

# IPWorks 3GPP AAA Server-CA OCSP Interface

---

## INTERWORK DESCRIPTION

**Copyright**

© Ericsson AB 2015. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document IPWorks Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Scope	1
1.2	Prerequisites	1
1.3	Related Information	1
<b>2</b>	<b>Interface Overview</b>	<b>3</b>
2.1	Interface Role	3
2.2	Encapsulation and Addressing	3
2.3	Operation Descriptions	4
<b>3</b>	<b>Information Model</b>	<b>5</b>
3.1	Request	5
3.2	Response	6
3.3	Cryptographic Algorithms	9
3.4	Extensions	10
<b>4</b>	<b>Related Standards</b>	<b>11</b>
	<b>Reference List</b>	<b>13</b>





# 1 Introduction

This document describes the OSCP interface between the 3GPP AAA server and ECAS (Ericsson Certificate Administration Server).

## 1.1 Scope

The scope of this document includes the implementation of OSCP (Online Certificate Status Protocol) interface protocol in IPWorks AAA according to RFC 6960.

## 1.2 Prerequisites

Not Applicable.

## 1.3 Related Information

Trademark information, typographic conventions, definition, and explanation of acronyms and terminology can be found in the following documents:

- *Trademark Information*, Reference [1]
- *Glossary of Terms and Acronyms*, Reference [2]
- *Typographic Conventions*, Reference [3]





## 2 Interface Overview

The interface is used by 3GPP AAA server to obtain timely information of the revocation status of certificates from ECAS.

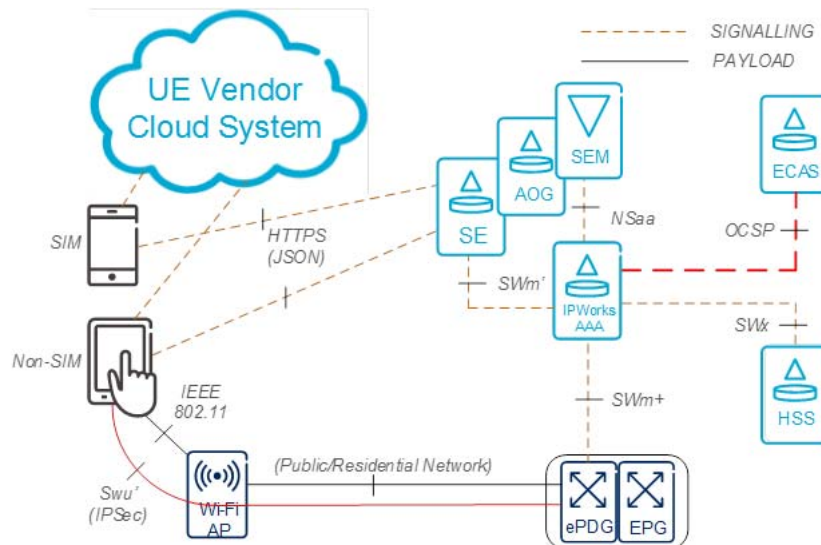


Figure 1 OCSF Interface in NSDS Solution

For OCSF interface, IPWorks AAA only supports some basic procedures as following:

- Send out OCSF request with optional nonce.
- Receive and check OCSF response.
  - Handle Exception Cases.
  - Check time validity.
  - Verify the signature in OCSF response.

### 2.1 Interface Role

In OCSF interface, the IPWorks AAA server takes the role of OCSF client and queries the certificates status from ECAS in Wi-Fi Calling for Multi-Device Solution.

### 2.2 Encapsulation and Addressing

The OCSF interface uses the lower-level protocol:

- TCP
- HTTP

## 2.3 Operation Descriptions

This procedure is triggered when 3GPP AAA server performs the authentication in Wi-Fi Calling for Multi-Device Solution and tries to fetch the certificate status from ECAS.

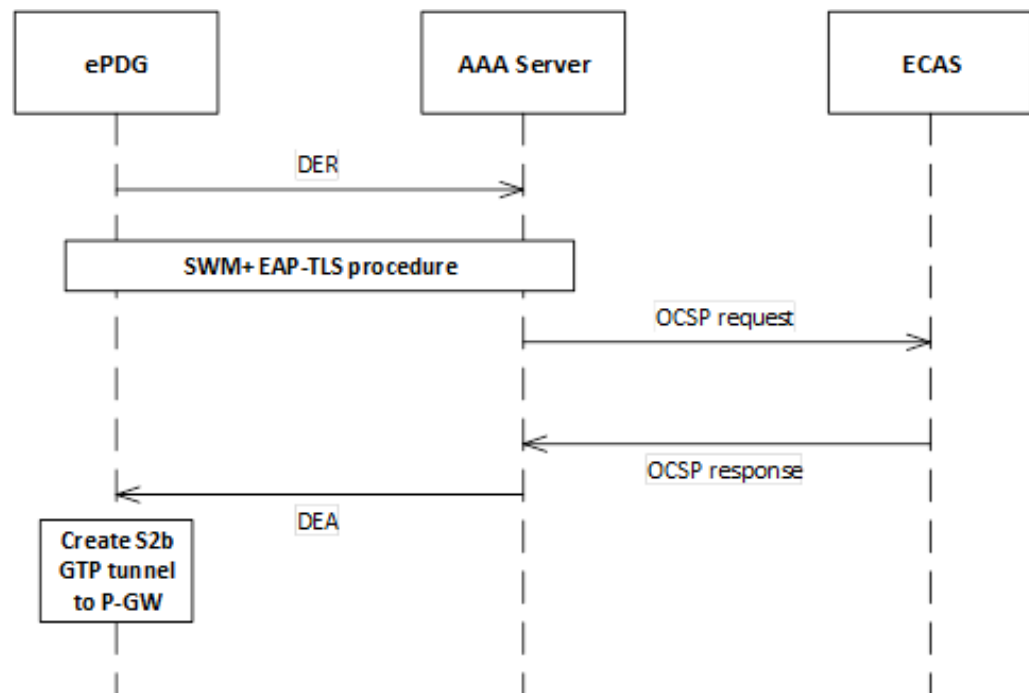


Figure 2 OCSF Checking Procedure





## 3 Information Model

This section describes the OCSP request and response which are supported by IPWorks AAA.

### 3.1 Request

#### 3.1.1 Request Syntax

The ASN.1 structure corresponding to the OCSPResponse is as follows:

```

OCSPRequest      ::= SEQUENCE {
    tbsRequest          TBSTRequest,
    optionalSignature   [0] EXPLICIT Signature OPTIONAL }

TBSTRequest      ::= SEQUENCE {
    version              [0] EXPLICIT Version DEFAULT v1,
    requestorName        [1] EXPLICIT GeneralName OPTIONAL,
    requestList          SEQUENCE OF Request,
    requestExtensions    [2] EXPLICIT Extensions OPTIONAL }

Signature        ::= SEQUENCE {
    signatureAlgorithm    AlgorithmIdentifier,
    signature             BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate
    OPTIONAL}

Version          ::= INTEGER { v1(0) }

Request          ::= SEQUENCE {
    reqCert              CertID,
    singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL }

CertID           ::= SEQUENCE {
    hashAlgorithm         AlgorithmIdentifier,
    issuerNameHash        OCTET STRING, -- Hash of issuer's DN
    issuerKeyHash         OCTET STRING, -- Hash of issuer's public key
    serialNumber          CertificateSerialNumber }
  
```

#### 3.1.2 OCSP Request Implement

An OCSP request contains the following data:

- Protocol version
- Service request
- Target certificate identifier
- Optional extensions, which might be processed by the OCSP responder

IPWorks AAA supports to generate and send OCSP request including some basic information. The IPWorks AAA sends the OCSP request to ECAS

by querying one by one from CAs and until finding the CA corresponding information.

For `requestExtensions`, IPWorks only supports optional `nonce` extension.

**Note:** Currently, IPWorks AAA does not support optional `signature`.

## 3.2 Response

### 3.2.1 Response Syntax

The ASN.1 structure corresponding to the `OCSPResponse` is as follows:

```
OCSPResponse ::= SEQUENCE {
    responseStatus      OCSPResponseStatus,
    responseBytes       [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
    successful          (0), -- Response has valid confirmations
    malformedRequest    (1), -- Illegal confirmation request
    internalError       (2), -- Internal error in issuer
    tryLater            (3), -- Try again later
                        -- (4) is not used
    sigRequired         (5), -- Must sign the request
    unauthorized        (6)  -- Request unauthorized
}
```

The value of `responseBytes` consists of an Object Identifier and a response syntax identified by that OID encoded as an OCTET STRING.

```
ResponseBytes ::= SEQUENCE {
    responseType      OBJECT IDENTIFIER,
    response           OCTET STRING }
```

For a basic OCSP responder, the value of `responseType` is `id-pkix-ocsp-basic`.

```
id-pkix-ocsp          OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic    OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
```

The OCSP responders must be capable of producing responses of the `id-pkix-ocsp-basic` response type. Correspondingly, IPWorks AAA receives and processes responses of this response type.

The value of `response` must be the DER (Distinguished Encoding Rules) encoding of `BasicOCSPResponse`.

```
BasicOCSPResponse ::= SEQUENCE {
    tbsResponseData      ResponseData,
    signatureAlgorithm    AlgorithmIdentifier,
    signature             BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }
```

The value of `signature` must be computed on the hash of the DER encoding of `ResponseData`. The responder might include the certificates in the `certs` field of `BasicOCSPResponse` that help the OCSP client verify the responder signature.

If no certificates are included, the field `certs` must be absent.

```
ResponseData ::= SEQUENCE {
    version                [0] EXPLICIT Version DEFAULT v1,
    responderID            ResponderID,
    producedAt             GeneralizedTime,
    responses              SEQUENCE OF SingleResponse,
    responseExtensions     [1] EXPLICIT Extensions OPTIONAL }

ResponderID ::= CHOICE {
    byName                [1] Name,
    byKey                  [2] KeyHash }
KeyHash ::= OCTET STRING -- SHA-1 hash of responder's public key
(excluding the tag and length fields)

SingleResponse ::= SEQUENCE {
    certID                CertID,
    certStatus            CertStatus,
    thisUpdate            GeneralizedTime,
    nextUpdate            [0] EXPLICIT GeneralizedTime OPTIONAL,
    singleExtensions      [1] EXPLICIT Extensions OPTIONAL }

CertStatus ::= CHOICE {
    good                  [0] IMPLICIT NULL,
    revoked               [1] IMPLICIT RevokedInfo,
    unknown               [2] IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
    revocationTime        GeneralizedTime,
    revocationReason      [0] EXPLICIT CRLReason OPTIONAL }

UnknownInfo ::= NULL
```

**Note:** There is no limit to the number of OCSP responders that IPWorks can manage.

## 3.2.2 OCSP Response Implement

This section describes how IPWorks AAA server handles the OCSP response from OCSP responder.

### 3.2.2.1 Handling Exception Cases

In case of errors, the OCSP responder might return an error message which is not signed. For the detail of handling action, refer to the following Table 1:

*Table 1 Error Handling*

Error Type	Handling Action
malformedRequest	<p>A server produces the <code>malformedRequest</code> response if the received request does not conform to the OCSP syntax.</p> <p>If IPWorks AAA server receives such response, it will reject the authentication request.</p>



Error Type	Handling Action
<code>internalError</code>	<p>A server produces the <code>internalError</code> response if a OCSP responder reaches an inconsistent internal state.</p> <p>If IPWorks AAA server receives such response, it will retry the query on another OCSP responder.</p>
<code>tryLater</code>	<p>If an OCSP responder is operational but unable to return a status for the requested certificate, the <code>tryLater</code> response can be used to indicate that the service exists but is temporarily unable to respond.</p> <p>After receiving such response, IPWorks AAA server will retry the query on another OCSP responder.</p>
<code>sigRequired</code>	<p>The response <code>sigRequired</code> is returned in cases the server requires the client to sign the request to construct a response. Currently, IPWorks AAA does not support to sign the OCSP request.</p> <p>If IPWorks AAA server receives such response, it will reject the authentication request.</p>
<code>unauthorized</code>	<p>The response <code>unauthorized</code> is returned if the client is not authorized to make this query to this server or the server is not capable of responding authoritatively.</p> <p>If IPWorks AAA server receives such response, it will reject the authentication request.</p>

### 3.2.2.2

#### Time Validity

The Responses contains four kinds of time type:

- `thisUpdate`
- `nextUpdate`
- `producedAt`
- `revocationTime`

The `thisUpdate` and `nextUpdate` fields can define a recommended validity interval. This interval corresponds to the `{thisUpdate, nextUpdate}` interval in CRLs.

Following two Response are considered as unreliable response:

- Whose `nextUpdate` value is earlier than the local system time value.
- Whose `thisUpdate` value is later than the local system time value.



If `nextUpdate` is not set, the responder indicates that the newer revocation information is available all the time.

But such interval between the two times might be a few seconds. In practice, the OCSP responder and IPWorks AAA clocks might not be precisely synchronized. Then, such a check might fail. To avoid this, IPWorks AAA allows a graceful error range when it checks the time validity, such time error range is 5 minutes.

### 3.2.2.3 Authorized Responders

The key to sign a certificate status information can be different with the key to sign the certificate. It is necessary to ensure that the entity signing this information is authorized. Therefore, a certificate issuer must do one of the following:

- Sign the OCSP responses itself.
- Explicitly designate this authority to another entity.

During verifying the OCSP response process, IPWorks AAA supports to check the OCSP response signature and the OCSP response certificate of signer. This certificate must be issued directly by the CA that is identified in the request.

**Note:**

- Currently, IPWorks AAA does not support to configure the additional delegation signer certificate.
- Currently, IPWorks AAA does not support additional local configuration to verify the response which is signed by OCSP responder itself.

### 3.2.2.4 Limitation for Revocation Check of an Authorized Responder

According to protocol RFC 6960, IPWorks AAA, as an OCSP client, should know how to check that an Authorized Responder certificate has not been revoked.

**Note:** Currently, IPWorks does not support this functionality.

## 3.3 Cryptographic Algorithms

As an OCSP client, IPWorks AAA server can process responses signed by followed cryptographic Algorithms:

- RSA with SHA-256 (identified by the `sha256WithRSAEncryption` OID specified in [RFC 4055]).
- Clients SHOULD RSA with SHA-1 (identified by the `sha1WithRSAEncryption` OID specified in [RFC 3279])



- Digital Signature Algorithm (DSA) with SHA-1 (identified by the `id-dsa-with-sha1` OID specified in [RFC 3279]).

## 3.4 Extensions

OCSP supports some standard extensions, based on the extension model employed in X.509 version 3 certificates (see [RFC 5280]). Supporting to all extensions is optional for both clients and responders.

**Note:** Currently, IPWorks AAA only supports optional nonce extension.

### 3.4.1 Nonce

IPWorks AAA supports to include optional nonce extension in OCSP request. The nonce cryptographically binds a request and a response to prevent replay attacks. The nonce is included as one of the `requestExtensions` in requests, while in responses it would be included as one of the `responseExtensions`.



## 4 Related Standards

X.509 Internet Public-Key Infrastructure Online Certificate Status Protocol RFC 6090.







# Reference List

## IPWorks Library Documents

- [1] *Trademark Information*
- [2] *Glossary of Terms and Acronyms*
- [3] *Typographic Conventions*

## Standards

- [4] [X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP RFC 6960](#)
- [5] [Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List \(CRL\) Profile RFC 5280](#)
- [6] [Additional RSA Algorithms and Identifiers RFC 4055](#)
- [7] [Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List \(CRL\) Profile RFC 3279](#)