OMA-DLDRM-2006-0525R02-
INP_Presentation_of_Consolidated_AKA_Scheme

Consolidated AKA Scheme

Submitted To: OMA-BAC-DLDRM

Date: 08 Dec 2006

Availability: [X] Public [ ] OMA Confidential

Contact: Hagai Bar-El, hbarel@discretix.com

Source: Discretix Technologies Ltd.

USE OF THIS DOCUMENT BY NON-OMA MEMBERS IS SUBJECT TO ALL OF THE TERMS AND CONDITIONS OF THE USE AGREEMENT (located at http://www.openmobilealliance.org/UseAgreement.html) AND IF YOU HAVE NOT AGREED TO THE TERMS OF THE USE AGREEMENT, YOU DO NOT HAVE THE RIGHT TO USE, COPY OR DISTRIBUTE THIS DOCUMENT.

THIS DOCUMENT IS PROVIDED ON AN "AS IS" "AS AVAILABLE" AND "WITH ALL FAULTS" BASIS.

Intellectual Property Rights

Members and their Affiliates (collectively, "Members") agree to use their reasonable endeavours to inform timely the Open Mobile Alliance of Essential IPR as they become aware that the Essential IPR is related to the prepared or published Specification. This obligation does not imply an obligation on Members to conduct IPR searches. This duty is contained in the Open Mobile Alliance application form to which each Member's attention is drawn. Members shall submit to the General Manager of Operations of OMA the IPR Statement and the IPR Licensing Declaration. These forms are available from OMA or online at the OMA website at www.openmobilealliance.org.

Overview

This presentation describes the consolidated AKA Scheme that is specified in **OMA–DLDRM–2006–0474**.

AKA Authentication and Key Exchange. The process that allows each device[1] to assure the compliance of the other device (by identification), and to establish a shared key that is used to secure the transport.

consolidated Joint work by all interested individuals, using elements of several proposed schemes.

[1] 'device' = 'Device' or SRM

Threats

The threats to the system are as following:

- A user (legitimate or not) obtains content keys
- A user manipulates state values that are sent between the devices, such as:
  - play counters
  - move counters
- A user takes advantage of a protocol flaw to effectively duplicate rights
  e.g. by replaying an exchange with a device that is no longer physically present.
- A user introduces *fake* rights into a device

Security Objectives

The proposed scheme should thus achieve:

- Confidentiality of objects against content key leakage
- Authentication of objects against impersonation of devices or of rights
- Integrity assurance for objects against data alteration
- Replay-protection of objects transport against impersonation of devices
- Must be renewable
  Must support newer versions with protection against rollback.

Required Features

The protocol must also provide these features:

- Separation between the *initiator* and the *sender*. When connecting to an SRM, the device is always the initiator, even when it is the entity receiving the RO[2].
- Ability to use one exchange for sending several ROs. The key exchange may take a while. We don't want to wait this while per each RO sent...
- Security association shall be trust model specific

[2] The term 'RO' here refers to Rights, encapsulated in some data object; not to the RO object in ROAP.

Resource Considerations

The protocol should also...

- Use the least number of messages possible
- Perform the least crypto operations possible
- Reuse existing primitives, where possible
- Be as simple as possible (code size)

Trust Models

- A *trust model* of an RO is the head of the certificate trust chain that attests for the origin of the RO.
- The trust model is indicated by the hash of the root key (the “self–signature” key).
- A trust model applies also for devices (incl. SRMs). They too have certified keys with certificate chains.
- A security association that is used to secure an RO delivery must be based on credentials that are of the same *trust model* as the one of the RO.

  E.g., when RO from RA *A* is transferred, security negotiation shall be done with device credentials issued by RA *A*.

Method of Introducing Security

Securing the existing protocol

- Security is added by encrypting message bodies. No new transport protocol is defined.
- Two sub–protocols are defined: *AKA* and *Reuse*

  AKA Establishes a *cryptographic context* for a particular trust model, and uses it.

  Reuse Reuses an existent cryptographic context.
- Cryptographic contexts are generated once per session per trust model, and are cached throughout the session.
- Each time RO information needs to be transferred, the relevant context data is used to protect the information.

Cryptographic Contexts

- Each execution of the AKA protocol results in an agreed *Cryptographic Context* that consists of:
  - A shared key
  - An HMAC key
  - A counter prefix
- The context data is specific for the trust model with which the AKA was carried out. It is never shared.
- The cryptographic context is cached until the devices are disconnected.
- Whenever RO information of that trust model needs to be transferred, the context data is reused. . . and then updated. . .

Control Options

- The protocol negotiates Control Options between the parties
- The *Control Options* structure is a list of options, sent by the initiating party, for:
  - Supported protocol versions (e.g. `(1.0)`)
  - Supported cipher-suites, each one containing
    - an algorithm (only AES for now), and
    - a mode (CBC and CTR are supported for now)
- The *Chosen Options* structure is a subset of *Control Options* in which the other party chooses one of each option category
  E.g. `V(1.0),AES,CTR`
- The supported and selected options are used in a way that prevents roll-back attacks

AKA Protocol

AKA Protocol

1. $A \rightarrow B \quad Chain_A, ControlOpt$
2. $A \leftarrow B \quad Chain_B, E(Pu_A, (R_B|ControlOpt|ChosenOpt))$
3. $A \rightarrow B \quad E(Pu_B, (R_A|R_B))$
4. $A \leftarrow B \quad H(R_A|R_B)$

Description

1. A's chain (by relevant trust model), supported versions and cipher-suites
2. B's chain (same trust model as of A's), B's random, and selections
3. Encrypted A's random, and also B's for verification
4. Hash of randoms, as key confirmation

Generation of Cryptographic Context

Cryptographic Context data is computed in two steps:

1. Key material is generated like this:
   $KM = KDF(R_A, R_B, ControlOpt, ChosenOpt)$

   KDF is the Key Derivation Function defined in the DRM V 2.0 specification.

   R_A and R_B form Z, and the options structures form *otherInfo*.
2. Context data is derived from *KM* as follows:
   - *Key* $\leftarrow$ 128 bits of *KM*
   - *HMACKey* $\leftarrow$ 160 bits of *KM*
   - *CtrA* $\leftarrow$ 96 bits of *KM*

Purpose of the Reuse Protocol

- Once AKA is done, a cryptographic context is available for the trust model.
- (Multiple AKA executions may occur for multiple trust models.)
- If another RO (of a trust model for which AKA was done) needs to be transferred...
- ...there is no need to run AKA again. The *Reuse* protocol is called instead; to save time.
- The Reuse protocol is simple:
  - The initiator specifies the trust model of which existing context data shall be reused.
  - The responder either agrees to reuse, or requests that AKA is run once again for this trust model.

Reuse Context Protocol

Reuse Context Protocol

1. $A \rightarrow B \quad TMID$
2. $A \leftarrow B \quad [ACK|NACK]$

Description

1. The trust model ID (hash of public key of the root RA)
2. Acknowledged or not. If Not — a new AKA execution is triggered for *TMID*.

Questions?

Questions?