European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Large Scale Integrating Project

EXALTED

Expanding LTE for Devices

FP7 Contract Number: 258512

WP5 – Security, authentication, provisioning

Deliverable 5.2

Solutions for Broadcast / Multicast and Device Management

Contractual Date of Delivery:	Feb 28th 2013
Actual Date of Delivery:	Apr 3rd 2013
Responsible Beneficiary:	Gemalto
Contributing Beneficiaries:	CEA, Gemalto, TKS, Unis, Vodafone
Estimated Person-Months:	18
Security:	Public
Nature	Report
Version:	1.0

Document Information

Authors

Name	Organisation	Email
Nick Bone	Vodafone	Nick.Bone@vodafone.com
Herve Ganem	Gemalto	Herve.ganem@gemalto.com
Bojana Jakovljevic	TKS	bojanaja@telekom.rs
Gorica Nikolic	TKS	gorican@telekom.rs
Aleksandar Obradovic	TKS	aleksandarob@telekom.rs
Shahab Mirzadeh	UNIS	s.mirzadeh@surrey.ac.uk
Oualha Nouha	CEA	nouha.oualha@cea.fr
James Raeburn	Vodafone	james.raeburn@vodafone.com

Approvals

	Name	Organisation	Date	Visa
Internal Reviewer 1	Javier Valiño	TST	02/27/2013	OK
Internal Reviewer 2				
Internal Reviewer 3				
Technical Manager	Pirabakaran Navaratnam	UNIS	02/28/2013	OK
Project Manager	Djelal Raouf	SWIR	03/04/2013	OK

Executive Summary

This deliverable is dealing with the security aspects of Machine to Machine (M2M) device management/provisioning via broadcast/multicast.

In a first step, the motivations for using broadcast/multicast techniques to remotely manage/provision M2M devices are explained and the expected benefits, primarily related to enhanced scalability described.

A security analysis, identifying the threats associated to broadcast/multicast techniques, results in the definition of a set of security requirements.

The topic (device management/provisioning via broadcast/multicast) is decomposed into 2 distinct problems addressed separately:"**device management**" and "**dissemination**"

The "**dissemination**" problem relates to the distribution of device management commands or provisioning data to a large number of devices. An overview of candidates broadcast/multicast protocols or services is provided.

The device management problem relates to the identification of solutions suitable to actually manage a large number of M2M devices. The report extends the work of Work Package 4 on device management and describes promising device management solutions, including two solutions dedicated to efficient software distribution in capillary networks.

The second part of the report contains the description of original solutions proposed by WP5 workgroup:

In 3GPP networks (2G, 3G, 4G networks), the Multimedia Broadcast and Multicast Service (MBMS) is proposed as a dissemination solution in conjunction with OMA device management solutions and we investigate the security aspects of this proposal. The Generic Bootstrapping Architecture (GBA) is at the core of the MBMS security and a lightweight version of the GBA is proposed, based upon CoAP (Constrained Application Protocol) mapping, replacing the traditional HTTP mapping.

Considering device provisioning in generic IP networks, this report examines security solutions to remotely manage devices located behind a gateway. In this case the specificities and the threats associated to the LAN part of the transmission (transmission of data behind the gateway to the devices) will deeply influence the global solution.

WP5 contributions on this topic are focused upon 3 points:

- How can a capillary device bootstrap its security with a remote device management server located in a wide area network when broadcast/multicast is being used. In particular we discuss how The ETSI (European Telecommunications Standards Institute) M2M architecture may be enhanced via the definition of new service capability layers enabling to bootstrap broadcast security.
- The initial security bootstrap usually results in the secure transmission of one or several group keys to all devices candidates to receiving data. A solution is proposed (based on the use of Key Derivation Keys) to minimize the key distribution for applications involving multiple group keys.
- Group keys have usually to be renewed whenever devices leave or enter the distribution group, possibly resulting in significant overhead. Solutions are proposed to reduce this overhead.

Table of contents

1 Introduction

The term Broadcast originates from the combination of the adjective “broad” and the verb “cast”. This combination expresses clearly the underlying idea: perform a large diffusion of some information from one source to multiple recipients.

This terminology was originally adopted by radio engineers as broadcasting was initially used to distribute analogue radio signals carrying voice or image communications. The radio spectrum is ideally suited to spread a signal throughout space.

Sometimes however there is a need to distribute the information only to a defined subset of the possible receivers. This is the purpose of “data multicast”.

Multicasting protocols generally use a data distribution structure to limit the diffusion of the messages to a subset of receiver nodes. However, the distribution and processing of this maintenance structure does not come for free and generally produces an overhead which in some situations can be overwhelming. It may actually be more efficient to use network wide broadcast even in order to distribute the information to a few selected nodes as explained in [1].

Broadcast or multicast communication modes are essentially push technologies enabling the asynchronous transmission of information from the source to the receivers. Push communications are well suited to support event based communications involving the unsolicited ”push” of information to the receiver upon the occurrence of an event. An alternative model to push communication is the pull model achieved with bidirectional communications where each recipient periodically poll the source to check for pending data and request transmission.

As M2M applications and more specifically Internet of Things applications are being deployed, the number of devices connected to public or private wireless networks increases sharply. Some of the devices may be very small and constrained either in computing power or in energy. There is need to identify lightweight, scalable, remote management and provisioning solutions to manage them. EXALTED has proposed a number of options and solutions for M2M devices management, and also addressed the problem of scalability of those solutions.[2]. This report will build upon and extend this work.

One important aspect of M2M device management is the distribution of new software/firmware updates. This involves the dissemination of the **same** data to a large number of recipients. However the fact that recipients may be small and energy constrained gives a particular flavour to the problem and push to investigate solutions eliminating redundancy from data transmission. As a matter of fact, broadcast/multicast techniques have been precisely designed for efficiently disseminating the same message to a possibly very large number of destinations.

Using those techniques to distribute the same commands/data to a lot of devices may therefore be attractive. However we need to make sure data is transmitted securely. This is precisely what this report is investigating.

In order to address the security of device management and provisioning via broadcast/multicast, Chapter 3 provides a security analysis. Section 3.1 lists the threats to be addressed and section 3.2 derives the corresponding security requirements.

Chapter 4 provides an overview of existing broadcast protocols and services. IP (Internet Protocol) Multicast, application and overlay multicast are discussed. We also provide an overview of the Multimedia Broadcast and multicast service (MBMS) in 3GPP networks and describe the underlying security mechanism based upon the General bootstrapping

architecture (GBA) defined by 3GPP. Although chapter 4 contains a mere description of existing solutions, It was felt appropriate to do so as chapter 6 is proposing to use the MBMS in 3GPP networks to perform the distribution of device management commands/updates.

Regarding device management/provisioning, section 5.1 provides an overview of (some) candidates protocol including solutions proposed by EXALTED, and the solutions under definition by the Open Mobile Alliance (OMA). Also part of this section, two protocols specifically designed to perform efficient distribution of software updates in energy constrained capillary networks: **Deluge and trickle**.

Chapter 6 contains the description of original solutions proposed by the group.

Section 6.1 investigates device provisioning occurring in 3GPP networks and proposes to use the MBMS to disseminate one single device management payload (possibly a software update) to a large number of nodes. For that purpose and in order to cope with small and constrained devices, we propose to implement the security with a lightweight version of the GBA using CoAP rather than HTTP as the transport protocol. A sample HTTP based GBA exchange is compared with the CoAP equivalent in order to show the interest of the proposal.

Section 6.2 is considering the more global picture of multicasting device management data from a central node located in a wide area network (typically a device management server located on the Internet) to devices located in a capillary network **behind a gateway**. The need to reach devices behind the gateway requires taking into account specific requirements linked to the LAN part of the transmission

Section 6.2.1 considers the simplest way to achieve that goal: combine existing WAN specific device management solutions with solutions specifically designed for LANs after a protocol translation in the gateway. In this type of solution, the gateway is running specific application software to perform the protocol translation. One benefit of this approach is its compatibility with non IP based capillary networks.

Section 6.2.2 considers end to end solutions to manage and provision capillary devices through IP networks via broadcast/multicast from a central server, and more specifically solutions which enable to use a ”generic” gateway supporting only standard protocols without the need for a specific gateway application. CoAP is a popular solution in 6LoWPAN capillary networks and EXALTED has already proposed a device management solution based upon CoAP. Furthermore the upcoming lightweight OMA protocol is also based upon CoAP. We propose therefore the use of CoAP in conjunction with IP multicast to disseminate the device management payload from the device management server directly to 6LoWPAN devices. The security implications of this proposal are examined. This leads to address two specific problems: Authenticating the source of the transmission and achieve an early detection of a bogus or corrupted broadcast. In the later case, the solution proposed is an adaptation of a solution already proposed in the literature [72]

A common approach for protecting information transmitted via broadcast/multicast typically involves the dissemination of a group key to all devices (possibly a very large number) part of the transmission group. In some cases multiple group keys may be involved according to the protection scheme wanted.

Section 6.2.2.3 is investigating the problem of reducing the group key distribution overhead using Key derivation keys for application involving the distribution of several group keys.

We also investigate how to cope with group member leaving (or excluded from) the group and propose a cluster based key distribution system witch avoid the costly operation of redistributing the group key to the whole group after the departure of one group member.

The security solutions presented in chapter 6 above often assume that an initial security bootstrap has been performed between M2M devices and a remote provisioning server.

Chapter 6.2.2.3 considers the possibility to facilitate the development of M2M applications using broadcast/multicast by enhancing the ETSI architecture with a broadcast service capability. This service capability could be made available by M2M services providers to their customers. This chapter presents one step towards that goal, focusing only on the security aspect of this service.

Finally, chapter 8 is providing an overview of the work schedule in standardization groups.

2 EXALTED architecture

The EXALTED System Architecture as defined in Deliverable D2.3 [3] is depicted in Figure 2-1

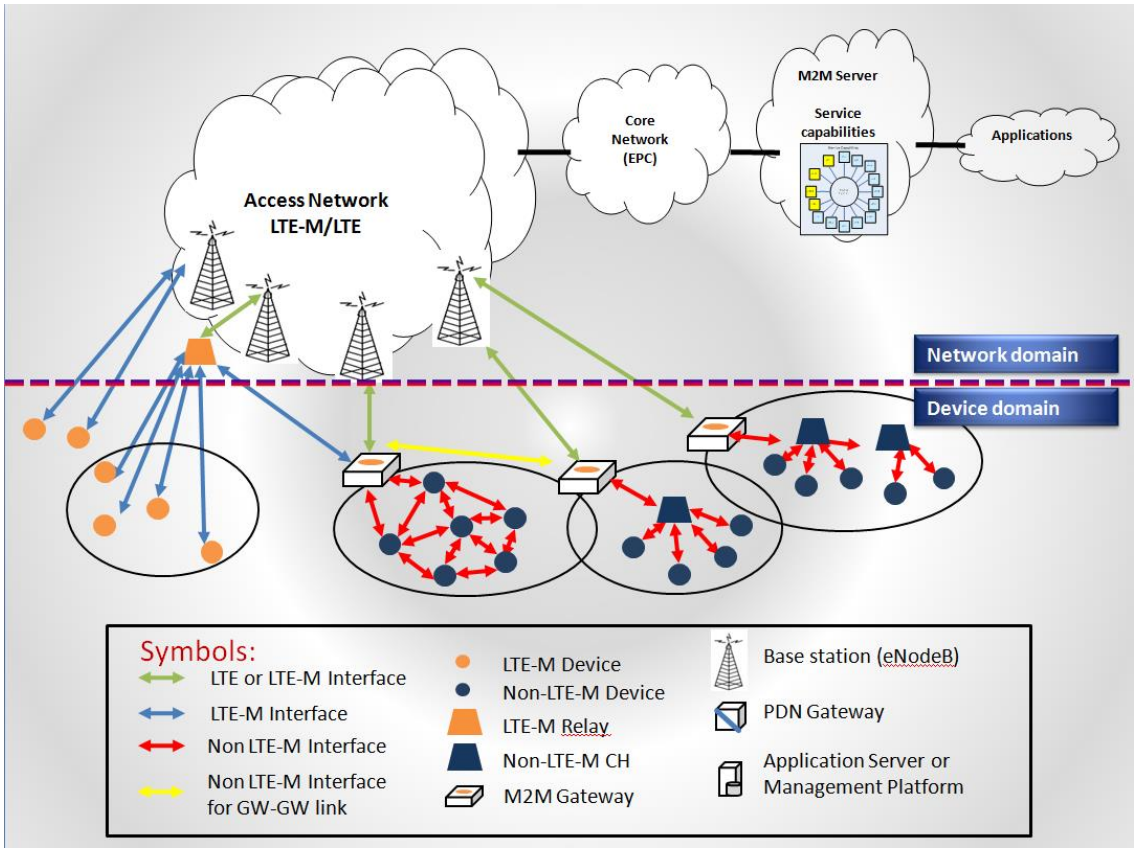


Figure 2-1: EXALTED System Architecture

The System Architecture comprises 2 main domains: the Network Domain (ND) and the M2M Device and Gateway Domain (DD).

In this memo, we consider two options for the wide area access network

1. The wide area access network is a 3GPP (3rd generation Partnership Project) network. It can be a 2G, 3G, 4G including LTE-M network, and will justify the use of architecture elements specified by 3GPP
2. The wide area access network is a “Generic” IP access network. It is not necessarily a 3GPP network although it may be so. In this case the solution will be based on IP protocols and will not use 3GPP defined architecture elements.

The entities considered in the device domain will result from the choice of the access network. (For example, considering an LTE-M gateway only makes sense if the access network is a 3GPP network).

2.1 EXALTED Broadcast/multicast use cases

Two uses cases justifying the use of broadcast/multicast have been identified in EXALTED Work Package 2 [33]:

- Vehicle collision management :

This scenario is addressing pre-crash sensing warning and collision detection, including the use of car sensors, and involves broadcasting collision information to vehicles nearby the collision. The scenario envisages the collision warning alert to be sent by the back end system, but a variant of this scenario may see the vehicles themselves broadcast the collision warning message to nearby vehicles. In both cases, the driving factor for using broadcast lies in the fact that there is no list of intended recipients for the message. The need is to push the message as quickly as possible to all recipients in a given area, and cannot be fulfilled easily with point to point communications.

- Using broadcast to send software updates to a large number of devices.

In this case the need is to push the same software update to multiple devices as efficiently as possible. A typical implementation of this scenario will involve a Device management server sending information over an IP based wide area network to devices located behind a gateway in a capillary network. The last part of the transmission may involve the software update being disseminated via broadcast within the capillary network. A driving factor to do so is that the dissemination via broadcast may be more energy efficient than a point to point transmission of the information to devices. The Capillary networks rely on wireless technologies (e.g., ZigBee, Wi-Fi, Bluetooth, LTE-M) that are inherently broadcast in nature, this means that if a node transmits a packet to a destination node, the packet will be overheard by many nodes in the surrounding of the sender. If the packet is intended for multiple recipients, unicast communication mode results in unnecessary expenditure of node energy since the same packet is sent multiple times. Unicast communication is particularly costly for recipients if nodes are in synchronous duty cycles (i.e., they wake-up simultaneously to listen to incoming packets). Using broadcast or multicast mode instead of unicast mode allows reducing the number of duplicate packets transmitted on the same physical link, and thus the network bandwidth and the power consumed by nodes are also reduced. The following sections will focus on this particular application of broadcast/multicast.

3 Security analysis

This report is mainly dedicated to securing device provisioning using broadcast or multicast techniques. In order to define the security requirements to be addressed we start by an analysis of the threats specific to those techniques.

3.1 Threat analysis: Broadcasting/Multicasting

Broadcast/multicast communications are prone to various potential threats. In particular the threats against network broadcasting have already been analysed in some detail by 3GPP, in particular connection with the “Multimedia Broadcast and Multicast Service” (MBMS), [16][17]

A detailed threat analysis is contained in the stage 3 document: TS 133 246 [30]. The threats can be grouped into attacks on the radio interface versus attacks on other part of the broadcast system. The main threats are similar in each case, and involve:

- Unauthorized access to data;
- Unauthorized access to services;
- Unauthorized insertion of user data and key management data;
- **Data eavesdropping and privacy violation**: radio communications are open by nature and wire communication may be intercepted. Any device with a transceiver can eavesdrop on ongoing transmissions, inject spurious messages, or block the transmission of legitimate ones. The device is able to track the transmissions of a particular node in order for example to learn private and contextual information about the user to which node is attached (e.g., identity, interests, location).
- **Threats to integrity**: If data is being intercepted, then intruders may be able to modify data or content.
- **Denial of service (DoS) attacks**: these attacks aim at node resource depletion (either at routers or at the edge nodes); thus reducing the network lifetime. Several DoS attacks are possible in broadcast/multicast communication mode:
    - Endless re-broadcasting: broadcast/multicast solutions generally limit re-broadcasting through overhearing approaches or using multicast group lists. Nevertheless, an attacker may continuously request the same message that it has already received. To mitigate this type of attacks, a common mean is to use an incremental counter at each router that gives the number of request messages for the same data by the same node. If the counter reaches a given threshold, its requests will not be answered.
    - Flooding the network with bogus data: an attacker may send large amount of bogus data pretending they come from the control system. In addition to depleting node resources consumed while receiving and processing these data, this type of attacks may lead also to erroneous commands in an actuator network for example.
    - Jamming attacks: jamming attacks aim at disrupting wireless transmission. The attacker simply listens to the open radio medium and broadcasts in the same frequency band as the network using a high transmission power signal that corrupts a communication link. The attacker may also launch selective jamming attacks in which it targets specific packets (e.g., ACK) or nodes.
- **Generating traffic keys for malicious use**: A malicious device may query the traffic key generation function for keys to use later on in an attack (e.g. for an unauthorized data insertion attack).

- **Routing attacks**: these attacks target the reliability of communications.
  - Insider attackers can refuse to forward packets to nodes.
  - Outsider attackers may target a particular node in the network. They may collude to target the same node, for example using flooding attacks. If this node plays an important functionality in communications (e.g., router, branching node), it will not be able to forward packets. This kind of attacks may degrade network reach ability, since some edge nodes may be isolated from the network, and data cannot be forwarded through them.
- **Man-in-the-middle and impersonation attacks**: an attacker may step between the control system and the node, and send bogus messages. On one side, it impersonates the control system and sends erroneous commands to nodes, and on the other side, it impersonates some nodes and sends false measurements to the control system.
- **Privacy violation**: The user identity could be exposed to the content provider, and open the possibility to track a user link a user by linking a user identifier to content.

3.1.1 Relevance of those threats to M2M?

It will be seen that many of these threats assumed a conditional access or “pay TV” model, rather than a “free-to-air” model. There is a significant concern about users having unauthorized access to content (such as watching TV or films without having paid for them).

The relevance of conditional access to LTE-M and M2M is less clear. One of the main use-cases for broadcast in M2M is delivering software updates, firmware upgrades etc. to machines, and it is not usually a concern that people who have not paid for these updates will see them. The concern is to get them out to as many machines as possible as quickly as possible. In many cases the device manufacturer, Operating System vendor etc. will have published a “patch” to their web-site anyway, so the software is not confidential. If it is being distributed to millions of machines, it should be assumed to be non-confidential (it is likely at least one recipient “leaks” the patch).

Still there are legitimate concerns with establishing a truly “free to air” broadcast facility across a cellular network, since that would then be available to non-subscribers of the network as well (such as unattached devices, devices in “limited service” mode which have failed authentication, or are not allowed to roam onto the visited network). It is hard to see why any cellular operator would launch such a service as a free public good (rather than as a value-add for its own customers), except in the case of an emergency broadcast system (see section A1.2.2.1).

Also, suppose an M2M Application Provider (someone delivering an M2M service) wants to distribute an application update to its own devices (a particular set of smart meters, environmental monitoring sensors, health devices etc.) through a public wireless network. The cellular network operator operating this network will presumably charge for such a broadcast event. The M2M application provider may not want every other M2M application provider to benefit from it (especially not commercial rivals who may be using the same sorts of devices). There is therefore a risk that no-one wants to pay for the broadcast, or that the smaller application providers free-load on the larger ones (who feel they have to broadcast anyway, since the costs of point-to-point update across so many devices are prohibitive, and the updates will take too long).

Some restricted access modes should therefore be supported, where only subscribers of a network get a key to recognize/decrypt the broadcast, or only targeted machines/devices for a particular application owner. However, the heavy security overhead of pay TV solutions involving custom smart-cards, set-top boxes, etc should be avoided. In particular Digital

Right Management concerns (trying to prevent the recipient from leaking the content) usually will not be relevant.

If an update is needed for a very large number of machines (e.g. to smart meters across a country), then putting the update on a true broadcast channel accessible to all subscribers of the LTE-M network makes sense. If it is going to fewer, targeted machines, then multicast to a closed group (in those cells with large numbers of target devices), combined with unicast to the remainder (in cells with only one or a few target devices) may make more sense. Individual application owners could then pay to add their own machines to an existing group, or pay to set-up their own group.

Since the MBMS architecture supports both broadcast and multicast use cases, and also considers threats around data integrity and source authentication (which are highly relevant to software update), it seems to be rather a good fit.

3.2 Security requirements

3GPP documents have identified security requirements for the broadcast and multicast solution. The Stage 1 document TS 122 146 [31] describes some high level security requirements (including availability). Another Stage 1 document TS 122 246 [32] also identifies some security and privacy requirements. A list of generic requirements for the security of broadcast/multicast systems has been compiled starting from these requirements, and considering the list of threats presented in section 3.1.

This list of requirements is shown on Table 3-1. The word “Service” should be understood as “broadcast/multicast service”.

Table 3-1: Requirements on Authentication

#	Description
R1.	If provided by a service provider, broadcast or multicast Services shall be available to all receivers that have registered to the service
R2. .	Receivers of service data should be authenticated before receiving a specific Service data.
R3.	Authentication should be performed not only at receivers, but also at intermediate nodes (i.e., routers, forwarding nodes). • *End-to-end authentication:* the receiver should be able to verify the source and the integrity of received messages before processing them. • *Hop-by-hop authentication:* due to the resource constrains of nodes in the capillary network, forwarding nodes should allow only legitimate communications to pass through. Packets originating from unauthenticated nodes should be dropped and not forwarded further in the network. Therefore, network access control should be provided in the network.
R4.	The origin of broadcast/multicast messages should be authenticated.

Table 3-2 : Requirements on Privacy

#	Description

R5.	Privacy : The User/device identity with the broadcast/multicast service provider should not be exposed to the content provider

Table 3-3 : Requirements on Key management

#	Description
R6.	The transfer of keys between the key generator and the receiver device shall be confidentiality protected
R7.	The transfer of the keys between the key generator and the receiver device shall be integrity protected.
R8.	It should be possible to perform re-keying as frequently as it believes necessary to ensure that: • receivers that have joined a Service, then left, shall not gain further access to the Service • Receivers joining a Service shall not gain access to data from previous transmissions in the service. • The effect of subscribed receivers distributing decryption keys to others is controllable and can be prevented
R9.	Only authorized receivers that have joined a Service shall be able to receive keys delivered from the key generator
R10.	All keys used for the Service shall be uniquely identifiable. The identity may be used by the receiver device to retrieve the actual key (based on identity match, and mismatch recognition) when an update was missed or was erroneous/incomplete
R11.	The function of providing traffic keys to the terminal shall only deliver a traffic key if the input values used for obtaining the key were fresh (have not been replayed) and came from a trusted source

Table 3-4 : Requirements on integrity protection of Service data

#	Description
R12.	It shall be possible to protect against unauthorized modification, insertion, replay or deletion of Service data sent to the receiver device
R13.	The Service data may be integrity protected with a common integrity key, which shall be available to all receivers that have joined the Service

Table 3-5 : Requirements on protection of Service data

#	Description
R14.	It shall be possible to protect the confidentiality of Service data on the radio interface.
R15.	The Service data may be encrypted with common encryption keys, which shall be available to all receivers that have joined the Service.
R16.	It shall be infeasible for a man-in-the-middle to bid down the confidentiality protection used to protect the Service
R17.	It shall be infeasible for an eavesdropper to break the confidentiality protection of the Service when it is applied.

Table 3-6 : Requirements on Reliability

#	Description
R18.	Reliability: broadcast/multicast communication should be reliable, for instance through redundancy of routing devices and routes and dynamic routing in reaction to network outages or nodes failures or compromising. Retransmission of messages should be secured to avoid the abuse and misuse of network resources, for example acknowledgement messages should be integrity protected...

Table 3-7 : Requirements on energy efficiency

#	Description
R19.	Energy efficiency: lightweight cryptographic primitives (e.g., symmetric cryptography, hash) should be favoured for secure capillary networks. The overhead of the secure broadcast/multicast protocol in terms of number rounds and packet size should be optimized.

Those requirements will apply whenever part of the data transmission occurs on low power and lossy networks. The energy efficiency requirement applies then to that part of the transmission.

4 Broadcast/Multicast protocols

This section provides a state of the art review of broadcasting solutions. It attempts to answer the following questions:

- What are the well known protocols to achieve the information dissemination required for broadcast or multicast in infrastructure networks? What are the existing underlying security mechanisms?
- What broadcasting higher level systems are based upon those protocols?

We will start by listing low level protocols used to perform the required information diffusion and will work up the value chain by describing higher level services systems leveraging those low level protocol and applications built with those services.

A number of protocols have been proposed to achieve the diffusion of information required by broadcast or multicast in infrastructure networks

4.1 Broadcast and multicast protocols in IP networks

Multicast was initially implemented at the IP layer. However, IP multicast has seen slow deployment due to several issues discussed in [38]. These issues are related to the open model and complexity of functionality to be implemented by Internet Service Providers (ISP's) (intelligence at the core network), for examples, group and network management (authorization for group creation, receiver authorization, and sender authorization, billing and service charge), distributed multicast address allocation, and security. To resolve the deployment issues of IP multicast, other solutions based on application layer (intelligence at the end points) and overlay multicast (hybrid approach) have been proposed.

Three types of protocols will be described:

- IP multicast
- Overlay Multicast
- Application multicast

4.1.1 IP multicast

IP multicast requires a mechanism to build multicast distribution trees. Distribution trees define a unique forwarding path between the source's subnet and each receiver's subnet. They are constructed such as the multicast content is not forwarded more than one time at the same tree branch.

Several routing protocols have been proposed for IP multicast that can be divided into Dense-Mode (DM) protocols e.g., (Multicast Open Shortest Path First(MOSPF) [6], Distance Vector Multicast Routing Protocol (DVMRP) [7]), and Protocol Independent Multicast (PIM-DM) [8]) and Sparse-Mode (SM) protocols (e.g. Core Based Trees (CBT) [9] and PIM-SM [10]). DM protocols assume that the majority of routers in the network need to disseminate multicast traffic for each group; whereas SM protocols assume that only few routers in the network will need to distribute multicast traffic. The DM protocols are generally used in Local Area Networks (LAN) environments and SM protocols in Wide Area Networks (WAN) environments.

These routing protocols either use the current unicast forwarding tables at routers (e.g., PIM, CBT) or specific unicast routing tables (e.g., distance vector routing for DVMRP, link-state table for MOSPF). PIM is currently the most widely used protocol.

PIM-DM [8] uses the underlying unicast routing information to flood routers with multicast traffic. Routers that are not interested in the multicast traffic will prune the distribution tree.

On the other hand, PIM-SM [10] requires from routers to send PIM join messages to be able to receive the multicast traffic. Receivers join the group by sending join messages to their designated routers (i.e., gateway multicast router) using either IGMP protocol if IPv4 is used or MLD protocol if IPv6 is used. The designated routers send PIM join messages to the source's designated router, which allows distribution trees to be updated. The source sends the multicast traffic to a rendezvous point that forwards the traffic on the shared distribution tree. PIM-SM is the current standard for Internet Service Providers (ISPs) supporting multicast communications.

Like in the Internet, IP multicast can be applied to capillary networks. Messages are transmitted by setting IP address to a selected multicast IP group address. The multicast approach is efficient since it is realized at the IP layer where only new messages are routed via the same physical link.

The IPv6 routing protocol for low-power and lossy networks (RPL) [39] supports the IP multicast operation. Nodes are able to advertise their IP multicast groups of interest via RPL DAO (Destination Advertisement Object) messages. For multicast forwarding, as discussed in [40], RPL IP multicast may rely on local unicast communications at link-layer, or preferably, on link-layer 802.15.4 broadcast frames.

RPL supports three security modes:

- Unsecured: unsecured mode does not imply all messages are sent without any protection, since the network could be using other security mechanisms, such as link-layer security.
- Preinstalled: to join a RPL Instance as either a host or a router, a node must have a preinstalled key used to provide message confidentiality, integrity, and authenticity. These protections are applied over the unsecured IPv6 packet before any compression that lower layers may apply (mutable IPv6 fields are considered to be filled with zeroes).
- Authenticated: Compared to preinstalled security mode, the authenticated mode requires two types of keys: the first key (generally symmetric) is used by the host to join the network, and the second one (can be asymmetric) is used by the router. to join a RPL Instance as a host, a node must have a preinstalled key used to provide message confidentiality, integrity, and authenticity. To join the network as a router, a node must obtain a second key from a key authority. As described in the current RPL specification, the authenticated mode is not supported, since it requires asymmetric cryptography algorithms.

4.1.2 Overlay multicast

Overlay multicasting allows providing transmission of IP multicast over IP networks that do not support network layer multicasting. Generally, the approaches rely on middleware-based relay agents (e.g., [86] that provide multicasting by means of tunnelling IP multicast over IP unicast. The relay agents interact with subnet hosts using IGMP; thus, allowing them to transparently forward multicast messages as if they were interacting with network-layer multicast routers. The messages are then forwarded using IP multicast over IP unicast tunnels.

The overlay multicasting approximates the network-layer multicasting and may reduce the network traffic overhead produced by application-layer multicast whereby end host send unicast messages to multiple hosts. Even though, this type of multicasting can be promising for networks not enabled with IP multicasting, little effort has been put to specify and implement overlay multicasting.

With respect to security, overlay multicast network may be vulnerable to DoS, spoofing, and collusion attacks targeting communications between the end hosts and their corresponding

relay agents. Implementations should identify participating overlay relay agents and allow their authentication.

In constrained networks, overlay multicast rely on special nodes called proxies (or service nodes) different from end hosts. These nodes are generally more powerful than end host to support the routing functionality. They are strategically deployed within the network, such that receivers form clusters around their closest proxies. There are two types of proxies: forwarding and branching nodes. Forwarding nodes simply retransmit packets from one neighbour to another one. On the other hand, branching nodes play a role in the multicast operation by keeping state information about the group and other branching nodes and by duplicating packets.

Proxies should be carefully chosen to efficiently manage the multicast trees i.e., by avoiding transmitting the same data over the same physical link multiple times. Fortunately, the so-deployed overlay network could support multiple groups or/and applications.

For example, in the proposed scheme in [19], multicast trees are built dynamically after the receivers have joined the multicast group. This process is either sender-driven or receiver-driven. In the sender-driven way, the source sends the list of receivers for the group; whereas, the receiver appends the group ID to its join message. The overlay multicast solution focuses on providing reliability since it relies as transport protocol on User Datagram Protocol (UDP) for performance considerations. It ensures reliability through a simple NACK-based mechanism.

4.1.3 Application multicast

The simplest way to route data from one source to multiple destinations is to maintain a list of recipients and send the data independently to each recipient in unicast mode.

This technique is not very efficient in terms of network utilization as it involves many redundant transmissions of the same data. It has however the advantage of simply requiring from the infrastructure a mere point to point communication capability.

The price to pay for the redundancy of the data transmission increases with the size of the payloads and the number of recipients. It should be noted that very large networks such as twitter function with this principle.

In application-layer based multicast [11], multicast trees are built by the end nodes themselves (i.e., group members), instead of routers in IP multicast. Multicast data is delivered through unicast tunnels between nodes. Group management and multicast routing are handled by the end nodes in a distributed fashion; thus providing more efficient data delivery. Depending on the group management protocol, the application-layer based approach could be affected by a scalability issue. For example, the scheme proposed in [12] attempts to overcome this problem by building a low overhead overlay multicast approach by means of a layered and hierarchical approach. Approaches in particular, structured peer-to-peer overlay networks (e.g., [13]) are scalable and moreover ensure support path redundancy in their architecture to guarantee failure tolerance. Multicast routing could be also a detrimental issue in overlay networks. For this, approaches like [14] propose a topology-aware overlay network that exploits the underlying network topology information for a more efficient multicast tree construction.

With respect to security, application-layer multicast approaches may be prone to denial-of-service (DoS) attacks. These attacks are mitigated by means of user authentication at the edge of the overlay network; hence, only legitimate group members can access the overlay network. Authentication is sometimes combined with user anonymity (e.g., Onion routing, Tor [15]) to prevent traffic analysis and user profiling in the Internet.

In capillary networks, application-layer multicast solutions (refer to the survey in [41]) have emerged in order to support group-based applications without relying on IP multicast. In application-layer multicast, nodes arrange themselves to form a logical overlay network and transfer data within the overlay network.

As demonstrated in [42], these solutions inquire a large amount of control communication overhead to maintain multicast delivery structures, and they are generally recommended only for small groups.

4.2 Broadcast/multicast services

4.2.1 MBMS overview

4.2.1.1 Introduction

Mobile 3GPP networks have evolved from mere voice telephony networks into data and multimedia delivery networks; they are now on their way to address the problem of M2M data delivery to a very large quantity of M2M devices.

The MBMS story started back in 2002, when the delivery of mobile TV and multimedia services to mobile devices was seen to be a big business opportunity. Adding broadcast and multicast support to 3GPP networks was a way to address efficiently the situation where many users are receiving simultaneously the same data.

Standardization of the"Multimedia Broadcast and Multicast service" (MBMS) started in 2002. The First version of the standard was frozen in 2005 and finalized in 2006. A first prototype of MBMS was demonstrated in GSM world congress in 2007.

Today, with the development of M2M services, we see a similar need emerge: the need to distribute provisioning and software update information to a large number of similar M2M devices. MBMS can be an option to solve the problem of distributing the same software update to a large number of devices and this solution will be detailed later.

This section will first describe MBMS and its main features, focusing particularly on its security features.

MBMS focuses on the transport aspects only of broadcast and multicast services. It provides a transport bearer along with two transmission modes over which IP packets may be delivered:

MBMS broadcast mode: The broadcast mode is a unidirectional point-to-multipoint transfer of multimedia content from a single source entity to all users in a broadcast service area. Data is transferred over a common radio channel due to optimized usage of radio network. The broadcast traffic is not guaranteed and because of that the receiver should know data loss has occurred. The M2M operator will provision one or more broadcast services within capillary network. A broadcast area is configured separately for each broadcast service. Broadcast areas are independent of different broadcast services. The broadcast mode needs neither service subscription nor charging.

MBMS multicast mode: The multicast mode is a unidirectional point-to-multipoint transfer of multimedia content from a single source entity to a multicast group in a multicast service area. It is very similar to IP multicast. User Equipment (UE) wanting to receive particular information joins a multicast channel in order to subscribe to the information carried by this channel.

4.2.1.2 MBMS Architecture

Different MBMS user services can consist of these broadcast and multicast MBMS bearer services. MBMS does not require architectural changes to 3GPP architectures. It introduces

a new functional node which is introduced in the network; the **Broadcast and Multicast-Service Centre** (BM-SC). Also, MBMS controlling functions are added to the existing core network elements. Additional functionalities must be added for UE to support the MBMS.

A simplified MBMS architecture depicted in Figure 4-1: [16]

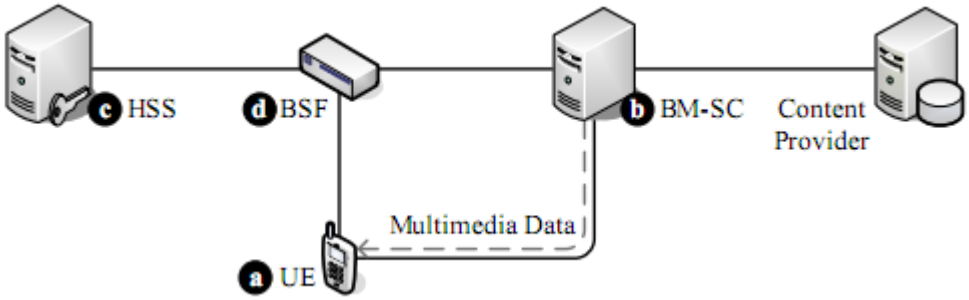


Figure 4-1: MBMS architecture

The BM-SC is usually presented as application server which acts as MBMS data source. The UE receives the MBMS application (also known as MBMS User Service) from the BM-SC. The BM-SC starts the establishment of the MBMS transmission bearer, after that it sends content to the joined UE (UE joining the multicast group for a specific MBMS User Service). The Bootstrapping Server Function (BSF) is a security server function responsible for establishing shared secrets between the BM-SC and the UE [16].

MBMS provides support for two types of services: streaming and download. Streaming will be most useful to implement mobile TV and multimedia services, while download will be used to download the same data to a large number of devices. For M2M purposes, we will be primarily interested by the data transfer aspect and we will focus on download services.

MBMS uses three errors recovery methods to cope with errors occurring in data transmission:

- The first one is passive and uses Forward Error Correction coding (FEC).
- The second is a point to point connection service to request retransmission of missing or erroneous part of the data download
- The third is a point to multipoint bearer to deliver missing data to several terminals simultaneously

4.2.1.3 MBMS Key Management

Figure 4-2 shows an overview of the MBMS security functions and data flows [17].

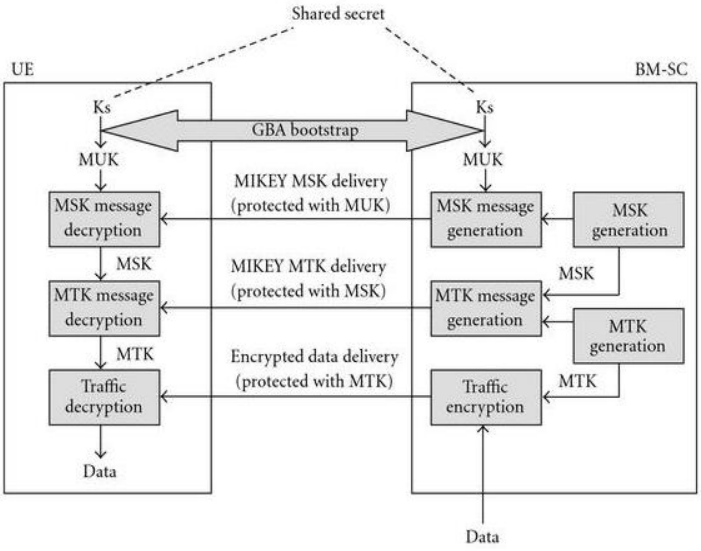


Figure 4-2 : MBMS Security Overview

MBMS will rely upon a hierarchy of keys to secure data transmission. The key management scheme is shown on Figure 4-2 and is based upon the use of SIM and USIM holding 3GPP credentials. Here is a brief overview of this scheme, in the anticipation of an improvement to it that will be proposed later.in the upcoming sections to propose an improvement to it.

- At first the 3GPP credentials used in the SIM/USIM are used in a GBA bootstrap process to generate a long term shared key: the MBMS User Key (MUK) between the UE and the BM-SC. GBA is a security bootstrapping process defined by the 3GPP. We provide an overview of the GBA in the next section. The MUK is really attached to a 2G/3G user subscription
- The MUK is used to protect the transmission via a Multimedia Internet KEYing (MIKEY) scheme of a shorter lifetime key: the MBMS Service Key (MSK). This key is attached to the subscription of the user to a particular MBMS service and will not be renewed if the user stops subscribing to the service.
- The MSK is used to protect the transmission from the BM-SC to the user terminal of an ever shorter lifetime key: The Traffic Encryption Key (TEK) used to cipher the data sent vie the MBMS service. Rekeying of the TEK occurs frequently whenever there is a need to avoid a subscriber passing the TEK to other users to provide them illegitimate access to MBMS data. MTK, protected by MSK is also transmitted from the BM-SC to the UE using a MIKEY scheme

4.2.1.4 GBA (Generic bootstrapping architecture)

The Generic Bootstrapping Architecture (GBA) specifies a framework enabling to leverage 3GPP authentication credentials and protocols to authenticate a terminal (UE) and define a shared key between this terminal and “Someone” implementing a so called “NAF” function .In the case of MBMS, the NAF function is implemented in the BM-SC and provide a convenient way for the BM-SC to leverage the 3GPP credentials stored in the SIM/USIM to authenticate the user and derive a shared key(The TUK) with the UE. This key will be used as explained above to derive in turn the MSK and the TEK.

A simplified GBA architecture, showing nodes and interfaces is shown on Figure 4-3

The following elements appear on this figure:

- The NAF and the UE: the purpose of the procedure is to enable the NAF to authenticate the UE based upon the AKA credentials and derive from this authentication a shared key between the NAF and the UE.
- The HSS: It is holding in a secure storage the subscriber AKA authentication credentials and provides authentication vectors which are the basis to craft AKA challenges.
- The BSF. This server is key to the GBA functioning. It will interface to the HSS to obtain authentication vectors, and communicates with the UE and the NAF.

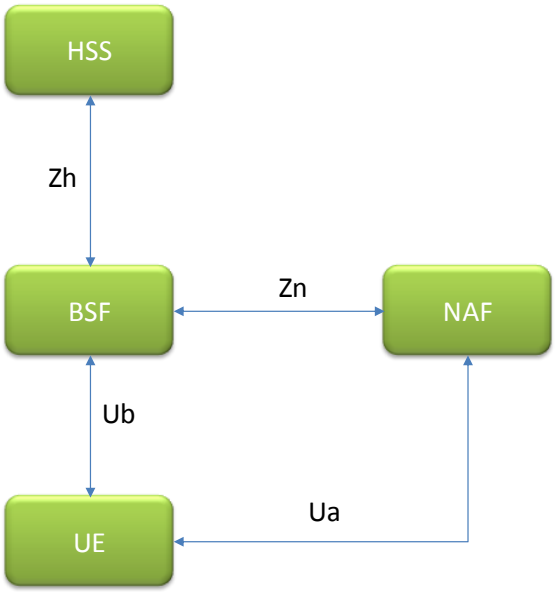


Figure 4-3 : GBA Architecture model

The GBA mechanism describes how the UE performs a security bootstrapping operation with the BSF via the Ub interface in order to define a long term key KS shared between the UE and the BSF. The derivation of this long term key is based upon an authentication challenge to the UE crafted by the BSF after obtaining authentication vectors form the HSS. The dialog between the HSS and the BSF to obtain those authentication vectors uses the Zh interface. At the end of the bootstrapping procedure, both BSF and UE have a security association that includes bootstrapping transaction identifier (B-TID) and a long term key material (Ks) [1].

In a second step the UE will be indicating to the NAF what key to use using the B-TID while NAF will be obtaining this key from the BSF.

Figure 4-4 provides an overview of the GBA authentication and key management process

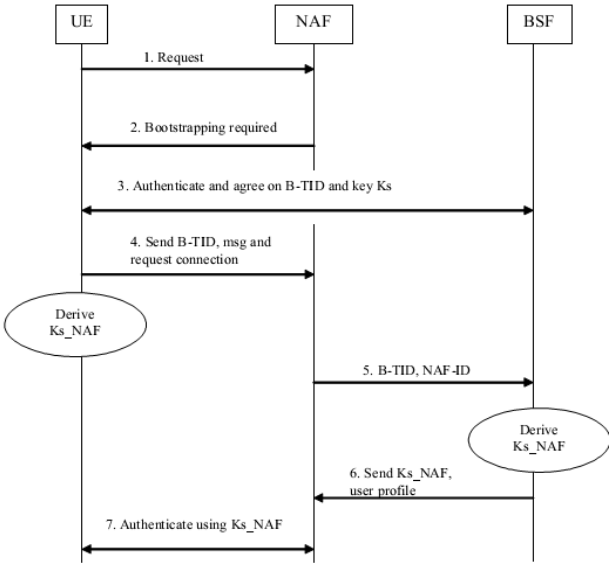


Figure 4-4: overview of GBA authentication and key agreement process

HTTP is commonly used as the transport protocol to carry the exchanges on the Ua and Ub Interfaces as this will be detailed later.

4.2.2 Secure broadcast in constrained networks

This section surveys multiple solutions to secure broadcast/multicast communications respecting the requirements discussed in the previous section. Solutions realizing source authentication of broadcast messages and group key management solutions to ensure confidentiality protection of broadcast messages are presented. Solutions that use broadcast communication for secure pairing are discussed. Finally, availability and reliability requirements for broadcast messages through secure message acknowledgements are discussed.

4.2.2.1 Secure authentication broadcast solutions

Broadcast messages intended for multiple recipients could be source authenticated using signatures. To avoid performing double verification of the signed message, nodes generally keep in their memory the public key of the source. To reduce the number of these public keys in multi-user broadcast authentication, a Bloom filter vector could be preloaded at sensor nodes by the gateway (or another entity, like a cluster-head or even a security server at the infrastructure) [43]. The Bloom filter vector is a space-efficient data structure that allows performing membership tests. It allows each node to efficiently verify the binding between the source node ID and its public key. Revocation of nodes is performed by the gateway that re-evaluates and broadcasts the Bloom filter vector signed to nodes.

Even though, the solution based on Bloom filters is efficient with respect to computation and storage resources, the maximum number of supported nodes is limited. Merkle hash trees [43] could be used instead and constructed from the bindings between the node IDs and their public keys. Still, both schemes rely on public-key cryptography for source authentication.

In contrast, the Timed Efficient Stream Loss Tolerant Authentication (TESLA) approach [44] relies on symmetric cryptography through the use of hash chain keys. Broadcast messages are authenticated based on the keys of the hash chain used in reverse order. In each broadcast message, the key used to authenticate the broadcast message is disclosed in a delayed time interval (e.g., in the next broadcast message). To improve over TESLA that authenticates the initial packet with a digital signature, which is too expensive for sensor

nodes, µTESLA [45] uses only symmetric mechanisms even for initial packet authentication. µTESLA still requires loose synchronization between nodes. Multi-level µTESLA [46] proposes to use multiple levels of key chains where low-level key chains are intended for authenticating broadcast messages, while high-level key chains are used to distribute and authenticate commitments of the low-level key chains. Loose time synchronization between nodes and the source is required in these schemes. Moreover, the delayed authentication of the messages may pave the way to energy-depletion DoS attacks.

When broadcast packets are known in advance by the source e.g., corresponding to pages from the upgraded software, the delayed authentication and time synchronization between nodes are not required if hash chains are used. Each broadcast packet may include a packet authentication segment and a key update segment, as in [47]. The packet authentication segment comprises a hash of the firmware packet along with a chain key. The key update segment allows authenticating the used key based on the key used in the previous sent packet. The first packet will be associated with the commitment hash chain key stored at sensors beforehand. To prevent man-in-the-middle attacks due to multi-hop communication, nodes are divided into different groups according to their hop distance from the gateway. Each node group uses a different hash chain. Packet authentication and key update segments associated with each group are included in the broadcast packet. With a large number of hops to the gateway (i.e., large number of node groups) and the lack of packet optimization, the scheme may become inefficient.

The authors in [48] propose several optimizations to hash tree and hash chain broadcast authentication techniques in different network topologies, namely linear, tree, and fully connected topologies; though they have assumed fixed topology classes, with the sender having, in some protocols, an already established knowledge of the network topology. In the hash tree based schemes, intermediate nodes can piggyback their values in the tree along with the broadcast message; thus the protocol requires only 2 passes. Partitioning the set of receivers into multiple subsets helps also reducing the number of passes. For hash chain based schemes, acknowledgement of receipt can be aggregated back to the sender based on a hash tree over acknowledgements or a simple XOR operation. The hash chain key can be encrypted and progressively decrypted until reaching leaf nodes. These nodes send the computed decryption key back upstream. In the same way, partitioning the receivers set reduces the number of passes in the protocol.

4.2.2.2 Group key management schemes for confidentiality protection

A primary method of limiting access to broadcast/multicast messages is through encryption and selective distribution of the keys used to encrypt messages. Then, the problem of confidentiality protection becomes a group key management issue.

Several solutions are proposed for group key management. Tree-based solutions like the Logical Key Hierarchy (LKH) approach [49] allow reducing the number of communication overhead during re-keying. In this approach, a rooted tree (from the bottom up) is generated with each leaf corresponding to one user. Each user establishes a unique pair-wise key with the server, using for example a public key exchange protocol. The pair-wise key is used to transmit keys (called, key encryption keys-KEKs) corresponding to those nodes above the leaf (i.e., path to the tree root). The re-keying process in the LKH approach is proportional to the tree degree (i.e., depth).

Another tree-based approach is the subset-difference trees used for example by AACS (Advanced Access Content System) [80] which is a standard for content distribution and digital rights management for pre-recorded DVD contents. The key provisioning approach is based on the broadcast encryption scheme. Each player belongs to a set of groups. It is provisioned with the keys associated with this set of groups. Content is encrypted under the title-specific key, which is itself encrypted under each group key. Thus, each disc contains a

collection of several hundred encrypted keys. Based on subset-difference trees, licensors are able to revoke individual players and trace the traitors, without re-provisioning legitimate players with new keys.

In a basic approach, MIKEY (Multimedia Internet KEYing) [52] is an IETF standard where group key management is realized through unicast communication. A group key, TGK (TEK Generation Key), and a set of security parameters are exchanged between the initiator and a receiver (in unicast mode). Several methods are proposed for key exchange based on either symmetric encryption (MIKEY-PSK), or asymmetric encryption (MIKEY-RSA, MIKEY-RSA-R), or Diffie-Hellman (MIKEY-DHSIGN, MIKEY-DHHMAC). From the TGK, TEKs (Traffic Encryption Keys) are derived locally by receivers for each session. TEKs and security parameters are used as an input to a security protocol e.g., Secure Real-time Transport Protocol (SRTP).

MIKEY is used by the MBMS Service specified in the 3rd Generation Partnership Project (3GPP) as a registration protocol and to convey keys and related parameters needed to secure multimedia streams. The RFC 4563 [53] specifies a new extension to MIKEY that consist of a new Type (the Key ID Information Type) for MIKEY General Extension Payload which provides indication of the type and the identity of the MBMS keys being carried in a MIKEY message. Three types are considered: MBMS Key Domain, MBMS Service Key (MSK), and MBMS Traffic Key (MTK). The MBMS user key (MUK) is used as a pre-shared key to run MIKEY with the pre-shared key method, and to deliver (point-to-point) the MSK. The same MSK is pushed to all clients, to be used as a group key. The MSK is used to push to all the clients an MTK, the actual group key that is used for the protection of the media traffic.

We propose here an extension to this allowing MTK rekeying using both unicast and multicast communications. MTK delivery is protected by a group MAC (Message authentication code), keyed by the group key (MSK), so the delivery is not source origin authenticated.

The adapted MIKEY (AMIKEY) protocol in [54] enables the original MIKEY protocol to efficiently and simultaneously manage the key parameters of multiple security protocols, in particular those related to low-power and lossy networks (LLNs). These protocols will share a common group key, but use different traffic encryption keys. With AMIKEY, communication overhead associated with used to generate keys for multiple protocols. However, the protocol renders more complex key management at the AMIKEY entity (i.e., the node) that has to manage several protocols with different considerations (in terms of security and performance).

5 Candidate technologies for device management and provisioning

This report is addressing the problem of securing M2M device provisioning via broadcast. This section intends therefore to provide a brief overview of candidate technologies that can be used for device provisioning.

OMA (Open Mobile Alliance) is currently defining device management solutions suited to M2M devices, and this is a major and important offering even if OMA solutions are **not** for the time being addressing transport via broadcast/multicast. A status of this work is provided below.

Two protocols which are also focused around the distribution of software updates in multihop capillary networks: Deluge and Trickle are also described. They are seen here as device provisioning solutions for capillary networks although they address as well and very much so the ”data dissemination” problem.

5.1 OMA device management

OMA (Open Mobile alliance) is not a newcomer in the area of device management and has already delivered specifications for device management solutions primarily aimed at managing mobile handsets. (OMADM 1.0, 1.1, 1.2) ([81][82])

OMA DM solutions address both the semantic aspect of Device management via the description of specific management objects, as well as the specification of the transportation mechanisms to carry device management objects. The goal of OMA Device Management is to address mobiles, smartphones, Machine-to-Machine (M2M) equipment and in general any device capable to connect to data network; it includes (but is not limited to):

- Setting, installation and management of initial and operational configuration information related to device capabilities and applications functionalities
- Firmware and software update
- Retrieval of management information from devices
- Processing events and alarms generated by devices
- Running diagnostic tests and monitoring tasks
- Controlling device capabilities and applications
- Controlling and managing how applications running on the devices uses and interacts with underlying capabilities

In the scope of Device Management, information includes (but is not limited to):

- Configuration settings
- Operating parameters
- Software installation and parameters
- Software and firmware Updates
- Application settings and interfaces
- User preferences

In order to address more specifically the management of M2M devices, OMA has undertaken 3 distinct actions:

- Definition of a specific object: The OMA gateway management object (GWMO). It will be implemented in gateways which will act as relays to manage devices located behind

them. OMA DM gateway is compatible with existing OMA DM specification (1.2,1.3)[81][82])

- Definition of two new protocols suitable for managing M2M devices: OMA DM 2.0 and OMA DM lightweight for resources constrained devices

5.1.1 OMA DM 2.0

The DM 2.0 Enabler is based on a RESTful architecture. This protocol allows simpler implementations of both DM clients and DM servers by reusing widely deployed standard base technologies, such as HTTP, and JSON data representation.

OMA DM Version 2.0 also introduces new user interaction method on Device Management using Web Browser Component.

DM 2.0 enabler requires a high level of security, due to the data that is being handled

Service provided:

- Mutual authentication between DM Server and DM Client.
- Support of mutual authentication between Data Repository and DM Client.
- Rejection of un-authorized access from DM Server to DM Client.
- Rejection of un-authorized access from DM Client to DM Server.
- Secure communication channel between DM Server and DM Client.
- The DM 2.0 Enabler can handle existing Management Objects (MO) which are designed for working with DM 1.x Enabler.

At the time of writing, there is no requirement in OMA DM2.0 stating the need to be able to distribute Management objects via broadcast.

5.1.2 Lightweight M2M (LWM2M)

The LWM2M specification defines the application layer communication protocol between the LWM2M Server and the LWM2M Client which is placed in the LWM2M Device. In contrast to the OMA DM specification which mainly concentrates on managing mobile devices, the OMA Lightweight M2M specification focuses not only on management but also on service enablement for LWM2M Devices. The Lightweight M2M specification will provide a solution for Resource Constrained LWM2M Devices which will greatly reduce the costs of deploying M2M services. This advantage will be of great benefit to every stakeholder in the M2M industry. Moreover, the Lightweight M2M specification minimizes the traffic impact on the communication network caused by the growing number of M2M Devices. Furthermore, power consumption of M2M Devices will be reduced.

The Lightweight M2M specification supports secure communication between the LWM2M Client and LWM2M Server. This secure communication contains authentication, authorization, data integrity, confidentiality and replay attach protection.

Four data formats are defined by the LWM2M specification:

- plain text
- opaque
- TLV
- JSON

The LWM2M protocol is based on CoAP (IETF **Constrained Application Protocol**) principles. CoAP runs over UDP. The UDP binding is used in NoSec (no security) mode. Reliability over the UDP transport is provided by the built-in retransmission mechanism of CoAP.

The UDP channel security is defined by the Datagram Transport Layer Security (DTLS) [83] which is the equivalent of TLS v1.2 [84] for HTTP.

DTLS is a long-lived session based security solution for UDP. It provides a secure handshake with session key generation, mutual authentication, data integrity and confidentiality. DTLS is also used for authorization on individual CoAP resources.

DTLS supports most of the Cipher Suites defined in TLS. Considering that every M2M devices can be managed by an LWM2M server the choice of Cipher Suites is not limited to the list defined in Section 9 of CoAP.

CoAP can also be used over SMS by placing a CoAP message in the SMS payload using 8-bit encoding... The LWM2M Client MAY interact with the server using both UDP and SMS bindings.

At the time of writing, there is no requirement in OMA LWM2M stating the need to be able to distribute Management objects via broadcast.

5.1.3 OMA Gateway management object (GWMO)

This specification focuses on enabling a DM Server to manage devices that are behind a gateway and not directly accessible to the OMA DM Server. This gateway is managed by an OMA DM Server; in turn, the gateway manages other devices under it.

This document also provides specification for management of devices in a Machine to Machine (M2M) ecosystem (for example, fanning out DM commands from a DM Server to multiple End Devices and aggregating responses from multiple End Devices so that a consolidated response is sent back to the DM Server).

The main difference between GWMO 1.0 & GWMO1.1 is GWMO1.1 is supporting a hierarchical gateway structure

The Gateway MO documents specify 3 modes of operations:

- Transparent mode (The DM Gateway does not participate in the management session that gets established between the DM Server and the End Device),
- Proxy mode (Two related DM sessions are established: one is between the DM Server and the DM Gateway; the other is between the DM Gateway and the End Device(s).),
- Adaptation mode (The DM Gateway manages End Device(s) behind the DM Gateway on behalf of the OMA DM Server over a non-OMA DM protocol)

In 1.1 specifications, a gateway maybe connected through other gateways.

Non OMA-DM protocol targeted between OMA-DM Gateway and End-devices may be supported such as ZigBee, Bluetooth and some others

5.2 Deluge

Deluge [71] addresses the problem of reliable remote device programming (or reprogramming) over multihop wireless sensors networks. An epidemic data distribution approach is used to achieve robustness against lossy communications and nodes failures.

Wireless sensor nodes are often energy constrained and need to minimize the energy used for their operation.

To support network reprogramming, requires to setup reliable communications to all sinks. Low bandwidth and high loss rates common to wireless sensor networks force the use of solutions different than those used for traditional wired networks

Deluge divides a data image into packets of a fixed size. The packet is the smallest unit of reliability that Deluge considers:

Deluge is a is a NACK-based protocol that relies on periodic advertisements to keep nodes informed of their neighbour's state

The cycle of communication proposed by Deluge can be summarized as follows:

1. Nodes periodically broadcast advertisements containing a software image version number and a bit vector describing which pages of that version the advertiser has already received completely.
2. When a node has determined from advertisements that it needs to upgrade some part of its image to match a newer version, it determines the lowest numbered page it requires, waits for a fixed amount of time, listening to advertisements to find out which of its neighbours are offering to send that page
3. When this period is up, the node determine a particular sender and sends a request to that sender indicating the page and packets within that page that it needs.
4. Upon receiving such a request, the sender does not send the packets immediately, but waits for a while to get an overview of which page and which packets are most requested. At the end of this observation period, the sender has selected a page to send. The packets sent in that page will be the union of all received requests for the page.
5. The sender broadcasts a data packet for every requested packet of the selected page.
6. After a node receives the last packet it needs to complete a page, it broadcasts an advertisement before attempting to request any packets it still needs. Proposed solutions for device provisioning via broadcast/multicast

A method for securing the deluge protocol has been proposed in [72]. The problem is that traditional techniques based upon digital signature for authenticating a program binary are not well suited to resource-constrained sensors. The solution proposed is to digitally sign only the first message of the transmission, the one containing the program version number and advertisement. In this first message is inserted a hash of the 2nd message which contains itself a hash of the third message and so on. If at a point in time the hash of a packet appears to be wrong, then this packet only is discarded. This avoids having to wait for the end of the whole transmission to perform integrity check.

5.3 Trickle

Trickle [73], is an algorithm for propagating and maintaining code updates in wireless sensor networks, borrowing techniques from the epidemic/gossip, scalable multicast, and wireless broadcast literature,

Trickle uses a "polite gossip" policy, where devices periodically broadcast a code summary to local neighbours but stay quiet if they have recently heard a summary identical to theirs

It assumes that devices can succinctly describe their code with metadata, and by comparing two different pieces of metadata can determine which device needs an update. To that end, a device transmits periodically code metadata, but only if it has not heard a few other devices transmit the same thing. Trickle sends all messages to the local broadcast address. There are two possible results to a Trickle broadcast: either every device that hears the message is up to date or a recipient detects the need for an update. Detection can be the result of either

an out-of-date device hearing someone has new code, or an updated device hearing someone has old code. As long as every device communicates somehow – either receives or transmits – the need for an update will be detected. When a device hears that a neighbour is behind the times (it hears older metadata), it brings everyone nearby up to date by broadcasting the needed pieces of code. When a device hears that it is behind the times, it repeats the latest news it knows of (its own metadata); this triggers devices with newer code to broadcast it.

Instead of flooding a net-work with packets, the algorithm controls the send rate so each device hears a small trickle of packets, just enough to stay up to date. For example If a device with code version x hears a summary for version x-y, it broadcasts the code necessary to bring version x-y up to version x. If it hears a summary for version x+y, it broadcasts its own summary, triggering the device with version x+y to send updates. With this simple mechanism, Trickle can cope with huge scale up in network density.

6 Proposed solutions for device provisioning via broadcast/multicast

This chapter proposes solutions for efficient device provisioning via broadcast/multicast. In a first step, section 6.1 deals with device provisioning occurring in 3GPP networks. Section 6.2 is considering the more global picture of multicasting device management data from a central node located in a wide area network (typically a device management server located on the Internet) to devices located in a capillary network **behind a gateway**. The need to reach devices behind the gateway requires taking into account specific requirements linked to the LAN part of the transmission

6.1 Broadcast device provisioning in 3GPP networks

This section will focus upon device management occurring within 3GPP networks. It was seen in section 5.1 that OMA is specifying 2 solutions suitable for M2M device management: OM DM 2.0 and OMA DM lightweight.

As described above both OMA DM 2.0 and OMA DM lightweight currently envisage device provisioning operations solely performed over unicast links.

EXALTED project has also described two possible lightweight device management solutions. [2]

- The first one: ELFOMA (EXALTED Lightweight device management For OMA) is a lightweight Device Management solution which is more efficient comparing to OMA-DM v1.x as the size of DM messages is reduced by 85%. The ELFOMA solution enables operators to reuse existing OMA-DM v1.x servers to incrementally handle new constrained M2M devices. The cost is limited to ELFOMA proxy adapter which converts OMA-DM v1.x Sync based messages (XML) to ELFOMA's Comma Separated Value (CSV) messages. Parsing CSV message is much less complex than parsing XML messages. Additionally, an ELFOMA message is six times smaller than an OMA-DM message. EXALTED Lightweight DM is used as underlying protocol to support other service capabilities, such as data collection, device-to-device messaging and device addressing. ELFOMA system concept is depicted on the Figure 6-1

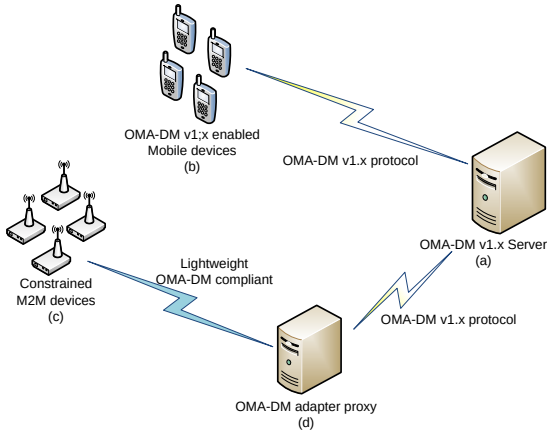


Figure 6-1 : ELFOMA as a OMA proxy

- The second one is more lightweight and suited to small devices. More complex scenarios for DM (i.e. update of device software/hardware) can be also employed by using CoAP protocol, to manage a device as well as a group of devices. The CoAP protocol can be utilized for these operations due to available multicast and unicast requests, header

option fields and the simple congestion control mechanism are applicable for the remote monitoring and the diagnostic of the devices. In order to employ CoAP for device management, protocol features are mapped with required device management procedures and DM module based on REST is defined

The rapid take off of M2M communications may lead to a sharp increase of the number of managed M2M devices. Also, the need to remotely manage very low end devices (i.e. sensor networks), possibly located behind a gateway within capillary networks, may put a heavy burden upon device management server, and lead to investigate alternative methods to distribute device management commands, and specially software updates.

In this situation, broadcast or multicast techniques may provide attractive solutions to manage a large number of identical devices.

We propose now a possible solution for wide scale distribution of device management objects to a large number of similar devices connected to 3GPP networks. This solution consists in using the MBMS (Multimedia Broadcast and Multicast service) for distributing the device management commands and objects, and to use a 'lightweight' version of the GBA based upon CoAP to perform security bootstrap.

6.1.1 Proposal to use MBMS for device provisioning

It is envisaged that the solution described here could be applied readily with any of the Device management solutions proposed by EXALTED. It can be applied as well to both OMA DM 2.0 and OMA DM lightweight, but will require the definition of appropriate extensions discussed below.

This proposal can be applicable to manage LTE-M enabled devices directly connected to the LTE core network. It can also apply to non LTE-M devices connected to the Internet through an LTE-M gateway.

6.1.1.1 OMA-DM required extensions

In this proposal we use the term «Payload" to refer to the commands or objects description which need to be pushed to the devices. The Payload may be carrying parameters settings, software or firmware update, and more generally any type of information required to configure or provision the devices.

In order to perform payload dissemination, two different scenarios can be envisaged:

Scenario one: A single payload is generated that is later disseminated to a large number of similar devices. This scenario may be applied to the distribution of a single software update to a large number of similar devices. The generated payload , will carry the software update which may be can be encrypted with a group key, while the all the recipient devices are provided with the same group key distributed in an initial security bootstrapping phase. Therefore, devices entitled to receive the packet can do so thanks to this group key.

Scenario two: The payload is ciphered with the private key of the single each device, resulting in different content being sent to each device. In this scenario, the use of broadcast is **not** possible, simply because the data sent to every device is different.

So, in order to be compatible with Broadcast/multicast, a device management protocol (such as OMA protocols), should support the concept of the single payload protected by a group key.

According to OMA specifications and Section 5**, it seems that neither OMA-DM2.0 nor OMA DM Lightweight support at this point a distribution mode via broadcast/multicast. Furthermore, even the concept of a single payload be distributed to multiple devices is not currently supported.**

The proposal to use the MBMS both for payload distribution and for payload protection can therefore enable the two OMA-DM protocols under definition to be compatible with broadcast/multicast **assuming action is taken to adapt the protocol to support the concept of a single payload** disseminated to a large number of devices.

6.1.1.2 Architecture description

The architecture of the solution proposed is illustrated on the Figure 6-2:

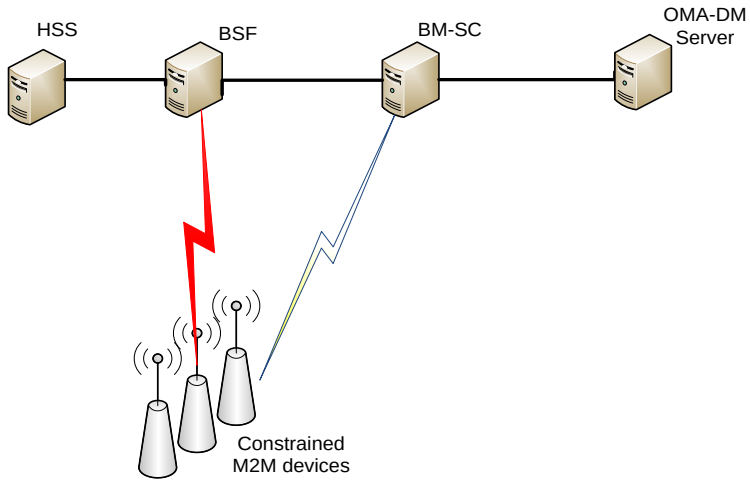


Figure 6-2: MBMS for management objects distribution

This architecture opens the possibility to use OMA DM2.0 or OMA DM Lightweight protocol for device provisioning by using MBMS for broadcasting of DM commands towards the constrained M2M devices.

The OMA DM server, in particular, could support either OMA DM 2.0 or OMA Lightweight DM and it will be used for sending DM commands towards BM-SC. The BM-SC is an entity within MBMS broadcasting mechanism, dedicated for data dissemination via broadcast. Therefore, the BM-SC will be used for further dissemination of DM commands towards constrained M2M devices in the capillary network. The solution proposed here would enable OMA DM 2.0 and OMA Lightweight DM support for broadcast and multicast mode of communication. Such possibility should make things easier, when it comes to DM of the large number of constrained devices deployed in the capillary network. The process itself, regarding provisioning of such devices is expected to be more efficient than a unicast provisioning.

The general idea is depicted on the Figure 6-2. In the proposed architecture, the DM server, from the MBMS perspective can be considered as a broadcaster.

The Discovery and Announcement functions in MBMS can be very useful if MBMS is integrated with the DM server. When the UE has subscribed to a SW update, it can be notified of the availability of the update.

The Figure 6-3 depicts a derived architecture using the ELFOMA protocol proposed by EXALTED WP4 deployed over MBMS.

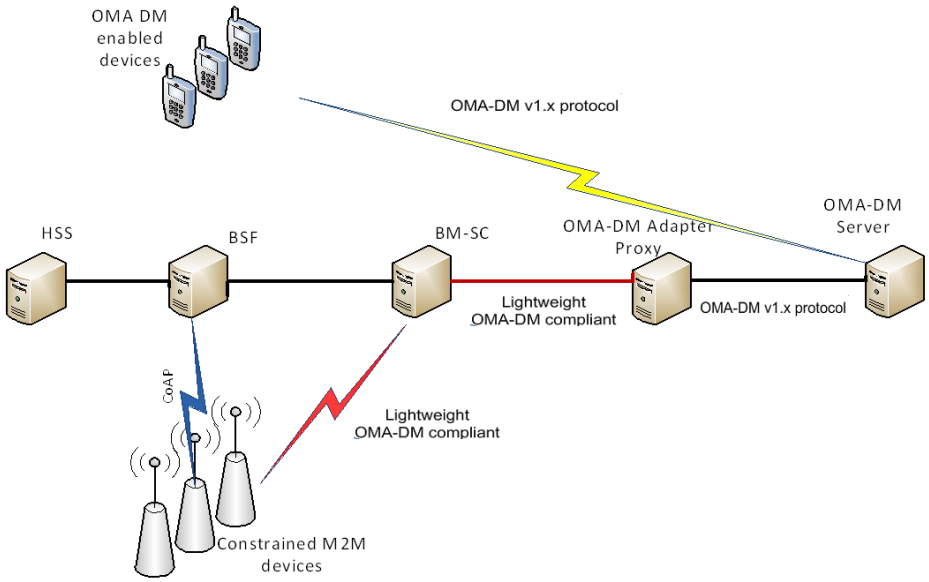


Figure 6-3: ELFOMA over MBMS

In order to deploy ELFOMA, EXALTED has proposed the combination of the regular OMA DMv1.x server together with the OMA DM Adapter Proxy. These two entities together constitute the ELFOMA solution. This solution has its advantages because it allows the independent use of already deployed OMA DMv1.x server in the network for DM of OMA DM Enabled devices. However, our focus was more concentrated to the DM of the constrained M2M devices. In this case OMA DMv1.x server is used all together with OMA DM Adapter Proxy. The role of the OMA DM Adapter Proxy is to perform a protocol translation between OMA DMAv1.x and Lightweight OMA DM and vice versa.

Large payloads to be sent to devices may be divided into a large number of smaller packets. The message is reassembled on the reception side and data integrity is checked. The receiver can then ask retransmission of bad packets. MBMS provides a possibility to ask a packet retransmission in unicast mode which can be used for that purpose. On this way, the problem of reliability and data retransmission can be solved.

EXALTED is proposing the integration of MBMS architecture with ELFOMA solution, since, MBMS is a 3GPP mechanism that is used for dissemination scheme and it can propagate DM using hierarchical organization which can be a good solution.

The integration point between ELFOMA and MBMS would be an interface between BM-SC and OMA DM Adapter Proxy. This interface would be Lightweight OMA DM based. In this solution BM-SC would be used for broadcasting of DM commands towards constrained M2M nodes. Therefore, MBMS as broadcast mechanism can be used to disseminate DM commands, and MBMS security can be bootstrapped using GBA over CoAP. This proposal is now described:

6.1.1.3 Proposal of GBA with CoAP mapping

As was explained in section 4.2.1.3, the MBMS security bootstrap procedure consists of three steps:

- First, it uses GBA in order to establish key which is tied with a user (in this case a constrained device). This key is changing only when user subscription changes.

- Second, this user key is then used for a provisioning of a second key – service key. This service key is also attached to the user when the user “asks” for a specific service. The service can be MBMS, but it can also be a M2M service provider, and then the same schema can be used for obtaining the bootstrapped association with M2M service provider.
- Finally, the service key is used to generate another key – this time that a traffic encryption key. However, this key is a short term key. It is good to emphasize that ETSI M2M architecture proposes to resort to the same type of procedure when performing bootstrapping between a device and M2M service provider.

As mentioned in section 5.1.2, OMA-DM lightweight will use CoAP as a transport protocol;

The EXALTED DM lightweight solution described in EXALTED D4.3 report proposes to use CoAP as a transport protocol.

GBA is typically using HTTP as the transport protocol to implement the Ua and Ub interface described on Figure 4-3 . In order to open the possibility of compact and lightweight security bootstrap with the GBA, we investigate here the use of GBA along with CoAP and the transport protocol.

6.1.1.3.1 Constrained Application Protocol (CoAP)

CoAP, defined by IETF, is a protocol for constrained nodes, that should fit in most M2M use cases and it should answer the challenges associated with the use of device on such devices... One of important advantages of CoAP is that it can easily be translated into HTTP protocol providing considerably lowered overhead for M2M devices comparing to HTTP protocol. Among other capabilities CoAP provides a constrained web protocol for M2M use cases, UDP binding with support for unicast and multicast messages, low header overhead and parsing complexity, etc. Another important advantage of CoAP is its possibility to work in asynchronous mode, which means it can broadcast messages without standard data integrity and ordering messages created by HTTP/TCP protocol. CoAP is based on UDP and UDP is appropriate solution for answering on small queries initiated from a large number of devices. UDP does not have reliability mechanisms installed but CoAP has built in an error detection and correction mechanism. CoAP message format can be found in [61]

Each HTTP method can be mapped using a ”restful approach” into adequate M2M use cases frames [62].CoAP redefines GET, POST, PUT and DELETE semantic in order to answer the initial goals. The main goal of CoAP is to develop generic web protocol for constrained nodes taking in account energy consumption, automation processes as well as M2M applications. The goal is not to compress HTTP but to make REST interworking with HTTPS (Hypertext Transfer Protocol Secure) in order to support it for M2M use. Using CoAP instead of HTTP results in low overhead and simplicity for the implementation of a business logic. Most important features which CoAP protocol brings are freshness and validation models. CoAP uses Max-Age option to determine how fresh information is. The option is defined on server side with default value of 60 seconds. Alternatively, ETag option could be used to determine freshness of data, but this option is more appropriate to situations when one GET request may produce a number of responses and it is important to allow server to select a stored response and update its freshness. CoAP is suitable for those complex Device Management operations due to its support for multicast and unicast requests and congestion control mechanism [63].

6.1.1.3.2 CoAP vs. HTTP performance

The compact data encoding makes possible to transport data faster over network connections. On that aspect, the superiority of CoAP over HTTP is confirmed by two

independent researches. Team from University of Bremen [64] has presented results of their research which shows that CoAP is approximately three times faster than bare HTTP (TCP) protocol and more than 30 times faster than Apache2-Firefox HTTP protocol.

Another research at University of Belgrade shows that CoAP transmission time is less than half of HTTP transmission time for the same data transmitted [65].

Both researches confirm that CoAP is easy do deploy, and that CoAP has advantages over HTTP in reduction of transmitted data which in turn leads to reduction of required bandwidth and devices power and memory usage.

EXALTED test in real environment using Android based device also shows that CoAP header size is ten times smaller than HTTP header size and payload size is less than half of payload size when HTTP protocol is used. All these researches and tests lead to a conclusion that CoAP has significantly better performance in constrained environment. It is important to highlight that CoAP is not just compressed HTTP, but efficient management of HTTP data which optimizes the data for usage in constrained networks.

CoAP is based on request/response model, but with a major difference compared to HTTP - response. CoAP messages are sent asynchronously, while HTTP protocol uses already established connection. CoAP- HTTP mapping is straightforward, but only at certain extend, as CoAP supports limited subset of HTTP features.

6.1.1.3.3 Proposed - GBA over CoAP solution

EXALTED believe the benefits of CoAP may be applied also to reduce the data overhead involved in security bootstrapping operations with the GBA: Typically, both Ua and Ub interfaces in GBA architecture are implemented using the HTTP protocol.

The idea explored here is therefore to use CoAP, as it is a lightweight transport mechanism, for carrying GBA exchanges over Ua and Ub interfaces. This solution will be examined by comparing a sample GBA exchange over Ua interface implemented with HTTP and CoAP.

The solution can be implemented in two phases:

Phase I (Figure 6-2) would enable to reuse existing implementations of both the BSF and NAF. The architecture would have a CoAP Proxy placed between the UE and NAF. The role of the Proxy is doing protocol translation between CoAP and HTTP and vice versa. In phase I, it is possible to use CoAP for the communication between the Proxy and the UE (in both directions), and HTTP for the commutation between the proxy and the NAF (in both directions). In this case, there is no change for NAF, and NAF itself is not aware of the CoAP use, therefore NAF is acting on the same way as if it were speaking with an HTTP client.

The CoAP request for an HTTP resource is the same as for the CoAP resource. The mapping between these two protocols is done on the Proxy (either CoAP-HTTP or HTTP-CoAP). In case of a CoAP GET request with an HTTP URI, a Proxy-URI Option is forwarded to the Proxy where the mapping to HTTP takes place. It is then sent to the HTTP server. In the proposed case, where the role of the HTTP server is given to NAF, HTTP server will respond back to the Proxy, where the reverse mapping is taking place. Finally, a response is forwarded back to the node that initiated the request. The process described is for a GET method and it is similar for POST, PUT or DELETE ones. The signalling flow is described in more details in section 6.1.1.3.4

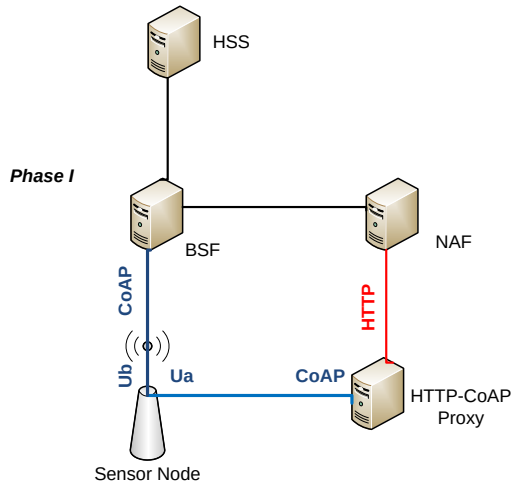


Figure 6-2 : Phase I of the proposed solution

Phase II would require specific version of BSF and NAF supporting the CoAP protocol. This will remove the need for an independent proxy. That would require certain changes within GBA architecture itself. Phase II architecture proposal is shown on Figure 6-3

The communication with BSF over Ub interface is done over CoAP in both phases.

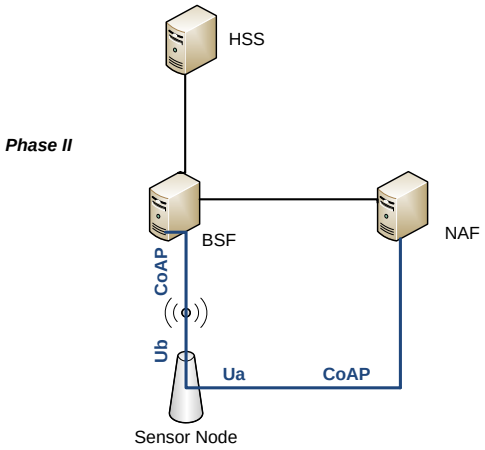


Figure 6-3 : Phase II of the proposed solution

With these proposals our goal is to set preconditions that are needed to be achieved in order to make possible the use of CoAP in GBA. CoAP introduction in GBA will result with further benefits which can be expressed in terms of savings regarding the number of transmitted bits or in terms of response time reduction.

6.1.1.3.4 HTTP-CoAP sample message exchange

As the GBA has been originally designed to be able to authenticate web based applications, both the Ua and Ub interfaces above described, GBA can be implemented using an http protocol mapping.

Digest authentication can be used on all HTTP based protocols which support mutual authentication process between UE and NAF. UE must inform NAF that its supports HTTP-bootstrapping based digest authentication. Specific 3GPP http headers are used to carry the GBA challenge and response using HTTP digest authentication. The protocol stack when HTTP Digest authentication is used is shown on Figure 6-4 [66][67]:

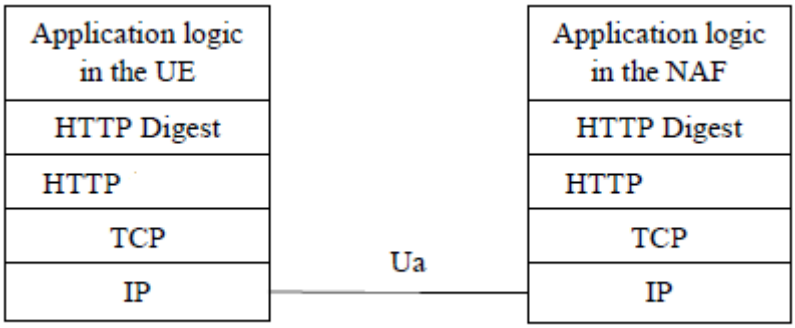


Figure 6-4 : HTTPS Digest Authentication

HTTP headers are built using "option fields" carrying parameters. One such parameter: the "The realm" contains two parts which are separated by "@" character. The first part is "3GPPbootstrapping" or "3GPP-bootstrapping-uicc" string, and the second part is the FQDN of the NAF. NAF informs UE about the key used by the first part of the realm parameter, so UE and NAF can verify each HTTP request and response. There are few ways how the verification can be done: shared key-based UE authentication (HTTP Digest) with certificate-based NAF authentication (TLS); shared key-based mutual authentication between UE and NAF (PSK TLS); certificate based mutual authentication between UE and AS. More about these authentication types can be found in 3GPP TS 24.109 [68]

The purpose of this section is to provide a simple message exchange when CoAP is used according to the solution , described in Section 6.1.1.3.3.

Below is the example of the signalling flow in the existing GBA architecture model, together with HTTP messages, demonstrating a successful HTTP Digest Authentication is shown of the Figure 6-5 [68].

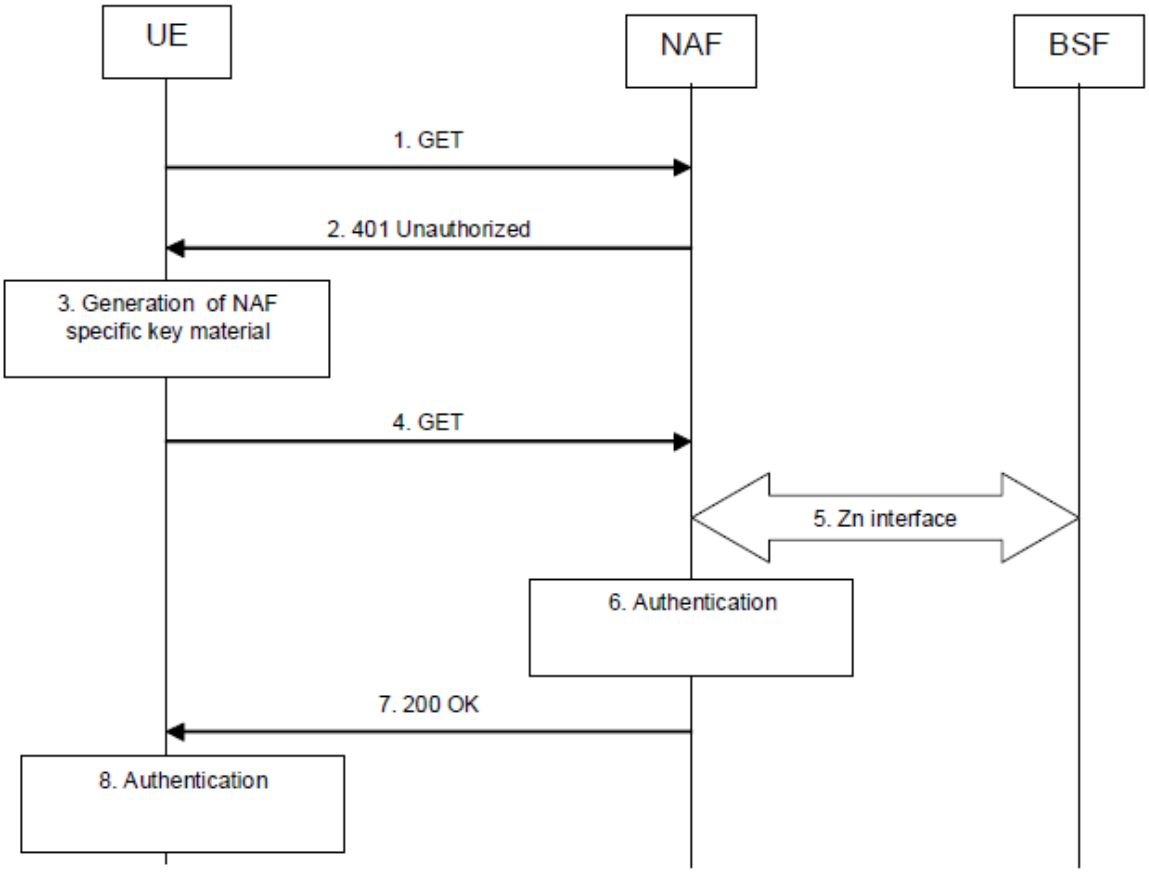


Figure 6-5 : HTTP authentication signalling flow for GBA

1. An initial GET request from UE to NAF:

GET / HTTP/1.1
Host: naf1.home1.net:1234
User-Agent: NAF1 Application Agent; Release-6 3gpp-gba
Date: Thu, 08 Jan 2004 10:50:35 GMT
*Accept: */**
Referrer: http://naf1.home1.net:1234/service

2. 401 Unauthorized response (NAF to UE)

HTTP/1.1 401 Unauthorized
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
WWW-Authenticate: Digest realm="3GPP-bootstrapping@naf.home1.net", nonce="6629fae49393a05397450978507c4ef1", algorithm=MD5, qop="auth,auth-int", opaque="5ccc069c403ebaf9f0171e9517f30e41"

3. Generation of NAF specific keys at UE

4. GET request (UE to NAF)

GET / HTTP/1.1
Host: naf1.home1.net:1234

User-Agent: NAF1 Application Agent; Release-6 3gpp-gba
Date: Thu, 08 Jan 2004 10:50:35 GMT
*Accept: */**
Referer: http://naf1.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@naf.home1.net", nonce="a6332ffd2d234==", uri="/", qop=auth-int, nc=00000001, cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1", opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=MD5

5. NAF retrieves the NAF specific key material (Ks_NAF or Ks_ext_NAF) from the BSF.

6. Authentication at NAF by using the bootstrapping transaction identifier B-TID and the key material Ks obtained from BSF.

7. 200 OK response (NAF to UE)

HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27Content-Type: text/html
Content-Length: 1234
Authentication-Info: qop=auth-int, rspauth="6629fae49394a05397450978507c4ef1", cnonce="6629fae49393a05397450978507c4ef1", nc=00000001
Date: Thu, 08 Jan 2004 10:50:35 GMT
Expires: Fri, 09 Jan 2004 10:50:36 GMT
<SERVER PAYLOAD>

According to Section 6.1.1.3 proposal to use CoAP in GBA instead of HTTP in two phases, the signalling flow and mapping of the messages are provided on the following figures respectively.

Figure 6-8 depicts the signalling flow for a successful authentication when CoAP Proxy is used for protocol translation between CoAP initiated as transport protocol from the M2M device and HTTP used from the Proxy towards NAF (and vice versa).

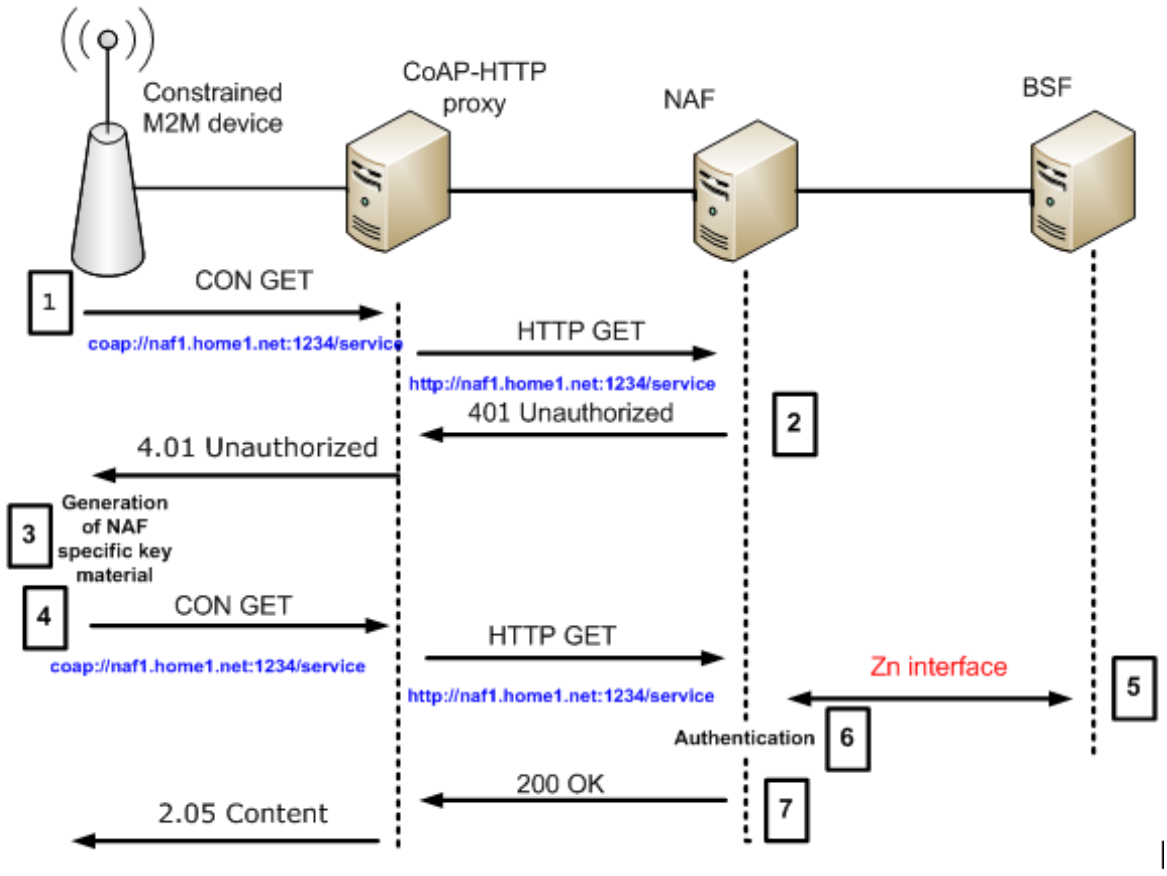


Figure 6-8: Authentication signalling flow in Phase I

1. An initial GET request from Constrained M2M device to NAF:

From Constrained M2M device towards CoAP-HTTP Proxy

```
  0               1               2               3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | 1 | 0 |  0  |    GET=1    |        MID=0x7d34         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

coap://naf1.home1.net:1234/service

From CoAP-HTTP Proxy towards NAF

```
GET / HTTP/1.1
```

http://naf1.home1.net:1234/service

2. 401 Unauthorized response (NAF to Constrained M2M device)

From NAF towards CoAP-HTTP Proxy

```
  0               1               2               3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | 1 | 2 |  0  |   4. 01= 129   |        MID=0x7d34         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

From CoAP-HTTP Proxy towards a constrained M2M device

```
HTTP/1.1 401 Unauthorized
```

3. Generation of NAF specific keys at UE

4. GET request (from Constrained M2M device to NAF)

From Constrained M2M device towards CoAP-HTTP Proxy

```
  0               1               2               3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |1 |0 |  0  |    GET=1    |        MID=0x7d34         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
coap://naf1.home1.net:1234/service
```

From CoAP-HTTP Proxy towards NAF

```
GET / HTTP/1.1
http://naf1.home1.net:1234/service
```

5. NAF retrieves the NAF specific key material (Ks_NAF or Ks_ext_NAF) from the BSF.

6. Authentication at NAF by using the bootstrapping transaction identifier B-TID and the key material Ks obtained from BSF.

7. 200 OK response (NAF to UE)

From NAF towards CoAP-HTTP Proxy

```
  0               1               2               3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |1 |2 |  0  |   2.05=69   |        MID=0x7d34         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

From CoAP-HTTP Proxy towards a constrained M2M device

```
HTTP/1.1 200 OK
```

Figure 6-6 corresponds to Phase II of the proposed solution, where protocol translation is not needed, therefore only CoAP is used from M2M devices towards NAF.

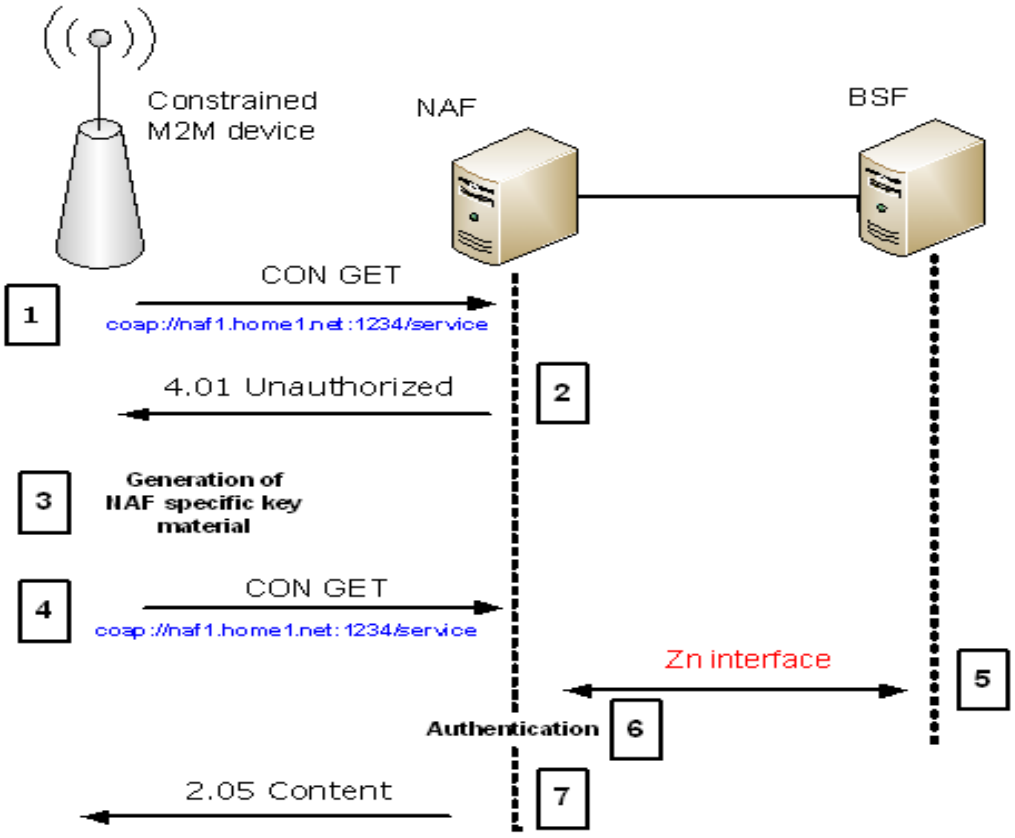


Figure 6-6 : Authentication signalling flow in Phase II

1. An initial GET request from Constrained M2M device to NAF:

```
 0               1               2               3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 0 |  0  |    GET=1      |        MID=0x7d34             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

coap://naf1.home1.net:1234/service

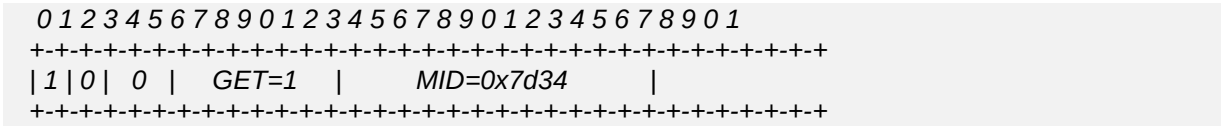
2. 401 Unauthorized response (NAF to Constrained M2M device)

```
 0               1               2               3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 2 |  0  |   4. 01= 129  |        MID=0x7d34             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3. Generation of NAF specific keys at UE

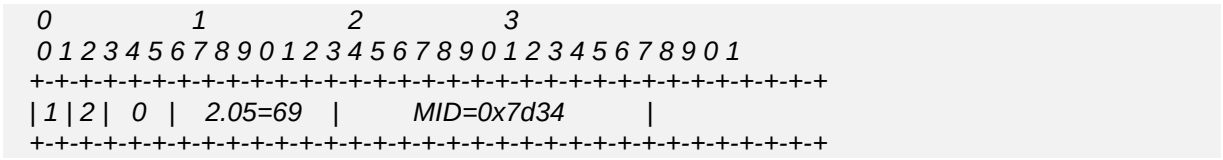
4. GET request (from Constrained M2M device to NAF)

```
 0               1               2               3
```



coap://naf1.home1.net:1234/service

5. NAF retrieves the NAF specific key material (Ks_NAF or Ks_ext_NAF) from the BSF.
6. Authentication at NAF by using the bootstrapping transaction identifier B-TID and the key material Ks obtained from BSF.
7. 200 OK response (NAF to UE)



CoAP headers are not yet fully standardized. Furthermore the HTTP headers used in the GBA exchange are specific 3GPP headers. They have been standardized by IETF upon request from 3GPP [68]. Such headers can only be understood by specific browser. For example, in GBA HTTP message, one of the Options there User-Agent is defining “NAF1 Application Agent; Release-6 3gpp-gba”, which clearly suggest that in this case it is a specific user agent able to understand those messages.

This is why EXALTED is proposing mapping between HTTP and GBA shown in Table 6-1 and trying to invoke IETF to take active role in this process.

Table 6-1 : Proposed mapping between 3GPP specific header parameters and coap option fields

HTTP Header in GBA	Proposed option in CoAP			
Name	**Name**	**Number**	**Existing (y/n)**	**Type**
Accept	Media-Accept	101	n	string
Content-Length	Content-Length	102	n	uint
Date	Date	103	n	HTTP-date (RFC1123)
Expires	Expires	104	n	HTTP-date (RFC1123)
Host	Uri-Host	3	y	string
Referrer	Uri-Host	3	y	string
Server	Server	105	n	string
User-Agent	User-Agent	106	n	string
WWW-Authenticate	Digest realm	110	n	string
	nonce	111	n	string
	algorithm	112	n	string
	qop	113	n	string
	opaque	114	n	string
Authorization	Digest username	115	n	string
	realm	116	n	string
	nonce	111	n	string
	uri	117	n	string
	qop	113	n	string
	nc	118	n	string
	cnonce	119	n	string
	response	120	n	string
	opaque	121	n	string
	algorithm	112	n	string

Authentication-Info	qop	113	n	string
	rspauth	122	n	string
	cnonce	119	n	string
	nc	118	n	string

The Table 6-1 represents one of many possible mappings between HTTP headers, from HTTP message exchange between UE and NAF in GBA, depicted on Figure 6-7, and CoAP Option fields described in TS 23.109 [68].

Options Host and Referrer in HTTP are mapped to the same CoAP option (Uri-Host), but in different formats. Referrer is equal to Uri-Host, while Host is its part.

Although Option Accept exists in both HTTP and CoAP by name, it cannot be equally mapped as CoAP version has different functionality.

Arbitrary number is given to all proposed CoAP option fields (101-106 and 110-122 in Table 6). EXALTED will initiate an action with IETF/IESG to address open issues of mapping between HTTP and CoAP.

WWW-Authenticate, Authorization and Authentication-Info HTTP Header are spitted in several new Option Fields in CoAP.

As GBA is used for bootstrapping and authentication procedures some of the newly proposed Options Fields in CoAP (especially WWW-Authenticate) are critically important and must be transmitted as such in order to make this security mechanism to work.

Each of the newly proposed Options in CoAP fits into the Option format structure as described in [68].

Table 6-2 presents the comparison between HTTP and CoAP headers in bytes, when particular GBA messages exchange is taken into account.

Table 6-2 : Header bytes savings comparison

Header savings in bytes	**HTTP bytes**	**CoAP bytes**
Initial GET (UE to NAF) - unsuccessful	70	30
401 Unauthorized response	50	38
GET (UE to NAF) -successful	80	70
200 OK Response (NAF to UE)	70	42
Total SUM of header bytes in one GBA exchange	270	180

The HTTP headers in GBA messages and corresponding CoAP Option Fields (according to the proposal in the Table 6) are only taken into consideration for the calculation, i.e. the payload itself is of the same size in both types of messages since it carries the overall information and cannot be compressed.

If taking into account [65], size of HTTP and CoAP header (20 and 10 bytes respectively) and the size of HTTP field headers and CoAP Field Options (10 and 4 bytes per each in the message respectively), it is possible to calculate the number of transmitted bytes in both cases and potential savings in terms of bytes if CoAP is used.

For example, if looking to GET message (UE to NAF, step 3 in HTTP Digest Authentication), there are 6 HTTP headers (Host, User-Agent, Date, Accept, Referrer, Authorization), and they are equivalent to 15 Option Fields in CoAP (according to the proposed Table 6). Each of the HTTP headers and Option Fields has its own header size which is taken into account. And even then, in this particular case (although this is the worst case scenario) we can save up to 10 bytes.

In the best case scenario (i.e. HTTP Digest Authentication messages exchanged) saving in header bytes can be up to 40 bytes per GET message if CoAP is used instead of HTTP and according to this proposal (Table 6-2).

The reduction in number of bytes would be even greater if CoAP "Options Fields“ could support "sub-options", as this would reduce the number of headers transmitted. This possibility would make possible for the various Option Fields proposed in CoAP (according to the Table 6-1) could be packed into single Option Field but in different sub-options. Therefore, the number of headers can be decreased to one header for a single Option Field with sub-options.

The adoption of sub-options within Option Fields in CoAP could express the true value of this lightweight protocol usage compared to HTTP. This is a proposal that can be submitted to the IETF as an enhancement of CoAP protocol.

6.1.1.4 Proposal for Offloading key management in MBMS

The key management in MBMS uses different types of keys: MBMS User Key (MUK), MBMS Service Keys (MSKs) and MBMS Traffic Keys (MTKs). MUKs are used to securely deliver MSKs to UE, whereas MSKs are used to secure the broadcast of MTKs used for MBMS content security (refer to sub-section 4.2.1.3); MTKs and MSKs are frequently changed to reduce content copy. MTKs are updated in a periodic-basis based on multicast communication. On the other hand, MSKs are updated between the BM-SC and each UE through unicast communications secured by MUKs (refer to Figure 6-7). Considered UE may consist of hundreds, thousands, or even more devices in highly-dense capillary networks (e.g., sensor/actuator networks) which render frequent unicast rekeying (i.e., MSKs' update) unpractical for these networks and may even cause network congestion.

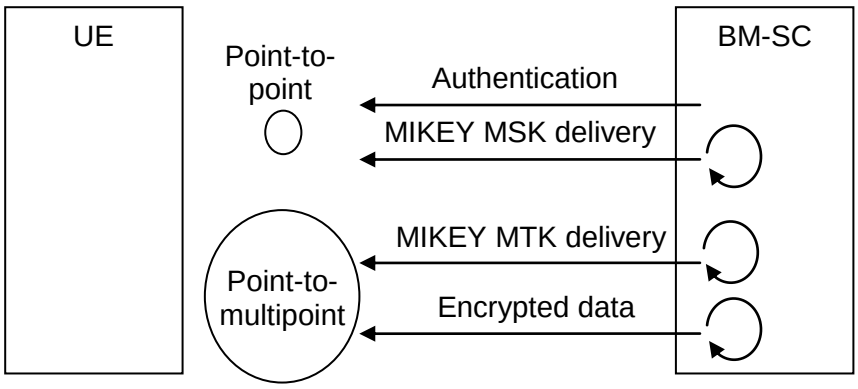


Figure 6-7: MBMS security functions.

A different approach to MIKEY [52] for rekeying consists in proactively storing multiple keys at nodes and use them when needed. For instance, devices can store multiple MBMS Service Keys (MSKs). When this is a need to refresh the MSK, the shared MSK that is not yet used by all nodes and eventually stored at these nodes, could be used in the following sessions. The number of stored MSKs is a parameter of trade-off between the memory space used and the communication overhead (e.g., the number of stored MSKs could be set to the number of clusters within the group). An alternative trade-off between computation resources and communication overhead can be achieved relying on key derivation keys (KDKs) securely delivered by the BM-SC to devices. Based on KDKs, devices could generate by themselves new MSKs.

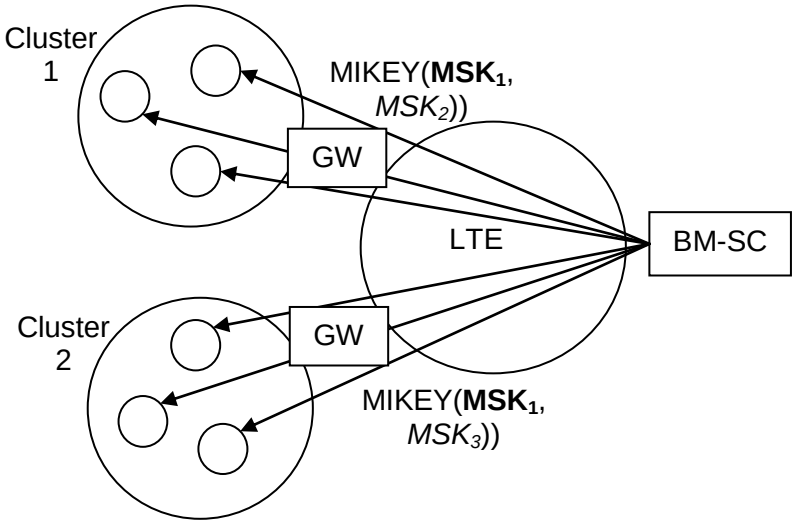


Figure 6-8: MIKEY MSKs delivery to the group protected with device MUK. MSK_1 is shared between all devices. MSK_2 (respectively MSK_3) is only shared within devices of cluster 1 (respectively cluster 2)

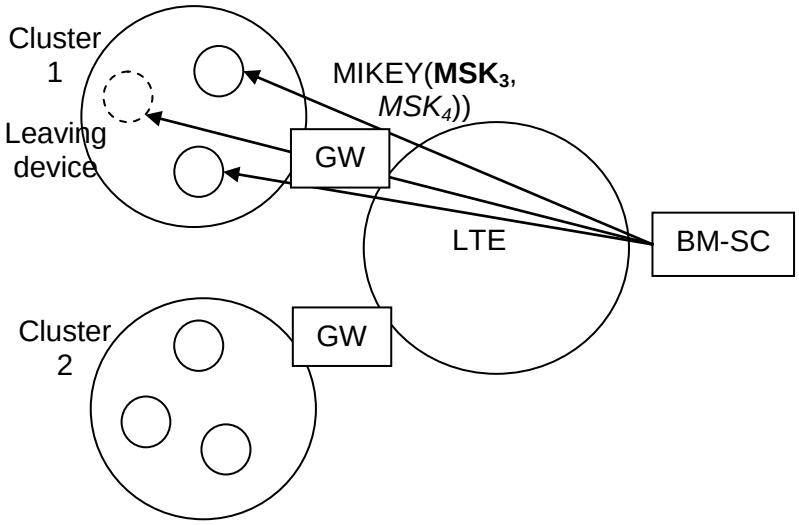


Figure 6-9: MIKEY MSK update at only clusters (cluster 1) where membership change occurs.

If group key update is performed because of membership changes due to device joining or leaving the group, key update requires the use of new MSKs that has not be used in previous sessions and particularly not known to the leaving node. In MBMS, this operation requires unicast MSK update at each group member. The MSK update can be performed immediately to provide forward secrecy, or directly after expiration of the validity of the old MSK. In order to reduce the overhead of this operation, the group could be divided into multiple clusters. The clustering of the group may take into account node type, its capabilities, its location, and/or mobility patterns, etc. These clusters hold some group keys that are not shared by all clusters. In this way, only clusters affected by membership changes undergo MSK update operation. The other clusters that share a common MSK not known to the affected clusters are notified of the membership change based on the MSK's key index.

The proposed approach may rely on pre-stored MSKs. The modification consists in generating multiple MSKs beforehand by the BM-SC. During MSK delivery, the BM-SC delivers, along with the current MSK, additional MSKs to each cluster. These MSKs are not shared with all clusters. They are used when the MSK need to be updated because of device joining or leaving (as an example, refer to Figure 6-8 and Figure 6-9). The number of MSKs stored at devices depends on the number of clusters formed by the BM-SC. A subset difference (SD) method [79] can be applied, so that devices store only a number of keys logarithmically proportional to the number of clusters. If needed, the number of keys stored at devices can be simply reduced by dividing the group into larger clusters.

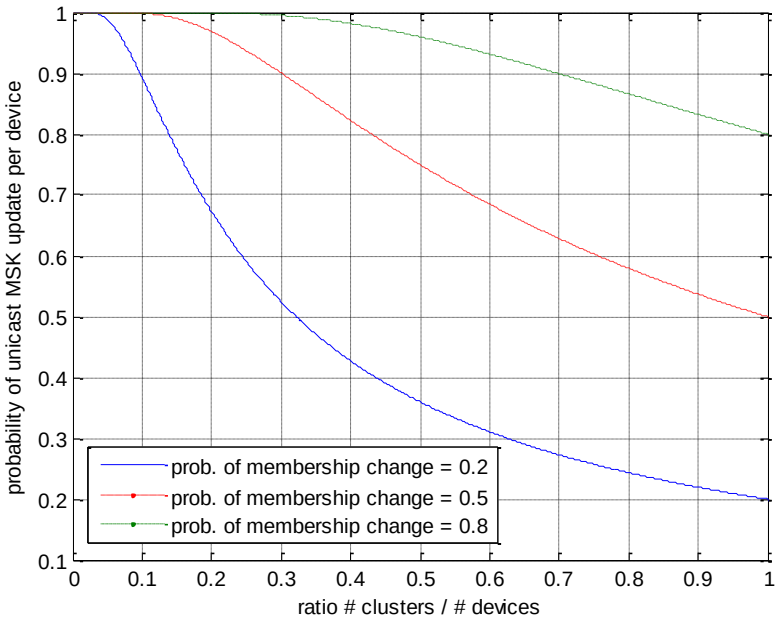


Figure 6-10: Probability of unicast MSK update per device over the number of considered clusters.

The main advantage of this solution is to allow unicast MSK update only for a subset of the group of nodes (i.e., clusters where membership change occurs), not for the whole group as in the current MBMS standard. The remainder of nodes will be simply notified of the new MSK that will be used to securely deliver MBMS Traffic Keys (MTKs), using its key index. As a result, the proposed approach limits the one-affects-all problem present in the MBMS key management. As example, the Figure 6-10 shows the average computed number of unicast MSK update messages per device with respect to different probabilities of device membership change. The figure demonstrates that clustering allows to reduce drastically the number of MSK update messages, while the number of required keys to be stored per device (not shown in the figure) increases only logarithmically with the number of considered clusters (based on the SD method [79]). If the capillary network churn out is high (high probability of membership change e.g., 0.8), the approach starts to be less advantageous with respect to reducing the number of update messages exchanged.

6.2 Broadcast device provisioning in IP networks

This section will focus upon device provisioning occurring in IP, not necessarily 3GPP networks. Hence the solutions described here solution do not depend from the availability of a 3GPP infrastructure.

6.2.1 Protocol translation from WAN to LAN protocols performed by the gateway

One way to scale up device management solutions in order to cope with the steep increase of the number of devices is to combine two types of device management protocols:

- Protocols suited to manage devices over Wide Area IP networks
- Protocols suited to manage devices over capillary networks

Section 5 described some candidate's technologies for device management and provisioning. Among those, the solutions proposed by OMA are suitable to manage devices over wide area networks. We have seen also that OMA has defined a Gateway management object (GWMO) enabling Device management server to use the gateway as an intermediate node to reach devices located in capillary networks behind the gateway.

Also in Section 5 , two protocols: deluge and trickle are described which are primarily focused upon the energy efficient distribution of software updates to devices located in multihop capillary network.

One possibility towards energy efficient management of devices from a central device management server located on the cloud is to combine those two types of protocol together.

Figure 6-11 shows how The GWMO object defined by OMA can be used in conjunction with deluge to perform software updates

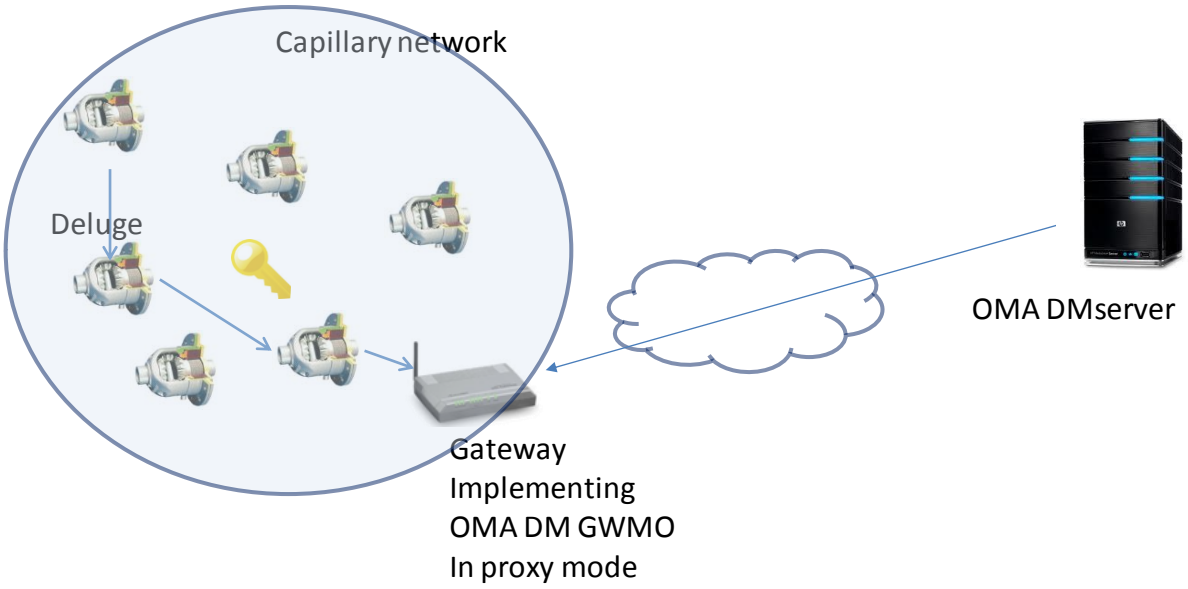


Figure 6-11 : Combination of OMA-DM and deluge

This solution is described by detailing a typical sequence of operations occurring when distributing a software update. This software update is possibly managed by an M2M service provider and is to be pushed to all devices located behind all gateways of the M2M service provider customers.

The software update is packaged according to the OMA DM format for a GWMO object. It is being sent from the central device management server to all the subscribers' gateways.

The gateways are assumed to function in Proxy Mode. So a typical gateway will extract the software update included in the GWMO object and will push the update within the capillary network using the deluge protocol.

The advantage of this proposal is that it constitutes a workable solution to provide central management of a lot of devices located behind many gateways (typical situation if an M2M service provider wants to get involved in capillary device management). Furthermore this solution can leverage existing device management servers.

The security to transmit the software update from the central device management server to capillary devices may be designed in two hops; the first hop is securing the link between the DM server to the gateway and the second hop is securing the transmission from the gateway to the device.

In Particular the method described in [72] may be used in this context a follows:

- The gateway extract the Software update from the GWMO,
- It decomposes this update in pages and packets,
- It inserts in each packet of the suite the hash of the following packet,
- It signs the first packet
- It begins the transmission inside the capillary network using deluge protocol.

The advantage of this two hop security link is that each leg of the hop is designed to cope with specific threats associated to that hop:

- OMA-DM credentials and security to secure the WAN part of the transmission
- Lightweight security to secure the LAN part.

The combination of a WAN and a LAN device management protocols with a protocol conversion performed by a gateway application is well suited to provide support for non IP based capillary networks (for example networks using Bluetooth or ZigBee). In this case the gateway decodes the GWMO objects and transcribes the commands before forwarding them to the capillary devices

6.2.2 IP multicast-based device management

With the advent of 6LoWPAN [76], M2M devices are increasingly becoming IP-connected with CoAP [77] being the de facto standard allowing these devices to communicate interactively with the Internet. In the literature, a hybrid solution for broadcast and multicast communications, i.e., overlay multicast, is generally proposed because it provides a good trade-off between multicast efficiency and its deployment complexity. Overlay multicast relies on a set of proxies that are in charge of transmitting multicast packets to recipients. These schemes choose the placement of proxies such that the number of duplicate packets is reduced. Nevertheless, the IP multicast approach is still the most efficient routing protocol for multicast communications. Most of proposed techniques in standards like CoAP tend to opt for IPv6 based multicast protocols. Indeed CoAP can be used for group communications [40]. In this context, the standard recommends the use of IP multicast as the underlying group communication mechanism, since this latter produces minimal complexity at the end device which makes it attractive for constrained networks. If PIM-SM is generally used in the Internet, lightweight routing protocols like RPL and MPL [39] enable IP multicast routing in resource-constrained networks. Efforts for securing IP multicast in such networks focused on the use of DTLS as a basis [78], since this security protocol is mandated by the standard CoAP. The method described in [78] manages group keys using unicast DTLS-protected communication channels between the controller and CoAP devices. The group keys (TGKs) are used to generate locally by each device of two keys: Traffic Encryption Keys (TEKs) and Traffic Authentication Keys (TAKs). These keys are used to secure multicast communications. Bootstrapping of M2M devices is discussed in detail in chapter 7

One of the device management (DM) protocols proposed in the EXALTED deliverable D4.3 [70] relies on CoAP for remote control and monitoring of devices and software/hardware update. The proposed approach is lightweight thanks to CoAP that allows avoiding verbose messaging while supporting group communications and providing security [78]. The approach employs gateways at the capillary network side in Transparent mode, i.e., that do not participate in DM sessions and uses the communication stack shown on Figure 6-12

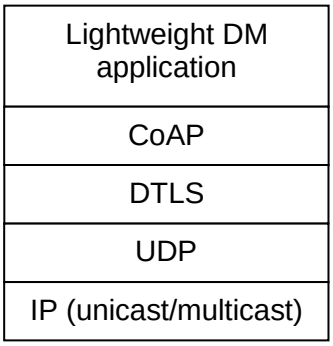


Figure 6-12: IP-based DM communication stack.

This section discusses security techniques that aim at providing authenticated and reliable device management, in particular with respect to firmware update applications, and access control enforcing group key management.

6.2.2.1 Source-authenticated broadcast.

This subsection introduces an approach which is derived from the scheme described in [72], adapting it to the EXALTED DM mechanism. In the proposed approach, the firmware update image is divided into fixed-size blocks, called pages [72], as in Deluge which is one of the most popular over-the-air (OTA) reprogramming protocols in constrained networks. Pages are further divided into fixed-size packets.

Since all packets of the firmware pages are known beforehand by the M2M application provider, source authentication and integrity protection of packets could be provided through a reverse-order hash chain. By considering N ordered packets $\{P_i\}_{1\leq i\leq N}$ composing firmware pages (P_1 being the first packet to be transmitted), N new packets $\{P'_i\}_{1\leq i\leq N}$ are derived such that:

$$P'_i = P_i \,||\, T_{i+1}, \text{ for every } 1\leq i\leq N-1 \qquad (6\text{-}1)$$
$$\text{where } T_{i+1} = H(P'_{i+1}), \text{ and } P'_N = P_N \,||\, 0$$

H being a cryptographic hash function e.g., HMAC.

The packets $\{P'_i\}_{1\leq i\leq N}$ are broadcast from the M2M application provider to devices in the capillary network through IP multicast communications. Each packet is linked to the previously transmitted packet. If the first packet is authenticated, the remaining packets are authenticated by induction (see Figure 6-13).

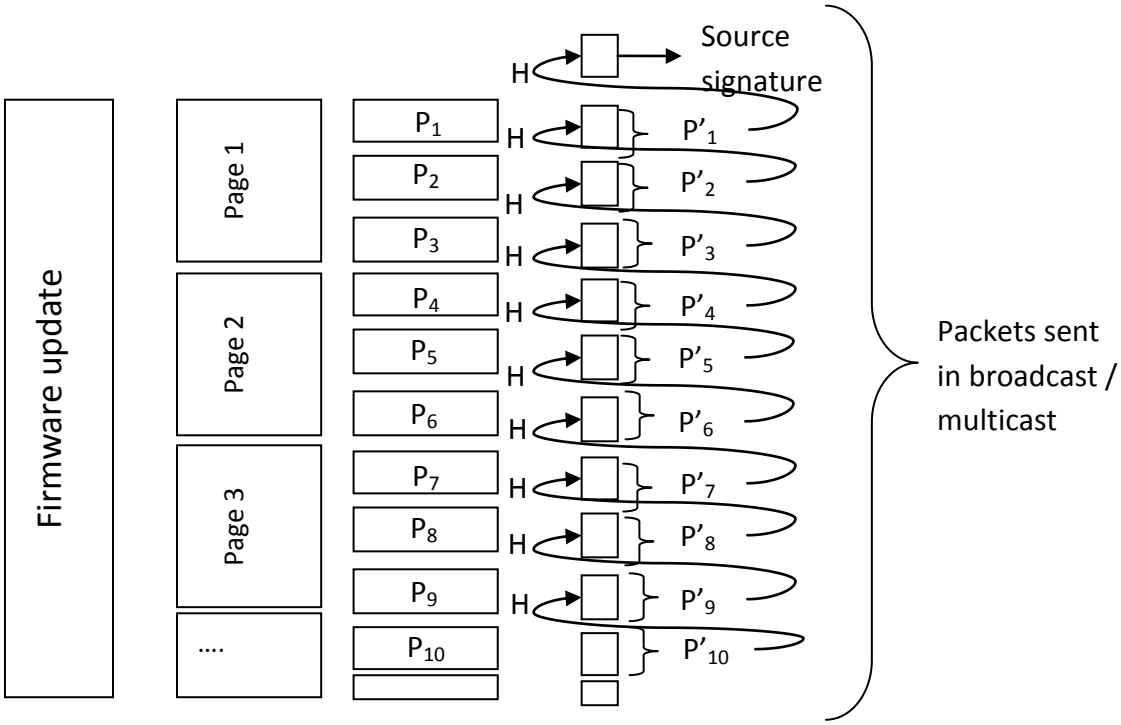


Figure 6-13: Authenticated firmware updates [72]

The authentication of the first packet is performed based on DTLS-based protected communication channels between the M2M provider and each device. For instance, the M2M provider could transmit the tag $T=H(P'_1)$ to each device in an integrity protected channel. To authenticate packets, each device verifies the authenticity of the first packet, and then, it only checks the hash part for the remaining packets. Alternatively, in order to avoid transmitting unicast messages for each device per firmware update, the tag could be simply signed by the M2M provider and broadcast to all devices based on DTLS-protected IP multicast communications. The public security parameters (i.e., public key, certificate, trust anchors) can be sent beforehand with the group key

6.2.2.2 Checking payload integrity

In the scheme proposed in the previous section, a device can authenticate a packet, only if the previous packet has been received. Because devices are generally limited in terms of buffer size, the number of firmware update packets received in disorder should be reduced. To overcome this problem, two solutions can be proposed. From the security server side, it may redistribute the same firmware update packets several times in the network. If the device has received the packet previously, it may just drop it. Alternatively at devices' side, devices may acknowledge the successful receipt of packets and/or pages.

For the considered application, i.e., firmware update, and for other applications like sending commands to actuators, reliability of broadcast messages is a strong requirement. Since CoAP is carried over the poorly reliable UDP, the standard proposes an error checking and correction mechanism in which messages that are marked CONfirmable (CON) should be answered by the recipients with a message marked ACKnowledgement (ACK). If the message is still not answered after a timeout, then the source resends the message again.

To confirm the receipt of broadcast messages, group members are required to answer using secure acknowledgement messages. As proposed in [55], acknowledgements may consist of

message authentication codes (MACs) using a pair-wise keys AKs (Acknowledgement Keys) shared between recipients and the sender. These acknowledgments can be aggregated through the reverse multicast tree using the simple bitwise-XOR operation.

With respect to security, the authors in [55] prove that the aggregation process cannot be forged. However, the scheme is prone to denial of service (DoS) attacks e.g., pollution attacks whereby intermediate nodes combine the aggregate MAC with random values to make not viable acknowledgement verification. The scheme proposed in [56] limits these DoS attacks. The aggregation process consists of shift bitwise XORing MACs. With this scheme, the transmission overhead is reduced thanks to MAC aggregation, while nodes combine their MACs only with a small number of other nodes' MACs, thus mitigating the spread of pollution attacks.

Each DM device should handle different types of keys. DTLS-based secure multicast keys (i.e., TGKs, TEKS, and TAKs) and unicast keys e.g., AKs (DM operations and used keys are summarized in Figure 6-14).

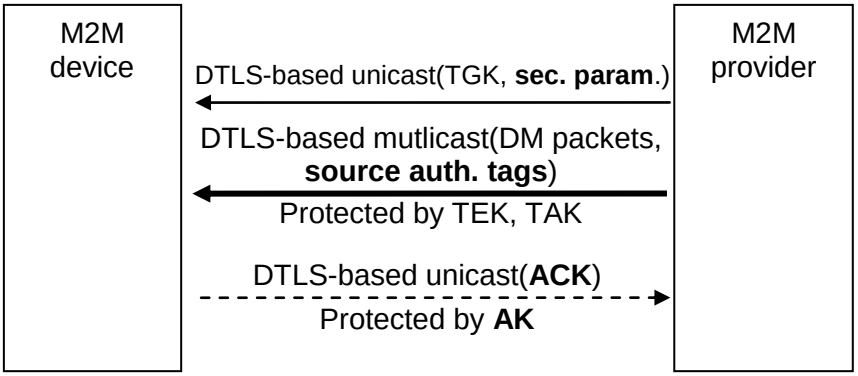


Figure 6-14: Reliable and authenticated DM operations.

6.2.2.3 Access control enforcing group key management

In the IP-based device provisioning, M2M devices are directly managed by the remote M2M provider. A main concern consists of the overhead produced by group key management that concerns generally a large number of M2M devices for one application and, moreover, it is handled using unicast communications for initial keying and rekeying e.g., key management in MIKEY. This subsection proposes solutions aiming at reducing group key management overhead.

In the EXALTED architecture described in [69], devices may have different capabilities. In the device domain there are: LTE-M and non-LTE-M enabled devices, and between M2M gateways, cluster-heads and resource-constrained devices. These different device capabilities generally match different device behaviour patterns. For example, the M2M gateway is generally more powerful than an ordinary device, relying for example on more robust security primitives and algorithms. In the general case, a single M2M device may run multiple M2M applications with multiple M2M providers over multiple M2M network operators (an example illustrated in Figure 6-15). Therefore, one approach is to catch such device heterogeneity by delivering diversified keys to devices based on different access levels with respect to applications. A straightforward solution may use multiple clusters for each type of M2M device and manage keys as proposed in subsection 6.1.1.4. Another alternative approach allows reducing the number of distributed keys by using key derivation keys (KDKs). The approach aims at simplifying key management at these heterogeneous devices, in particular by reducing the number of group key update messages. Indeed, a single

instance of MIKEY [52] is used to share distinct keys with the M2M devices based on their type.

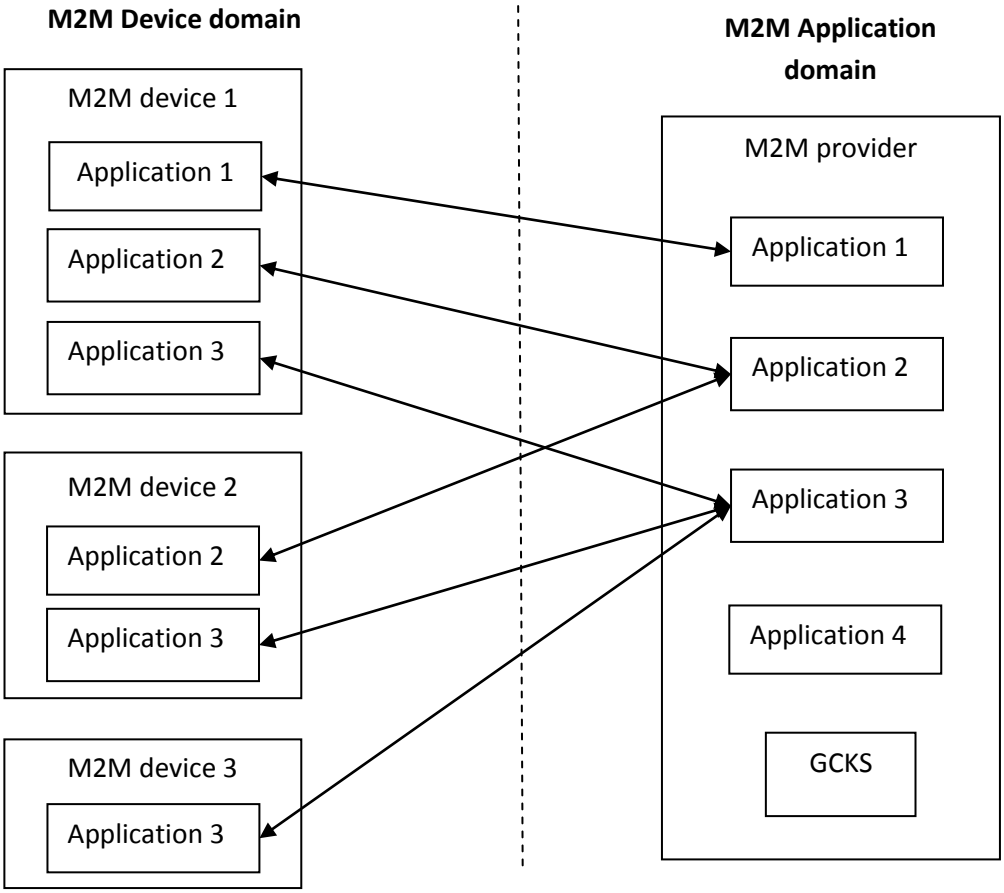


Figure 6-15: Heterogeneous M2M devices (e.g., of 3 types) may share common M2M applications or different ones

In the proposed approach, a linear key derivation key (KDK) structure like in [74] and [75] is used to map group keys managed by MIKEY to different device types associated to the applications to which devices are subscribed to. These levels in the key structure may correspond to different authorization levels that allow varied access to data i.e., firmware update images corresponding to different M2M applications.

Key structure: The following KDK structure is generated using a cryptographic hash function such that at each a parent node p in the structure, the KDK of the child node i below p is computed as:

$$KDK(i) = H(KDK(p), ch);$$

where the number ch is used to differentiate children nodes of the same parent node and H is a cryptographic hash function. Group keys are generated at each node i in the structure as:

$$TGK = H(KDK(i), KI);$$

where KI (key index) is the identifier of the generated key TGK (refer to [54]).

Group key exchange: The GCKS sends a set of key derivation keys (KDKs) accordingly to the device type (refer to Figure 6-16 and Figure 6-17) Based on these initial KDKs, nodes can derive lower-level KDKs (e.g., in the sub-tree rooted by their level) i.e., {KDK(j), all levels j lower than i}, to finally generate group keys TGKs (refer to Figure 6-17).

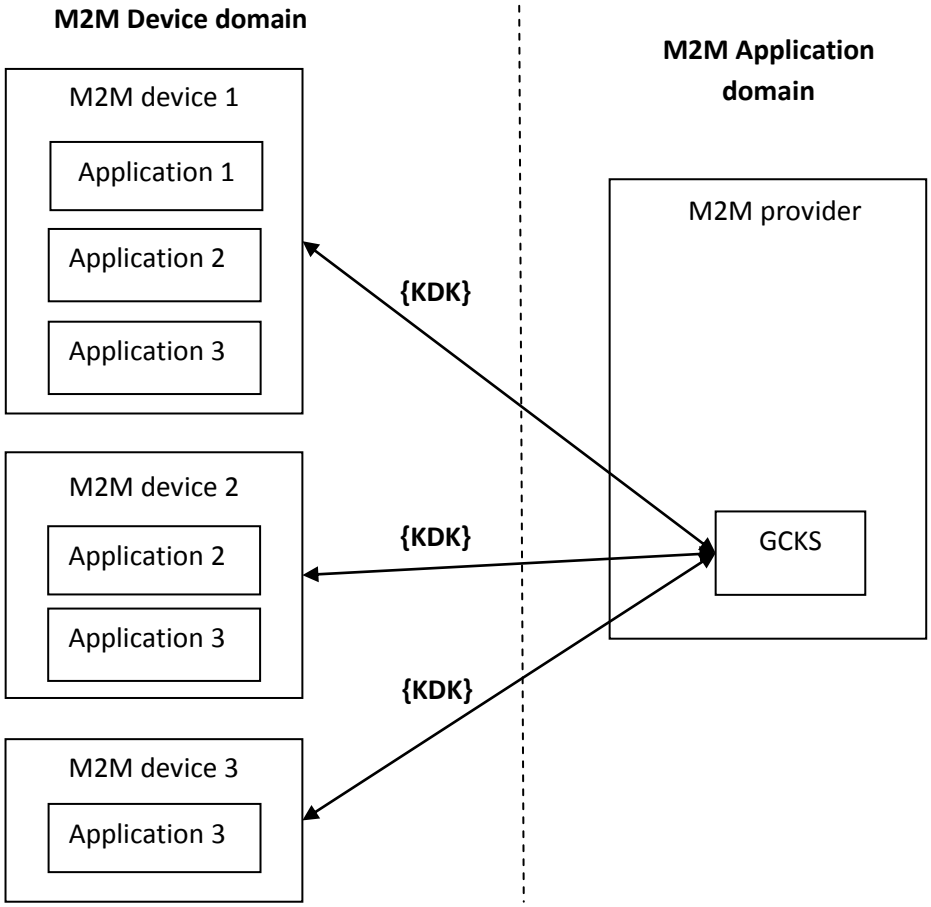


Figure 6-16: Distribution of KDK for group keys to devices

TGKs are used to generate the corresponding TEKs (Traffic Encryption Keys) that allows decrypting firmware images that the node is authorized to access.

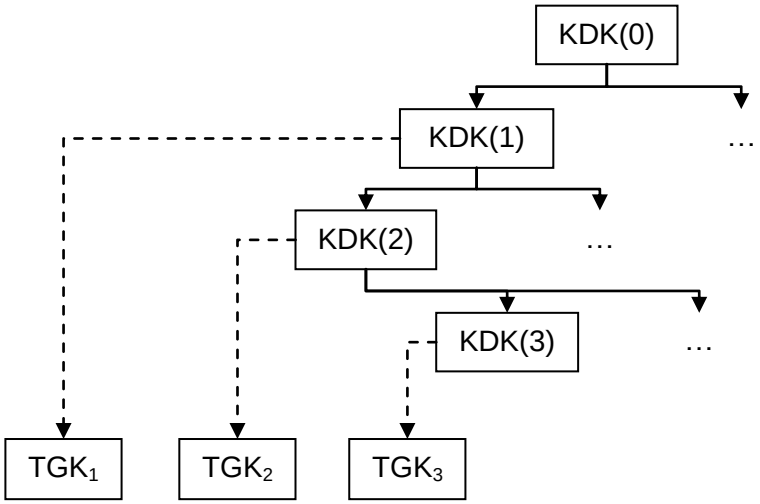


Figure 6-17: KDK structure: M2M devices of type 1 (respectively, 2, 3) can derive group keys {TGK1, TGK2, TGK3} (respectively {TGK2, TGK3}, {TGK3})

Group key update: When a node leaves or joins the group, group keys TGKs should be updated, and subsequently KDKs in the KDK structure that are associated with levels below the level of the device type where the change occurred. Nodes of the same type should receive new KDKs based on unicast communications, as in the initial group key exchange phase. On the other hand, the remaining nodes will be notified of the new TGK through a new KI.

With the proposed approach, nodes in the highest level (i.e., levels above the level of change) keep most of their TGKs and generate the remaining TGKs locally. Consequently, the rekeying operation triggered by group member join or leave requires less rekeying messages than in the base standard protocol MIKEY.

7 Implementing broadcast service capability within ETSI M2M architecture

The security solutions presented above assume that an initial security bootstrap has been performed between M2M devices and a remote provisioning server.

This initial security bootstrap result in the definition of a master secret which is typically used to secure the transmission of a broadcast group key which is used by the recipients to decode data received via broadcast/multicast.

On the other hand, ETSI M2M group has defined an M2M platform architecture specifying a number of “service capabilities”, the goal being to improve scalability, interoperability and ease of development of M2M applications.

In order to facilitate the development of applications relying upon broadcast/multicast it could be valuable to provide a broadcast service capability accessible from the ETSI M2M service platform. This service capability could be made available by M2M services providers to their customers.

This section presents one step towards the enhancement of the ETSI M2M architecture to offer a broadcast service capability. It will focus only on the security aspect, and more specifically the security bootstrap.

Two cases will be envisaged:

- Broadcasting to/from devices directly connected to the ETSI M2M platform from which EXALTED architecture is inspired
- Broadcasting to/from devices located behind a gateway itself connected to the ETSI M2M service platform.

7.1 Devices directly connected to the ETSI M2M platform

This paragraph describes one solution to secure data transmitted via multicast from the NSCL (Network service Capability layer) to the M2M nodes within the ETSI M2M architecture. It essentially relies upon the use of a master key to distribute a broadcast key to the devices.

The ETSI M2M architecture is relying upon a hierarchy of keys which are used for different levels of authentication and authorization.

In a security bootstrapping phase, the device defines a shared key **Kmr** used for mutual authentication and key agreement between the M2M node and the M2M service provider.

This long term key is later diversified into a shorter lifetime session key **Ka** which is then used to protect the data connection between the M2M node and the NSCL.

Figure 7-1 represents a typical situation where several M2M nodes (numbered i) have secured their connection with an NSCL using different key **Kai**. As illustrated, the devices may be either standalone devices directly accessible from the M2M server, for example LTE-M devices or gateway devices.

The keys Kai cannot be used for the NSCL to transmit common information to all the nodes via multicasting as they are all different.

However, the NSCL can use the keys **Kai**. To protect the transmission to each M2M node numbered **i** of a common broadcast key **kb** sent in unicast mode.

This key may be used to protect the broadcasted information from the NSCL to all devices

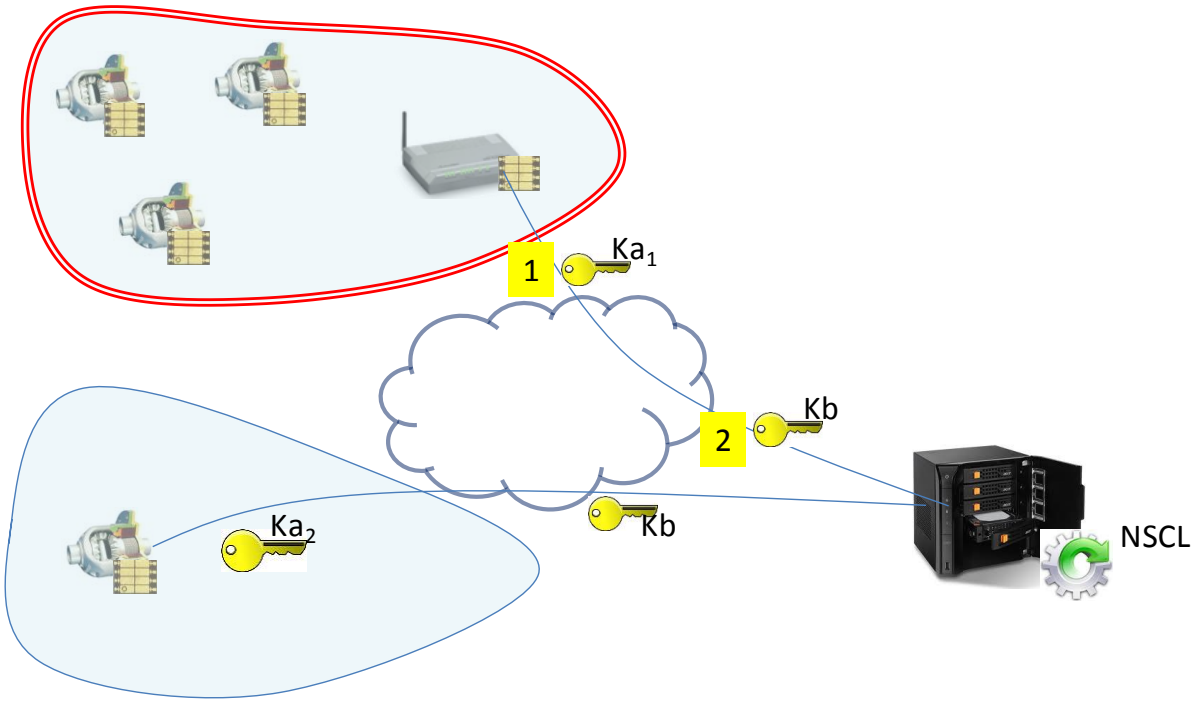


Figure 7-1: Securing broadcasted data

The broadcast key **Kb** can be sent to each device in unicast mode ahead of the transmission of the broadcasted data. With this scheme however the data broadcast or multicast cannot begin before all nodes have received the broadcast key without a risk of data loss

Another solution is to include in the data broadcasted or multicasted to the devices a pointer to the key **Kb**. Upon reception of multicast data, the M2M node will contact an authorization server and authenticate with its key **Kai** in order to obtain the key **Kb**.

7.2 Devices located behind a gateway

This subsection investigates the particular case of M2M devices are located behind a gateway which is connected to an ETSI compatible M2M service platform

In this scenario the capillary devices are located behind a gateway connected to an IP network. The gateway bootstraps its connection with the M2M services provider and both the gateway and the provider sides then derive a key **Ka** that will be used to secure the data communication between the gateway and the NSCL.

The key **Ka** can be used to protect the transmission from the NSCL to the gateway of a broadcast or multicast group key (**Kb**) that will protect data broadcasted or multicasted from the NSCL. The gateway then relays information to the capillary devices.

Two options may then be envisaged for the gateway to secure the propagation of multicasted information from the NSCL inside the capillary network:

1. The gateway, acting as a group leader, has performed a security bootstrap inside the capillary network which will result in the definition of a local group key Kgl shared by the gateway and all capillary devices. Different methods to perform this security bootstrap have been described in [34]. Depending on security requirements of the use-case, the gateway will then rekey each message received from the NSCL and ciphered with Kg

before rebroadcasting it secured with the Kgl key within the capillary network. Most likely, integrity protection of the message is more important than re-encryption of the message. This method has the drawback of requiring a rekeying operation at the level of the gateway. It is illustrated on Figure 7-2

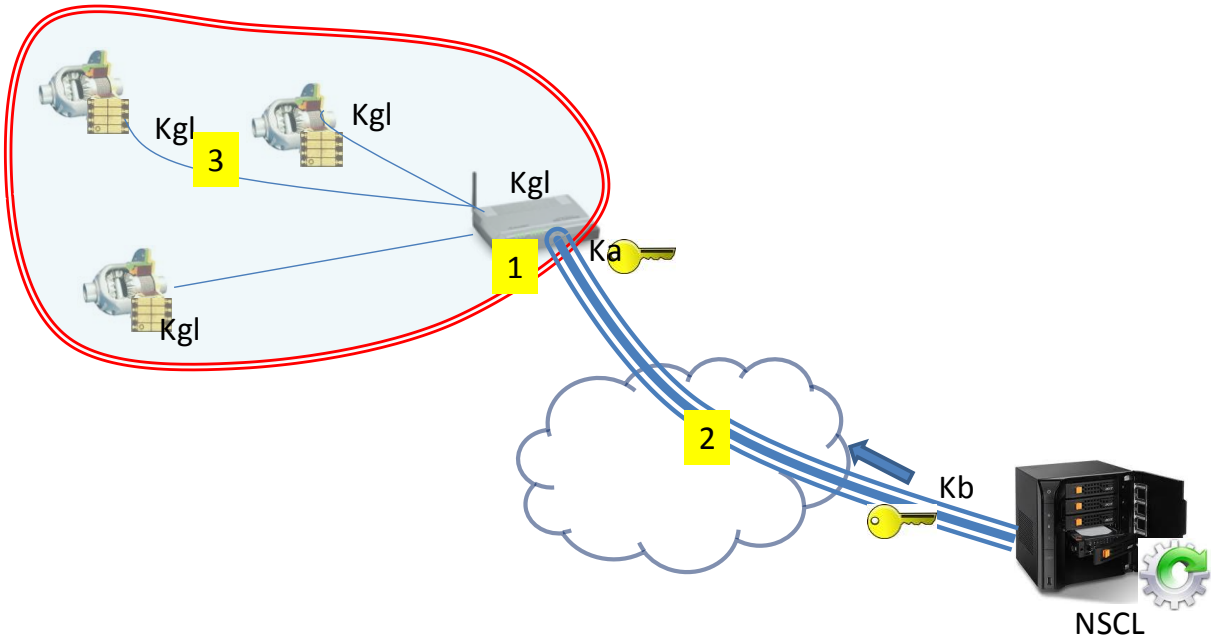


Figure 7-2: Broadcast to capillary devices located behind a gateway (option 1)

2. The gateway acting as a group leader will perform a security bootstrap inside the capillary network which will result in the definition of a group key Kgl shared by the gateway and all capillary devices and use this group key to transmit the key Kb to all devices in the capillary network. The advantage of this method is to avoid rekeying the data multicasted from the NSCL which can then be rebroadcasted without modification of the payload to the capillary devices. However, note that if there is any need to reformat the payload data (e.g. re-partition data blocks, protocol translation, insertion or removal of headers) then this won't work. This method is illustrated on Figure 7-3

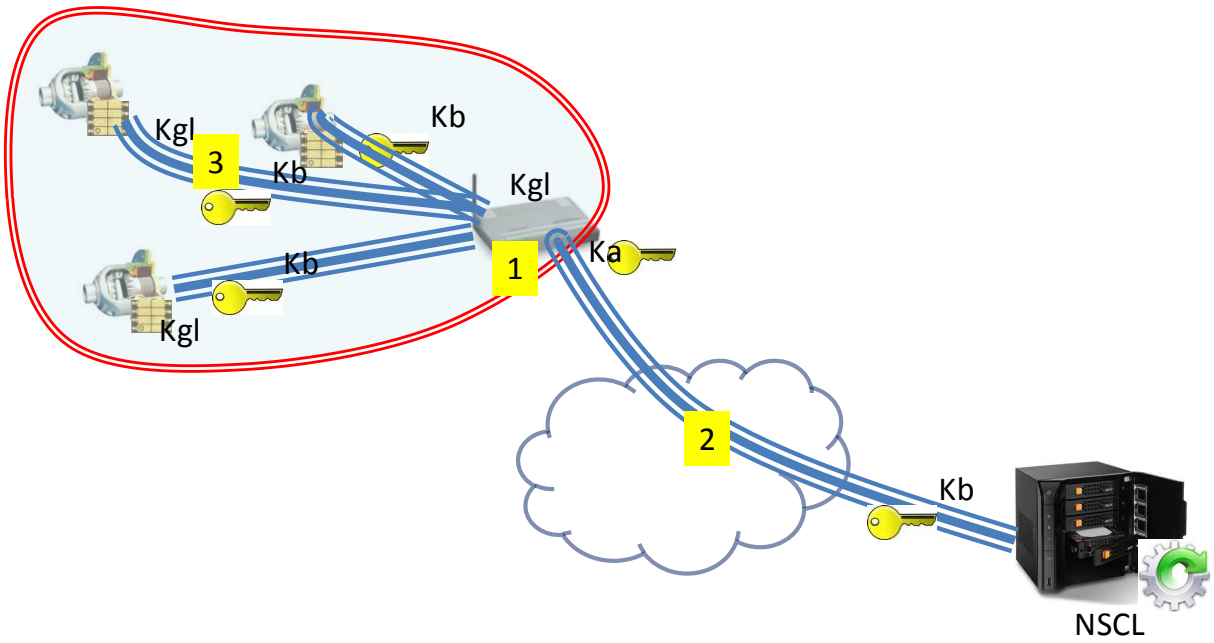


Figure 7-3 : Broadcast to capillary devices behind a gateway (option 2)

7.2.1 LTE-M enabled capillary devices connected to the LTE-M network

This scenario represents a special case of gateway relayed broadcast. It involves LTE-M enabled devices directly connected to the LTE-M network. Although there does not seem to be any gateways involved in this, the LTE-M network is actually relaying the broadcasted data over the last communication hop (LTE-M radio link) by radio broadcasting data to all devices in the same cell.

As in the previous case each of the devices has bootstrapped the security of its connection with the M2M services provider and both the device and the provider sides have derived a key **Kai** that will be used to secure the data communication between the device and the NSCL during a session.

The key **Kai** can be used to protect the transmission from the NSCL to the device number i of a multicast group key Kb that will protect data multicasted from the NSCL.

As in the previous section, the key **Kai** can be used to protect the transmission of a broadcasting group key **Kg** to the device numbered **i**.

The MBMS system described above may be used to multicast data from the NSCL to the devices using the evolved MBMS facility to broadcast data to LTE-M enabled devices, using Broadcast facilities from eNodeB

Again, as described in Section 4.2.1.2 above, the NSCL itself will need to be a BM-SC or to have BM-SC capabilities, which requires tight integration between the M2M service provider (NSCL) and the network operator. The bootstrapping and derivation of the **Kai** keys will need to be based on GBA, and aligned with the key hierarchy described in Figure 7-4 this requires the **Kai** to coincide with the MUK and the key **Kg** to coincide with an MSK.

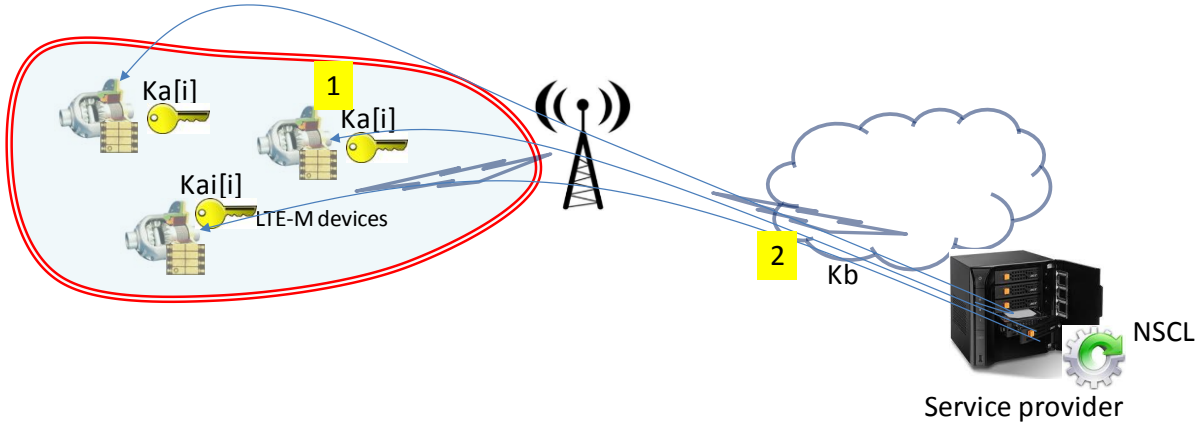


Figure 7-4 protected transmission of a broadcast key to M2M devices

8 Standardization activity

This section the standardization opportunities related to the topics covered in this report along with a possible associated timeline.

8.1 ETSI (ONE M2M) M2M working Group

ETSI TC M2M has defined an end to end M2M functional architecture designed to make use of an IP capable underlying network. The main focus is with IP service provided by 3GPP, TISPAN and 3GPP2 compliant systems, but the use of other IP capable networks is not excluded. This effort is now being transferred to the oneM2M International partnership, which is still at the requirement stage.

The whole architecture revolves around the notion of service capability. Service capabilities provide functions that have to be shared by different applications. They expose those capabilities through a set of open interfaces to facilitate application development. Service capabilities also allow to simplify and optimize applications development and deployment and to hide network specificities to applications. Capabilities may be M2M specific or generic, i.e. providing support to other than M2M applications. Examples include: data storage and aggregation, unicast and multicast message delivery, etc.

The requirements for this platform are described in document TS 102 689 [58] . Section 4.3 of this document states the need to support broadcast and multicast data: "*The M2M system shall support anycast, unicast, multicast and broadcast communication modes. Whenever possible a global broadcast should be replaced by a multicast or anycast in order to minimize the load on the communication network*".

It does not seem that this requirement is explicitly addressed by the current architecture. On the other hand, timing of the ETSI M2M group has focused towards the finalization of release 2 and transferring its existing work to oneM2M.

The definition of a new M2M architecture backed by OMEM2M (and probably largely inspired from the ETSI M2M architecture) in 2013 should provide opportunities to submit contributions related to the use of broadcast or multicast to distribute data from/to devices

8.2 OMA device management

3 devices Management solutions currently been defined by OMA have been mentioned in this document
Here is some information regarding the schedule of the specifications releases.

8.2.1 DM 2.0 delivery schedule

Table 8-1 : OMA 2.0 delivery schedule

Type of document	status
Requirements	approved 2011-12-20
Architecture	approved 2012-03-06
Technical Specification	targeted approval 2013-06-09

8.2.2 LWM2M delivery schedule

Table 8-2 : LWM2M delivery schedule

Type of document	status
Requirements	approved 2012-10-02
Architecture	approved 2012-11-27
Technical Specification	targeted approval 2013-05-30

8.2.3 GWMO

Table 8-3 : LWM2M delivery schedule

Type of document	status
Requirements version 1.0	approved 2012-10-02
Requirements version 1.1	targeted approval 2013-02-01
Architecture 1.0	approved 2010-11-02
Architecture 1.1	targeted approval 2012-03-05
Technical Specification 1.0	approved 2012-03-06
Technical Specification 1.1	targeted approval 2014-10-15

8.3 Emergency broadcast

The existing 3GPP documents provide very little information about securing emergency broadcast systems. The only relevant security requirement appears to be the one from TS 122 268 [59]. Section 4.8 which mentions that:

Security requirements are as follows: PWS shall only broadcast warning notifications that come from an authenticated authorized source.

NOTE 1: This requirement is subject to regulatory policies.

NOTE 2: The authentication and authorization of the source are outside the scope of 3GPP specifications.

This is not precise, and sets no firm requirements on the system. (The need to authenticate and authorize a broadcast service provider was already identified by MBMS and BCAST, so it is covered in EXALTED by the requirements already listed in section 3.) For an apparently critical public service, there is embarrassingly little about security here.

There is a parameter “Warning Security Information” included in another 3GPP specification TS 123 041 [60]. This specification is called “Technical Realization of Cell Broadcast Service (CBS)” i.e. the emergency broadcast system is overlaid on the old cell broadcast service, which has been in existence since GSM. CBS has low capacity, and isn’t used to carry much information (e.g. advice on area codes). The “Warning Security Information” is squeezed into a very small set of fields: 7 byte timestamp and a 43 byte digital signature. It is not defined how to construct it, and there is no space for certificate chains etc.

This leads to the following standardization actions:

1. Enhance the statement of requirements for emergency broadcast, so they are more comprehensive, and analogous to the requirements for MBMS, including integrity protection, protection against denial of service, spoofing etc.
2. Specify the format of the warning security information to ensure it is constructed in a genuinely secure manner. Possibilities here include an elliptic curve (EC-DSA) signature, perhaps with message recovery to add space, or a Message Authentication Code,

constructed using a key that is encrypted under an existing broadcast group key (there is probably sufficient space to send an encrypted key block as well as the MAC). A symmetric key imposes greater risk of a fake broadcast, but clever approaches would ensure that the actual MAC key itself is not distributed until there is an emergency broadcast to secure.

3. Specify mechanisms to distribute the initial keying information at network attachment, or perhaps via the UICC (e.g. an authenticated elliptic curve public key, or a broadcast group key together with a digest of the MAC key that will be used in the event of an emergency broadcast). One issue is that the emergency broadcast may need to reach unattached subscribers, or subscribers in “limited service mode”. However, as long as the device has attached to any network in recent history, it may retain a record of the public key or group key until needed.

9 Conclusion

This report has been dealing with M2M device management/provisioning solutions using broadcast/multicast. It was primarily focused upon the security aspect of the broadcast/multicast transmission, and this has led to distinguish various scenarios and make a number of proposals that we attempt to recapitulate:

When devices are connected to a 3GPP network, the use of MBMS to distribute the same device management payload (such as a software update) to a large number of devices has been investigated.

The solution would fit quite well with device management solutions offered by OMA, and particularly OMA-DM lightweight, providing there is support for the notion of group key, critical to make broadcast possible. As it stands, it seems that this notion of group key is **not** supported as OMA-DM solutions rely solely upon unicast communication.

MBMS could be used to disseminate the payloads (and in particular software updates) should OMA-DM lightweight support the group key based security, and in this case GBA would be a good solution to use for security bootstrap. A lightweight version of the GBA well suited to small M2M devices has been proposed and the analysis of a sample exchange has shown that it results in more compact exchange than the traditional GBA HTTP mapping. A problem in defining CoAP mapping has been however the lack of standardized CoAP "options fields". Should the idea of GBA/CoAP be adopted, there would be a need to standardize the corresponding CoAP options (equivalent of HTTP headers).

The management IP connected devices and in particular to devices located behind a gateway has been studied. It turns out that specific requirements of the LAN part of the transmission such as: energy efficiency, computing power restrictions, need to route through multihop networks, will shape completely the overall device management solution.

One possibility is to avoid designing an end to end solution and use the gateway as a proxy device (combining a WAN specific solution with a different LAN specific one). We have investigated this option. One big advantage to it is that it provides a simple solution to deal with non IP enabled devices located behind the gateway. It requires however to execute a specific application on the gateway.

The use of generic gateways, which do not hold custom applications, has been investigated along with end to end device management solutions. A possible architecture using CoAP in combination with IP multicast has been briefly described and this led to discuss solutions to achieve broadcast source authentication and early detection of bogus transmissions. Those proposals will have to be explored further to take them to a higher level of implementation.

Broadcast/multicast transmissions are generally secured using a group key distributed ahead of transmission to all members of the multicast group. Improvements to this group key distribution model have been proposed, as well as the possibilities to better deal with the situation of group members entering or leaving the group. Again those proposals will need maturing.

Finally, the problem of the very first security bootstrap has been tackled and the benefits brought by an interoperable M2M platform such as the one proposed by ETSI M2M discussed .The suggestion to implement a new service capability layer inside the ETSI architecture to simplify the development of M2M applications using broadcast/multicast was formulated and the security aspects of this proposal investigated. This solution would need to be standardized in order to become readily useable.

Appendix 1 broadcast/multicast usages other than device provisioning

In this appendix, we describe how broadcast/multicast techniques can be used for applications different that device management/provisioning.
2 such applications are described:

- The first one relates to security bootstrapping in capillary network. In this case the interest of broadcast relates to the fact that information broadcasted cannot easily be suppressed or modified by an adversary power as this may be done in man in the middle attacks for peer to peer communications.
- The second describes the use of broadcast in public infrastructure towards the implementation of public warning systems.

A1.1 Broadcast to achieve security pairing in self organized capillary networks.

Pairing is the process of key establishment between two or more devices without relying on pre-shared secrets or Public Key Infrastructure (PKI).The outcome of a pairing mechanism can be pair-wise keys, group key, or authenticated public keys. Pairing protocols have been studied in detail in EXALTED D5.3 [34]and we have recommended the Hash Commitment before Knowledge (HCBK) protocol for pairing in capillary networks. In this section we investigate the added value of using broadcast instead of unicast in this protocol.

Pairing requires exchanging different messages between paired devices. In case of capillary network where a number of numerous devices want to be paired with each other or with a centralized gateway, using broadcast communication reduces the number of required messages and protocol's stages. Figure 9-1 shows the HCBK protocol. In HCBK, initially every node (sensors) including the group leader (gateway) broadcasts its public key and identifier. Then the group leader broadcasts a secure hash of a freshly generated long key K and group members (sensors) acknowledge its receipt by using Out-of-Band (Oboe) channel. With getting the entire group member's acknowledgement, the group leader reveals its key K which is used by everyone to derive a short digest on all the public keys. The nodes compare their digests over the Oboe channel e.g. they present it to the user who accepts or rejects the pairing process based on the results.

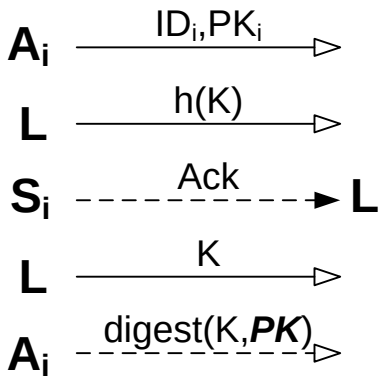


Figure 9-1 HCBK based group device pairing protocol

One of the main reasons of choosing HCBK protocol was its communication efficiency because of its proper use of broadcasting. HCBK has three rounds of communication over the wireless channel. In this protocol every nodes except the group leader only broadcast one message. The group leader broadcasts three messages. As the protocol requires the receipt of messages by all nodes using unicast instead of broadcast, the communication involves n communications per node and 3n communications for gateway (where n is the size of capillary network).

The protocol's usability can be improved by applying new in-band wireless pairing method [50] In this approach, instead of presenting the digests to "user" for checking, the leader can broadcast its digest to the group over insecure wireless channel in a series of noisy bursts (ones) separated by silences (zeros). The group check that they receive the bursts at the right times from the other devices. As discussed in [50], this approach exploiting the Integrity Codes to protect digest's integrity over wireless channel and it is immune from Man-In-The-Middle (MITM) attack, since a MITM can only insert additional bursts into the silences and he/she cannot change bursts into silences and then the hashes will not match.

A1.2 Public warning broadcast systems

From 2007 it has become common requirement for the phone manufactures to enable receiving warning messages to their mobile devices therefore alerting systems will have a secure place in the future. Public warning systems are developed in many countries and we list some of them together with their capabilities and general requirements.

A1.2.1 General requirements for the Public Warning System (PWS)

One of the first questions that emerged was how to push a large number of important notifications to end users and how to ensure that they are received in shortest possible time. Possible way to push emergency notifications towards many users is by using SMS (Short Message Service), TR 122.968 [24].

SMS is defined according to 3GPP recommendations as a Store and Foreword Service; therefore SMSs are just temporary stored until they are foreword to the user. In regular situations SMSs are delivered in a few seconds time, but in some particular cases it is possible that the SM (Short Message) delay is far longer (even few days). It is even possible that the SMS Centre (SMSC) ends up deleting the message. Since SMS is a Point to Point service (P2P), therefore in some cases it is possible to use bulk messaging that is proprietary and not defined in 3GPP.

However, SMS is not ideal to be used where real time is a criterion. Also, SMSC can push messages to the users located in some area, but obtaining the list of the users in some particular area is very slow and hard.

One of the most important parameters when speaking of alert notification delivery is time. It depends on various factors but the most important ones are: amount of information that is going to be sent, size of the warning notification, priority of notification itself, number of users that is going to be delivered, UE capabilities and techniques for notification formatting, RAN, technology used for broadcasting (SMS, MMS, MBMS etc..), TR 122.968 [24].

What are the expectations? For systems designed for public warning it is required to send multiple users notifications simultaneously and acknowledgement is not needed. A representative set of requirements is provided in the specification TS 122.268 [25]:

- Sending of concurrent broadcast notifications to many users
- Sending of notifications to many users in certain defined geographical area
- UE must be capable to receive notifications in idle state
- System works on FIFO (first in-first out) basis
- Received warning notifications must not interrupt voice or data session

Notification content shall contain the following elements, TS 22.268 [25]:

- Event Description
- Area Affected

- Recommended Action
- Expiration Time
- Sending Agency

Warning systems must support the capability to send various different notifications as well as a capability to update existing notifications that are being sent or to cancel some of the previous defined for sending.

Requirements for UE capabilities are listed in TS 122.268 [25]:

- UE shall be possible to display warning notification without any interaction from the user side
- UE shall not have the capability to copy, foreword, or to replay on any of the received warning messages
- UE will automatically ignore all duplicate incoming warning notifications
- UE will have a special attenuation tone and possible vibration dedicated only for warning notifications and they can be switched off only by the user, and even then they are not turned off for the new warning notifications possible to come

Note: It is possible that the user itself can disable in the user menu the reception of the warning messages if the regulatory agencies allow that. Similar the HPLMN operator can instruct UE not to receive (to ignore) incoming warning messages.

For the roaming it is possible to retrieve warning message when the user is in the VPLMN if the user is allowed to receive warning notifications in HPLMN and if both operators support PWS (Public Warning System).

In some particular cases, depending on regulation policies in each country and depending of the type of alerting system it is possible to define time window in which notifications should be delivered as well as the priority in congestion cases for each of the notification messages.

Parameters for configuration provisioning are defined according to 3GPP, TR 122.968 [24]: duration of delivery time, defining of the Notification Area, information element and size, indication with user interface, indication with interaction with the other services activation in the handset, priority, security, Warning Notification cancellation, delivery and receipt confirmation etc.

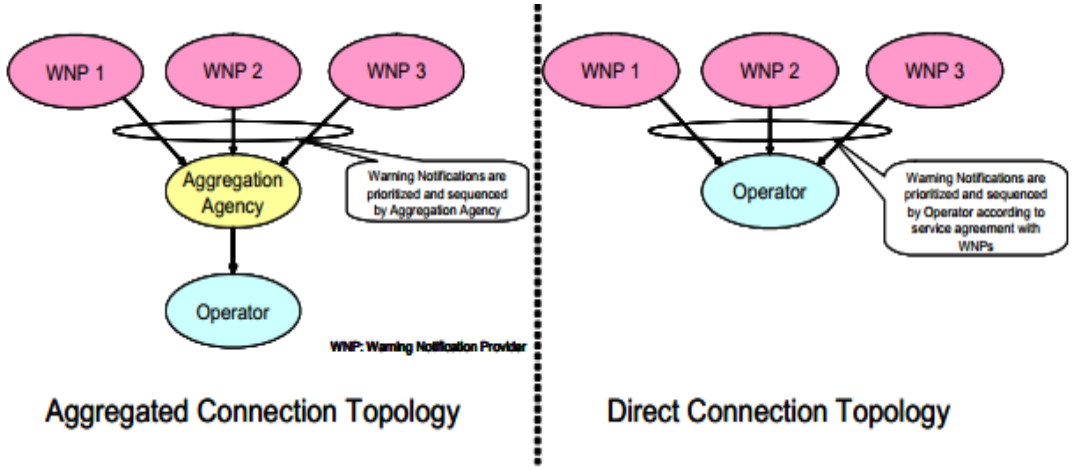


Figure 9-2 : Connection types between PLMN and WNP **[24]**

There are two possible kinds of connections between PLMN and WNP (Wireless Network Providers), TR 122.968 [24]. The first scenario includes that WNPs are connected to an

Aggregation Agency in which warning notifications are basically relayed towards PLMN operator. The other scenario includes a direct connection between WNP and PLMN. The advantage of the first scenario is that aggregation agency can provide the sequencing based on priority of the incoming warning notification in cases if they are sent simultaneously.

A1.2.2 Specific Public Warning Systems

A1.2.2.1 Emergency Broadcast System

EBS (Emergency Broadcast System) system was used as an emergency warning system in USA from 1961 until 1997 when it was replaced by Emergency Alert System [26].

The role of EBS was to provide the efficient communication link between the government and people in emergency situations like war or some natural disasters. It was used mostly for purpose of warning on weather disasters. The system was designed on a way to provide efficient and quick communication. The special broadcast frequency was assigned to alert other broadcast stations on the incoming message that is going to be broadcasted. The system used frequencies from 853 to 960 Hz. Decoders placed on relay stations would sound an alarm. This is used in order to alert operator on the station of the upcoming alert message. After that relay stations would broadcast the received alarm tone and continue with rebroadcasting the emergency message from the primary station.

A1.2.2.2 Emergency Alert System

EBS was replaced with Emergency Alert System (EAS) in 1997 [26]. The role of EAS is similar to EBS, to do public alert on whether emergencies alerts like tornadoes and to broadcast message from the government officials in cases of national interests. The work of EAS is standardized and regulations are made from the appropriate government agencies in the USA.

EAS uses messages that are made of four different parts [27]:

- Digitally encoded SAME header – this part of the message is defining the information regarding who started the alert, alert description, the area affected, the time period and other important parameters according to specification
- An attention signal
- Audio announcement
- Digitally encoded end of message marker.

The header is repeated three times in order to minimize the possibility of errors. This process also has its side effects, because it adds redundant information to the signal. The role of EAS decoders is to compare received headers to some referent header. If the two matches then the errors are eliminated. The decoder has the right to decide whether to send the message or to ignore it.

After initial header burst follows the attention signal that lasts for 25 seconds (1050 HZ). The signal on commercial broadcast station consists of a signal combined of two frequencies (853 Hz and 960 Hz). Attention signal is then followed by a voice message which is caring relevant details about the alert.

According to the regulations all broadcast stations are required to install EAS coders and decoders at their headends. Their role is to supervise the signals from nearby broadcast stations and to detect EAS messages. In order to increase reliability at least two broadcast stations are monitored.

A1.2.2.3 Commercial Mobile Alert System

The Commercial Mobile Alert System (CMAS) [26] is another network used for alerting needs. It is used for sending emergency alerts to mobile phones. The structure of the network is designed on a way to provide the sending of emergency messages firstly to wireless operators participating in the CMAS network system, and then each of the operators will push alerting message further to their subscribers by using SMS. There is a possibility to have different levels of priority between messages that are sent. For example messages with lower priority can be blocked by the end user, but those messages that are highest prioritized cannot be blocked and they will be delivered to each customer. The customers are therefore automatically provisioned to receive CMAS alerts. Many of the operators joined to participate in this alerting system such as: Verizon, T-Mobile, AT&T etc.

Alerting messages are delivered by E-UTRAN. CMAS is based on capability of mapping CMAS messages into 3GPP defined Broadcast functionality from the point of reception of the CMAS alert message by the WNP (Warning Notification Provider) to the point where CMAS capable mobile devices have received the alerting message, TS 122.268 [25].

The picture illustrating the process is Figure 9-3.

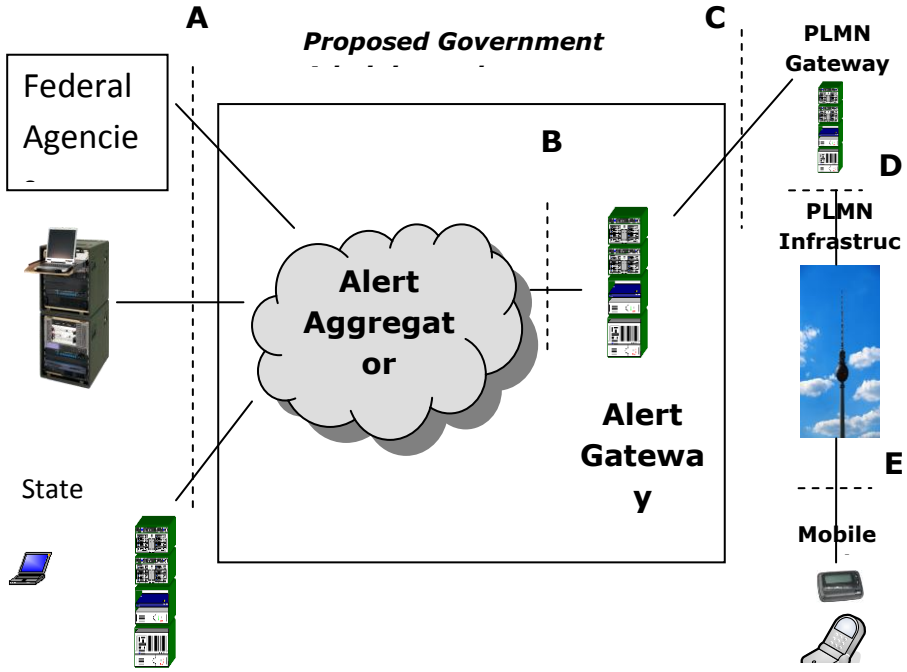


Figure 9-3 : CMAS architecture [25]

CMAS is designed on the assumption that point to point (P2P) or unicast techniques are not adequate for wireless alerts. P2P if used will lead to quick congestion in the network, and congestion will lead to message delays or even lost. CMAS do not requires ACK (acknowledgment) from the user and it is based on unidirectional delivery from wireless operator network to users.

CMAS is using Alerting GW which acts as an interface in between the aggregator and wireless operator network and he should support multiple media profiles like text, audio, video, multimedia etc.

The format of alerting messages should be in such form so that it does not require any modification by the operators (they are just forwarding to the users notifications as they are).

Alerting GW must be redundant and reliable. Multiple access points and delivery points should be supported as well.

Each message (within the same notification) must have the same format and the message identifier when sending to the operator.

A1.2.2.4 Europe Alerting System

Europe Alerting System is being used in some European countries, TS 102.900 [27]. It introduces three levels of severity.

Public Warning Systems (PWS) should be capable to send messages to a large numbers of individuals within special affected areas, within planned time frame to large number of users.

The system supports alert messages according to 3GPP TR 122.968 [24]. The specification is defining the delivering of the alert notifications:

- to 50% of the users within 3 minutes, and
- to 97% of the users within 5 minutes

Ack can be used if supported but it is not necessary. In case when not used then the delivery can be repeated as long as message notification is valid.

ENS (Emergency notification systems) are dealing with an overall protection of the data that are used for proper operation of the system. These systems must be designed on a way to provide tracking, capturing and performance information feedback.

A1.2.2.5 Digital Emergency Alert Systems

Digital Emergency Alert Systems is one more alerting system. It is designed for alerting purposes and it relays on EAS system for sending text, voice, video and other digital content to mobile phones, television, radios etc. Some of the disadvantages of earlier alerting systems like EBS and EAS are corrected when using DEAS. DEAS allows broadcasting of "bottomless" audio messages and streaming [28].

A1.2.2.6 Common Alerting Protocol

Common Alerting Protocol (CAP) [24][29] is commonly used for public warnings. It is based on XML and it allows alerting message to be sent simultaneously by many alerting systems on a consistent way. CAP is good for detecting hostile acts, trends and patterns in warning activity.

Similar to this alerting systems from USA, there is an alerting system in Japan [24], designed for early earthquake warning. Also other countries are developing alerting systems for their own purposes.

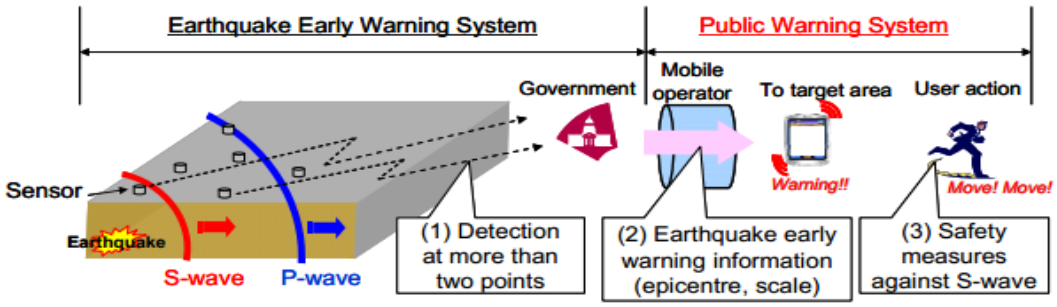


Figure 9-4 : Warning system interaction [24]

Acronyms

ACK	Acknowledgment
BCAST	Mobile Broadcast Services Enabler Suite
BCMCS	Broadcast Multicast Services
BDS	Broadcast Distribution Systems
CAP	Common Alerting Protocol
BSF	Bootstrapping Server Function
CMAS	Commercial Mobile Alert System
DEAS	Digital Emergency Alert Systems
DVBH-IPDC	Digital Video Broadcasting- IP Data casting
DVB-SH	Digital Video Broadcasting – Satellite services to Handhelds
e2e	end to end
EAS	Emergency Alert System
EBS	Emergency Broadcast System
ENS	Emergency notification systems
FIFO	First in-first out
GPI	GBA push info
HPLMN	Home PLMN
HLR	Home Location Register
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ISP	Internet Service provider
Ks_NAF	NAF-key in GBA_ME mode
LAN	Local Area Network
MBMS	Multimedia broadcast/multicast service
MBS	Multicast Broadcast Services
MIKEY	Multimedia Internet KEYing
MMS	Multimedia Message Service
MSE	Mobile Services Enabler
MSK	MBMS service key
MTK	MBMS Traffic Key
MUK	MBMS user key
NAF	Network application function
OMA	Open Mobile Alliance
OMA DM	Open Mobile Alliance - Device Management
P2P	Point to Point service
PLMN	Public Land Mobile Networks
PWS	Public Warning System
RAN	Radio Access Network

RMS	Rich Media Solution
SA	Security association
SM	Short Message
SMS	Short Message Service
SMSC	SMS Centre
UE	User Equipment
UICC	UMTS Integrated Circuit Card
VOD	Video On Demand
VPLMN	Visited PLMN
WAN	Wide Area Network
WAP	Wireless Application Protocol
WiMAX	Worldwide Interoperability for Microwave Access
WNP	Wireless Network Providers
WNP	Warning Notification Provider
WWW	World Wide Web

References

[1] Bora Karaoglu, Wendi Heinzelman: Multicasting vs. Broadcasting: What are the Trade-offs?

[2] FP7 EXALTED consortium: “D4.3 - Device Management” 2012-10

[3] FP7 EXALTED; “D2.3 – EXALTED System Architecture” project report, August 2012

[4] YU-CHEE TSENG, SZE-YAO NI, YUH-SHYAN CHEN, JANG-PING SHEU, “The Broadcast Storm Problem in a Mobile Ad Hoc Network” Wireless Networks 8, 153–167, 2002

[5] Fei Ye, Raymond Yim, Jinyun Zhang, Sumit Roy: “Congestion Control to Achieve Optimal Broadcast Efficiency in VANETs” IEEE International Conference on Communications 2010

[6] J. Moy. Multicast Extensions to OSPF. IETF RFC 1584, March 1994.

[7] D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol. IETF RFC 1075, November 1988

[8] A. Adams, J. Nicholas, and W. Siadak. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). IETF RFC 3973, January 2005

[9] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing. IETF RFC 2189, September 1997

[10] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). IETF RFC 4601, August 2006

[11] Jinu Kurian and Kamil Sarac. 2010. A survey on the design, applications, and enhancements of application-layer overlay networks. ACM Comput. Surv. 43, 1, Article 5 (December 2010).

[12] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. Scalable application layer multicast. In Proceedings of the 2002 conference on Applications.

[13] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany: 329–350.

[14] Minseok Kwon and Sonia Fahmy. Topology-aware overlay networks for group communication. In Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video (NOSSDAV '02). ACM, New York, NY, USA, 127-136.

[15] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: the second-generation onion router. In Proceedings of the 13th conference on USENIX Security Symposium - Volume 13 (SSYM'04), Vol. 13. USENIX Association, Berkeley, CA, USA, 21-21.

[16] Key Management for UMTS MBMS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 7, NO. 9, SEPTEMBER 2008, Shin-Ming Cheng, Member, IEEE, Wei-Ru Lai, Member, IEEE, Phone Lin, Senior Member, IEEE, and Kwang-Cheng Chen, Fellow, IEEE

[17] MBMS—IP Multicast/Broadcast in 3G Networks, International Journal of Digital Multimedia Broadcasting Volume 2009 (2009), Article ID 597848,Frank Hartung, Uwe Horn, Jörg Huschke, Markus Kampmann, and Thorsten Lohmar Ericsson GmbH, Eurolab R&D, Ericsson Allee 1, December 2009

[18] 3GPP TS 133.223: “Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push function”

[19] GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)"

[20] GPP TS 33.223: " Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push function”

[21] OMA-AD-BCAST-V1_0-20090212-A “Mobile Broadcast Services Architecture”

[22] OMA-RD-BCAST-V1_1 – latest candidate version 20100914-C, see http://www.openmobilealliance.org/technical/release_program/bcast_v1_1.aspx

[23] OMA-RD-BCAST-V1_0-20090212-A “Mobile Broadcast Services Requirements”, Approved Version 1.0 – 12 Feb 2009, Approved Version 1.0 – 12 Feb 2009

[24] 3GPP TR 122.968 “Study for requirements for a Public Warning System (PWS) service” -version 10.0.0, Release 10

[25] 3GPP TS 122.268 “Public Warning System (PWS) requirements” - version 10.3.0, Release 10

[26] Federal Communications Commissions, FCC, USA http://www.fcc.gov/topic/emergency-communications

[27] Emergency Alert System 2001 AM & FM Handbook -"Emergency Alert System 2001 AM & FM Handbook“, United States, Federal Communications Commission. 2001

[28] Sarkar, Dibya "FEMA tests digital alert system - Technology will send messages to wireless devices, radio, TV and the Internet". Federal Computer Week. Archived from the original on 2008-09-07. (2005-04-11).

[29] Common Alerting Protocol, v. 1.1 OASIS Standard CAP-V1.1, October 2005; https://www.oasis-open.org/committees/download.php/14759/emergency-CAPv1.1.pdf

[30] 3GPP TS 133 246 “Security of Multimedia Broadcast/Multicast Service (MBMS)” - latest release v10.0.0, 2011-5

[31] 3GPP TS 122 146 “Multimedia Broadcast/Multicast Service (MBMS); Stage 1”- latest release v10.1.0, 2012-01

[32] 3GPP TS 122 246 “Multimedia Broadcast/Multicast Service (MBMS) user services; Stage 1” - latest release v10.0.0, 2011-5

[33] FP7 EXALTED consortium: “D2.1 - Description of baseline reference systems, scenarios, technical requirements & evaluation methodology”

[34] FP7 EXALTED consortium: “D5.3 - Security and Provisioning Solutions project report, Feb 2013

[35] Raja Jurdak, Antonio G. Ruzzelli, Gregory M. P. O'hare, and Russell Higgs. 2010. Directed broadcast with overhearing for sensor networks. ACM Trans. Sen. Netw. 6, 1, Article 3 (January 2010), 35 pages.

[36] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. The Trickle Algorithm. IETF RFC 6206, March 2011.

[37] J. Hui and R. Kelsey. Multicast Forwarding Using Trickle. IETF Internet-Draft draft-ietf-roll-trickle-mcast-00, April 11, 2011.

[38] C. Diot, B. N. Levine, B. Liles, H. Kassem, and D. Balensiefen. Deployment issues for the IP multicast service and architecture. IEEE Network, vol. 14, no. 1, pp. 78-88, Jan. 2000.

[39] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550, March 2012.

[40] A. Rahman and E. Dijk. Group Communication for CoAP. IETF Internet-Draft draft-ietf-core-groupcomm-01, CoRE Working Group, March 9, 2012.

[41] M.Hosseini, D.T. Ahmed, S. Shirmohammadi, and N.D. Georganas. A survey of application-layer multicast protocols. In: Communications Surveys and Tutorials, IEEE, vol. 9.

[42] Li Lao , Jun-hong Cui , Mario Gerla , and Dario Maggiorini. A Comparative Study of Multicast Protocols: Top, Bottom, or In the Middle? In Proceedings of the 8th IEEE Global Internet Symposium (GI'05) in conjunction with IEEE INFOCOM'05, 2005.

[43] Kui Ren. Communication Security in Wireless Sensor Networks. PhD Dissertation, Worcester Polytechnic Institute, May, 2007.

[44] A. Perrig, R. Canetti, J. Tygar and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In: IEEE Symposium on Security and Privacy (2000).

[45] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. SPINS: security protocols for sensor networks. Wirel. Netw. 8, 5 (September 2002), 521-534.

[46] D. Liu and P. Ning. Multi-level µTESLA: Broadcast authentication for distributed sensor networks. ACM Trans. Embedded Computing Systems (TECS), vol. 3, no. 4, pp. 800-836, 2004

[47] Hailun Tan, John Zic, Sanjay K. Jha, and Diethelm Ostry. Secure Multihop Network Programming with Multiple One-Way Key Chains. Mobile Computing, IEEE Transactions on Mobile Computing, vol.10, no.1, pp.16-31, Jan. 2011, doi: 10.1109/TMC.2010.140

[48] Haowen Chan and Adrian Perrig. "Round-Effcient Broadcast Authentication Protocols for Fixed Topology Classes." Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, May 2010.

[49] Wallner, D., Harder E., and Agee R. Key Management for Multicast: Issues and Architectures. IETF Internet Draft, Informational, September 1998.

[50] S. C˘ apkun, M. C˘ agalj, R. Rengaswamy, I. Tsigkogiannis, J.-P. Hubaux, and M. Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. IEEE Transactions on Dependable and Secure Computing, 5(4):208–223, October–December 2008.

[51] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In USENIX Security Sym., 2011

[52] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman. MIKEY: Multimedia Internet KEYing. IETF RFC 3830, August 2004.

[53] E. Carrara, V. Lehtovirta, and K. Norrman. The Key ID Information Type for the General Extension Payload in Multimedia Internet KEYing (MIKEY). IETF RFC 4563, June 2006

[54] R. Alexander and T. Tsao. Adapted Multimedia Internet KEYing (AMIKEY): An extension of Multimedia Internet KEYing (MIKEY) Methods for Generic LLN Environments. Internet-Draft, July 26, 2011

[55] Jonathan Katz and Andrew Y. Lindell. 2008. Aggregate message authentication codes. In Proceedings of the 2008 The Cryptopgraphers' Track at the RSA conference on Topics in cryptology (CT-RSA'08), Tal Malkin (Ed.)

[56] Vladimir Kolesnikov, Wonsuck Lee, and Junhee Hong. MAC aggregation resilient to DoS attacks. IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp.226-231, 17-20 Oct. 2011, doi: 10.1109/SmartGridComm.2011.6102323

[57] ETSI TS 102 690 (2010-09) : Machine- to- Machine communications (M2M); Functional architecture

[58] ETSI TS 102 689 Machine to Machine communications (M2M); M2M service requirements V1.1.1

[59] 3GPP TS 122 268 “Public Warning System (PWS) requirements” - latest release v10.3.0, 2012-04

[60] TS 123 041 “Technical Realization of Cell Broadcast Service (CBS)” - latest release v10.0.3, 2012-03.

[61] Z. Shelby, K. Hartke, C. Borman, B. Frank “Constrained Application Protocol (CoAP) draft-ietf-core-coap-13”, December 2012;

[62] IETF RFC 6690: “Constrained RESTful Environments (CoRE) Link Format”;

[63] Group Communication for CoAP, http://tools.ietf.org/html/draft-rahman-core-groupcomm-07;

[64] K. Kuladinithi, O. Bergmann, T. Pötsch, M. Becker, C. Görg “Implementation of CoAP and its Application in Transport Logistics”;

[65] T. Dimčić, S. Krčo, N. Gligorić “CoAP (Constrained Application Protocol) implementation in M2M Environmental Monitoring System

[66] 3GPP TS 24.109 V8.3.0, “Bootstrapping interface (Ub) and network application function interface (Ua)”;(Release 8), June 2010;

[67] 3GPP TS 33.222 V11.2.2: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)"; (Release 11), November 2012;

[68] 3GPP TS 24.109 V8.3.0, “Bootstrapping interface (Ub) and network application function interface (Ua)”;(Release 8), June 2010;

[69] ”The EXALTED system architecture”, EXALTED deliverable D3.2, 31 August 2012.

[70] ”Device Management”, EXALTED deliverable D4.3, 31 october 2012.

[71] Adam Chlipala, Jonathan Hui, Gilman Tolle :Deluge: Data Dissemination for Network Reprogramming at Scale : http://www.cs.berkeley.edu/~jwhui/deluge/cs262/cs262a-report.pdf

[72] Prabal K. Dutta, Jonathan W. Hui, David C. Chu, and David E. Culler. 2006. Securing the deluge Network programming system. In Proceedings of the 5th international conference on Information processing in sensor networks (IPSN '06). ACM, New York, NY, USA, 326-333.

[73] P. Levis et al., “Trickle: A Self-Regulating Algorithm for Code Propagation andMaintenance in Wireless Sensor Networks,” Proc. 1st Usenix/ACM Symp. Networked Sys- temsDesign and Implementation (NSDI 04), Usenix Assoc., 2004, pp. 15–28

[74] Alessandro Sorniotti, Refik Molva, and Laurent Gomez. Efficient access control for wireless sensor data. In Proceedings of PIMRC, 2008; IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 15–18 September, 2008, Cannes, France.

[75] Hani Ragab Hassen, Hatem Bettahar, Abdalmadjid Bouadbdallah, and Yacine Challal. 2012. An efficient key management scheme for content access control for linear hierarchies. Comput. Netw. 56, 8 (May 2012), 2107-2118. DOI=10.1016/j.comnet.2012.02.006 http://dx.doi.org/10.1016/j.comnet.2012.02.006

[76] J. Hui and P. Thubert, " Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", IETF RFC 6282, September 2011.

[77] Z. Shelby, K. Hartke, C. Bormann, and B. Frank. " Constrained Application Protocol (CoAP)", IETF Internet-Draft draft-ietf-core-coap-13, December 6, 2012

[78] S. Keoh, O. Garcia-Morchon, S. Kumar, E. Dijk. " DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs)". IETF Internet-Draft draft-keoh-tls-multicast-security-00, October 15, 2012

[79] T. Martin. "A Set Theoretic Approach to Broadcast Encryption". Ph.D Thesis, Royal Holloway University of London, 2005.

[80] http://www.aacsla.com/home

[81] OMA: Device Management V1.2; Approved Enabler-Release Date 17 June 2008. http://www.openmobilealliance.org/Technical/release_program/dm_v1_2.aspx

[82] OMA DM Representation Protocol; Approved Version 1.2.1, 17 June 2008. http://www.openmobilealliance.org/Technical/release_program/docs/DM/V1_2_1-20080617-A/OMA-TS-DM_RepPro-V1_2_1-20080617-A.pdf

[83] [RFC6347] Rescorla, E. and Modadugu, N. (2012) RFC6347. Datagram Transport Layer Security

[84] RFC5246] Dierks, T. and Rescorla, E. (2008) RFC5246. The Transport Layer Security (TLS) Protocol Version 1.2

[85] Z. Shelby, K. Hartke, C. Bormann, and B. Frank. Constrained Application Protocol (CoAP). draft-ietf-core-coap-09, 2012

[86] S. A. Camtepe and B. Yener, "Key Management in Wireless Sensor Networks", In Book: "Wireless Sensor Network Security", Javier Lopez and Jianying Zhou (editors), ISBN 978-1-58603-813-7, Cryptology & Information Security Series, IOS Press, 2008.