

Configuração do OmniAUSF

A configuração é lida da chave de ambiente da aplicação `:omniausf`, populada a partir de `config/runtime.exs`. A maioria das chaves de tempo de execução é respaldada por uma variável de ambiente do sistema operacional lida na inicialização, mostrada abaixo; as duas chaves do context-reaper são lidas diretamente do ambiente da aplicação `:omniausf` e não são respaldadas por variáveis de ambiente.

```
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
```

Referência de Parâmetros

Parâmetro	Tipo	Requerido	Padrão
udm_uri	string	Não	"http://127.0.0.12:7
sbi_scheme	string	Não	"http"
sbi_addr	string	Não	"127.0.0.19"
sbi_port	integer	Não	7777

Parâmetro	Tipo	Requerido	Padrão
nrf_uri	string	Não	"http://127.0.0.1:75
mcc	string	Não	"999"
mnc	string	Não	"70"
prometheus_metrics_port	integer	Não	9568
heartbeat_interval	integer (ms)	Não	10000
context_ttl_seconds	integer (s)	Não	3600

Parâmetro	Tipo	Requerido	Padrão
<code>context_reap_interval_ms</code>	integer (ms)	Não	<code>60000</code>

Nota sobre o padrão de `nrf_uri`: `config/runtime.exs` define `NRF_URI` como `http://127.0.0.1:7777`. Em uma implantação padrão de loopback, o NRF é acessado em `http://127.0.0.10:7777`; defina `NRF_URI` explicitamente para corresponder ao seu endereço NRF.

Registro

Os logs são emitidos como JSON (`OmniLogger.JsonFormatter`) com todos os metadados incluídos. As linhas de log no caminho de autenticação carregam

contexto de sessão (`supi`, `auth_ctx_id`, `procedure`) para que uma única troca de autenticação possa ser rastreada de ponta a ponta pelo `auth_ctx_id` ou SUPI/SUCI. O nível de log pode ser alterado em tempo de execução através da API de Gerenciamento (veja abaixo).

API de Gerenciamento (Porta 8443)

Além do listener do SBI e da porta de métricas, o OmniAUSF expõe uma API de gerenciamento/OAM HTTPS na porta `8443` (TLS, nome do produto `OmniAUSF`). É usada para inspeção operacional e controle em tempo de execução, e não faz parte do SBI 3GPP. Principais endpoints:

Método	Caminho	Propósito
GET	/api/status/api	Status do serviço da API
GET	/api/status/nrf	Status do registro do NRF
GET	/api/status/nf	Resumo do NF (tipo AUSF, serviço nausf-auth, contagem de contextos de UE ativos)
GET	/api/ausf_sessions	Lista sessões de autenticação ativas (id do contexto, SUCI/SUPI, nome da rede servidora, tipo/resultado da autenticação)
DELETE	/api/ausf_sessions/{id}	Limpar uma sessão de autenticação específica pelo id do contexto
POST	/api/ausf_sessions	Limpar todas as sessões de autenticação ativas
GET	/api/statistics	Contadores de sessões de autenticação (total / sucesso / falha / pendente)
GET	/api/oam/config	Visualizar configuração de tempo de execução efetiva (sbi_addr, sbi_port, nrf_uri, udm_uri, capacidade/prioridade do NF, nível de log)
PATCH	/api/oam/config/{id}	Atualizar configuração de tempo de execução: udm_uri, nrf_uri, nf_capacity, nf_priority, log_level

Método	Caminho	Propósito
POST	<code>/api/oam/log_level</code>	Alterar o nível de log em tempo de execução (<code>emergency</code> ... <code>debug</code>)
GET	<code>/api/oam/resources</code>	Uso de recursos da VM BEAM (memória, contagens de processos/portas, versão OTP/BEAM)
GET	<code>/api/oam/diagnostics</code>	Diagnósticos da VM BEAM (fila de execução, I/O, GC, nós conectados)
POST	<code>/api/oam/nrf/reregister</code>	Forçar re-registro com o NRF

Mudanças em tempo de execução feitas através de `PATCH /api/oam/config` e `POST /api/oam/log_level` se aplicam apenas ao nó em execução e não são persistidas; elas reverterão aos valores configurados na reinicialização.

Configurações apenas OAM

Dois valores são expostos através de `GET/PATCH /api/oam/config`, mas não são parâmetros de inicialização na tabela acima - eles residem no perfil do NRF em vez da configuração central `:omniausf`:

Campo	Tipo	Significado
<code>nf_capacity</code>	integer (0-65535)	Capacidade relativa anunciada no perfil do NRF do AUSF, usada pelos consumidores como um peso de balanceamento de carga quando várias instâncias do AUSF estão registradas. Um <code>PATCH</code> reanuncia o novo valor ao NRF
<code>nf_priority</code>	integer (0-65535)	Prioridade de seleção relativa anunciada no perfil do NRF; menor é preferido. Um <code>PATCH</code> reanuncia o novo valor ao NRF

`PATCH /api/oam/config` valida esses valores antes de aplicar:

`nf_capacity`/`nf_priority` devem ser inteiros (uma string decimal é aceita e analisada), e `log_level` deve ser um nível conhecido. Um valor inválido rejeita toda a solicitação com `400` sem aplicar nada.

Inspecionando sessões de autenticação

`GET /api/ausf_sessions` retorna uma entrada por contexto de autenticação de UE **em memória**, com os seguintes campos:

Campo	Significado
ctx_id	O authCtxId (UUID); passe para DELETE /api/ausf_sessions/{ctx_id} para limpar uma sessão
suci	O supi0rSuci que o AMF apresentou na inicialização
supi	O SUPI puro resolvido pelo UDM (nulo até que a resposta do vetor chegue)
serving_network_name	O nome da SN ao qual as chaves da sessão estão vinculadas
auth_type	5G_AKA uma vez que o vetor é armazenado
auth_result	Resultado final, ou nulo enquanto a sessão ainda está em andamento (pendente)
has_kausf	Se K_AUSF está mantido para esta sessão - indica que o vetor foi buscado e as chaves derivadas. O material da chave em si nunca é exposto

Como os contextos são excluídos no momento em que uma confirmação é estabelecida, as entradas nesta lista normalmente têm vida curta; uma sessão que persiste é aquela que o AMF iniciou, mas nunca confirmou ou desregistrou.

[← Visão Geral](#)

Métricas do OmniAUSF

As métricas são expostas em `http://{host}:`
`{prometheus_metrics_port}/metrics` (porta padrão `9568`).

Métricas do AUSF

Métrica	Tipo	Rótulos	Descrição
<code>omni_ausf_auth_count</code>	contador	<code>result</code>	Operações de autenticação UE. <code>result</code> é um dos <code>initiated</code> (vetor obtido, desafio retornado), <code>success</code> (RES corresponden a XRES*), <code>failure</code> (erro do UDM, veto malformato c discrepância RES*), <code>resync</code> (uma resincroniza AUTS/SQN foi detectada e retransmitida UDM), ou <code>resync_inval</code> (um blob de resincroniza malformato f descartado)

Métrica	Tipo	Rótulos	Descrição
<code>omni_ausf_nrf_registration_status</code>	gauge	<code>nf_type</code>	Status de registro do NF (1 = registrado, 0 = não registrado)
<code>omni_ausf_active_contexts_count</code>	gauge	-	Número de contextos de autenticação UE ativos. Gauge ao vivo impulsionado pelo evento de telemetria <code>[:omniausf, :ue_contexts :active]</code> , emitido em cada criação, exclusão de contexto e em cada varredura do ContextRepository. Portanto, rastreia a contagem atual de contextos em andamento.

`omni_ausf_active_contexts_count` é um gauge ao vivo. Ele é impulsionado pelo evento de telemetria `[ :omniausf, :ue_contexts, :active ]`, que o armazenamento de contexto do AUSF emite sempre que o conjunto de contextos ativos muda: na criação de contexto, na exclusão (confirmação, desregistro ou exclusão de resultado de autenticação) e em

cada varredura do ContextReaper que coleta contextos ociosos. Portanto, o gauge reflete a contagem atual de contextos de autenticação ativos e é seguro para alertas e dashboards. A API OAM permanece disponível para verificações cruzadas: `GET /api/status/nf` (inclui uma contagem de contexto de UE ativo) ou `GET /api/statistics` (contadores totais / sucesso / falha / sessão pendente).

`omni_ausf_auth_count` cobre o fluxo de autenticação primário mais o caminho de resincronização. Os rótulos `resync` e `resync_invalid` contam as tentativas de resincronização SQN detectadas na solicitação de iniciação (validadas e retransmitidas, e malformadas e descartadas, respectivamente). Os caminhos de desregistro e exclusão de resultado de autenticação emitem seus próprios eventos de telemetria, mas atualmente não estão anexados a nenhuma métrica Prometheus exportada.

Telemetria interna (não uma métrica Prometheus). O coletor de contexto emite um evento de telemetria `[ :omniausf, :ausf, :context_reap ]` carregando a contagem de contextos coletados por varredura. Esta é uma telemetria interna para operadores que conectam seus próprios manipuladores; não está registrada como uma métrica Prometheus exportada.

Exemplo PromQL

- Taxa de sucesso de autenticação em 5 minutos:

```
sum(rate(omni_ausf_auth_count{result="success"}[5m])) /  
sum(rate(omni_ausf_auth_count{result="initiated"}[5m]))
```

- Falhas de autenticação por segundo:

```
sum(rate(omni_ausf_auth_count{result="failure"}[5m]))
```

- AUSF atualmente registrado com o NRF:

```
omni_ausf_nrf_registration_status == 1
```

- Tentativas de resincronização SQN por segundo (válidas vs malformadas):

```
sum(rate(omni_ausf_auth_count{result="resync"}[5m])) e  
sum(rate(omni_ausf_auth_count{result="resync_invalid"}[5m]))
```

- Contextos de autenticação ativos em andamento:

`omni_ausf_active_contexts_count`

Para uma verificação cruzada independente da contagem de contextos em andamento, a API OAM (`GET /api/status/nf` ou `GET /api/statistics`) relata a mesma figura.

Métricas da BEAM VM

Métrica	Tipo	Descrição
<code>beam_memory_total</code>	gauge	Memória total da BEAM em bytes
<code>beam_memory_processes</code>	gauge	Memória usada por processos Erlang
<code>beam_memory_system</code>	gauge	Memória do sistema (ETS, átomos, código)
<code>beam_processes_count</code>	gauge	Número de processos Erlang
<code>beam_vm_uptime</code>	gauge	Tempo de atividade da VM em segundos

Operações do OmniAUSF

OmniAUSF implementa a Função do Servidor de Autenticação (AUSF) do Core 5G, executando a autenticação primária 5G-AKA entre o AMF (N12) e o UDM (N13). Ele solicita um vetor de autenticação do UDM, retransmite o desafio RAND/AUTN (e HXRES*) para o AMF, verifica a resposta do UE, deriva a chave âncora K_SEAF e relata o resultado da autenticação de volta ao UDM.

Referências de Papel e Especificação 3GPP

Especificação	Relevância
TS 23.501	Arquitetura do sistema - papel do AUSF no Sistema 5G
TS 33.501	Arquitetura de segurança - autenticação 5G-AKA (6.1.3), hierarquia de chaves K_AUSF/K_SEAF, HXRES*/HRES* (Anexo A.5), derivação de K_SEAF (Anexo A.6)
TS 29.509	API do serviço Nausf_UEAuthentication (HTTP SBI)
TS 29.503	API do serviço Nudm_UEAuthentication - geração de vetor de autenticação e relatório de resultado de autenticação
TS 29.510	APIs de serviço Nnrf_NFManagement e Nnrf_NFDiscovery - registro/heartbeat de NF e descoberta de UDM
TS 24.501	NAS - formato do nome da rede de serviço (5G:mnc<MNC>.mcc<MCC>.3gppnetwork.org)

Interfaces

Interface	Par	Direção	Transporte	Propósito
N12 (Nausf_UEAuthentication)	AMF	Entrada	HTTP SBI	Iniciação da autenticação de acesso e confirmação AKA
N13 (Nudm_UEAuthentication)	UDM	Saída	HTTP SBI	Geração de vetores de autenticação e envio de relatório de resultado de autenticação
Nnrf	NRF	Saída	HTTP SBI	Registro de NF, heartbeat e descoberta de par (UDM) (Nnrf_NFDiscovery TS 29.510)

Endpoints SBI

Todos os endpoints SBI são servidos sob a URL base

`{sbi_scheme}://{sbi_addr}:{sbi_port}` e usam `Content-Type:`

`application/json` na solicitação. A resposta de criação bem-sucedida usa

`application/3gppHal+json` conforme TS 29.509.

Método	Caminho	Serviço	
POST	<code>/nausf-auth/v1/ue-authentications</code>	Nausf_UEAuthentication	Ir d R e s
PUT	<code>/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation</code>	Nausf_UEAuthentication	C A
DELETE	<code>/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation</code>	Nausf_UEAuthentication	E r d A
POST	<code>/nausf-auth/v1/ue-authentications/deregister</code>	Nausf_UEAuthentication	R d p R

Resumo de Solicitação / Resposta

Iniciar autenticação do UE - o corpo da solicitação contém `supiOrSuci` (SUPI ou SUCI; o campo `suci` também é aceito) e `servingNetworkName`; um objeto opcional `resynchronizationInfo` é passado para o UDM para a resincronização do SQN. O AUSF solicita um vetor 5G-AKA do UDM, então retorna `201 Created` com um corpo `UEAuthenticationCtx`:

- `authType`: "5G_AKA"
- `5gAuthData`: { `rand`, `autn`, `hxresStar` } (strings hexadecimais)
- `_links.5g-aka.href`: a URL de confirmação absoluta para este contexto
- `servingNetworkName`

Um cabeçalho `Location` aponta para o recurso de contexto criado (`.../ue-authentications/{authCtxId}`).

Confirmação 5G-AKA - o corpo da solicitação contém `resStar` (hex). O AUSF compara o `resStar` decodificado com o `XRES*` armazenado (TS 33.501 6.1.3.2; o AMF é responsável pela verificação de hash HRES*/HXRES*). Ele registra o resultado no UDM (`auth-events`) e sempre retorna `200 OK` com um `ConfirmationDataResponse`:

- Em caso de sucesso: `{ authResult: "AUTHENTICATION_SUCCESS", supi, kseaf }` (K_SEAF como uma string hexadecimal).
- Em caso de falha: `{ authResult: "AUTHENTICATION_FAILURE", supi }`. Observe que uma falha de autenticação é sinalizada no corpo via `authResult`, **não** por um status de erro HTTP.

O contexto de autenticação é deletado após a confirmação, independentemente do resultado.

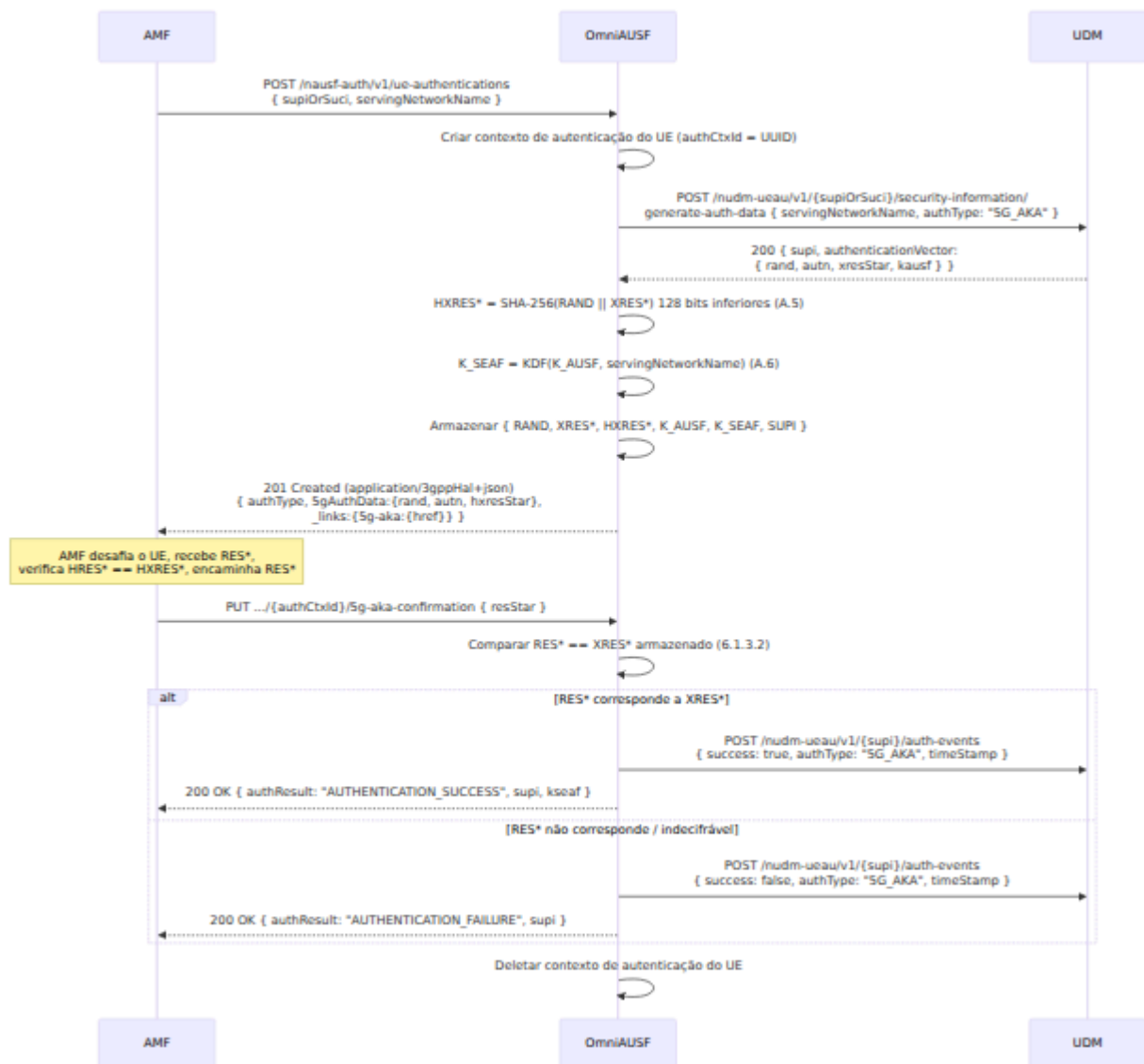
Respostas de Erro

Os erros são retornados como corpos `ProblemDetails`.

Status HTTP	Causa	Condição
400	<code>MANDATORY_IE_MISSING</code>	Um campo obrigatório está faltando (<code>supiOrSuci/servingNetworkName</code> na iniciação; <code>resStar</code> na confirmação; <code>supi</code> na desregistro)
404	<code>USER_NOT_FOUND</code>	O UDM retornou 404 para o assinante durante a geração do vetor
404	<code>CONTEXT_NOT_FOUND</code>	O <code>authCtxId</code> em uma solicitação de confirmação não corresponde a nenhum contexto ativo
500	<code>SYSTEM_FAILURE</code>	UDM inacessível, ou o UDM retornou um vetor de autenticação malformatado

Procedimentos Chave

Autenticação 5G-AKA (Seção 6.1.3 da TS 33.501)



Iniciação. Ao `POST .../ue-authentications`, o AUSF cria um contexto, então solicita um vetor 5G-AKA do UDM em `/nudm-ueau/v1/{supiOrSuci}/security-information/generate-auth-data`, incluindo o id da instância NF do AUSF e qualquer `resynchronizationInfo` fornecido pelo AMF. O UDM retorna um vetor de autenticação (`rand`, `autn`, `xresStar`, `kausf`) e o `supi` resolvido (desconhecido do SUCI). O AUSF calcula `HXRES*` a partir de `RAND` e `XRES*`, deriva `K_SEAF` de `K_AUSF` vinculado ao nome da rede de serviço, armazena o material e retorna o desafio ao AMF. Os campos do vetor são decodificados em

hexadecimal defensivamente - um vetor ausente ou malformado produz um `500 SYSTEM_FAILURE` limpo em vez de uma falha.

Confirmação. Ao `PUT .../{authCtxId}/5g-aka-confirmation`, o AUSF procura o contexto, decodifica `resStar` e o compara diretamente com o `XRES*` armazenado. Ele relata o resultado ao UDM via `auth-events` (esforço máximo; um relatório falho é registrado, mas não altera a resposta ao AMF) e retorna o resultado no corpo. O contexto é então deletado. Um `authCtxId` desconhecido retorna `404 CONTEXT_NOT_FOUND`.

Nome da rede de serviço. Construído a partir de `mcc/mnc` como `5G:mnc<MNC>.mcc<MCC>.3gppnetwork.org` conforme TS 24.501, com MNCs de dois dígitos preenchidos com zeros para três dígitos. O nome da rede de serviço vincula `K_SEAF` à rede de serviço, portanto, o AMF e o AUSF devem concordar sobre a identidade PLMN.

Ressincronização do SQN. Quando o AMF inclui `resynchronizationInfo` (o `RAND` e `AUTS` produzidos pelo UE) na solicitação de iniciação, o AUSF detecta a condição de ressincronização, valida que tanto `rand` quanto `auts` estão bem formados em hexadecimal e retransmite o par validado para o UDM/ARPF na solicitação `generate-auth-data` para que o UDM possa re-derivar o número de sequência (TS 33.501 6.1.3.3). A entrada de ressincronização malformada é descartada e não é retransmitida. O AUSF não calcula o número de sequência, mas não é uma passagem cega: ele inspeciona e valida o blob de ressincronização antes de retransmiti-lo.

Hierarquia de Chaves e Divisão de Derivação (Anexo A da TS 33.501)

A hierarquia de chaves 5G-AKA é dividida entre o UDM e o AUSF. O AUSF nunca vê a chave de longo prazo (K_i) ou CK/IK - ele recebe `K_AUSF` já derivado pelo UDM dentro do vetor de autenticação e deriva apenas a chave âncora `K_SEAF` a partir disso. O AMF (não o AUSF) continua a cadeia até `K_AMF` e as chaves do estrato de acesso.

K_i / OP_c
(chave de longo prazo)

UDM / ARPF

Computado no UDM

CK, IK

$K_{AUSF} = KDF(CK \parallel IK,$
 $SNN, SQN \oplus AK)$
FC 0x6A (A.2)

K_{AUSF}

$K_{SEAF} = KDF(K_{AUSF},$
 $SNN)$
FC 0x6C (A.6)

Computado no AUSF

K_{SEAF}

$K_{AMF} = KDF(K_{SEAF},$
 $SUPI, ABBA)$
FC 0x6D (A.7)

Computado no AMF

K_{AMF}

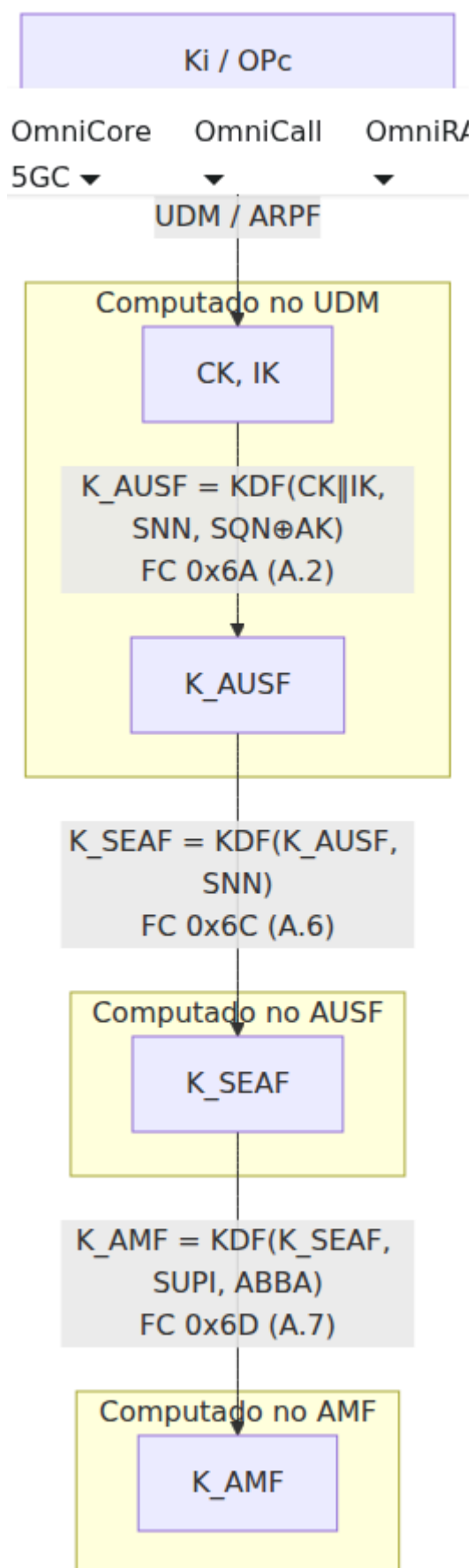
Chave / valor	Onde derivada	Derivação	Propósito no AUSF
K_AUSF	UDM (ARPF)	$KDF(CK \parallel IK, \text{servingNetworkName}, SQN \oplus AK)$, FC 0x6A (A.2)	Recebida no vetor; a chave âncora do AUSF da qual K_SEAF é derivada. Mantida no contexto, nunca retornada ao AMF
K_SEAF	AUSF	$KDF(K_AUSF, \text{servingNetworkName})$, FC 0x6C (A.6)	A chave âncora da rede de serviço. Retornada ao AMF apenas em AUTHENTICATION_SUCCESS; o AMF deriva K_AMF a partir dela
HXRES*	AUSF	128 bits inferiores de $SHA-256(RAND \parallel XRES^*)$ (A.5)	Retornada ao AMF na iniciação para que o AMF possa corresponder ao HRES* do UE antes de encaminhar RES*
XRES*	UDM (ARPF)	$KDF(CK \parallel IK, SNN, RAND, XRES)$, FC 0x6B (A.4)	Recebida no vetor; a resposta esperada que o AUSF compara RES* durante a confirmação

Porque K_SEAF está vinculado ao servingNetworkName (e K_AUSF a montante está vinculado ao mesmo nome de SN e a SQN⊕AK), o AMF, AUSF e UDM devem concordar sobre a identidade PLMN para que as chaves derivadas correspondam às do UE. Um desvio no nome da rede de serviço aparece como uma AUTHENTICATION_FAILURE na confirmação, não como um erro explícito.

Deletar Resultado de Autenticação 5G-AKA (TS 29.509)

O AMF pode deletar explicitamente um resultado/contexto de autenticação sem uma troca de confirmação. Isso mapeia para

`Delete5gAkaAuthenticationResult` na TS 29.509.

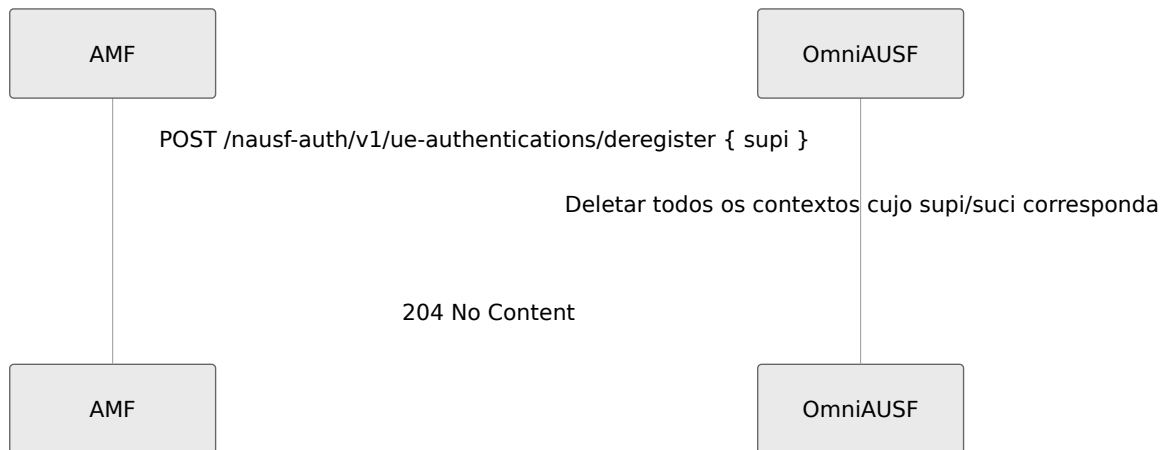


O AUSF deleta o contexto incondicionalmente (um `authCtxId` desconhecido é uma operação sem efeito) e sempre retorna `204 No Content`; nenhuma

chamada ao UDM é feita.

Desregistro (TS 29.509)

Remove qualquer(s) contexto(s) de segurança do UE que o AUSF possui para um SUPI dado.



`supi` é obrigatório (um valor ausente/vazio retorna `400 MANDATORY_IE_MISSING`). O AUSF itera seus contextos em memória e deleta qualquer um cujo `supi` ou `suci` armazenado seja igual ao SUPI solicitado; a operação é local apenas e não chama o UDM. Ele sempre retorna `204 No Content`, mesmo quando nenhum contexto correspondente existe.

Contexto de Autenticação do UE

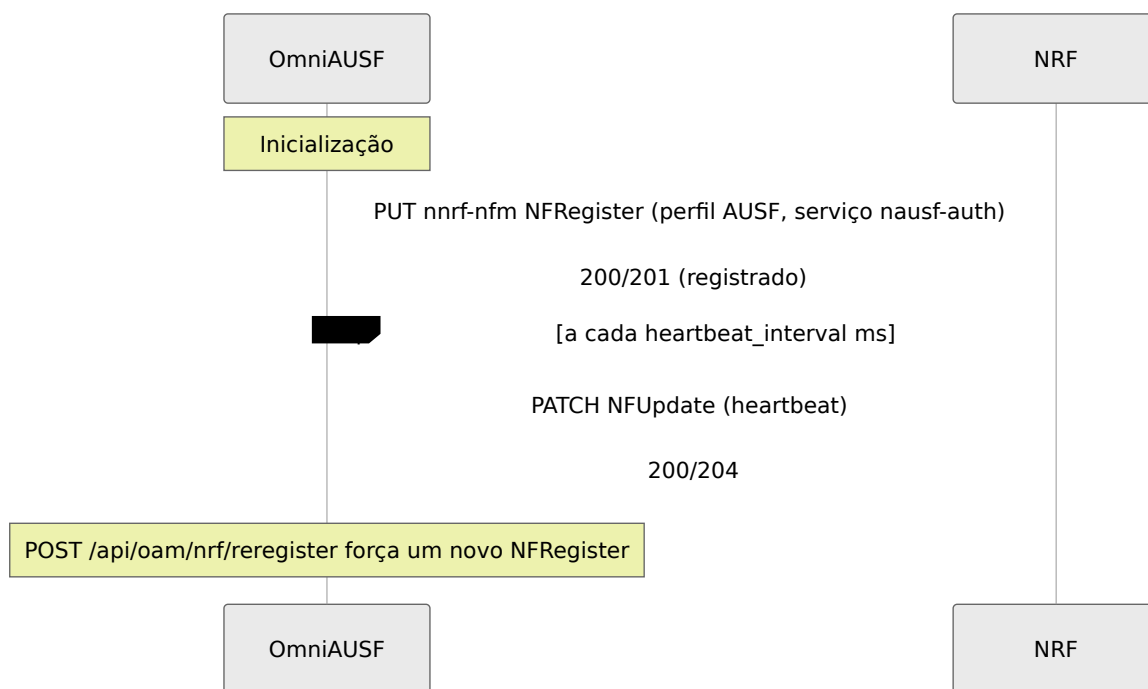
Cada iniciação cria um contexto de autenticação do UE em memória, indexado por um `authCtxId` gerado (UUID). O contexto contém o material necessário para completar a troca e é a unidade que os endpoints da sessão OAM (`GET /api/ausf_sessions`, `DELETE /api/ausf_sessions/{id}`) listam e limpam.

Campo	Definido em	Significado
ctx_id	iniciação	O UUID authCtxId; aparece na URL de confirmação e na lista de sessões OAM
suci	iniciação	O supi0rSuci originalmente apresentado pelo AMF (SUCI ou SUPI)
supi	resposta do vetor	O SUPI simples resolvido (desconhecido) pelo UDM
serving_network_name	iniciação	O nome da SN ao qual as chaves estão vinculadas
rand, xres_star, hxres_star, kausf, kseaf	resposta do vetor	O vetor e as chaves derivadas mantidas durante a troca
auth_type	resposta do vetor	5G_AKA
auth_result	-	Preenchido para uma sessão resolvida; um contexto ainda em andamento não tem resultado e é contado como pending em GET /api/statistics

Um contexto normalmente existe apenas entre a iniciação e a resolução. Ele é deletado na confirmação (sucesso **ou** falha), na DELETE .../5g-aka-confirmation explícita, no deregister para seu SUPI e na reinicialização do processo. Além disso, contextos ociosos ou abandonados são coletados por um coletor periódico supervisionado: um contexto que o AMF inicia, mas nunca confirma, é coletado uma vez que excede context_ttl_seconds (padrão 3600 segundos), com o coletor varrendo a cada context_reap_interval_ms (padrão

60000 ms). Isso limita a vida útil de contextos abandonados sem intervenção do operador. Veja [Contagem de contexto ativo crescente](#) para como inspecionar e limpar contextos manualmente, se necessário.

Registro NRF



Na inicialização, o AUSF registra um perfil NF AUSF com o NRF anunciando o serviço nausf-auth (v1, 1.0.0), seu endereço/porta/esquema SBI e o PLMN configurado. Ele envia batimentos a cada heartbeat_interval milissegundos. O registro pode ser forçado novamente em tempo de execução via POST /api/oam/nrf/reregister. O AUSF também usa o NRF para descobrir o UDM: antes de cada chamada Nudm, ele consulta Nnrf_NFDiscovery (TS 29.510) por uma instância udm e direciona-se ao UDM descoberto. Somente quando a descoberta não gera nenhuma instância (ou o NRF está inacessível) é que ele recorre ao udm_uri configurado estaticamente, portanto, implantações de UDM único e estático continuam funcionando.

Solução de Problemas do OmniAUSF

A autenticação falha com 404 USER_NOT_FOUND

O UDM retornou 404 para o assinante durante `generate-auth-data`. Confirme:

1. `udm_uri` está correto e acessível a partir do host OmniAUSF.
2. O assinante (IMSI) está provisionado no backend UDM/UDR/HSS.
3. O SUCI apresentado pelo AMF está bem formado e pode ser decifrado pelo UDM.

Os logs do AUSF `AUSF auth failed for {supi0rSuci}: user not found at UDM`.

A confirmação retorna AUTHENTICATION_FAILURE (HTTP 200)

O `resStar` decodificado não é igual ao `XRES*` armazenado. Isso é sinalizado no corpo (`authResult: "AUTHENTICATION_FAILURE"`), não como um erro HTTP.

Causas comuns:

- As credenciais do UE (Ki/OPc) não correspondem às provisionadas no backend.
- O RAND/AUTN ou RES* foram alterados em trânsito, ou o AMF/AUSF discordam sobre o nome da rede de serviço (o que produziria um vetor incompatível).

- Um `resStar` malformatado/não hexadecimal é tratado como uma incompatibilidade.

Os logs do AUSF `AUSF 5G-AKA authentication FAILED for {supi}: RES* mismatch`.

A confirmação retorna 404 CONTEXT_NOT_FOUND

O `authCtxId` na solicitação PUT não corresponde a nenhum contexto ativo. Contextos são excluídos após qualquer confirmação (sucesso ou falha) e são perdidos na reinicialização do AUSF. O AMF deve reiniciar a autenticação a partir de `POST .../ue-authentications`.

A iniciação retorna 500 SYSTEM_FAILURE

Ou o UDM está inacessível em `udm_uri`, ou o UDM retornou um vetor de autenticação malformatado (faltando/não hexadecimal `rand`, `xresStar`, ou `kausf`, ou um `autn` ausente). Verifique a conectividade do UDM e se a geração de vetor do UDM está saudável. Os logs do AUSF `AUSF auth failed ...` com a razão, ou `AUSF auth failed: malformed authentication vector from UDM`.

AUSF não descobrível / status NRF mostra não registrado

Verifique `GET /api/status/nrf` e a métrica `omni_ausf_nrf_registration_status`. Verifique se `nrf_uri` está correto (note a ressalva do padrão de tempo de execução na referência de configuração) e acessível, então force o re-registro com `POST /api/oam/nrf/reregister`.

Crescimento da contagem de contexto ativo

Os contextos de autenticação em andamento devem retornar a quase zero em repouso. Contextos são removidos na confirmação, exclusão ou desregistro. Se o AMF abandonar autenticações sem confirmar, esses contextos são automaticamente coletados pelo coletor de contexto periódico uma vez que excedam `context_ttl_seconds` (padrão `3600` segundos); o coletor varre a cada `context_reap_interval_ms` (padrão `60000` ms). Contextos obsoletos, portanto, se limpam sozinhos após o TTL sem intervenção do operador.

Se você precisar inspecionar ou limpar contextos mais cedo, use a API OAM:

`GET /api/status/nf` ou `GET /api/statistics` para as contagens ao vivo, `GET /api/ausf_sessions` para listá-los, e `DELETE /api/ausf_sessions/{id}` (ou `POST /api/ausf_sessions` para limpar todos) para removê-los imediatamente.

O medidor `omni_ausf_active_contexts_count` rastreia essa contagem diretamente (é emitido na criação, exclusão de contexto e em cada varredura do coletor), então você pode monitorá-lo no Prometheus, bem como via API OAM. Se a contagem subir constantemente mais rápido do que o TTL pode drená-la, verifique se o AMF está repetidamente iniciando autenticações que nunca confirma.

