
# OMA Device Management Security

Candidate Version 1.3 – 06 Mar 2012

**Open Mobile Alliance**

OMA-TS-DM_Security-V1_3-20120306-C

# Contents

1. SCOPE ..... 4
2. REFERENCES ..... 5
2.1 NORMATIVE REFERENCES ..... 5
2.2 INFORMATIVE REFERENCES ..... 6
3. TERMINOLOGY AND CONVENTIONS ..... 7
3.1 CONVENTIONS ..... 7
3.2 DEFINITIONS ..... 7
3.3 ABBREVIATIONS ..... 7
4. INTRODUCTION ..... 8
5. OMA DEVICE MANAGEMENT SECURITY ..... 9
5.1 CREDENTIALS ..... 9
5.2 INITIAL PROVISIONING OF CREDENTIALS ..... 9
5.3 MESSAGE SECURITY ..... 9
5.4 AUTHENTICATION ..... 9
5.4.1 Digest Authentication string format in OMA DM ..... 10
5.4.2 Computation of the Digest ..... 11
5.4.3 Password and nonce usage ..... 11
5.5 NEXT NONCE CONSIDERATIONS ..... 11
5.6 TRANSPORT NEUTRAL INTEGRITY ..... 12
5.6.1 HMAC-SHA256 Integrity ..... 12
5.7 NOTIFICATION INITIATED SESSION ..... 13
5.8 MANAGEMENT OBJECT ENCRYPTION ..... 13
5.9 CONFIDENTIALITY OF INFORMATION BETWEEN DM SERVERS ..... 13
5.10 SECURITY FOR BOOTSTRAP OPERATION ..... 13
5.10.1 Bootstrap via CP ..... 13
5.10.2 Bootstrap via DM ..... 14
APPENDIX A. CHANGE HISTORY (INFORMATIVE) ..... 17
A.1 APPROVED VERSION HISTORY ..... 17
A.2 DRAFT/CANDIDATE VERSION 1.3 HISTORY ..... 17
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ..... 19
B.1 SCR FOR DM CLIENT ..... 19
B.2 SCR FOR DM SERVER ..... 20
APPENDIX C. DM INTERACTION WITH GBA PUSH (INFORMATIVE) ..... 22

# Figures

Figure 1: DM-GBA interaction with added NAF into the DM Server (DMS) ..... 22
Figure 2: DM-GBA interaction via external NAF ..... 23

# 1. Scope

This document describes OMA-DM security requirements in general, and provides description of transport layer security, application layer security, etc. It also describes security mechanisms that are used to provide for integrity, confidentiality and authentication.

# 2. References

## 2.1 Normative References


**[DMDICT]** "OMA Device Management Dictionary, Version 1.0". Open Mobile Alliance™. OMA-SUP-DM_Dictionary-v1_0. URL:http://www.openmobilealliance.org

**[DMBOOT]** "OMA Device Management Bootstrap, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_Bootstrap-V1_3. URL:http://www.openmobilealliance.org

**[DMMETA]** "Device Management Meta Information, version 1.3". Open Mobile Alliance™. OMA-TS-DM_ _MetaInfo-V1_3. URL:http://www.openmobilealliance.org

**[DMNOTI]** "OMA Device Management Notification Initiated Session, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_Notification-V1_3. URL:http://www.openmobilealliance.org

**[DMPRO]** "OMA Device Management Protocol, Version 1.3". Open Mobile Alliance™. OMA-DM_Protocol-V1_3. URL:http://www.openmobilealliance.org

**[DMREPPRO]** "OMA Device Management Representation Protocol", Open Mobile Alliance™, OMA-TS-DM_RepPro-V1_3, URL:http://www.openmobilealliance.org/

**[DMSTDOBJ]** "OMA Device Management Standardized Objects, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_StdObj-V1_3. URL:http://www.openmobilealliance.org

**[DMTND]** "OMA Device Management Tree and Description, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_TND-V1_3. URL:http://www.openmobilealliance.org

**[HTTPBIND]** "OMA Device Management HTTP Binding Specification", Open Mobile Alliance™, OMA-TS-DM_HTTPBinding-V1_3, URL:http://www.openmobilealliance.org/

**[OBEXBIND]** "OMA Device Management OBEX Binding Specification", Open Mobile Alliance™, OMA-TS-DM_OBEXBinding-V1_3, URL:http://www.openmobilealliance.org/

**[PROVBOOT]** "Provisioning Bootstrap 1.1". Open Mobile Alliance™. OMA-WAP-ProvBoot-v1_1. URL:http://www.openmobilealliance.org

**[PROVSC]** "Provisioning Smart Card Specification Version 1.1". Open Mobile Alliance™. OMA-WAP-ProvSC-v1_1. URL:http://www.openmobilealliance.org

**[RFC1321]** "The MD5 Message-Digest Algorithm". Network Working Group. April 1992. URL:http://www.ietf.org/rfc/rfc1321.txt

**[RFC2045]** "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies" - N. Freed. November 1996. URL:http://www.ietf.org/rfc/rfc2045.txt

**[RFC2104]** "HMAC: Keyed-Hashing for Message Authentication". Network Working Group. February 1997. URL:http://www.ietf.org/rfc/rfc2104.txt

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

**[RFC2616]** "Hypertext Transfer Protocol – HTTP/1.1". Network Working group. June 1999. URL:http://www.ietf.org/rfc/rfc2616.txt

**[RFC4648]** "The Base16, Base32, Base64 Data Encodings". S. Josefsson. October 2006. URL:http://www.ietf.org/rfc/rfc4648.txt

**[RFC5843]** "Additional Hash Algorithms for HTTP Instance Digests", A. Bryan, April 2010. URL:http://www.ietf.org/rfc/rfc5843.txt

**[RFC6234]** "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", D. Eastlake 3rd, etc. May 2011
URL:http://www.ietf.org/rfc/rfc6234.txt

**[SCRRULES]** "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures,
URL:http://www.openmobilealliance.org

**[SHA]** "Secure Hash Standard", NIST FIPS PUB 180-1, National Institute of Standards and Technology, U.S. Department of Commerce, DRAFT, May 1994.
URL: http://www.itl.nist.gov/fipspubs/fip180-1.htm

**[TS33.220]** "Generic Bootstrapping Architecture (GBA)"
URL:http://www.3gpp.org

**[TS33.223]** "Generic Bootstrapping Architecture (GBA) Push function"
URL:http://www.3gpp.org

**[WAP-219-TLS]** OMA Wireless Public Key Infrastructure V1.0, Open Mobile Alliance™, WAP-219_100-TLS
URL:http://www.openmobilealliance.org/

**[WBXML1.1]** "WAP Binary XML Content Format Specification", WAP Forum™.
SPEC-WBXML-19990616.pdf.
URL:http://www.openmobilealliance.org

**[WBXML1.2]** "WAP Binary XML Content Format Specification", WAP Forum™. WAP-154-WBXML.
URL:http://www.openmobilealliance.org

**[WBXML1.3]** "WAP Binary XML Content Format Specification", WAP Forum™. WAP-192-WBXML.
URL:http://www.openmobilealliance.org

**[WSPBIND]** "OMA Device Management WSP Binding Specification", Open Mobile Alliance™,
OMA-TS-DM_WSPBinding-V1_3,
URL:http://www.openmobilealliance.org/

**[WTLS]** "Wireless Transport Layer Security", Open Mobile Alliance™, WAP-261-WTLS,
URL:http://www.openmobilealliance.org


## 2.2 Informative References

None.

# 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.

## 3.2 Definitions

Kindly consult [DMDICT] for all definitions used in this document.

## 3.3 Abbreviations

Kindly consult [DMDICT] for all abbreviations used in this document.

# 4. Introduction

OMA DM is a protocol based upon DM representation [DMREPPRO]. Its purpose is to allow remote management of any device supporting the OMA DM protocol. Due to the vast range of data needing to be managed on current and future Devices, it is necessary to take account of the value of such data. In many situations, the data being manipulated within a Device (or being transferred to/from the device) is of high value. In some cases this is confidential data and some degree of protection regarding the confidentiality of that data should be offered. In another case, the integrity of the data being transferred must be maintained, since deliberate or accidental corruption of this data can result in lost revenue or subsequent exploits being facilitated. Finally it's important that both entities (the DM Client and the DM Server) have confidence in the authenticity of the other entity.

# 5. OMA Device Management Security

## 5.1 Credentials

Four examples of suitable credentials exchanged between DM Clients and DM Servers are shown in the following list.

1. A username (AAUTHNAME in [DMSTDOBJ]) that uniquely identifies the DM Server [DMTND] to a DM Client), a password (AAUTHSECRET in [DMSTDOBJ]) – to be coupled with the username, and a nonce (AAUTHDATA in [DMSTDOBJ]) – to prevent replay attacks where hashing algorithms are used with static data.

2. A username (AAUTHNAME in [DMSTDOBJ]) that identifies the DM Client to the DM Server), a password (AAUTHSECRET in [DMSTDOBJ]) – to be coupled with username, and a nonce (AAUTHDATA in [DMSTDOBJ]) – to prevent replay attacks where hashing algorithms are used with static data.

3. A certificate, as specified in [WAP-219-TLS]

4. A network, transport or server specific mechanism, for example WAP.

For the purpose of DM Server to DM Client authentication, if username, password and nonce are used, the DM Server MUST use a different password for each DM Client it serves, in order that a DM Client (which possesses a shared secret based on this password) cannot pose effectively as this DM Server in a interaction with another DM Client.

## 5.2 Initial Provisioning of Credentials

The initial provisioning of the credentials for a DM Server, so that the DM Client may be capable of authenticating a specific DM Server, is documented in [DMBOOT]. However, other techniques outside of these specifications are not excluded.

Essentially, any suitable technique will deliver at least the bare minimum of information required to establish the DM session. This, of course, includes the DM Server credential and the DM Client credential.

## 5.3 Message Security

Message security is comprised of three functions: Authentication, Integrity, and Confidentiality. All three functions are necessary to provide a safe and secure method of managing a device. Appropriate security MUST be employed when sending a DM Message to the DM Client.

If the transport is able to provide authentication and integrity, the transport authentication and integrity MUST be used.

If the transport is able to provide confidentiality, the transport confidentiality MUST be used.

If the transport is unable to provide authentication and integrity, transport neutral integrity MUST be used.

Transport specific security is documented in the transport binding documents [HTTPBIND], [OBEXBIND], [WSPBIND].

## 5.4 Authentication

Both the DM Client and the DM Server MUST be authenticated to each other. Authentication can be performed at different layers. DM Servers MUST support both DM Client and DM Server authentication at the transport layer. DM Servers MUST request client authentication at the transport layer when transport layer security is requested by the DM Client during session establishment. Some DM Clients may not support transport-layer client authentication. DM Servers MUST authenticate such DM Clients at the application layer. If the transport layer does not have a sufficiently strong authentication feature, OMA DM Protocol layer authentication MUST be used.

Either the DM Client or the DM Server MAY send credentials to each other or challenge the other to send them.

The Basic scheme is identified by the URI `syncml:auth-basic`. This authentication scheme is a Base64 character encoding, as defined in [RFC4648], of the concatenation of the originator's userid, followed by the COLON (i.e., ":")

separator character, followed by the password associated with the specified userid. This authentication scheme is susceptible to the threat of network eavesdrop, but is simple to implement. However, take care when using this scheme. For example, a user is strongly advised to consider using additional security considerations, such as an encrypted transport connection.

The MD5 Digest scheme is identified by the URI `syncml:auth-md5`. The computation of MD5 digest is specified in section 5.4.2. The maximum duration that the nonce string can be used by the originator is the current DM Session. Note that issuing a nonce does not constitute use – a nonce MAY be issued for use in the next session. More frequent changes to the nonce string can be specified with the `NextNonce` element type within the `Meta` element type of the `Chal` element type. The MD5 digest algorithm and a publicly available source code for generating MD5 digest strings are specified by [RFC1321]. The MD5 digest, a 128-bit binary value, MUST be Base64 character encoded when transferred as clear-text XML. For WBXML representation, the additional Base64 character encoding is not necessary.

The SHA2 Digest scheme is identified by the URI syncml:auth-sha256. The computation of SHA-256 digest is specified in section 5.4.2. The SHA-256 digest algorithm and a publicly available source code for generating digest string is specified by [RFC6234]. The SHA-256 digest, a 256-bit binary value, MUST be Base64 character encoded ([RFC4648]) when transferred as clear-text XML. For WBXML representation, the additional Base64 character encoding is not necessary.

Other authentication schemes can be specified by prior agreement between the originator and the recipient.

The DM Clients that do not support client authentication at the transport layer MUST support DM syncml:auth-md5 type authentication. The DM Clients that support mutual authentication at the transport layer MAY support OMA DM authentication mechanisms such as the syncml:auth-md5 type. The DM Server MAY still issue a MD5 challenge when transport layer mutual authentication has already been completed but the session MUST be terminated if the client does not respond with the requested authentication type. The provisioning of credentials/certificates for transport layer authentication is beyond the scope of OMA DM Security.

It is assumed that DM Protocol will often be used on top of a transport protocol that offers session layer authentication so that authentication credentials are exchanged only at the beginning of the session (like in TLS or WTLS). If the transport layer is not able to provide session authentication, however, each request and response MUST be authenticated.

### 5.4.1 Digest Authentication string format in OMA DM

To specify the userid for the credentials, when the credentials do not include it in the resolvable form, the userid MUST be transferred in the `LocName` element of `Source` in `SyncHdr`. MD-5 authentication [RFC1321] or SHA-256 authentication [RFC5843] works by supplying primitive `userid:password` in the `Cred` element of the `SyncHdr`. A SyncML message example is shown below.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.3</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
    </Target>
    <Source>
      <LocURI>IMEI:493005100592800</LocURI>
      <LocName>Bruce1</LocName>    <!-– userid -->
    </Source>
    <Cred>
      <Meta>
        <Type xmlns=’syncml:metinf’>syncml:auth-md5</Type>
        <!-- digest scheme name -->
        <Format xmlns=’syncml:metinf’>b64</Format>
      </Meta>
      <Data>18EA3F……</Data>
          <!-- base64 formatting Digest Computation result -->
    </Cred>
```

```
    <Meta>
      <MaxMsgSize xmlns='syncml:metinf'>5000</MaxMsgSize>
    </Meta>
  </SyncHdr>
        <!—- regular body information here -->
  <SyncBody>
  </SyncBody>
</SyncML>
```

### 5.4.2 Computation of the Digest

The digest supplied in the `Cred` element is computed as follows:

Let H = the MD5 or SHA-256 Hashing function.
Let Digest = the output of the Hashing function.

Let B64 = the base64 encoding function.

Digest = H(B64(H(username:password)):nonce)

This computation allows the authenticator to authenticate without having knowledge of the password. The password is neither sent as part of the `Cred` element, nor is it required to be known explicitly by the authenticator, since the authenticator need only store a pre-computed hash of the username:password string.

### 5.4.3 Password and nonce usage

Both password and nonce are RECOMMENDED to be at least 128 bits (16 random octets) in length.

The nonce value MUST be issued in a challenge from either the DM Client or the DM Server. In the case of the credentials being sent prior to a challenge being issued, then the last nonce used SHALL be reused. The authenticator MUST be aware that the issuer of the credentials may be using a stale nonce (that is to say, a nonce that is invalid due to some previous communications failure or a loss of data). Because of this, if authentication fails, one more challenge, along with the supply of a new nonce, MUST be made.

A new nonce SHOULD be used for each new session. The sequence of nonce values (as seen across sessions) SHOULD be difficult to predict.

The authenticator MUST be aware that the issuer of the credentials may be using a stale nonce (that is to say, a nonce that is invalid due to some previous communications failure or a loss of data). Because of this, if authentication fails, one more challenge, along with the supply of a new nonce value given by NextNonce element, MUST be made. If second challenge failed to authenticate the Server, the Client MUST NOT try further authentication.

## 5.5 Next nonce considerations

OMA-DM supports two mechanisms for updating the nonce. The updated nonce value can be either provided by the value of the NextNonce meta element [DMMETA] in the SyncML message or by the value of the AAuthdata node in the DMAcc MO [DMSTDOBJ].

The DM Client automatically sets the value of the AAuthData node to the value of the NextNonce element in the SyncML message if one of the following authentication schemes is in effect for the current DM Session:

- DIGEST
- DIGEST-SHA256
- HMAC

Since other authentication schemes do not support the nonce, if any other authentication scheme is in effect for the current DM Session, the value of the NextNonce element is ignored.

# 5.6 Transport Neutral Integrity

Transport neutral integrity of DM messages is achieved using a HMAC [RFC2104].

The transport neutral integrity SHOULD be used when the transport is unable to provide neither authentication nor integrity. It is RECOMMENDED to apply the transport neutral integrity for Sessionless DM and Sessionless DM Reporting.

## 5.6.1 HMAC-SHA256 Integrity

The HMAC-SHA256 is the mechanism for message integrity using SHA-256 cryptographic hash function (see [RFC6234]).

The use of HMAC-SHA256 MAC integrity is identified as 'syncml:auth-hmac-sha256'. The support of 'syncml:auth-hmac-sha256' is OPTIONAL.

### 5.6.1.1 How the integrity information is provided

The integrity information MUST be transported along with the original OMA DM message. This is achieved by inserting the HMAC into a transport header called `x-syncml-hmac`. This technique works identically on HTTP, WAP, and OBEX. The HMAC is calculated initially the sender using the entire message body, either in binary form (WBXML) or text form (XML). The receiver applies the same technique to verify the incoming message.

The header x-syncml-hmac contains multiple parameters, including the MAC data itself, the '`Algorithm`' parameter, the '`ServerId`' parameter and '`Timestamp`' parameter.

The value of the x-syncml-hmac header is defined as a comma separated list of attribte-values pairs. The rule "#rule" and the terms "token" and "quoted-string" are used in accordance to their definition in the HTTP 1.1 specifications [RFC2616].

Here is the formal definition:

```
x-syncml-hmac = #syncml-hmac-param

syncml-hmac-param = (algorithm | serverid | timestamp | mac)

algorithm = 'algorithm' '=' ( 'HMAC-SHA256' |  token)

serverid = 'ServerId' '=' quoted-string

timestamp = 'Timestamp' '=' quoted-string

mac = 'mac' '=' base64-string
```

Example:

x-syncml-hmac: algorithm="HMAC-SHA256"; ServerId="dm.foo.org"; Timestamp="2012-01-01T13:04:20Z"; mac="
NTI2OTJhMDAwNjYxODkwYmQ3NWUxN2RhN2ZmYmJlMzkay2="

### 5.6.1.2 How integrity is verified

Definition of HMAC-SHA256 function is found in [RFC6234].

MAC information for 'syncml:auth-hmac-sha256' is calculated using three parameters.

The integrity key(K): The binary data which is provided as AAuthSecret

The message (Msg): The message to authenticate

The timestamp (Timestamp): Current UTC time text string which is represented in format of 'YYYY-MM-DDThh:mm:ssZ' (See [ISO8601]). Note that the Timestamp parameter is provided as opaque parameter value.

MAC = HMAC( K, Timestamp:Msg )

Where HMAC(K, X) is the result of HMAC calculation with SHA-256 hash function.

The message receiver can verify the integrity of the incoming OMA DM message with checking whether the calculated mac value is matched with supplied mac value.

If HMAC-SHA256 is expected, but x-syncml-hmac is not provided by the sender, the message SHOULD be rejected to be processed.

If the receiver failed to verify the integrity on supplied x-syncml-hmac header, the message MUST be rejected to be processed.

# 5.7 Notification Initiated Session

OMA DM offers the ability for a DM Server to make a request to a DM Client to establish a Management Session. The security of this message depends upon verifying integrity of the message using the value of digest field. The specification of this message can be found in [DMNOTI].

# 5.8 Management object encryption

OMA DM fully supports the use of encrypted management objects, which MAY remain encrypted within the DM Tree, or be decrypted upon receipt by the DM Client or the DM Server.

Depending upon implementation, an object MAY be encrypted prior to transmission over a unencrypted transport layer, and remain encrypted in storage space within either the DM Server or the DM Client, or, it MAY be decrypted immediately after receipt, and stored internally in unencrypted format.

No restrictions are placed upon the encryption technique used, since this is independent of the OMA DM protocol itself.

# 5.9 Confidentiality of information between DM Servers

OMA DM offers the ability for a CM Server to make private any data that is stored under DM control from another DM Servers. This is facilitated by the use of an ACL (Access Control List) that allows the protection of any group, or any individual MO.

#### 5.9.1.1 The Access Control List

The Access Control List allows a hierarchical assignment of Access Rights based upon DM Server Identifiers (Unique identifiers for the DM Servers [DMTND]). A detailed description of the ACL can be found in [DMTND].

# 5.10 Security for Bootstrap Operation

Bootstrapping is a sensitive process that may involve communication between two parties without any previous relationship or knowledge about each other. In this context, security is very important. The receiver of a bootstrap message needs to know that the information originates from the correct source and that it has not been tampered with en-route. The sender also wants end-to-end confidentiality to prevent impersonation by eavesdroppers who could see the contents of the bootstrap message containing credentials to access the DM Server. It is important that DM Clients accept bootstrapping commands only from authorized DM or CP Servers.

## 5.10.1 Bootstrap via CP

The CP bootstrap mechanism is defined in [PROVBOOT].

#### 5.10.1.1 Smartcard

The CP Bootstrap mechanism from the smartcard is defined in [PROVSC].

## 5.10.2 Bootstrap via DM

### 5.10.2.1 Transports

The Bootstrap message MUST be sent to the DM Client via secure transport.

Transport neutral security MAY also be applied to the Bootstrap message.

Transport specific security is documented in the transport binding documents [HTTPBIND], [OBEXBIND] and [WSPBIND].

### 5.10.2.2 Smartcards

Smartcard is a secure transport. A Smartcard allows for very secure delivery of a Bootstrap message.

For specific specification of a Smartcard, please refer to [DMDICT].

Bootstrap data MAY be stored on the smartcard. The behaviour of a DM Client regarding bootstrap data is specified in [DMBOOT].

### 5.10.2.3 Transport Neutral Security

It is very important for the DM Client to be able to verify the data integrity and the authenticity of the Bootstrap Message. OMA-DM defines the following types of shared secrets for use between the DM Client and the network provider during the DM Bootstrap process. It needs to be emphasized that other types of shared secrets are not precluded.

> **NETWORKID**: In this case the shared secret is known by the Device and the network provider before the bootstrap process starts. What the shared secret actually is depends on the network provider and the particular Device. This could be things like IMSI (for GSM) or ESN (for CDMA). One advantage with this method is that it does not involve any user intervention.
>
> **USERPIN**: In this case the shared secret is selected by the user or selected by the network operator and then communicated to the user. The PIN is either communicated via some user out-of-band mechanism, or agreed upon before the bootstrap process starts.
>
> **USERPIN_NETWORKID**: In this case the shared secret is the concatenation of the USERPIN and the NETWORKID, delimited by the ":" character. For example, if the USERPIN is *abc* and the NETWORKID is *xyz*, the shared secret is "*abc:xyz*".

#### 5.10.2.3.1 Basic Requirements

The DM Server and DM Client MUST support NETWORKID and USERPIN.

The DM Server and DM Client MAY support USERPIN_NETWORKID.

Other methods MAY be used as long as they employ a level of security appropriate for bootstrap. The combined security of the secret (e.g., randomness, difficulty of obtaining, etc.), the transport and the environment of use needs to be among the considerations when a bootstrapping service is being implemented.

#### 5.10.2.3.2 HMAC Computation for Bootstrap

The HMAC for the DM Bootstrap Message is calculated in the following way:

First, the bootstrap document is encoded in the WBXML format [WBXML1.1], [WBXML1.2] or [WBXML1.3]. The encoded document and the shared secret are then input as the data and key, respectively, for the HMAC calculation [RFC2104], based on the SHA-1 algorithm [SHA], as defined in the WTLS specification [WTLS]. The output of the HMAC (Digest= HMAC-SHA(K,Msg)) calculation is encoded as a string of hexadecimal digits where each pair of consecutive digits represent a byte. The hexadecimal encoded output from the HMAC calculation is then included in the security information.

The key type used in the HMAC computation MUST be one of the following: (see section 5.10.2.3 for details)

- NETWORKID
- USERPIN
- USERPIN_NETWORKID
- A non-standard key type

The transport protocol used to send the Bootstrap Message to the device MUST be capable of transporting both the HMAC value and the OMA DM bootstrap package.

If the Bootstrap Message is sent over OMA Push, the following rules apply:

- the Bootstrap Message MUST be in the body of the Push message
- the value of the *Content-Type* header MUST conform to the ABNF syntax given in the following textbox.

```
Value = MIMEtype ";" Keytype ";" Digest
MIMEtype = "application/vnd.syncml.dm+wbxml"
            ; Cannot use XML for bootstrap
Keytype = "NETWORKID" / "USERPIN" / "USERPIN_NETWORKID" / Other_Key
            ; Indicates the type of shared secret that is employed
            ; for the Bootstrap process.  Keys other than NETWORKID,
            ; USERPIN and USERPIN_NETWORKID may also be used
Other_Key = 1*VCHAR
            ; Keys other than NETWORKID, USERPIN and
            ; USERPIN_NETWORKID may also be used
            ; (RFC 5234 defines VCHAR as any visible character
            ;  i.e. any character with ASCII value in the range
            ;  0x21 through 0x7E)
Digest=40HEXDIG
            ; The digest is the HMAC value that is computed as per the
            ; method described in this section
```

### 5.10.2.4 Network Dependent Security

#### 5.10.2.4.1 3GPP_GBA

This method applies only to 3GPP Networks and devices that support GBA Push.

It is assumed that the DM Server has access to both a DM Bootstrap Integrity Key (DMBIK) and a DM Bootstrap Encryption Key (DMBEK) which have been derived from the long-term secret that is shared between the device smartcard and the network using the 3GPP Generic Bootstrapping Architecture (GBA) Push specifications [TS33.223].

The GBA Push procedures MUST be executed prior to sending the bootstrap message itself in order for terminal and network to agree on the DMBIK and DMBEK that SHALL be used to protect the bootstrap message. For more information on how a DM Server can interact with GBA Push see Appendix C.

The security method and combined integrity and confidentiality are passed as parameters to the content type in the format like this:

Content-Type: MIME type; SEC=type

Where:

MIME type is '*application/vnd.syncml.dm+wbxml*'

SEC = "3GPP_GBA"

GBA Push allows the generation of a so called Ks_(ext/int)_NAF shared secret both in the network and in the device. From this master key Ks_(ext/int)_NAF, two shared secrets are generated: the DMBIK and the DMBEK.

This 3GPP_GBA method requires:

- The NAF_Id SHALL be constructed using as FQDN the DM Server FQDN and as GBA Ua security protocol identifier the one defined for DM in Open Mobile Naming Authority (OMNA).
- An integrity and confidentiality protected bootstrap message using DMBIK and DMBEK shared secrets and derived from the Ks_(ext/int)_NAF using the GBA key derivation function (see Annex B of [TS33.220] as follows (see notation style and how parameters are used in Annex B of TS 33.220):
    - FC = 0x01
    - For DMBIK: P0 = "dmbik" (i.e. 0x64 0x6D 0x2D 0x62 0x69 0x6B)
    - For DMBEK: P0 = "dmbek" (i.e. 0x64 0x6D 0x2D 0x62 0x65 0x6B )
    - L0 = length of P0 is 6 octets (i.e. 0x00 0x06).

  The Key to be used in key derivation SHALL be:

    - Ks_(ext/int)_NAF

  In summary, the DMBIK and DMBEK and SHALL be derived from the Ks_(ext/int)_NAF and static strings "dmbik" and "dmbek" respectively as follows:

    - DMBIK = KDF (Ks_(ext/int)_NAF, "dmbik")
    - DMBEK = KDF (Ks_(ext/int)_NAF, "dmbek")

The protocol used to send the bootstrap message MUST be capable of transporting the protected OMA DM bootstrap package.

# Appendix A. Change History (Informative)

## A.1 Approved Version History

| Reference | Date | Description |
|---|---|---|
| N/A | N/A | No prior DM 1.3 version |

## A.2 Draft/Candidate Version 1.3 History

| Document Identifier | Date | Sections | Description |
|---|---|---|---|
| Draft Versions<br>OMA-TS-DM_Security-V1_3 | 15 Oct 2008 | All | Baseline to v1.3 using OMA-TS-DM_Security-V1_2_1-20080617-A. |
| | 06 Jul 2009 | All | Applied<br>OMA-DM-DM13-2009-0031-CR_Add_Bootstrap_XML_Security<br>OMA-DM-DM13-2009-0019R05-CR_Secure_Server_Initiated_Bootstrap<br>OMA-DM-DM13-2009-0029R01-CR_Security_Cleanup<br>OMA-DM-DM13-2009-0044-CR_Transport_Specific_Security. |
| | 25 Nov 2009 | All | Applied<br>OMA-DM-DM13-2009-0108R02-CR_Security_Cleanup |
| | 12 Dec 2009 | 5.6 | Applied<br>OMA-DM-DM13-2009-0115R01-CR_Integrity_Encryption_Ordering. |
| | 28 Dec 2009 | All | Applied<br>OMA-DM-DM13-2009-0126-CR_VerProto_Bug_Fix_for_DM_Security. |
| | 04 Feb 2010 | All | Applied<br>OMA-DM-DM13-2010-0012-CR_Security_SCR<br>OMA-DM-DM13-2010-0002R01-CR_Security_Reference_Fix.zip |
| | 11 Feb 2010 | All | General editorial clean-up of formatting by DSO |
| | 15 Mar 2010 | All | Changed all text to UK, reapplied OMA-DM-DM13-2010-0002R01-CR_Security_Reference_Fix. |
| | 14 Apr 2010 | All | Applied OMA-DM-DM13-2010-0053R01-CR_Security_SCR__Correction. Also added editorial notes to XML-security sections. |
| | 23 Apr 2010 | All | Applied OMA-DM-DM13-2010-0056R02-CR_SMIME_Security. |
| | 26 Apr 2010 | All | Creation by DSO of a clean version without change marks |
| | 05 May 2010 | All | Double quotes changed to single quotes<br>Font changed from Courier to Courier New for snippets where relevant to harmonize the fonts in the document |
| Candidate Version<br>OMA-TS-DM_Security-V1_3 | 25 May 2010 | N/A | Status changed to Candidate by TP<br>Ref # OMA-TP-2010-0221-INP_DM_V1.3_ERP_and_ETR_for_Candidate_approval |
| Draft Versions<br>OMA-TS-DM_Security-V1_3 | 25 Aug 2010 | 3.2, 5.3, 5.11 | Applied OMA-DM-DM13-2010-0085R04-CR_Bootstrap_Transport |
| | 01 Oct 2010 | All | Applied OMA-DM-DM13-2010-0112-CR-Security-TS-Cleanup |
| Candidate Version<br>OMA-TS-DM_Security-V1_3 | 07 Dec 2010 | N/A | Status changed to Candidate by TP<br>Ref #OMA-TP-2010-0502-INP_DM_V1_3_ERP_and_ETR_for_Candidate_re_approval |
| Draft Versions<br>OMA-TS-DM_Security-V1_3 | 28 Apr 2011 | 2.2, 5.11.2.3 | Applied OMA-DM-DM13-2011-0024R03-CR_DMSEC_Bootstrap_Cleanup |
| | 30 May 2011 | 5.11.2.3.1 | Applied OMA-DM-DM13-2011-0045-CR_Removal_Encryption_Bootstrap |
| | 04 July 2011 | All | Applied OMA-DM-DM13-2011-0049R01-CR_Remove_XMLEnc_XMLSig |
| | 20 Jul 2011 | 2.1, 5.4, 5.4.2, 5.4.3, | Applied OMA-DM-DM13-2011-0047R03-CR_SHA2_DigestAuth |
| | 12 Jan 2012 | All | Applied OMA-DM-DM13-2011-0135R03-CR_CONR_Security |

| Document Identifier | Date | Sections | Description |
|---|---|---|---|
| | 30 Jan 2012 | All | Applied 2012 TS template to SCR tables according to AI DM-2012-A008 + added reference to SCRRULES and deleted reference to IOPROC.<br>Applied 2012 template to introduction section.<br>AI DM-2012-A015: hard coded references changed into proper cross-references in the whole document, added new reference [RFC2045] in section 2.1, fixed a typo in App C |
| Draft Version<br>OMA-TS-DM_Security-V1_3 | 08 Feb 2012 | All | Applied OMA-DM-DM13-2012-0038-CR_CONR_Security_round_2 as modified by R&A DM13-12-009 comment ("OMA DM Servers" changed into "DM Servers" in 5.4) |
| | 14 Feb 2012 | 2.1, 5.4.2, 5.9.2.2 | Applied OMA-DM-DM13-2012-0052-CR_Smartcard_definition_in_Security<br>Removed two erroneous cross-references in 5.4.2 . |
| | 22 Feb 2012 | All | Applied<br>OMA-DM-DM13-2012-0072-CR_Security_CONRR<br>OMA-DM-DM13-2012-0075-CR_NextNonce_and_AAuthData |
| | 23 Feb 2012 | 4.1, 5.5, App B | Header removed by DSO according to Action Item DM-2012-A030<br>Applied OMA-DM-DM13-2012-0030R03-CR_HMAC_SHA256_Auth |
| Candidate Version<br>OMA-TS-DM_Security-V1_3 | 06 Mar 2012 | N/A | Status changed to Candidate by TP<br>Ref # OMA-TP-2012-0084-INP_DM_V1_3_ERP_and_ETR_for_Candidate_re_approval |

# Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in **[SCRRULES]**.

## B.1 SCR for DM Client

| Item | Function | Reference | Requirement |
|---|---|---|---|
| DM-SEC-C-001-M-O | Client MUST authenticate itself to a Server | Section 5.4 | |
| DM-SEC-C-002-M | Client MUST authenticate a Server | Section 5.4 | |
| DM-SEC-C-003-O | Support for transport layer authentication | Section 5.3 | |
| DM-SEC-C-004-O | Support for the different layer authentication | Section 5.3 | DM-SEC-C-004-O OR DM-SEC-C-007-O |
| DM-SEC-C-005-O | Send credentials to Server | Section 5.4 | |
| DM-SEC-C-006-O | Challenge Server | Section 5.4 | |
| DM-SEC-C-007-O | Support for application layer authentication | Section 5.3 | DM-SEC-C-008-O AND DM-SEC-C-010-O |
| DM-SEC-C-008-O | Support for OMA DM syncml:auth-md5 type authentication | Section 5.4 | |
| DM-SEC-C-009-O | Accept challenges from Server that has not yet been successfully authenticated | Section **Error! Reference source not found.** | |
| DM-SEC-C-010-O | Integrity checking using HMAC- SHA256 | Section 5.6 | DM-SEC-C-011-O AND DM-SEC-C-012-O |
| DM-SEC-C-011-O | Inserting HMAC in transport | Section 5.6.1 | |
| DM-SEC-C-012-O | Using HMAC for all subsequent messages | Section 5.6.1 | |
| DM-SEC-C-018-O | Bootstrap Security for Bootstrap via DM Profile | Section 5.10.2.1 | DM-SEC-C-019-O OR DM-SEC-C-020-O |
| DM-SEC-C-019-O | Transport neutral security for Bootstrap via DM Profile | Section 5.10.2.3 | DM-SEC-C-021-O |
| DM-SEC-C-020-O | Transport layer security for Bootstrap via DM Profile | Section 5.10.2.1 | |
| DM-SEC-C-021-O | Use of NETWORKID, USERPIN, or USERPIN_NETWORKID when Bootstrapping via DM Profile | Section 5.10.2.3 | |
| DM-SEC-C-022-M | Support of NETWORKID method in Bootstrap via DM Profile | Section 5.10.2.3.1 | |
| DM-SEC-C-023-M | Support of USERPIN | Section | |

| Item | Function | Reference | Requirement |
|---|---|---|---|
| | method in Bootstrap via DM Profile | 5.10.2.3.1 | |
| DM-SEC-C-024-O | Support of USERPIN_NETWORKID method in Bootstrap via DM profile | Section 5.10.2.3.1 | |

# B.2 SCR for DM Server

| Item | Function | Reference | Requirement |
|---|---|---|---|
| DM-SEC-S-001-M | Different password for each client | Section 5.1 | |
| DM-SEC-S-002-M | Support for client authentication at the transport layer | Section 5.3 | |
| DM-SEC-S-003-M | Send credentials to client | Section 5.3 | |
| DM-SEC-S-004-O | Challenge Client | Section 5.3 | |
| DM-SEC-S-005-O | Support for clients authentication at the application layer | Section 5.3 | DM-SEC-S-006-O AND DM-SEC-S-009-O AND DM-SEC-S-010-O |
| DM-SEC-S-006-O | MD5 challenge to client | Section 5.3 | |
| DM-SEC-S-007-O | MD5 challenge to client in conjunction with transport layer security | Section 5.3 | |
| DM-SEC-S-008-M | Supply of a new nonce with one more challenge if authentication fails | Section 5.4.3 | |
| DM-SEC-S-009-O | Using new nonce for each new session | Section 5.4.3 | |
| DM-SEC-S-010-O | Accept challenges from clients that have not yet been successfully authenticated | Section **Error! Reference source not found.** | |
| DM-SEC-S-011-O | Integrity checking using HMAC-SHA256 | Section **Error! Reference source not found.** | DM-SEC-S-012-O AND DM-SEC-S-013-O |
| DM-SEC-S-012-O | Inserting HMAC in transport | Section 5.6.1 | |
| DM-SEC-S-013-O | Using HMAC for all subsequent messages | Section 5.5.1 | |
| DM-SEC-S-022-O | Bootstrap Security for Bootstrap via DM Profile | Section 5.10.2.1 | DM-SEC-S-023-O OR DM-SEC-S-024-O |
| DM-SEC-S-023-O | Transport neutral security for Bootstrap via DM Profile | Section 5.10.2.3 | DM-SEC-S-025-O |
| DM-SEC-S-024-O | Transport layer security for Bootstrap via DM Profile | Section 5.10.2.1 | |

| Item | Function | Reference | Requirement |
|---|---|---|---|
| DM-SEC-S-025-O | Use of NETWORKID, USERPIN, or USERPIN_NETWORKID when Bootstrapping via DM Profile | Section 5.10.2.3 | |
| DM-SEC-S-026-M | Support of NETWORKID method in Bootstrap via DM Profile | Section 5.10.2.3.1 | |
| DM-SEC-S-027-M | Support of USERPIN method in Bootstrap via DM Profile | Section 5.10.2.3.1 | |
| DM-SEC-S-028-O | Support of USERPIN_NETWORKID method in Bootstrap via DM profile | Section 5.10.2.3.1 | |

# Appendix C. DM Interaction with GBA Push (Informative)

This section describes the possible interactions between the DM Server and GBA according GBA Push specification [TS33.223]. There are two possible ways of interacting with the GBA infrastructure in order to obtain the necessary key material for protecting the bootstrap message.

The figure below illustrates the straightforward approach where the DM Server has to additionally implement the NAF functionality according to GBA Push [TS33.223] and also some mechanism to decide whether a particular device is GBA Push enabled.

This NAF functionality would:

- interface the BSF to obtain the GPI
- derive the integrity and confidentiality keys DMBIK and DMBEK
- Send the GPI to the device in order to trigger key derivation

This deployment has a larger impact on the DM Server since new interfaces need to be implemented towards GBA and device.

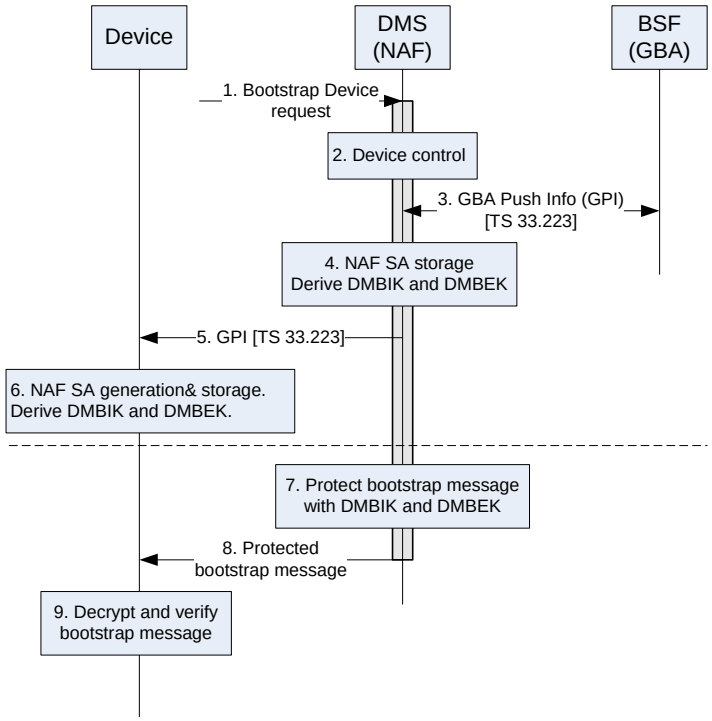


**Figure 1: DM-GBA interaction with added NAF into the DM Server (DMS)**

The figure below shows an alternative proxy NAF implementation that offloads the DM Server from GBA specifics. In this case, an external entity handles all the additional logic of BSF interfacing, key derivation and device triggering. The DM Server would use an interface to receive a request for bootstrapping a device with the attached integrity and confidentiality keys DMBIK and DMBEK. Such an interface is out of the scope of this specification.

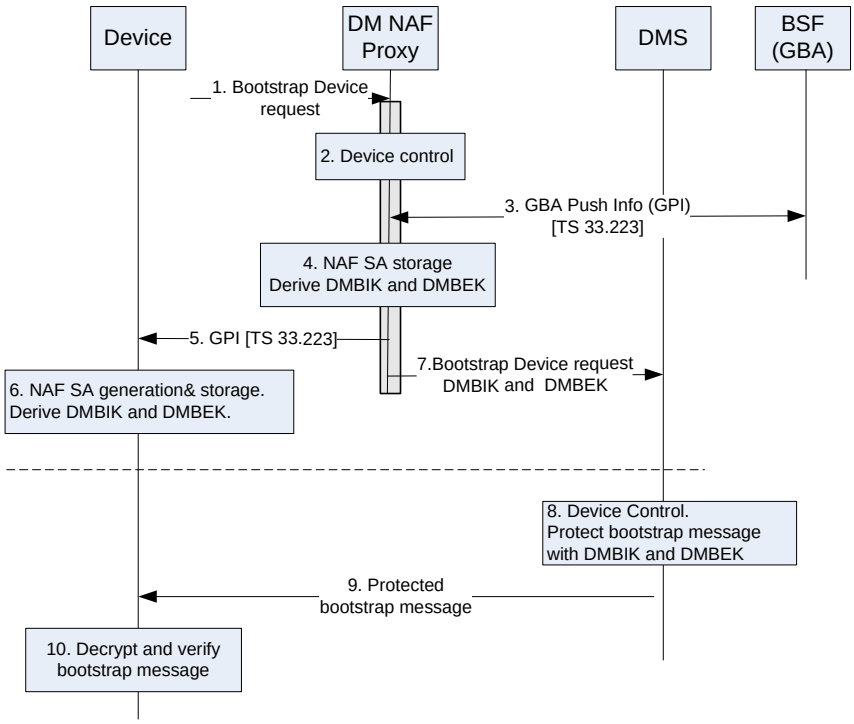


**Figure 2: DM-GBA interaction via external NAF**