API（`GET /api/status/nf` 或 `GET /api/statistics`）报告相同的数字。

BEAM VM 指标

指标	类型	描述
`beam_memory_total`	计量器	BEAM 总内存（字节）
`beam_memory_processes`	计量器	Erlang 进程使用的内存
`beam_memory_system`	计量器	系统内存（ETS、原子、代码）
`beam_processes_count`	计量器	Erlang 进程数量
`beam_vm_uptime`	计量器	VM 运行时间（秒）

← 概述

OmniAUSF 操作

OmniAUSF 实现了 5G 核心的认证服务器功能 (AUSF)，在 AMF (N12) 和 UDM (N13) 之间执行 5G-AKA 主要认证。它从 UDM 请求认证向量，将 RAND/AUTN 挑战 (和 HXRES*) 中继到 AMF，验证 UE 的响应，推导锚定密钥 K_SEAF，并将认证结果报告回 UDM。

3GPP 角色和规范参考

规范	相关性
TS 23.501	系统架构 - AUSF 在 5G 系统中的角色
TS 33.501	安全架构 - 5G-AKA 认证 (6.1.3)，K_AUSF/K_SEAF 密钥层次结构，HXRES*/HRES* (附录 A.5)，K_SEAF 推导 (附录 A.6)
TS 29.509	Nausf_UEAuthentication 服务 API (HTTP SBI)
TS 29.503	Nudm_UEAuthentication 服务 API - 认证向量生成和认证结果报告
TS 29.510	Nnrf_NFManagement 和 Nnrf_NFDiscovery 服务 API - NF 注册/心跳和 UDM 发现
TS 24.501	NAS - 服务网络名称格式 (`5G:mnc<MNC>.mcc<MCC>.3gppnetwork.org`)

接口

接口	对等方	方向	传输	目的
N12 (Nausf_UEAuthentication)	AMF	入站	HTTP SBI	UE 认证初始化和 5G-AKA 确认
N13 (Nudm_UEAuthentication)	UDM	出站	HTTP SBI	认证向量生成和认证结果报告
Nnrf	NRF	出站	HTTP SBI	NF 注册、心跳和对等 NF (UDM) 发现 (`Nnrf_NFDiscovery`, TS 29.510)

SBI 端点

所有 SBI 端点都在基础 URL `{sbi_scheme}://{sbi_addr}:{sbi_port}` 下提供，并在请求中使用 `Content-Type: application/json`。成功的创建响应使用 `application/3gppHal+json`，符合 TS 29.509。

□□	□□	□□	
POST	`/nausf-auth/v1/ue-authentications`	Nausf_UEAuthentication	□□ AU `su` `se`
PUT	`/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation`	Nausf_UEAuthentication	□□ `re`
DELETE	`/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation`	Nausf_UEAuthentication	□□
POST	`/nausf-auth/v1/ue-authentications/deregister`	Nausf_UEAuthentication	□□ □

□□/□□□□

□□□ **UE** □□ - □□□□□ `supiOrSuci` (SUPI □ SUCI□□□ `suci` □□□□) □ `servingNetworkName`□□□□□□ `resynchronizationInfo` □□□□□□ UDM □□ SQN □□□ □□AUSF □ UDM □□ 5G-AKA □□□□□□□ `201 Created` □ `UEAuthenticationCtx` □□

- `authType: "5G_AKA"`
- `5gAuthData`: `{ rand, autn, hxresStar }` (□□□□□□□)
- `_links.5g-aka.href`: □□□□□□□□□ URL
- `servingNetworkName`

`Location` □□□□□□□□□□ (`.../ue-authentications/{authCtxId}`)□

5G-AKA □□ - □□□□□ `resStar` (□□□□)□AUSF □□□□□ `resStar` □□□□ `XRES*` □□□□ (TS 33.501 6.1.3.2□AMF □□ HRES*/HXRES* □□□□)□□□ UDM □□□□□ (`auth-events`)□□□ □□□ `200 OK` □ `ConfirmationDataResponse`□

- □□□: `{ authResult: "AUTHENTICATION_SUCCESS", supi, kseaf }` (K_SEAF □□□□□□□□□)□

- 失败时：`{ authResult: "AUTHENTICATION_FAILURE", supi }`。请注意，认证失败在体中通过 `authResult` 信号，而不是通过 HTTP 错误状态。

认证上下文在确认后被删除，无论结果如何。

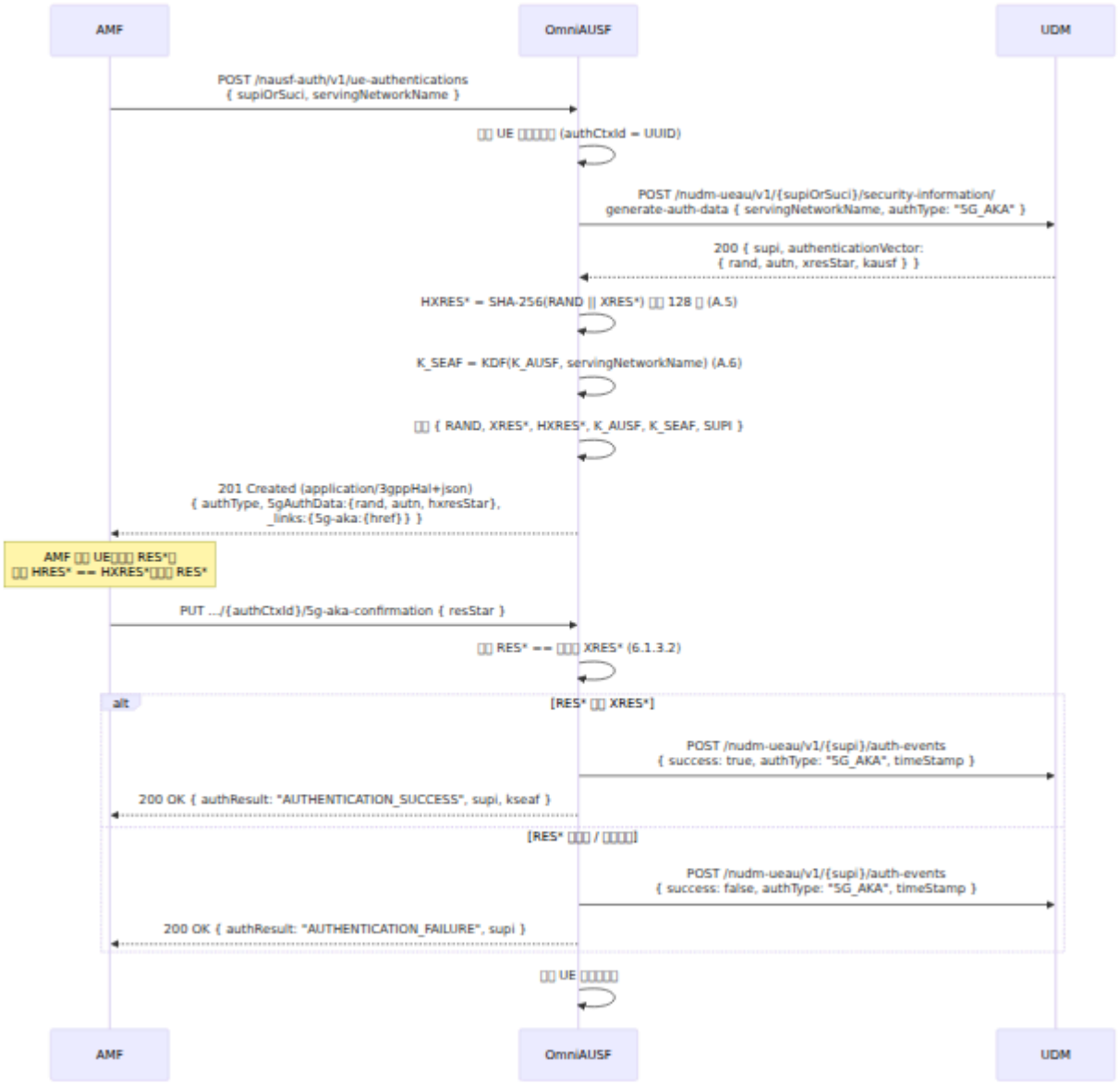
错误响应

错误作为 `ProblemDetails` 体返回。

HTTP 状态	原因	条件
400	`MANDATORY_IE_MISSING`	缺少必需字段 (`supiOrSuci`/`servingNetworkName` 在初始化时；`resStar` 在确认时；`supi` 在注销时)
404	`USER_NOT_FOUND`	UDM 在生成向量时返回 404，表示找不到订阅者
404	`CONTEXT_NOT_FOUND`	确认请求中的 `authCtxId` 与任何活动上下文不匹配
500	`SYSTEM_FAILURE`	UDM 无法访问，或 UDM 返回了格式错误的认证向量

□□□□

5G-AKA □□ (TS 33.501 □ 6.1.3 □)



□□□□ □ `POST .../ue-authentications` □□AUSF □□□□□□□□□□□ `/nudm-ueau/v1/{supiOrSuci}/security-information/generate-auth-data` □ UDM □□ 5G-AKA □□□□□ AUSF NF □□ ID □ AMF □□□□□ `resynchronizationInfo`□UDM □□□□□ □□□ (`rand`, `autn`, `xresStar`, `kausf`) □□□□□ `supi` (□ SUCI □□□)□AUSF □ RAND □ XRES* □□ `HXRES*`□□□□□□□□□□□□ `K_AUSF` □□□ `K_SEAF`□□□□□□□□□□□□□ AMF□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `500 SYSTEM_FAILURE`□

□□□ □ `PUT .../{authCtxId}/5g-aka-confirmation` □□AUSF □□□□□□□□ `resStar`□ □□□□□□□□ `XRES*` □□□□□□□□□ `auth-events` □ UDM □□□□ (□□□□□□□□□□□□□□□□□□□□□□□□ AMF □□□) □□□□□□□□□□□□□□□□□□□□□□□□ `authCtxId` □□ `404 CONTEXT_NOT_FOUND`□

服务网络名称。根据 `mcc`/`mnc` 构建为 `5G:mnc<MNC>.mcc<MCC>.3gppnetwork.org`，符合 TS 24.501，两个数字的 MNC 补零到三位数。服务网络名称将 `K_SEAF` 绑定到服务网络，因此 AMF 和 AUSF 必须就 PLMN 身份达成一致。

SQN 重新同步。当 AMF 在初始化请求中包含 `resynchronizationInfo` (UE 生成的 `RAND` 和 `AUTS`) 时，AUSF 检测到重新同步条件，验证 `rand` 和 `auts` 都是格式正确的十六进制，并将验证后的对发送到 UDM/ARPF 的 `generate-auth-data` 请求中，以便 UDM 可以重新推导序列号 (TS 33.501 6.1.3.3)。格式错误的重新同步输入会被丢弃，不会转发。AUSF 本身不计算序列号，但它并不是盲目的传递：它在转发之前检查和验证重新同步数据块。

密钥层次结构和推导分离 **(TS 33.501** 附录 **A)**

5G-AKA 密钥层次结构在 UDM 和 AUSF 之间分离。AUSF 从不看到长期密钥 (Ki) ��� CK/IK - 它接收 UDM 在认证向量中已经推导的 `K_AUSF`，并仅从中推导锚定密钥 `K_SEAF` ◦AMF (而不是 AUSF) 继续链条，直到 `K_AMF` 和接入层密钥。



UDM / ARPF



K_SEAF = KDF(K_AUSF,
SNN)
FC 0x6C (A.6)



K_AMF = KDF(K_SEAF,
SUPI, ABBA)
FC 0x6D (A.7)

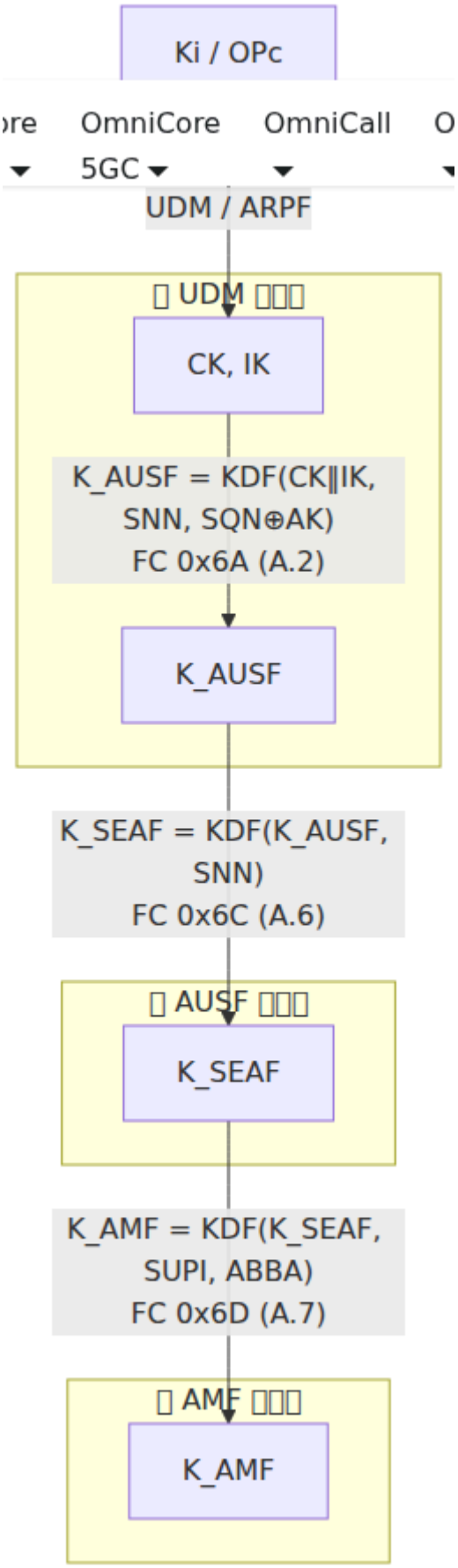


密钥/值	来源	推导	**AUSF** 中的目的
`K_AUSF`	UDM (ARPF)	`KDF(CK||IK, servingNetworkName, SQN⊕AK)`，FC `0x6A` (A.2)	在向量中接收；AUSF 锚定密钥，从中推导 `K_SEAF`。保存在上下文中，永不返回给 AMF
`K_SEAF`	AUSF	`KDF(K_AUSF, servingNetworkName)`，FC `0x6C` (A.6)	服务网络锚定密钥。仅在 `AUTHENTICATION_SUCCESS` 时返回给 AMF；AMF 从中推导 `K_AMF`
`HXRES*`	AUSF	`SHA-256(RAND || XRES*)` 的低 128 位 (A.5)	在初始化时返回给 AMF，以便 AMF 可以在转发 `RES*` 之前匹配 UE 的 `HRES*`
`XRES*`	UDM (ARPF)	`KDF(CK||IK, SNN, RAND, XRES)`，FC `0x6B` (A.4)	在向量中接收；AUSF 在确认时将 `RES*` 与之进行比较的预期响应

由于 `K_SEAF` 绑定到 `servingNetworkName`（而上游的 `K_AUSF` 绑定到相同的 SN 名称和 `SQN⊕AK`），AMF、AUSF 和 UDM 必须就 PLMN 身份达成一致，以便推导出的密钥与 UE 的匹配。服务网络名称不匹配在确认时表现为 `AUTHENTICATION_FAILURE`，而不是作为显式错误。

删除 **5G-AKA** 认证结果 **(TS 29.509)**

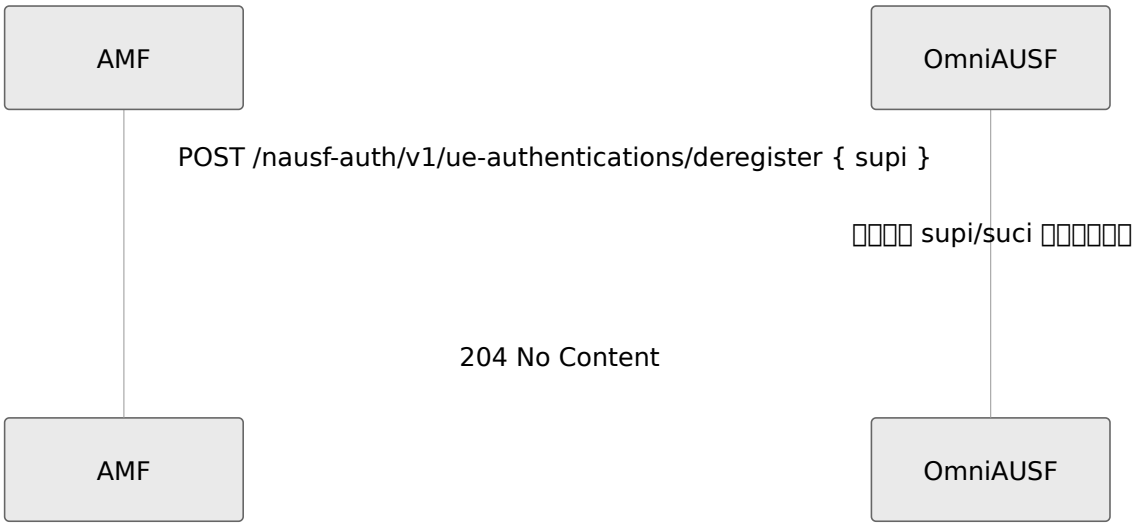
AMF 可以在没有确认交换的情况下显式删除认证结果/上下文。这映射到 TS 29.509 中的 `Delete5gAkaAuthenticationResult`。



AUSF □□□□□□□□ (□□□ `authCtxId` □□□□)□□□□□□ `204 No Content`□□□□ UDM □□□

注销 (TS 29.509)

删除 AUSF 为给定 SUPI 持有的任何 UE 安全上下文。



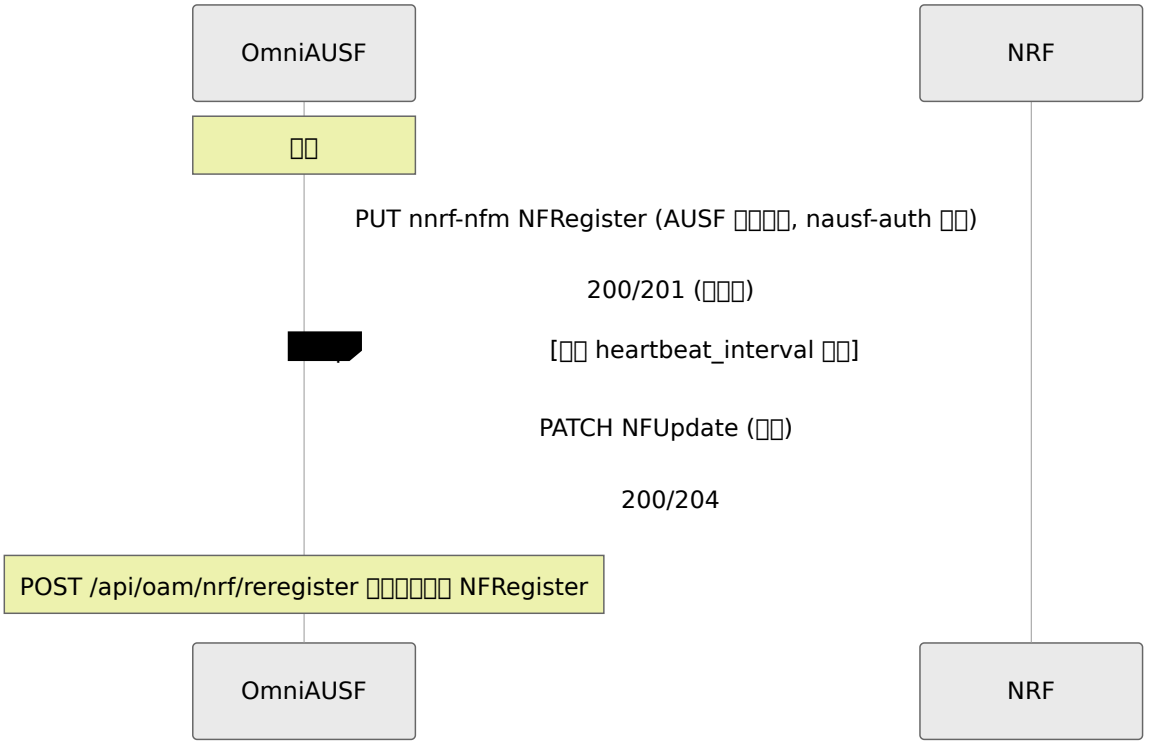
supi 是必需的 (缺失/空值返回400 MANDATORY_IE_MISSING)。AUSF 遍历其内存中的上下文，删除任何存储的 supi 或 suci 等于请求的 SUPI 的上下文；该操作仅限于本地，不会调用 UDM。它始终返回 204 No Content，即使没有匹配的上下文存在。

UE 认证上下文

每次初始化都会创建一个内存中的 UE 认证上下文，以生成的 authCtxId (UUID) 为键。上下文保存完成交换所需的材料，是 OAM 会话端点 (GET /api/ausf_sessions, DELETE /api/ausf_sessions/{id}) 列出和清除的单位。

字段	设置于	含义
`ctx_id`	初始化	`authCtxId` UUID；出现在确认 URL 和 OAM 会话列表中
`suci`	初始化	AMF 最初呈现的 `supiOrSuci`（SUCI 或 SUPI）
`supi`	向量响应	UDM 解析的纯 SUPI（解密）
`serving_network_name`	初始化	密钥绑定的 SN 名称
`rand`、`xres_star`、`hxres_star`、`kausf`、`kseaf`	向量响应	在交换期间持有的向量和推导密钥
`auth_type`	向量响应	`5G_AKA`
`auth_result`	-	为已解决的会话填充；仍在进行中的上下文没有结果，并在 `GET /api/statistics` 中计为 `pending`

上下文通常仅在初始化和结算之间存在。它在确认时被删除（成功或失败），在显式 `DELETE .../5g-aka-confirmation` 时，在其 SUPI 的 `deregister` 时，以及在进程重启时。此外，闲置或被遗弃的上下文由监督的定期清理程序进行垃圾收集：AMF 发起但从未确认的上下文在超过 `context_ttl_seconds`（默认 `3600` 秒）后被清理，清理程序每 `context_reap_interval_ms`（默认 `60000` 毫秒）扫描一次。这限制了被遗弃上下文的生命周期，而无需操作员干预。有关如何手动检查和清除上下文的信息，请参见 增长的活动上下文计数。



NRF 注册

在启动时，AUSF 注册一个 `AUSF` NF 配置文件到 NRF，广告 `nausf-auth` 服务 (`v1`, `1.0.0`)、其 SBI 地址/端口/方案和配置的 PLMN。它每 `heartbeat_interval` 毫秒发送心跳。可以通过 `POST /api/oam/nrf/reregister` 在运行时强制重新注册。AUSF 还使用 NRF 发现 UDM：在每个 Nudm 调用之前，它查询 `Nnrf_NFDiscovery` (TS 29.510) 以获取 `udm` 实例并以发现的 UDM 为目标。只有在发现没有实例 (或 NRF 无法访问) 时，它才会回退到静态配置的 `udm_uri`，因此单 UDM 和静态部署仍然可以正常工作。

← 概述

OmniAUSF 故障排除

身份验证失败，返回 404 USER_NOT_FOUND

在 `generate-auth-data` 期间，UDM 返回了 404 的订阅者。确认：

1. `udm_uri` 是正确的，并且可以从 OmniAUSF 主机访问。
2. 订阅者 (IMSI) 已在 UDM/UDR/HSS 后端中配置。
3. AMF 提供的 SUCI 格式正确，并且可以被 UDM 解密。

AUSF 日志显示 `AUSF auth failed for {supiOrSuci}: user not found at UDM`。

确认返回 AUTHENTICATION_FAILURE (HTTP 200)

解码后的 `resStar` 与存储的 `XRES*` 不相等。这在主体中以 `authResult: "AUTHENTICATION_FAILURE"` 的形式表示，而不是作为 HTTP 错误。常见原因：

- UE 的凭据 (Ki/OPc) 与后端中配置的凭据不匹配。
- RAND/AUTN 或 RES* 在传输过程中被更改，或者 AMF/AUSF 对服务网络名称存在分歧（这会导致向量不匹配）。
- 格式错误/非十六进制的 `resStar` 被视为不匹配。

AUSF 日志显示 `AUSF 5G-AKA authentication FAILED for {supi}: RES* mismatch`。

确认返回 404 CONTEXT_NOT_FOUND

PUT 请求中的 `authCtxId` 与任何活动上下文不匹配。上下文在任何确认（成功或失败）后被删除，并在 AUSF 重启时丢失。AMF 必须从 `POST .../ue-authentications` 重新发起身份验证。

发起返回 500 SYSTEM_FAILURE

要么 UDM 在 `udm_uri` 上不可达，要么 UDM 返回了格式错误的身份验证向量（缺少/非十六进制的 `rand`、`xresStar` 或 `kausf`，或缺少 `autn`）。验证 UDM 连接性以及 UDM 的向量生成是否正常。AUSF 日志显示 `AUSF auth failed ...` 及原因，或 `AUSF auth failed: malformed authentication vector from UDM`。

AUSF 无法发现 / NRF 状态显示未注册

检查 `GET /api/status/nrf` 和 `omni_ausf_nrf_registration_status` 指标。验证 `nrf_uri` 是否正确（注意配置参考中的运行时默认警告）并且可达，然后使用 `POST /api/oam/nrf/reregister` 强制重新注册。

活动上下文计数增加

正在进行的身份验证上下文应在静止时返回接近零的状态。上下文在确认、删除或注销时被移除。如果 AMF 在没有确认的情况下放弃身份验证，这些上下文将在超过 `context_ttl_seconds`（默认 `3600` 秒）后由定期上下文清理器自动垃圾回收；清理器每 `context_reap_interval_ms`（默认 `60000` 毫秒）进行一次清理。因此，过期的上下文在 TTL 过期后会自行清除，无需操作员干预。

如果您需要更早地检查或清除上下文，请使用 OAM API: `GET /api/status/nf` 或 `GET /api/statistics` 获取实时计数，`GET /api/ausf_sessions` 列出它们，`DELETE /api/ausf_sessions/{id}`（或 `POST /api/ausf_sessions` 清除所有）立即删除它们。`omni_ausf_active_contexts_count` 指标直接跟踪此计数（在上下文创建、删除和每次清理器清扫时发出），因此您可以在 Prometheus 中监视它，也可以通过 OAM API 监视。如果计数增长速度快于 TTL 可以消耗的速度，请检查 AMF 是否反复发起未确认的身份验证。