Kiuru AMS Release 4 Product Fact Sheet

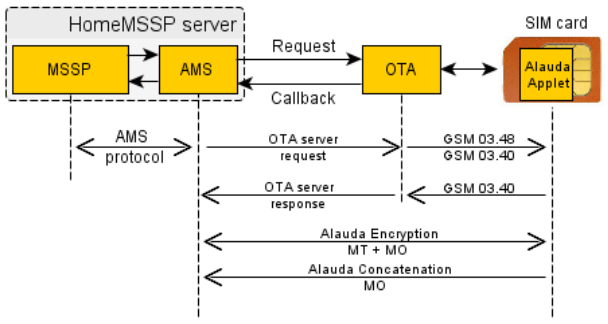
v1.0

Product Description

Kiuru HomeMSSP system has a build-in *Kiuru AMS* (Alauda Messaging Server) module which communicates via OTA/SMSC with an Alauda applet on SIM card.

Introduction

Parties involved in this communication are: HMSSP, AMS, OTA/SMSC, and SIM/Alauda applet.



Kiuru HMSSP produces Alauda protocol request message and submits it to AMS - internal or external. *Kiuru AMS* will assign Alauda Protocol TransactionId, and register callback information. Then *Kiuru AMS* picks up MT encryption counter for the message, and does Alauda Encryption. Finally AMS sends the message to OTA for sending the message to the SIM resident Alauda applet.

When the Alauda applet responds on request it has received, it uses same TransactionId value that was at the request, and *Kiuru AMS* can then use the Id and MSISDN to find out former request and all data recorded at it. Finally *Kiuru AMS* uses that reference information to find out where to deliver the callback: either over *AMS Interface* or internally.

Product Features

AMS Service

- WPKI Client request/response message pairing.
- Message encryption over OTA service.
- May handle MO SMS message concatenation on behalf of the OTA server.
- The Alauda encryption keys and related message counters.
- Alauda key management is integrated with MSSP provisioning tools.

Service Interface

- *Kiuru AMS* uses a SOAP interface allowing also 3rd parties (other HMSSPs) to send *Alauda protocol* requests to SIMs, and receive responses.
- WSDL + XML Schema for SOAP 1.2 service
- Support for WSSE Security extension

OTA server protocol

- The Alauda protocol minimizes OTA server protocol related requirements:
  - MT: The OTA server must provide request id for the request, and send an encrypted and concatenate SMS message to Alauda Applet.
  - MO: The OTA server must receive concatenated SMS response, provide response id, and optionally it may handle message concatenation.
- *Kiuru AMS* can use various proprietary http, SOAP1.2 (W3C) or Corba interfaces (JacORB).

AMS Proxy

- *Kiuru AMS* can be integrated with multiple simultaneous OTA environments. This is important when OTA service is segregated because of service performance or availability reasons.

Clustering

- *Kiuru AMS* is freely clustered meaning the requests and responses are free to go to any cluster member, and the only place with state tracking is the database.

Standards Conformance

- RFC 2246 SSL/TLS support
- Kiuru AMS Interface (alauda-ota-xml-1.0.wsdl)
- AlaudaClient v1.0 RC13

Software Requirements

This is part of *Kiuru HMSSP* installation. Kiuru HMSSP includes required AMS database schema extensions.

For More Information

See also Kiuru MSSP product fact sheet to find out how to build infrastructure for mobile signature services. For more information, call us at

tel.	+358 (0)9 5840 0188
e-mail	methics.sales@methics.fi

About Methics

Methics Ltd is a privately held company, which has developed PKI and wireless PKI products for application providers and operators and built unique security and performance testing tools for operators. The company focuses on security applications and digitizing customer relationship, processes and customer interaction. Kiuru is a registered trademark of Methics Ltd in Finland and other countries.

For the HMSSP server to be able to request services out of the WPKI applet, the OTA communication is needed. Although following description assumes classical SMS mediated messaging, also BIP is allowed.

Mobile Terminating (MT) direction

1. HMSSP composes a binary request (complete Alauda Protocol message) to be sent to the applet
2. HMSSP sends that request to internal helper service called "AMS" (Alauda Messaging Server) that will then:
   - Find Alauda Transport Encryption keys and counters for the particular SIM (there can be up to 16 key pairs and counter pairs for this)
   - Increment MT counter by one
   - Encrypt the request message with new counter value, and the MT key
   - Assign the request a new Transport Transaction ID
   - Record the outgoing request with <MSISDN, TransID> key, placing there callback reference information for response processing
   - Send the encrypted binary request to OTA
3. OTA server adds Applet related details like TAR and SMSC required details
4. OTA server sends the binary request to SIM at MSISDN
   - with 03.48 encryption + 03.40 segmentation
5. SIM receives the 03.40/03.48 request, and recognizes it to be processed by the Alauda Applet
6. The Alauda Applet receives the request message:
   - Decrypt the request content if it is using Alauda Transport Encryption

Mobile Originated (MO) direction

1. The Applet uses request originating SMSC and originating MSISDN as target for the responses (this may vary)
   - The service site MSISDN does not need to be hard-coded into SIMs for the Alauda Service to work
   - The service site MSISDN needs be coded on possible supporting SIM-menu service, which can send user originated service requests (change my Language, change my NoSpamCode, ...) to specified service MSISDN, which need not to be the same as OTAs use for applet communication
   - The Applet produces a response dataset in an internal buffer, wraps it with Alauda Transport Encryption using request key's MO direction version, and using key specific MO counter
   - Then the Applet uses 03.40 segmentation procedure to split the response to fitting bits for SMS transport, and sends them to request's MSISDN at request's SMSC (This needs an 8 bit counter separate from everything else.)
2. The OTA server receives these segments, and uses 03.40 concatenation procedure to combine them
   - When the full 03.40 concatenation result is available, and the OTA recognizes it as the Alauda Protocol response (probably by receiving MSISDN number)
   - OTA server sends it to configured AMS server URL address (aka callback)
   - Optionally the OTA sends the original response segments as is to AMS (HMSSP), which does the concatenation.
3. The AMS (HMSSP) server receives the response data
   - If it is in Alauda Transport Encryption wrapper, appropriate decrypt with MO counter validation is done
   - If the MO counter is not at least +1 of previous value, and at most +200 of previous, the response is acceptable and is processed onwards
   - The responses carry Transport Transaction ID, and now the AMS looks up <MISSDN,TransID> to find which system wants the response, and what supplemental transaction reference data needs to be supplied
4. The HMSSP receives the decrypted response along with original request contained reference data

OTA server protocol

The protocol needing to be defined on per OTA basis is called as the "OTA server protocol". Services needed for the OTA server protocol are:

1. **MT**
   Submit binary applet request to the MSISDN (OTA or AMS can add possible other parameters like TAR). Optional functions:
   - Submit TEXT messages (normal and "flash" message) to MSISDN
   - Cancel the delivery of previously submitted request
   - Query request submission status

2. **OTA**
   Send delivery success/failure notification to the AMS server.

3. **MO**
   Deliver received (and concatenated) response message to URL configured for AMS. Optionally the original segments can be delivered to AMS. AMS can do the 03.40 concatenation.