Security Common Functions Requirements

Candidate Version 1.0 – 08 Aug 2006

Open Mobile Alliance

OMA-RD-SEC_CF-V1_0-20060808-C

Contents

Figures

Tables

1. Scope (Informative)

The Security Common Functions (SEC_CF) will provide common security functions for OMA enablers. These functions shall not be specific to any particular application. The SEC_CF architecture will provide a common way to specify security functionality for different enabler deployment scenarios.

SEC_CF will be accompanied with several Technical Specifications (TS). While the Architecture Document (AD) intends to describe the high level architecture of the SEC_CF, and to provide architectural guidance for different enabler deployments, details of the security functions will be provided in separate technical specifications.

All these specifications shall be developed such that they can be applicable to any OMA protocols, i.e. they shall not be specific to any specific application. Specifications for protocols will either reference SEC_CF specifications, or include building blocks provided by the SEC_CF, or are designed according to guidelines that are provided by the SEC_CF.

This document, the Requirements Description (RD), will identify the requirements for the Security Common Functions. It will describe, in a generic way, for which entities and under which conditions identification, authentication, confidentiality and integrity are to be provided in protocols developed by OMA working groups.

The requirements for the SEC_CF will be developed from use cases as described later in this document. These use cases will, however, be more examples of applications of different security solutions than detailed and explicit application use cases. Security requirements as identified for already specified OMA enablers, such as e.g. Secure User Plane Location (SUPL), will be taken into consideration. Further requirements, however, will be added where necessary.

Development of Common Security Functions is an ongoing process as requirements of OMA enablers change. SEC_CF development will be phased into various enabler releases. Details of the defined phases and the relevant requirements can be found in Section 5.

The first enabler release of SEC_CF will provide security architectures based on the commonly deployed security architectures in the mobile industry. This is done in order to ensure maximum re-use of the existing security mechanisms that are already deployed. Any mechanisms that need further developments such as defining new security protocols or require adaptations to be used by the mobile industry are considered to be addressed in the future releases of SEC CF.

2. References

2.1 Normative References

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[OSE] "OMA Service Environment", Approved Version 1.0 – 07 Sep 2004 OMA-Service_Environment-V1_0-20040907-A

<< Add/Remove reference rows as needed! >>

2.2 Informative References

[Handbook] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone: Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7, Fifth Printing, August 2001

[DDOS] "Recommendations for the Protection against Distributed Denial-of-Service Attacks in the Internet", Bundesamt für die Sicherheit in der Informationstechnik, 2000. URL: http://www.iwar.org.uk/comsec/resources/dos/ddos_en.htm

[ETR 232] Security Techniques Advisory Group (STAG): Glossary of security terminology, ETSI Technical Report 232, November 1995

[RFC2828] R. Shirey: Request for Comments 2828: Internet Security Glossary. May 2000

[RFC3365] J. Schiller: Request for Comments 3365: Strong Security Requirements for Internet Engineering Task Force Standard Protocols. August 2002.

[RFC3552] E. Rescorla, B. Korver; Internet Architecture Board (IAB): Request for Comments 3552: Guidelines for Writing RFC Text on Security Considerations July 2003

[GBA] 3GPP TS 33.220 "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 6) "

URL: http://www.3gpp.org/ftp/Specs/html-info/33220.htm

[PSK-TLS] "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", IETF RFC 4279, December 2005

URL: http://www.ietf.org/rfc/rfc4279

[HTTP Digest] "HTTP Authentication: Basic and Digest Access Authentication", IETF RFC 2617, June 1999

URL: http://www.ietf.org/rfc/rfc2617

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Definitions

Anonymity

Anonymity provides protection of the identity of a party, against both eavesdroppers and peers

- **Identity Protection against Eavesdroppers**:
  An attacker (eavesdropper) should not be able to link the communication exchanged by one party to the real identity of the party.
- **Identity Protection against Peer**:
  The peer in a communication should not be able to link the communication exchanged by one party to the real identity of the party, but rather to an unlinked pseudonym or private identifier.

Authentication

Authentication is the process of verifying an identity (distinguishing identifier) claimed by or for a system entity, which may be a peer in a communication or the source of some data. This assured Identity may be well known (a real name, telephone number, mailing address, phone number, social security number, IP- or email address) or it can be an unlinkable identifier (like a pseudonym). The verification is achieved presenting authentication information (credentials) that corroborates the binding between the entity and the identifier. Authentication is usually divided into entity and message (or data) authentication. The main difference between the two is that message authentication provides no timeliness guarantee (the authenticated message may be old), while entity authentication implies actual communication with an associated verifier during execution of the current run of the protocol.
Authentication is usually unilateral (“Alice authenticates Bob”). **Mutual Authentication** refers to Authentication in both directions.

Authorization (by a Trusted Third Party)

Authorization is a right or a permission that is granted to a system entity to access a system resource. An "authorization process" is a procedure for granting such rights. In some protocols, a Trusted Third Party introduces one principal to another one, and assures to the first one that the second one is trusted and authorized to access the service or function.

Data Confidentiality

Data Confidentiality is the property that a particular data item or information (usually sent or received as part of the content of a “secured” message, or else constructed on the basis of exchanged data) is not made available or disclosed to unauthorized individuals, entities, or processes, and remains unknown to the intruder. We choose the convention that the secrecy of a session key generated during a key agreement is not considered here but in Goal “Key authentication” above. Also the secrecy of a long-term key used within a protocol is not part considered as a secrecy goal of the protocol.

Data Integrity	**Data Integrity** is a security service that protects against unauthorized changes to data, including both intentional change or destruction and accidental change or loss, by ensuring that changes to data are detectable. A data integrity service can only detect a change and report it to an appropriate system entity; changes cannot be prevented unless the system is perfect (error-free) and no malicious user has access. However, a system that offers data integrity service might also attempt to correct and recover from changes. Relationship between data integrity service and authentication services: Although data integrity service is defined separately from data origin authentication service and peer entity authentication service, it is closely related to them. Authentication services depend, by definition, on companion data integrity services. Data origin authentication service provides verification that the identity of the original source of a received data unit is as claimed; there can be no such verification if the data unit has been altered. Peer entity authentication service provides verification that the identity of a peer entity in a current association is as claimed; there can be no such verification if the claimed identity has been altered.
Denial-of-Service (DoS)	Denial of Service attacks target the valuable resources that are needed to provide services. A typical denial of service attack results in the excessive usage of a particular resource by a malicious entity in order to make that resource unusable for the rest of the legitimate users of the service. Below are few examples of DoS attack types: – DoS on memory allocation, – DoS on computational power, and – Overloading attacks on third parties: This is inducing one or several hosts to send large amounts of packets to a victim.
Entity authentication (Peer Entity Authentication)	**Entity authentication** is assuring one party, through presentation of evidence and/or credentials of the identity of a second party involved in a protocol, and that the second has actually participated during execution of the current run of the protocol. Usually this is done by presenting a piece of data that could only have been generated by the second party in question (as a response to a challenge, for instance). Thus, usually entity authentication implies that some data can be unequivocally traced back to a certain entity, which implies Data Origin Authentication.
Identity Module	A fixed or removable module keeping identity information and credentials, i.e. a SIM/USIM/ISIM or UIM/RUIM
Key Agreement	– An authenticated key agreement protocol has as goal the secure distribution of keys, and in particular most often session keys.
Message authentication (Data Origin Authentication)	The protocol must provide means to ensure confidence that a received message or piece of data has been created by a certain party at some (typically unspecified) time in the past, and that this data has not been corrupted or tampered with, but without giving uniqueness or timeliness guarantees. The confidence that data has been created by a certain party, but without the assurance that it has not been modified, is of no interest for us. Thus Message authentication implies integrity. Only very few Internet protocols offer Data Origin Authentication without providing Entity Authentication (IPsec AH or PKI Signatures would be examples).

Privacy Privacy is the right of an entity (normally a person), acting in its own behalf, to determine the degree to which it will interact with its environment, including the degree to which the entity is willing to share information about itself with others. (See: anonymity.)
In particular, privacy is the right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Public-key cryptography **Public-key cryptography** (also called asymmetric cryptography) is based on using a pair of two different keys (a public and a private key. A public key is called "public" because it is generally available to everybody and may be used either to encrypt messages intended for the owner of the corresponding private key or to verify the signature of that owner. Corresponding to the public key is a private key, typically known only to one principal. The private key is used to decrypt the message. Because it is uniquely bound to an individual a private key can also be used for a digital signature on a message. But often, for security reasons, different keys and different algorithms are used for decryption and digital signatures. In order to use a public key, the entity using it has to know which principal is bound to the public key. This binding is usually accomplished by a certificate, typically a record asserting such binding, containing an indicator of timeliness and signed by a well-known trusted third party.

Replay Protection In a replay attack, the attacker captures one or several messages plays them back to the party which originally received them. The attacker does not need to be able to understand the messages. A protocol provides Replay protection if it offers means to ensure confidence that a received message has not been recorded and played back by an adversary".
More precisely, replay protection is assuring one party that an authenticated message is not old. Depending on the context, this could have different meanings:

- that the message was generated during this session, or
- that the message was generated during a known recent time window, or
- that the message has not been accepted before.

Symmetric-key cryptography **Symmetric-key cryptography** (also called secret-key cryptography) relies on the same key for both encryption and decryption.

3.3 Abbreviations

IM Identity Module

OMA Open Mobile Alliance

4. Introduction (Informative)

4.1. Background and history of SEC_CF

OMA enablers typically comprise of protocols, such as MLP (Mobile Location Protocol), RLP (Roaming Location Protocol), PCP (Privacy Checking Protocol) in the Location enabler, SSI (Server-to-Server Interface) and CSI (Client-to-Server Interface) in the Presence enabler, SyncML in Device Management, or PAP (Push Access Protocol), which is used by multiple enablers. While the structure and the intended use of the protocols are diverse, there are a number of common features that occur repeatedly in most of them. Among these common features, mechanisms for the following security features are particularly important:

– identification and authentication of entities

– confidentiality

– integrity

– accountability (proof of origin, proof of delivery, proof of receipt)

The OMA Architecture working group (OMA-ARC), as an activity under the GOAL work item (Gaps and Overlaps Analysis, OMA work item #88), compared protocols in different OMA enablers (see OMA-ARC-2004-0018/OMA-SEC-2004-0010). The analysis showed that:

- There are multiple protocols where the endpoints need to identify and authenticate each other and where the endpoints need to protect the confidentiality and the integrity of protocol messages as they are exchanged between the endpoints,
- Different OMA enablers use different role models to categorize the actors that are involved when the enabler is being used. The different role models result in different approaches to solve identification, authentication, confidentiality and integrity issues,
- Organizations, other than OMA, are addressing the security aspects of protocols; they provide sophisticated solutions to security issues and have already achieved considerable awareness in the market

The current practice of addressing security features specifically in each OMA enabler creates the following problems:

- There is a significant amount of duplicated efforts being carried out by OMA working groups to address similar security concerns.
- Application developers using OMA enablers need to understand and use different ways to do the same thing.
- Vendors providing multiple enablers in their portfolios need to implement different solutions for the similar problems (one per protocol), rather than re-using security common functions across multiple protocols.
- There is a risk that the security features of OMA protocols are behind current state of the art, because the creators of the protocols are typically subject matter experts for a particular enabler, but not for security (e.g. they might lack knowledge about cryptography, or threat modelling).
- Since the security of a system is as strong as its weakest link, it is imperative that each enabler implements the best security standards, something that may be difficult to achieve if each working group is independently developing security enablers.
- Mobile Operators and Service Providers are finding it increasingly difficult to implement an array of different security solutions which are application specific that are solving similar security vulnerabilities, a more centralised approach is encouraged as suggested in this work item.

This document, the Requirement Description (RD) will identify the requirements for the SEC_CF. The resulting specifications will comprise a Common Function in the sense of WI #0062 (Interfaces to Common Functions). They will describe, in a generic way, how identification and authentication, confidentiality, integrity and accountability are to be provided in protocols developed by OMA working groups.

4.2 Details regarding the considerations in this document (RD)

This requirement specification aims at identifying key requirements on security architectures for OMA enablers using a client server operational model, and TCP as the transport protocol. By working with generic communication solutions for OMA enablers in general, the nature of the use cases will also be generic and abstract.

Even the simple case of establishing a secure connection for a client server application exhibits quite a number of options and conditions, which have to be taken into account when deciding on valid solutions. Factors considered here are

- Required **authentication mode**

Mutual authentication: The client should be able to authenticate the server to know that it connects to the intended server, and the server should be able to authenticate the client (user) to and verify authorization and give access to services.

Server authentication only: The client should be able to authenticate the server to know that it connects to the intended server but the server does not need to authenticate the client (user). A situation like this may occur when privacy and/or the integrity of the communication need protection but the service offered is open for everyone.

- Type of **credentials** that are used by the client and the server. For **mutual authentication** we recognize the following three types

Shared secret: A secret shared by the enabler and the client is used to establish a PSK-TLS session. The sharing of the secret key is a basis for mutual authentication of the endpoints. One example where shared secrets are used is when the client and the server use GBA keys with PSK-TLS.

Shared secret & server certificate, which is the most common method employed on the Internet. The client verifies the authenticity of the server in the TLS set-up by checking the server's certificate. The server uses some specific method based on a username and a shared secret (used as a password) to authenticate the user and authorize him for service access. An often used method for user authentication is HTTP Digest.

Client & server certificates: In the TLS set-up the client verifies the authenticity of the server by checking the server certificate and the server checks client certificate to authenticate the user. Authorization of user access to the services offered has to be performed on application level.

For **server authentication only**, we only consider

Server certificate, the client verifies the authenticity of the server by checking the server's certificate.

- Mode of **deployment**, which indicates the type of network domain relations between the involved entities, which in turn usually also implies the trust relations at hand.

The **client** (requestor) can be in a mobile terminal associated with a home network or in some cases it can be located in a network server. The **server** may be located in the **home network**, a **visited network** when the client is roaming, or in an **external network** (on the Internet).

Connections for payload traffic can be either **direct** between the terminal and the server or **mediated** via a proxy/gateway function.

Credentials for direct connections can be long term, short-lived based on direct recognition/sharing of credentials between the client and the server, or on credentials generated and distributed by a trusted party.

- **Session initiation** is about which party initiates the connection setup and the session.

Terminal initiated connections/sessions are when the user/client initiates a session to access a service. Normally there are no specific denial-of-service threats related to cases when users initiate sessions.

Network initiated connections/sessions are connections initiated by an entity in the network. To accomplish the connection establishment the initiating party usually sends a PUSH message to the client requesting it to set up a connection to the service. The PUSH message mechanism has to be protected to counter denial-of-service and replay attacks.

5. Use Cases (Informative)

The use cases described below will be more of example applications of different security solutions than detailed and explicit application use cases. These solutions are not enabler services by themselves; they are architectural components to be used by the enablers.

The first two cases cover requirements for shared key (e.g. smartcards) based security mechanisms.. The third case introduces client certificates for key management The fourth case handles the case when a component of the enabler helps in the establishment of a secure connection between entities not having a common trust base. The fifth use case looks at issues when there is a network initiated use of an enabler. The sixth and seventh use cases deal with provisioning of security parameters and secret keys respectively.

5.1 Shared Key based Security Establishment

5.1.1 Short Description

A client in a Mobile Terminal establishes a secure connection to an enabler in its home network where the mobile terminal has pre-established credentials. Credentials and their use for the establishment of the secure connection are based on a shared key mechanism such as the Generic Bootstrapping Architecture [GBA]. The connection is either protected with PSK-TLS [PSK-TLS], using a shared key for mutual authentication of the endpoints or TLS 1.0 [TLS] with server certificates for server authentication and a shared key HTTP Digest [HTTP DIGEST] for client authentication.

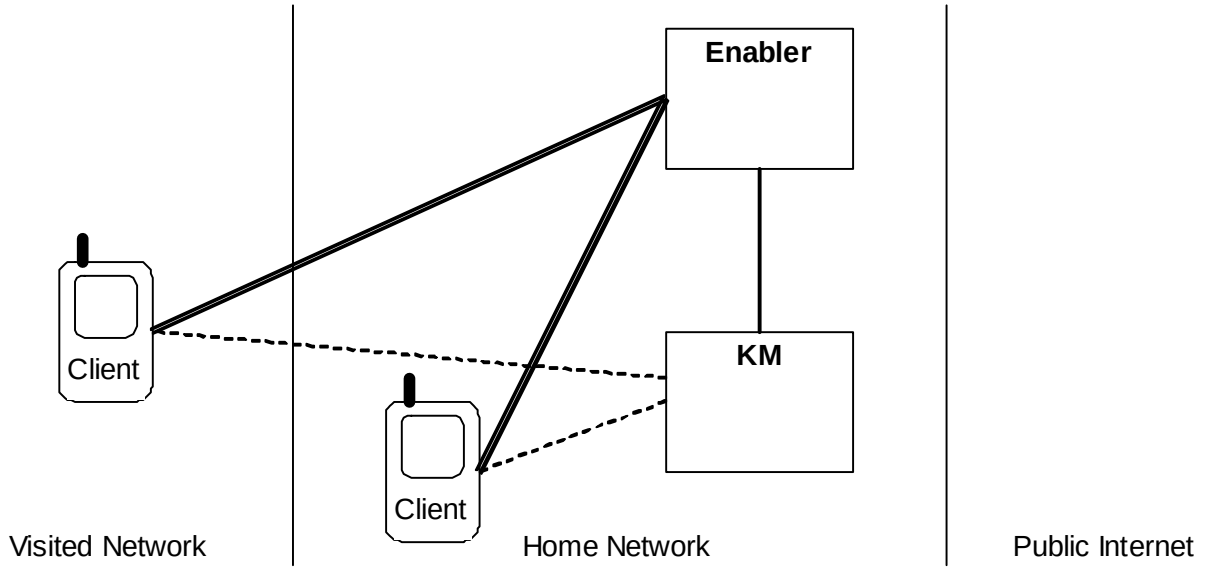

Figure 1: Secure access to OMA enabler using shared key based key management

5.1.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the Mobile Terminal (MT).
- The home network operator. The home network operator runs

  - The enabler function that performs the authentication e.g. in a GBA context is a Network Application Function (NAF)
  - The Key Manager, which performs key generation, management and distribution e.g. in a GBA context is the Bootstrapping Server Function (BSF).
- Possibly a visited network operator. The operator of the visited network is passive and only provides connectivity between the visited and the home network.

5.1.2.1 Actor Specific Issues

The MT and the home operator have to support a common shared key based mechanism such as GBA functionality.

5.1.2.2 Actor Specific Benefits

It is essential for users as well as home network operators that users can be offered secure access to services in the their home networks.

5.1.3 Pre-conditions

The MT can establish TCP connections to the Key Manager and the Enabler. In 3GPP networks, roaming terminals usually have their point of presence in their home networks, which would guarantee that both the Enabler and the Key Manager could be reached from the MT. However, it is sufficient that the MT can directly address the Key Manager and the Enabler and establish a TCP connection. This could always be achieved, even with NAT(P)s in the path, if the Key Manager and the Enabler interfaces had public IP addresses.

Here, it is of course assumed that the Enabler is allowed to use the enabler's key management/authentication functionality. We only note that operators most likely will set up policies governing which enablers that implement the authentication functionality.

5.1.4 Post-conditions

A TLS protected connection between the MT and the enabler exists. The end points have been mutually authenticated.

5.1.5 Normal Flow

The MT connects to the Key Manager (e.g.BSF) to retrieve a shared key (e.g. GBA key). The MT then connects to the Enabler and initiates a PSK TLS session, indicating that the key to be used is the retrieved the shared key. The Enabler connects (securely) to the Key Manager and retrieves the indicated shared key together with end-user identity information (anonymous use may be allowed). The shared key is then used in PSK-TLS to establish the payload data protection.

5.1.6 Alternative Flow

The MT connects to the Key Manager (e.g. BSF) to retrieve a shared key. The MT then connects to the Enabler and initiates a TLS session. The Enabler authenticates itself with a server certificate and requests client authentication with HTTP digest using the agreed/established shared key. The client should validate that the certificate of the Enabler. The Enabler connects (securely) to the Key Manager and retrieves the indicated Enabler (NAF) specific key to be used in the HTTP digest authentication.

5.1.7 Operational and Quality of Experience Requirements

The establishment of the secure connection should be automatic and invisible to the end-user.

5.2 Performing Authentication Using an Authentication Proxy

5.2.1 Short Description

The home operator runs several Enablers that are accessible via HTTP. The home operator uses a common Authentication Proxy (AP) for mutual authentication between enablers and clients. The protected connection between MT and Enabler is terminated in the AP.

A client in a MT establishes a secure connection to the AP in its home network. Credentials and their use for the establishment of the secure connection are based on a shared key management mechanism. The connection is either protected with PSK-TLS [PSK-TLS], using a shared key for mutual authentication of the endpoints or TLS 1.0 with server certificates for server authentication and a shared key HTTP Digest for client authentication.

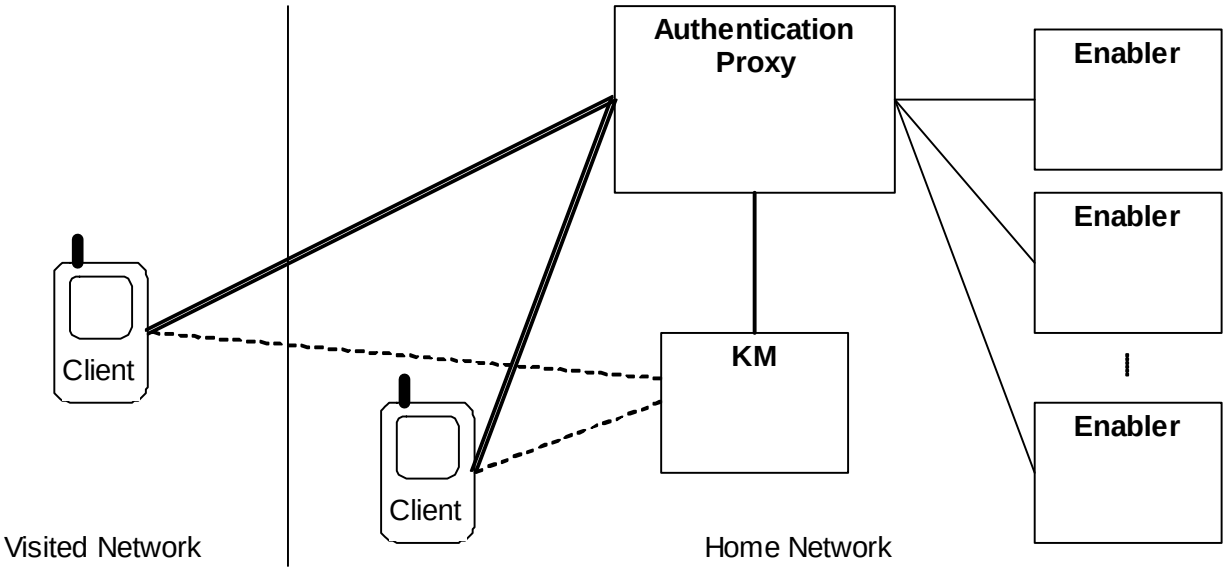

Figure 2: Access to Enablers via Authentication Proxy

5.2.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the MT.
- The home network operator. The home network operator runs
  - The Application Proxy
  - The Key Manager
- Operators of enablers. Usually it is the home network operator that will run the enablers but it is also possible to have 3rd party enablers.
- Possibly a visited network operator. The operator of the visited network is passive and only provides connectivity between the visited and the home network.

5.2.2.1 Actor Specific Issues

The MT and the home operator have to support a shared key based key management mechanism.

5.2.2.2 Actor Specific Benefits

It is essential for users as well as home network operators that users can be offered secure access to services in the their home networks. The use of an Authentication Proxy can offload Enablers authentication tasks.

5.2.3 Pre-conditions

The MT can establish TCP connections to the Key Manager and the AP. In 3GPP networks, roaming terminals usually have their point of presence in their home networks, which would guarantee that both the Enabler and the Key Manager could be reached from the MT. However, it is sufficient that the MT can directly address the Key Manager and the AP and establish a TCP connection. This could always be achieved, even with NAT(P)s in the path, if the Key Manager and the Enabler interfaces had public IP addresses.

Here, it is of course assumed that the AP is allowed to use the shared key management functionality. We only note that operators most likely will set up policies governing which enablers that may be allowed to use the shared key management.

Trusted channels between the AP and the Enablers exist.

5.2.4 Post-conditions

A TLS protected connection between the MT and the enabler exists. The end points have been mutually authenticated. The Enablers have information about the identity of the end-user, if required.

5.2.5 Normal Flow

The MT connects to the Key Manager to retrieve a shared key. The MT then connects to the Enabler. This connection is passed via the AP. The MT initiates a PSK-TLS session, indicating that the key to be used is the retrieved the shared key. . The AP connects (securely) to the Key Manager and retrieves the indicated shared key together with end-user identity information (anonymous use may be allowed). The shared key is then used in PSK-TLS to establish payload data protection between the MT and the AP. The AP proxies the traffic from the MT to the intended Enabler together with the user identity information.

5.2.6 Alternative Flow

The MT connects to the Key Manager to retrieve a shared key. The MT then connects to the Enabler. This connection is passed via the AP. The MT initiates a TLS session. The Enabler authenticates itself with a server certificate and requests client authentication with HTTP Digest with the agreed/established shared key. The client should validate that the server certificate. The AP connects (securely) to the Key Manager and retrieves the indicated Enabler specific key to be used in the HTTP digest authentication. The AP performs user authentication and proxies the traffic from the MT to the intended Enabler together with user identity information.

5.2.7 Operational and Quality of Experience Requirements

The establishment of the secure connection should be automatic and invisible to the end-user.

5.3 Certificate based end-user authentication (Optional)

5.3.1 Short Description

A client in a MT establishes a secure connection to an Enabler. Certificates are used as credentials to establish a TLS connection between the MT and the Enabler.

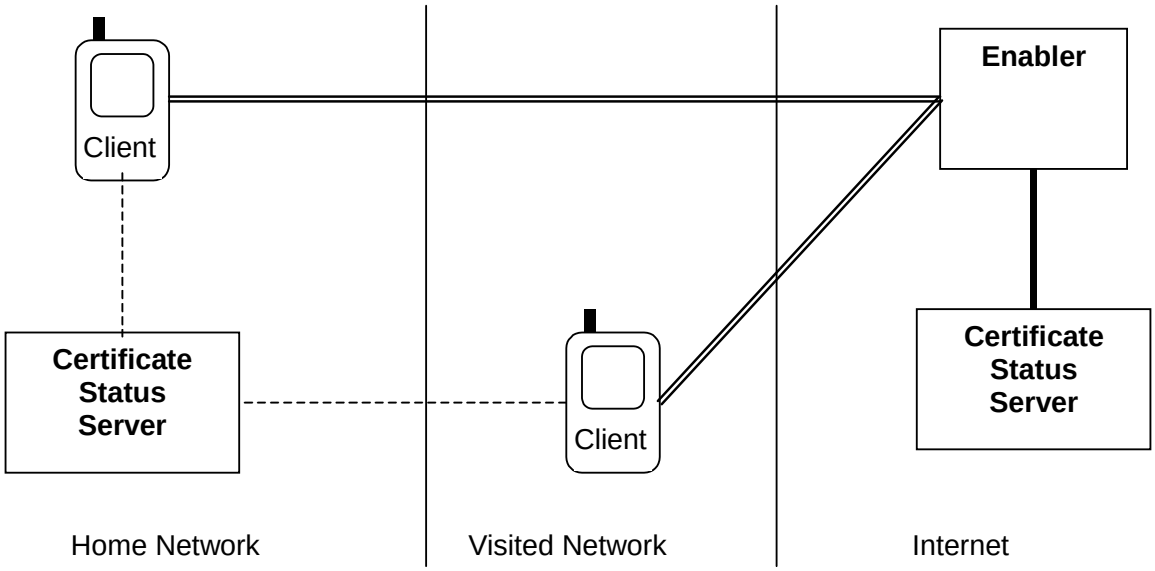

Figure 3: Use of certificates to establish TLS connection

5.3.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the MT.
- The home network operator
- Enabler operator
- Certificate status server operators

5.3.2.1 Actor Specific Issues

The Certificate Authorities iinvolved issuing client and server certificates have to provide a Certificate Status Service.

5.3.2.2 Actor Specific Benefits

It is essential for users as well as enabler operators that users can be offered secure access to services.

5.3.3 Pre-conditions

Client and Enabler certificates are pre provisioned.

The MT has access to a Certifiate Status Server for validation of Enabler certificates. Likewise the Enabler must have access to a Certifiate Status Server for validation of client certificates.

5.3.4 Post-conditions

A TLS protected connection between the MT and the enabler exists. The end points have been mutually authenticated.

5.3.5 Normal Flow

The MT connects to the Enabler and triggers a TLS set-up. Client and Enabler (server) certificates are used for mutual authentication.

5.3.6 Alternative Flow

Void

5.3.7 Operational and Quality of Experience Requirements

The establishment of the secure connection should be automatic and invisible to the end-user.

5.4 Distributed Enabler

5.4.1 Short Description

An enabler is distributed over cooperating parts in different operator domains (one example of such an enabler is Location). The client needs to establish a secure connection to the Enabler function in the visited network. The Enabler function in the home network facilitates the setup of a secure connection between the MT and the Enabler function in the visited network.

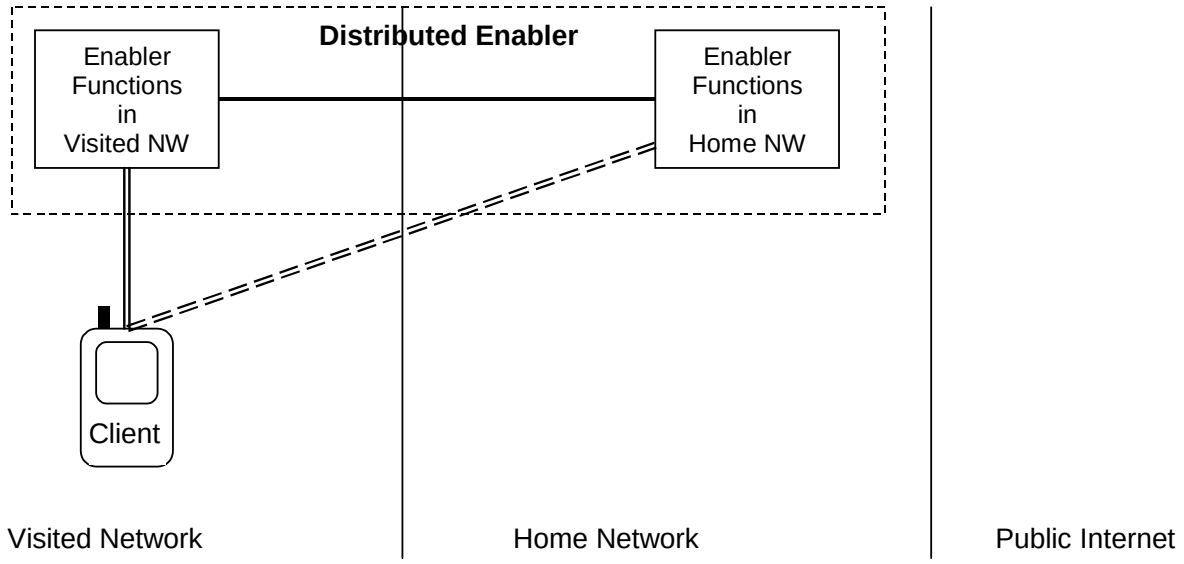

Figure 4. Distributed enabler accessed in Visited Network

5.4.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the MT.
- The home network operator running the Enabler function in the home network.
- The visited network operator running the Enabler function in the visited network.

5.4.2.1 Actor Specific Issues

5.4.2.2 Actor Specific Benefits

It is essential for users as well as visited network operators that users can be offered secure access to distributed enabler functions located in visited networks.

5.4.3 Pre-conditions

The Enabler functions in different operator domains have secure channels for inter domain communications. The MT can establish a secure connection to the Enabler function in the home network.

5.4.4 Post-conditions

A TLS protected connection between the MT and the Enabler function in the the visited network exists. The end points have been mutually authenticated. The Enabler function in the visited network has information about the identity of the end-user, if required.

5.4.5 Normal Flow

The MT connects securely to the Enbaler function in its home network. How this secure connection is achieved is out of scope in this use case; it could be by use of GBA based key management, preprovisioned secret keys or use of certificates. The Client indicates that it wants to connect to the Enabler function in the visited network. The enabler function in the home network verifies that the visited network enabler function is trusted and generates a key to be used for the setup of a PSK-TLS protected connection between the MT and the Enabler in the visited network. This key and its identity are sent to the MT and to the Enabler in the visited network. The Enabler in the visited network might also obtain information about the end-user identity or other information to authorize its use, if required. Then the MT establishes the PSK-TLS protected connection to the Enabler function in the visited network.

5.4.6 Alternative Flow

5.4.7 Operational and Quality of Experience Requirements

The establishment of the secure connection should be automatic and invisible to the end-user.

5.5 Network initated enabler access.

5.5.1 Short Description

A service in the network needs access to an enabler function involving the MT. The enabler in the network then initiates that the terminal connects to it by sending a a PUSH message to the MT. The MT connects to the enabler in the home network for verification of the request. Network initiation of services is susceptible to all kind of DoS and replay attacks. Thus the MT has to get help by the Enabler to verify that authencity of the initiation request.

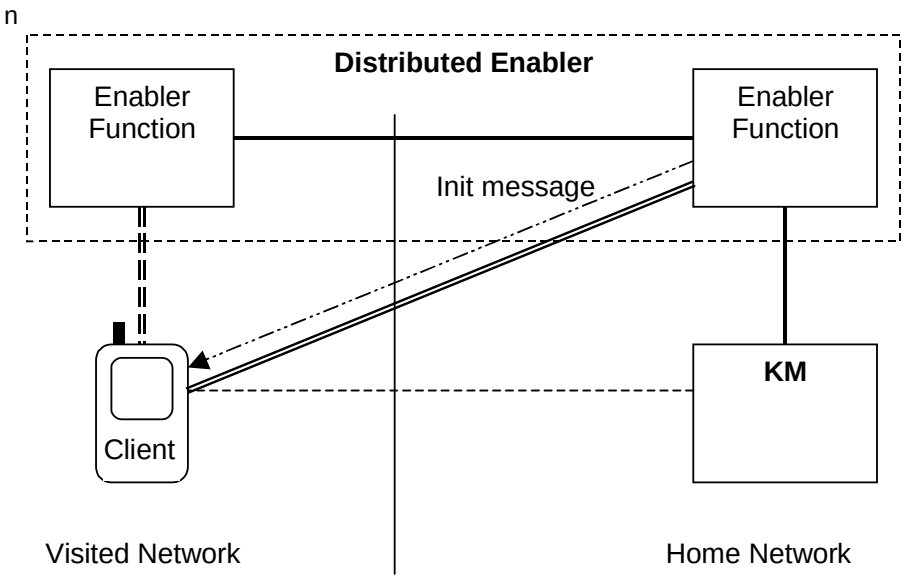

Figure 5: Network initiated connection between MT and Enabler.

5.5.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the MT.
- The home network operator. The home network operator runs
  - The the Enabler or the Enabler function in the home network if the enbaler is distributed over domains
- Possibly a n operator running the visited network part of a distributed Enabler

5.5.2.1 Actor Specific Issues

5.5.2.2 Actor Specific Benefits

It is essential for users as well as the operator(s) of an Enabler that there is a way to offer a secure and protected network initiated use of the Enabler.

5.5.3 Pre-conditions

The MT can establish a secure connection to the Enabler function in the home network. The Enabler functions in different operator domains have secure channels for inter domain communications.

5.5.4 Post-conditions

The Client in the MT has been assured (by the home network part of the Enabler, that the PUSH message initiating use of the Enabler is valid.

5.5.5 Normal Flow

The MT receives a PUSH message requesting initiation of a connection to one Enabler function (in the home or the visited network). The MT then connects over a secure channel to the Enabler function in the home network to have the request verified. The Enabler function in the home network indicates that the request is valid (or invalid) and the actual use of the enabler services proceeds.

5.5.6 Alternative Flow

5.5.7 Operational and Quality of Experience Requirements

The verification of the validity of the PUSH message initiating the use of the service should be automatic and invisible to the end-user.

5.6 Provisioning of security parameters.

5.6.1 Short Description

An Enabler may need to control that a client only can establish connections to or accept connections from trusted entities. Such security controls can be used to prevent the client from being tricked into connecting to fraudulent nodes acting as legitimate enabler entities. The security parameters are usually in the form of white-lists of trusted URL's/URI's for the Enabler, authorized initiators of message exchanges, etc.

This use case is only concerned with the use of device management functionality to achieve the distribution and management of security parameters.

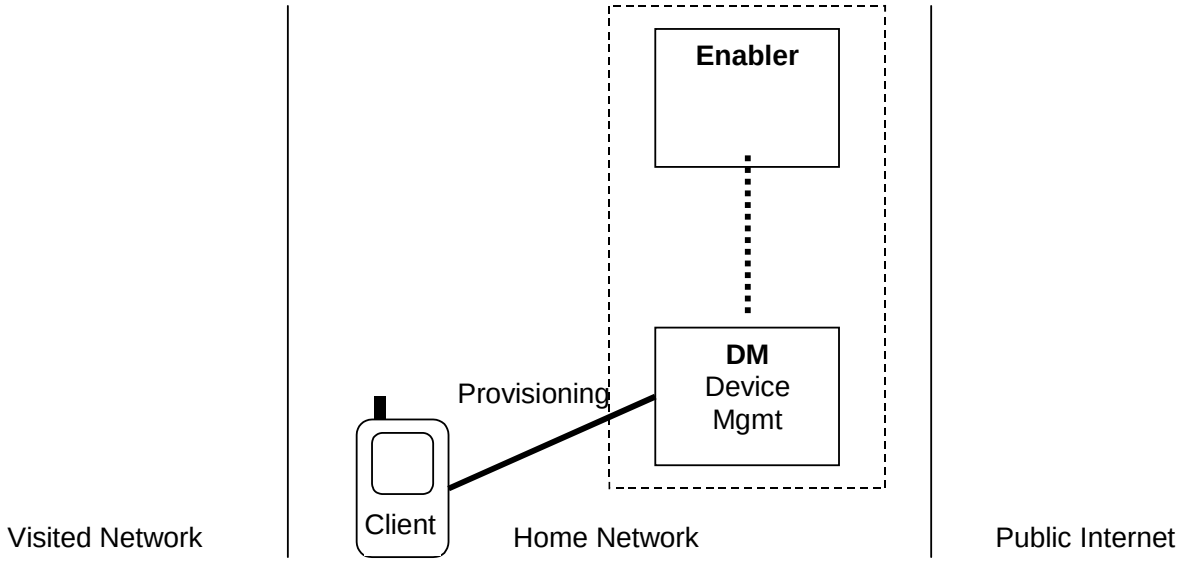

Figure 6: Provisioning of security parameters.

5.6.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the MT.
- The home network operator. The home network operator runs
  - The Enabler
  - The device management system as a stand alone functionality or as part of the Enabler

5.6.2.1 Actor Specific Issues

The provisioning of security parameters is based on device management and thus the MT and the Enabler / home network operator has to support a device management system.

5.6.2.2 Actor Specific Benefits

It is essential for users as well as the operator(s) of an Enabler that there is a secure way to provision the security parameters required for secure use of the Enabler.

5.6.3 Pre-conditions

The device management system (MT and Enabler / home network operator) is enabled and configured with adequate security settings. Managed objects for handling of the security parameters are defined.

5.6.4 Post-conditions

The Enablers managed objects for security parameters in the MT have been populated with data obtained from the Enabler / home network operator.

5.6.5 Normal Flow

The device management system in the Enabler / home network operator establishes a secure device mangement session with the MT. The device management system writes the Enablers security parameters into the relevant Managed Objects in the MT.

Whenever the Enabler functionality in the MT is invoked, the client in the MT reads the security parameters from the device management system and applies them in its local security control activities.

5.6.6 Alternative Flow

The device management client in the MT “bootstraps” the security parameters for the Enabler from the IM into its tree of managed objects.

Whenever the Enabler functionality in the MT is invoked, the client in the MT reads the security parameters from the device management system and applies them in its local security control activities.

5.6.7 Operational and Quality of Experience Requirements

The provisioning of security parameters should be automatic and invisible to the end-user.

5.7 Provisioning of keys.

5.7.1 Short Description

An Enabler requires that a secure connection can be established between clients and the enabler. The secret keys needed to establish such a secure connection are provisioned by the Enabler / home network operator. Naming of keys needs to be specified to be compliant with existing key management schemes.

This use case is only concerned with issues of how secret keys can be provisioned to MTs (and clients).

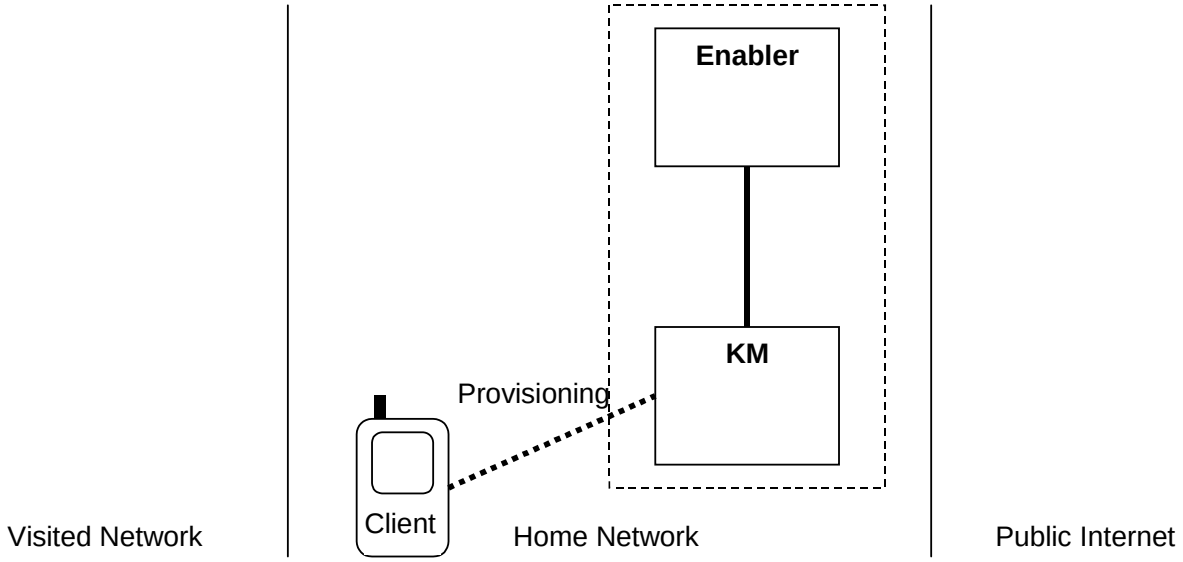

Figure 7: Provisioning of security parameters.

5.7.2 Actors

We have the following actors in this use case

- The end-user represented by the client in the MT.
- The home network operator. The home network operator runs
  - The Enabler
  - The Key Manager

5.7.2.1 Actor Specific Issues

The provisioning of keys to the MT from the KM may be proprietary and be defined by the home operator, the network or the Enabler. There has to be secure storage for the keys in the MT or in the IM.

5.7.2.2 Actor Specific Benefits

It is essential for users as well as the operator(s) of an Enabler that there is a secure way to provision the keys required for secure use of the Enabler.

5.7.3 Pre-conditions

The MT and/or the IM are enabled and configured to receive the keys to be used by the enabler.

5.7.4 Post-conditions

The MT has securely stored the keys used by the Enabler.

5.7.5 Normal Flow

The MT/client specific keys used by the Enabler are retrieved from the KM. These keys together with their key identifiers are installed in the MT and/or the IM. When the Enabler service is initiated, the MT uses the installed keys to establish a secure connection to the Enabler. The Enabler requests the corresponding keys from the Key Manager. The set up of the secure connection continues.

5.7.6 Alternative Flow

5.7.7 Operational and Quality of Experience Requirements

The provisioning of secret keys security parameters should be automatic and invisible to the end-user.

6. Requirements (Normative)

6.1.1 Security

Label	Description	Enabler Release
SEC_CF-S1	Any secret data needed to perform the SEC_CF MUST be *stored* such that no unauthorized entity can get access to this data.	SEC_CF 1.0
SEC_CF-S2	Any secret data needed to perform the SEC_CF MUST be *transmitted* such that no unauthorized entity can access this data.	SEC_CF 1.0
SEC_CF-S3	It MUST be possible for authorized entities to modify secret data in a secure way.	SEC_CF 1.0

Table 1: High-Level Functional Requirements – Security Items

6.1.1.1 Authentication

Label	Description	Enabler Release
SEC_CF-1.1	The SEC_CF MUST be able to provide authentication of the client (requestor) to the responder that makes use of the SEC_CF. Authentication credentials presented by the requestor MUST be communicated to the resource that makes use of the SEC_CF enabler. Mechanisms to communicate these authenticated identities MUST be defined in the SEC_CF specifications.	SEC_CF 1.0
SEC_CF-1.1 a	The SEC_CF MAY be able to provide authentication of the end-user to the resource that makes use of the SEC_CF , e.g. by entering a PIN code, by using biometrics if applicable or a username/password	Future
SEC_CF-1.2	The SEC_CF MUST be able to provide authentication of the resource that makes use of the SEC_CF to the requesting client. Authenticated identities presented by the resource MUST be communicated to the requesting client. Mechanisms to communicate these authentication credentials MUST be defined in the SEC_CF specifications.	SEC_CF 1.0
SEC_CF-1.2 a	It MUST be possible for Authentication (server to client, client to server, or mutual) to be performed via an authentication proxy..	SEC_CF 1.0
SEC_CF-1.2 b	It MUST be possible for authentication to be performed directly between a client and the resource that makes use of the SEC_CF without an authentication proxy.	Future
SEC_CF-1.2 c	In case that the enabler is distributed between the home network and visited network(s), the SEC_CF MUST be able to provide authentication of the servers (representing the resource) in the visited network to the requesting client. This may be done via a server in the home network assuming a secure connection between the servers is present.	SEC_CF 1.0
SEC_CF-1.3	The SEC_CF MUST be able to provide data origin authentication. This means, it MUST be possible to ensure confidence that a received message or piece of data has been created by a certain party, and that this data has not been corrupted or tampered with.	SEC_CF 1.0
SEC_CF-1.4	The SEC_CF MUST be able to provide replay protection to ensure confidence that a received message has not been recorded and played back.	Future
SEC_CF-1.5	The SEC_CF MUST be able to authenticate the source of the broadcast or streaming.	Future
SEC_CF-1.6	The SEC_CF MAY allow the user to authenticate himself to the client, e.g. by entering a PIN code or by using biometrics if applicable.	Future

Table 2: High-Level Functional Requirements – Authentication

6.1.1.2 Data Integrity

SEC_CF-3.1	The SEC_CF MUST be able to provide data integrity, i.e. protection against accidental or intentional changes to the data, by ensuring that changes to the data are detectable. The ability of data integrity must be provided for any data transmissions between any resources in either home or visited networks.	SEC_CF 1.0

Table 3: High-Level Functional Requirements – Data integrity

6.1.1.3 Confidentiality and Privacy

SEC_CF-4.1	The SEC_CF MUST be able to provide data confidentiality that ensures that *transmitted* information is not made available or disclosed to unauthorised individuals, entities, or processes. The ability of data confidentiality must be provided for any data transmissions between any resources in either home or visited networks.	SEC_CF 1.0

Table 4: High-Level Functional Requirements – Confidentiality and Privacy

6.1.1.4 Key Management

SEC_CF-5.1	The SEC_CF MUST be able to provide a secure means of key agreement prior to key usage. This ability is needed with respect to authentication keys as well as with respect to (temporary) encryption keys and keys needed for data integrity. Affected entities are any resources in either home or visited networks.	SEC_CF 1.0

Table 5: High-Level Functional Requirements - Privacy

6.1.2 Charging

Label	Description	Enabler Release
	(no charging requirements)	

Table 6: High-Level Functional Requirements – Charging Items

6.1.3 Administration and Configuration

Label	Description	Enabler Release
SEC_CF-A1	It MUST be possible to provide initial keys to the requesters and resources.	SEC_CF 1.0
SEC_CF-A2	It MUST be possible to change security algorithms in the servers and clients in a secure manner.	SEC_CF 1.0

Table 7: High-Level Functional Requirements – Administration and Configuration Items

6.1.4 Usability

Label	Description	Enabler Release
	(no particular usability requirements)	

Table 8: High-Level Functional Requirements – Usability Items

6.1.5 Interoperability

Label	Description	Enabler Release
	(no particular interoperability requirements)	

Table 9: High-Level Functional Requirements – Interoperability Items

6.1.6 Privacy

Label	Description	Enabler Release
	(no particular privacy requirements)	

Table 10: High-Level Functional Requirements – Privacy Items

6.2 Overall System Requirements

Label	Description	Enabler Release

Table 11: High-Level System Requirements

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft Version History

Document Identifier	Date	Sections	Description
OMA-RD_Security Common Function-V0_1-20040811-D	11 Aug 2004	1, 4, A	Initial input, based on WID
OMA-RD_SEC_CF-V1_0-20041104-D	04 Nov 2004	2, 3, 5, A.2	Some references and definitions added, Use cases added
OMA-RD_SEC_CF-V1_0-20050819-D	19 Aug 2005	1, 2, 3, 4, 6, A	New approach due to discussions within SEC and between SEC and other OMA WGs
OMA-RD_SEC_CF-V1_0-20050923-D	23 Sep 2005		Editorial changes. First version uploaded to permanent documents
OMA-RD_SEC_CF-V1_0-20051018-D	18 Oct 2005	2, 3, 4, 5, 6, A	Two additional use cases. Particularization of requirements. Editorial changes.
OMA-RD_SEC_CF-V1_0-20051215-D	15 Dec 2005	1, 2, 5, A	Editorial changes. Deletion of requirements foreseen for later versions.
OMA-RD_SEC_CF-V1_0-20060208-D	8 Feb 2006	5, 6, A	Editorial changes. Deletion and modification of some requirements.
OMA-RD_SEC_CF-V1_0-20060209-D	9 Feb 2006	4, 5, 6, A	Editorial changes.
OMA-RD_SEC_CF-V1_0-200603159-D	15 Mar 2006	1, 3, 4, 6, A	Modifications according to review, see http://www.openmobilealliance.org/ftp/PD/OMA-RDRR-SEC_CF-V1_0-20060223-D.zip
OMA-RD-SEC_CF-V1_0-20060614-D	14 Jun 2006	6,A	Modifications according to review, see OMA-RDRR-SEC_CF-V1_0-20060614-D
OMA-RD-SEC_CF-V1_0_20060711-D	12 Jul 2006	3,4,6,A	Modifications accorifng to the comments received from the REQ WG
OMA-RD-SEC_CF-V1_0-20060713-D	13 Jul 2006	6	Modifications according to the comments received from the REQ WG
OMA-RD-SEC_CF-V1_0-20060808-C	08 Aug 2006	n/a	TP approval: OMA-TP-2006-0284-INP_OMA_RD_SEC_CF_V1_0_for_approval_as_Candidate