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 5.1
Security and Provisioning Solutions

Contractual Date of Delivery:	29 February 2012
Actual Date of Delivery:	29 February 2012
Responsible Beneficiary:	Vodafone
Contributing Beneficiaries:	CEA, Gemalto, UNIS, Vodafone, TKS
Estimated Person Months:	24
Security:	Public
Nature	Report
Version:	1.0

Document Information

Document ID: EXALTED_WP5_D5.1
Version Date: 24 February 2012
Total Number of Pages: 115

Authors

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

Approvals

	Name	Organisation	Date	Visa
Internal Reviewer 1	Juan Rico	TST	13/02/2012	OK
Internal Reviewer 2	Eleftheria Vellidou	VID	22/02/2012	OK
Internal Reviewer 3				
Technical Manager	Pirabakaran Navaratnam	UNIS	29/02/2012	OK
Project Manager	Djelal Raouf	SCET	29/02/2012	OK

Executive Summary

This first full deliverable from the security working group addresses the security architecture for LTE-M networks in the scope of the EXALTED project.

Security is a transversal initiative. Hence, a major goal of this report is to compile an overview of security basics, particularly as related to existing cellular networks (section 2). We then provide a consolidated set of security requirements originating from the different EXALTED functional working groups and from other working groups on M2M (section 3).

We intend to make sure that the EXALTED project deliverables will fit nicely in the existing standardization landscape. EXALTED is not a standalone, isolated project, dealing with topics of its own. It is no surprise that we find that some of the problems addressed by the project are already partially covered by existing working groups. This is particularly relevant for the issues related to the security architecture. This architecture will not be created from scratch, but will likely result from the evolution of existing solutions already in place. Therefore, a large part of this report is devoted analysis of the state of the art in standards from a security standpoint.

One of the requirements set out by EXALTED regarding security is the need to achieve low cost security, but do so in a large-scale, complex environment. Further the goal of reducing cost must not lead to a lower overall security level (since M2M devices may represent critical infrastructure), nor can it impact on device usability (since many devices will be unattended through much of their lives). This conflict between important goals, and the vital need not to sacrifice any of them, is dealt with in section 4 of this report, which discusses the new security challenges for the EXALTED vision. Significantly, lowering the cost of security will require an awareness of the security processes in place in LTE networks and associated costs, and these are detailed in section 4.2.

Possible solutions to these challenges are described in Section 5. With the objective of reducing device cost of ownership, we propose in this project to have a particular focus to M2M communications occurring with M2M devices located behind a gateway serving as a bridge between the wide area (LTE-M) network and the local area (capillary) network. In this context, significant cost reduction and network offload can be expected. Mechanisms for efficiently securing such a local network are introduced (section 5.3) and will be detailed further in subsequent reports related to Broadcast and Relay security.

With the objective of achieving security from the ground up, while minimizing costs and security overheads, we propose a particular focus on optimizations to network access security, and ways to merge network access security with “service layer” security (section 5.1). Also, it is expected that cost savings can be achieved with a single secure element embedded in an M2M device or M2M appliance, which can be remotely administered (section 5.2). The secure element could be shared among very low cost devices e.g. each LAN or PAN might only need one device holding a secure element. The secure element could then be used for multiple purposes such as securing remote device administration (section 5.4), authentication of the LAN's subscription onto the wide area network (section 5.5), device authentication within the LAN (sections 5.3 and 5.6), and bootstrapping the application level security mechanisms that will guarantee a secure transmission of the dataflow originating from/to each device (section 5.7).

Regarding application level security we investigate new business models for M2M communications where the trust provider is a business entity (an authorization server) which can be distinct from the M2M service provider (see section 5.7.2). This enables “end to end” data encryption from the source to destination, in the sense that it allows M2M service providers to provide the M2M communication infrastructure while carrying M2M data opaque to them. We indicate how this capability could be implemented within ETSI M2M architecture.

Having proposed a number of different solutions in Section 5, we return to standards to give an indication of which of them could be standardized and by when (section 6).

Table of Contents

1. Introduction

This report deals with the security architecture used to secure LTE-M communications.

The starting point is an overview of the EXALTED system architecture, followed by an analysis of basic security concepts, and their applicability in existing mobile cellular networks. This leads us to define the security requirements to be taken into account for LTE-M networks. They result from an analysis of requirements already identified by standardization groups working on M2M, as well as new LTE-M requirements.

We then present an overview of the particular security challenges arising out of the EXALTED vision, especially the need to reconcile low cost with a high overall security level, and high usability in the case of unattended devices. This leads us to provide a summary of the existing authentication and provisioning mechanisms used in LTE networks and perform a cost analysis which will explain the difficulty to support an exponential growth of M2M communications using existing mechanisms.

Next, we investigate possible solutions to those challenges, focusing especially on optimizations to network access security, on the embedded SIM and ways to make best use of it as a general purpose secure element.

Finally we will detail how EXALTED WP5 group intends to contribute to various industry standardization bodies in order to define solutions compatible with the security requirements specific to LTE-M networks.

1.1 Overview of EXALTED System Architecture

As background material, we will quickly re-cap the proposed high-level architecture of the EXALTED M2M system, as shown in Figure 1.1 below:

The **M2M Device Domain** basically involves the following devices:

A. **M2M Gateway**: it provides the interconnection between the LTE-X (i.e. LTE/LTE-A/LTE-M) network and the capillary networks (consisting of one or more M2M devices). It can provide various functionalities, such as protocol translation, routing, resource management, device management, data aggregation, etc. In some cases, the Gateway may also act as an application server providing M2M services locally in the capillary network.

B. **M2M devices**, which are devices that can support one or more M2M applications. They can be categorised into:
   - **LTE-M devices**: they have LTE-M interface and can access the application domain, either by directly accessing the LTE-M network, or through a M2M Gateway.
   - **Non-LTE-M devices**: they do not have LTE-M interface, but form capillary network(s) using other short-range network access technologies, such as ZigBee, and IEEE 802.11x. They can access the application domain through a M2M Gateway, and run M2M applications locally.

C. **Cluster heads** (CHs): They are considered as more powerful M2M devices with some additional capabilities. Like regular M2M devices, they are also part of capillary networks and the communication from a regular M2M device will be directed through and managed by a CH. The functionalities of a CH may include data aggregation,

device management, routing, etc. Depending on the network access technology they implement, they can be named as LTE-M CHs, and non-LTE-M CHs.

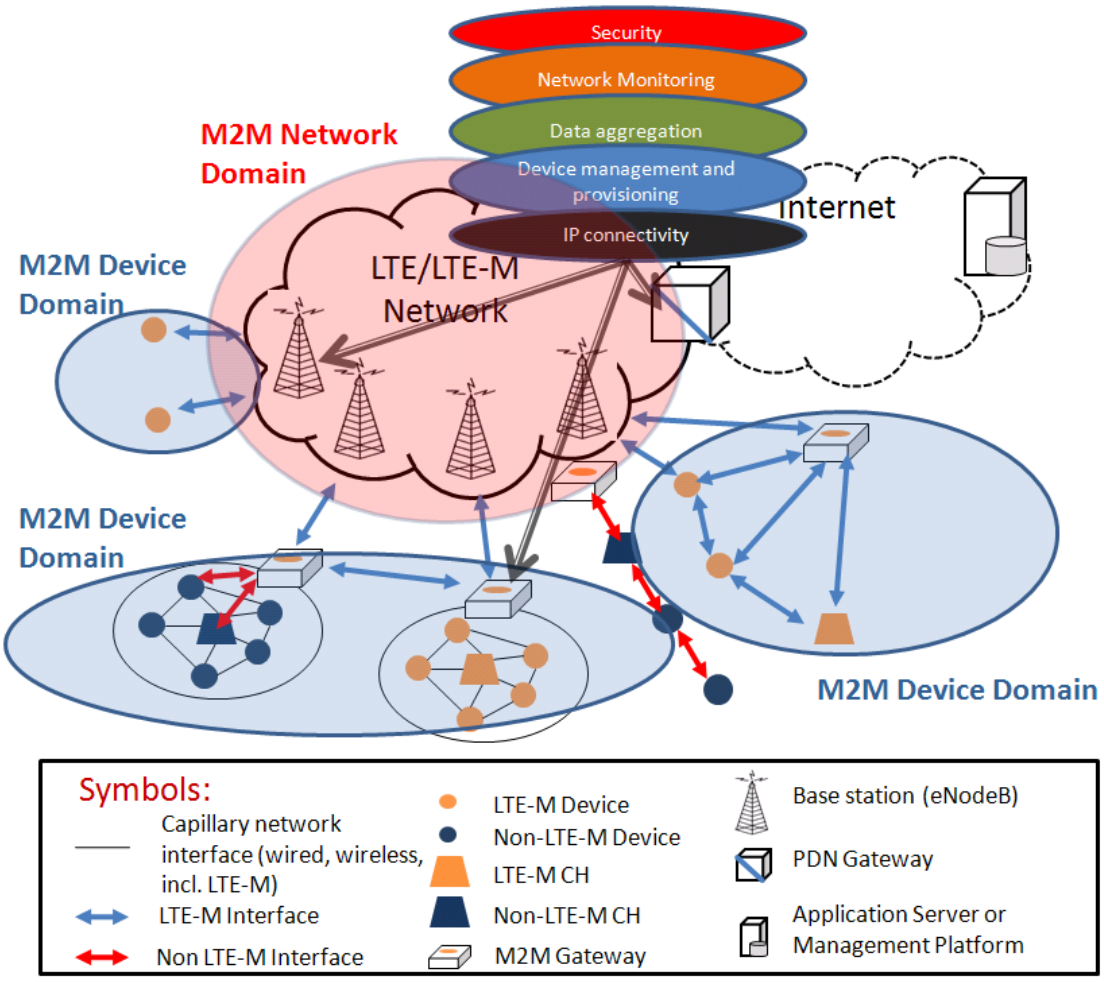

Figure 1.1: EXALTED system overview.

The core of the **M2M Network Domain** is actually the LTE/LTE-M system, involving all those elements that can be found in LTE (e.g. eNodeB, EPC (Evolved Packet Core)), but with the necessary enhancements in order to support M2M communications.

On the top of the underlying protocols and technologies, particular **M2M Application Servers** communicate with M2M Devices and Gateways that are involved in the same application.

Note that the applications may run on any functional element (i.e. on the M2M Gateway, the M2M devices or the cluster heads). Apart from the Application Servers, the architecture assumes other relevant servers for management and control, such as Device Management Server, which uses specifically designed protocol over the same network for communication with Devices and Gateways, and servers needed to fulfill the security requirements.

2. Security Basics

This section covers some fundamentals of security including: the various uses of cryptographic algorithms (and why they need a "key"); the differences between authentication and encryption and integrity; why we need a **unique** key to authenticate things and why the key needs to be stored safely. We also describe the need for session keys, key derivation and key distribution methods.

In the light of this general technical background, we raise (in Section 2.4) the various trade-offs that must be made when designing a security system, and the particular difficulty within EXALTED of reconciling different goals. We also discuss (in Section 2.5) the difficulties in reaching a consensus on security solutions in standards and getting the resulting solutions deployed.

We close the section with a "state of the art" review about how security is implemented in existing cellular networks (up to LTE), and we also comment on some of the practical lessons and experiences from deploying smart cards.

2.1 Security Services and Cryptographic Primitives

In general the most important security services are as follows:

- Authentication – verification of a claimed identity
- Confidentiality – ensure information is not available to unauthorized individuals, processes, entities etc.
- Integrity – ensure data has not been changed or destroyed in an unauthorized manner
- Non-repudiation – ensure that the author cannot later claim not to be an author of data that was sent over a network)
- Availability – assures that system assets are available each time when is needed to the authorized personnel
- Access control – granting access to data or performing certain action

To help deliver these services, cryptographic primitives are deployed. These include:

- Cipher – a method that provides confidentiality (via *encrypting* data)
- Message Authentication Code – a method that provides integrity of data (by appending a cryptographic check sum)
- Digital Signature – a method that provides non-repudiation of data
- Hash Function – a general term used for algorithms which transform strings of arbitrary length into strings of a fixed length

Robust cryptographic primitives are designed in such a way that their design does not need to be kept secret; this allows their properties to be publically reviewed and studied, and any flaws identified (or fixed). Only some parameter to the algorithm (called a "key") is kept secret. Each party (or pair or group of parties) wishing to use the algorithm chooses a **different** key (preferably at random), and other parties which do not know the key will then not be able to impersonate users, read confidential data, tamper with data etc.

2.1.1 Authentication and Keys

In most computer security contexts, user authentication is a basic building block and as such is the basis for most types of access control and for user responsibility. For network-based user authentication, a common secure way to proceed is to use 2 authentication factors: The

first one (what you have) is using something hard to duplicate, and is typically implemented using a cryptographic key embedded in a secure storage area such as a smart card to perform cryptographic computations. The smart card stores the key and also executes cryptographic primitives which make use of the key (as discussed above). The second factor relies on something the individual knows, such as a password.

In most cases, authentication is a prelude to transferring a message from (authenticated) person A to (authenticated) person B, and it then needs to be guaranteed that no other person can read (or change) the message even if it was intercepted. Confidentiality and integrity are the main issues.

2.1.2 Confidentiality

There are two ways to encrypt data: symmetrically or asymmetrically

1. Symmetric encryption - the same key is used for both encryption and decryption. (Symmetric ciphers include: RC4, DES, 3DES, AES)

2. Asymmetric encryption – a different key is used for encryption and decryption. (Asymmetric ciphers include: RSA and DSA)

Asymmetric cryptography is based upon the concept of private and public keys. Every party keep its private key secret and gives its public key to others. The public key is used with a function to transform plain text into cipher text. Anything encrypted with the public key can only be decrypted with the private key. Anything encrypted with the private key can only be decrypted with the public key. The keys are linked and generated at the same time. The safety of the encryption process is directly related to the reliability and privacy of the key storage and key distribution processes.

2.1.3 Integrity

Integrity checks aim at verifying that a message has not been tampered with, altered, or corrupted during its transmission from A to B. In terms of computer data integrity, message authentication is used to detect if even one single bit of a file or packet has been changed.

Message authentication primitives ensure that received messages are originated from the legitimate senders (data authenticity) and have not been modified en route (data integrity). The common approach is sending a message with an authentication tag that depends on the message and cannot be forged by an adversary. On the receiver side, the recipient checks the received tag and if it is correct, concludes (with high probability) that the message is really originated from the sender and has not been tampered with during transfer.

Similar to encryption, the authentication tag can be derived with symmetric or asymmetric primitives. In symmetric approaches, sender uses the shared key to drive the authentication tag on the message by a known function to the receiver who verifies the tag by its shared key. In asymmetric schemes, sender uses its private key to derive the authentication tag on the message and receiver verifies the tag by the sender's public key. Message Authentication Codes (MAC) and digital signatures are the main symmetric and asymmetric cryptographic primitives that provide message authentication. Digital signatures have the advantage that can also provide non-repudiation services which can be seen as undisputable form of message authentication.

2.2 Key Storage

2.2.1 Key storage basics

From the above discussion it should be clear that if an attacker ever recovers an authentication key (either symmetric or private), then the attacker can successfully impersonate the party being authenticated. This is referred to as "spoofing" or "cloning".

A party wishing to rely on an authentication key will therefore want assurance that the key has not been compromised or leaked. To provide such assurance, authentication keys should be stored in specially-secured hardware (referred to as a “secure element”), because such a hardware environment can be more easily secured and certified. In particular, the hardware can ensure non-exportability of private content (such as cryptographic keys). For instance, bank and credit cards, Pay-TV cards, transportation passes, employee ID cards, passports etc. all have the property that they need to store some form of persistent authentication credential, and it is very important that the credential is not spoofed or “cloned”.

Smart card technology has already become familiar to every person with GSM SIM cards, telephone cards and bankcards. The old magnetic stripe cards are being replaced with smart cards which have an integrated circuit merged into them, which makes them possible to be used for much complex operations. The IC card itself verifies the PIN code and the card can´t be copied; at least not so fast and cheap as the magnetic stripe card. The old systems had cryptographic systems in a secure place and the passcodes for the card were verified remotely from the card terminals. In smart card system, the card itself must be secured very carefully because the cryptographic keys are in the card itself and not in some safe deposit of the bank. One of the most important applications of the IC card is a general identification of the cardholder.

One of the most common ways of identification used for smart cards is by using Public Key Infrastructure (PKI). In that case the card stores a private key and digital certificate (a signed association between the public key and the corresponding card user) issued from the PKI provider. Client-side identification and authentication cards are the most secure way, especially for internet banking applications.

One of the first such projects is the smart card driver's license system, implemented in 1987 in Turkey. Since then the professional driver's licenses in Turkey are issued as smart cards and the driver is required to insert his driver's license into the digital tachograph which records speed violations for each driver and give a printed report. Also, in 2002 the Estonian government started to issue smart cards as primary identification for citizens to replace the usual passport. These cards are also widely used in internet banking, buying public transport tickets, authorization on various websites etc. Similar to this, since 2009, the entire population of Spain and Belgium is being provided with an electronic identity (eID) card that is used for certificate-based identification.

2.2.2 Physical Unclonable Function (PUF) Devices

A Physical Unclonable Function (PUF) is a function which is easy to evaluate but hard to predict. PUF is embodied in a physical structure. PUF devices must be easy to make but hard to duplicate. Basically hardware of PUF devices is based on “one way function”, making it possible to compute on every input, but hard to invert when given the image of a random input.

PUF devices are based on challenge-response authentication due to disadvantages of embodying a single cryptographic key. When a physical stimulus is applied to the device, it reacts in an unpredictable way due to the complex interaction of the stimulus with the physical microstructure of the device itself. The microstructure of the device depends on physical factors introduced during manufacture process and it is unpredictable. The applied stimulus is called the challenge, and the reaction of the PUF is called the response. A precise challenge together with its response forms a challenge-response pair. The PUF device identity is determined by the proprieties of a microstructure. The advantage here is that device is not revealed by the challenge-response procedure, so the PUF device is resistant to spoofing attacks.

The basic principle of challenge-response authentication is shown on Figure 3.1.

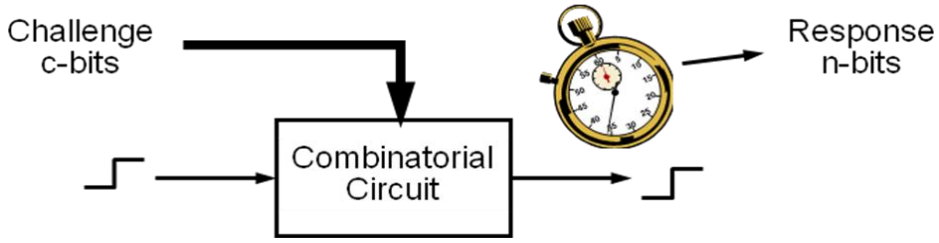

Figure 2-1: PUF device challenge-response authentication based on timing and delay information of silicon

The advantage when using PUF is that it requires a small hardware investment, due to the fact that PUF is installed in hardware based on proportion of the number of challenge and response bits.

Each PUF device has a unique and unpredictable way of mapping challenges to responses. The uniqueness is kept even in the situations where a similar device is produced in the same manufacturing process. It is impossible to produce two same PUF devices, because the exact control of manufacturing process is not possible, and according to already mentioned above, PUF function is based on „one way function“, thus making it impossible to compute unknown response based on random inputs from PUF device.

The combination of physical and mathematical unclonability is what truly makes PUF unclonable.

Different sources of physical randomness can be used in PUF. We must distinguish PUFs where physical randomness is explicitly introduced with PUFs where randomness is presented in a physical system itself.

Physical Unclonable Functions (PUFs) are innovative circuit primitives that extract secrets from physical characteristics of integrated circuits (ICs). PUF can be designed on the way that exploit inherent delay characteristics of wires and transistors that differ from chip to chip, so PUF can enable low-cost authentication of individual ICs and to generate volatile secret keys for cryptographic operations. As electronic devices become more and more the part of our daily life, with interconnection among them, people rely on integrated circuits (ICs) for performing security tasks and for handling sensitive information. For example, RFID is often used as a key card to control access to buildings, smart cards are used for financial transactions, mobile phones contain some confidential information such as personal e-mails etc... So, ICs are used for authentication purposes, authentication of devices and for protection of confidential information etc. The current best practice is to place secret key in non-volatile memory. But, unfortunately this option is difficult and expensive, as well as vulnerable to attacks.

PUFs are innovative primitives to derive secrets from complex physical characteristics of ICs rather than storing keys in digital memory. Another advantage of PUFs is that they do not require any specific manufacturing process or programming and testing steps.

The outputs when using PUF are likely to be slightly different on each evaluation, even on the same IC for the same challenge (due to noise). PUF can generate volatile secret keys used for cryptographic operation in two steps. First, the error correlation process, which consists of initialization and re-generation ensuring that PUF can produce the same output even when there are environmental changes such as voltage and temperature fluctuations. The second step is the generation process in which PUF outputs are converted into cryptographic keys.

The whole process is illustrated on Figure 2-2

Figure 2-2: Key generation – integration of PUFs with keyed crypto primitives

There are different types of PUF devices based on the material that they are built from (not only based on timing and delay information of silicon circuits – Silicon PUFs) such as optical PUF (it uses speckle patterns of optical medium for laser light), coating and acoustic PUF (measures the capacitance of a coating layer and the acoustic reflection) etc.

2.2.2.1 Relevance of PUF In the context of M2M communications

The context of M2M communications appears to be suitable to develop the PUF value proposition. M2M low end devices appear to present challenging security and privacy issues. Implementation of adequate security may involve embedding a secure element inside the M2M device, leading to solutions such as those described in section 5.3.1. The PUF Value proposition may be interesting in the context of low cost M2M devices for at least two reasons:

- These functions exploit the inherent variability of the hardware (wire delays and parasitic gate delays), and may be implemented with an order-of-magnitude reduction in gate counts compared with traditional cryptographic functions. This is particularly interesting when dealing with low cost M2M devices.
- Traditional secure elements involve the storage of cryptographic keys in non-volatile memory, with potential sensitivity to hardware probing and disassembly methods. PUFs on the other hand open the possibility to perform key derivation upon power up of the device; the cryptographic secrets defined are then stored only in volatile memory and cannot be accessed any more once the device power is turned off. This significantly increases the difficulty to compromise the device.

2.3 Provisioning

The topics of cryptographic key management and cryptographic key distribution are complex, involving cryptographic protocol and management considerations. As the entire operation is dependent upon the security of the keys, it is sometimes appropriate to invent a fairly complex mechanism to provision them. A number of exhaustive and proven key management systems are available for encryption of the keys. The two components required to encrypt data are an algorithm and a key. The algorithm is generally known, and the key is kept secret. The key is a very large number that should be impossible to guess, and therefore that makes comprehensive search impractical.

Keys should whenever possible be distributed by electronic means, enciphered under previously established higher-level keys (when no higher-level key exists and it is necessary to establish the key manually). A common way of doing this is to split the key into several parts (components) and delegate the parts to a number of key management personnel. The idea is that none of the key parts should contain enough information to reveal anything about the key itself, i.e. knowledge of keys should be restricted.

For link encryption, reasonable requirement is manual delivery of a key because each hop device must receive a key, and when the keys change, each must be updated since each link encryption device is going to be exchanging data only with its partner on the other end of the link. Also, packets are decrypted at each hop and thus, more points of vulnerability exist. However, for end-to-end encryption over a network, manual delivery is uncomfortable. In a distributed system, any given host or terminal may need to engage in exchanges with many other hosts and terminals over time. Thus, each device needs a number of keys supplied

dynamically. The problem is especially difficult in a wide-area distributed system. The scale of the problem depends on the number of communicating pairs that must be supported. If end-to-end encryption is done at a network or IP level, then a key is needed for each pair of hosts on the network that wish to communicate. If encryption is done at the application level, then a key is needed for every pair of users or processes that require communication. Thus, a network may have hundreds of hosts but thousands of users and processes. To reduce both communication and computation overhead of group key management schemes, there is a need to develop an efficient secure shared-key-based group-key management scheme in which the requirements in every phase of end-to-end key management will be studied, typical schemes of every category and their features and the critical techniques used in the end-to-end encryption key management scheme will be introduced and studied. The strength of any cryptographic system rests with the key distribution technique.

For end-to-end encryption, the encryption process is carried out at the two end systems. Encrypted data are transmitted unaltered across the network to the destination, which shares a key with the source to decrypt the data. Packet headers, addresses, and routing information are not encrypted, and therefore not protected. Also, it provides more flexibility to the user in choosing what gets encrypted and how. Higher granularity of functionality is available because each application or user can choose specific configurations. Each hop computer on the network does not need to have a key to decrypt each packet.

2.3.1 Session Key Distribution

The concept of a key hierarchy and the use of automated key distribution techniques greatly reduce the number of keys that must be manually managed and distributed. It also may be desirable to impose some control on the way in which automatically distributed keys are used.

To illustrate the value of separating keys by type, consider the risk that a master key is imported as a data-encrypting key into a device. Normally, the master key is physically secured within the cryptographic hardware of the key distribution centre and of the end systems. Session keys encrypted with this master key are available to application programs, as are the data encrypted with such session keys. However, if a master key is treated as a session key, it may be possible for an unauthorized application to obtain plaintext of session keys encrypted with that master key.

Central to the problem of session key distribution are two issues: confidentiality and freshness. To prevent compromise of session keys, essential identification and session-key information must be communicated in encrypted form. This requires the prior existence of secret or public keys that can be used for this target.

The second issue, freshness, is important because of the threat of message replays. Such replays, at worst, could allow an opponent to compromise a session key or successfully act as another party. At minimum, a successful replay can break operations by presenting parties with messages that appear authentic but are not.

Some examples of replay attacks are the following:

- **Simple replay** - when the opponent simply copies a message and replays it later.
- **Repetition that can be logged** - an opponent can replay a time stamped message within the valid time window.
- **Repetition that cannot be detected** - the original message did not arrive at its destination only the replay message arrives.
- **Backward replay without modification** - usage of symmetric encryption leveraging on the fact that the sender cannot easily recognize the difference between messages sent and messages received on the basis of content.

A two-level hierarchy of symmetric encryption keys can be used to provide confidentiality for communication in a distributed environment. In general, this strategy involves the use of a trusted key distribution centre (KDC) which serves to distribute session keys after authentication. Each party in the network shares a secret key, known as a master key, with the KDC. The KDC is responsible for generating short-lived keys, known as session keys, and for distributing those keys protected by the master keys. This approach is quite common. Each user must share a unique key with the key distribution centre for purposes of key distribution. The use of a key distribution centre is based on the use of a hierarchy of keys. Communication between end systems is encrypted using a temporary key, often referred to as a session key. Typically, the session key is used for the duration of a logical connection, such as a frame relay connection or transport connection, and then discarded. Each session key is obtained from the key distribution centre over the same networking facilities used for end-user communication. Accordingly, session keys are transmitted in encrypted form, using a master key that is shared by the key distribution centre and an end system or user.

For each end system or user, there is a unique master key that it shares with the key distribution centre. These master keys must be distributed in some fashion. If there are N entities that wish to communicate in pairs, then, as many as [N(N - 1)]/2 session keys are needed at any one time. However, only master keys are required, one for each entity. Thus, master keys can be distributed in some non-cryptographic way, such as physical delivery. The scenario assumes that each user shares a unique master key with the key distribution centre (KDC). It is not necessary to limit the key distribution function to a single KDC. Indeed, for very large networks, it may not be practical to do in such a way. As an alternative, a hierarchy of KDCs can be established. For example, there can be local KDCs, each responsible for a small domain of the overall internetwork, such as a single LAN or a single building. For communication among entities within the same local domain, the local KDC is responsible for key distribution. If two entities in different domains desire a shared key, then the corresponding local KDCs can communicate through a global KDC. In this case, any one of the three KDCs involved can actually select the key. The hierarchical concept can be extended to three or even more layers, depending on the size of the user population and the geographic scope of the internetwork. A hierarchical scheme minimizes the effort involved in master key distribution, because most master keys are those shared by a local KDC with its local entities. Furthermore, such a scheme limits the damage of a faulty or corrupted KDC to its local area only.

For connection-oriented protocols, one obvious choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session. If a logical connection has a very long lifetime, then it would be intelligent to change the session key periodically, perhaps every PDU (protocol data unit) sequence number cycle. For a connectionless protocol, such as a transaction-oriented protocol, there is no explicit connection initiation or termination. Thus, it is not obvious how often one needs to change the session key. The most secure approach is to use a new session key for each exchange. However, this negates one of the principal benefits of connectionless protocols, which is minimum overhead and delays for each transaction. A better strategy is to use a given session key for a certain fixed period only or for a certain number of transactions.

The key distribution centre has to be highly trusted, since it holds a copy of every master key. This requirement can be avoided if key distribution is decentralized, so avoiding a single point of failure problem. Although full decentralization is not practical for larger networks using symmetric encryption only (full decentralization effectively means no KDC at all) it may be useful within a local context. A decentralized approach requires that each end system is able to communicate in a secure manner with all potential partner end systems needed for session key distribution. Thus, there may need to be as many as [n(n-1)/2] master keys for a configuration with n end systems. Thus, although each node must maintain at most (n-1) master keys, as many session keys as required may be generated and used. Because the

messages transferred using the master key are short, cryptanalysis is difficult. As before, session keys are used for only a limited time, to protect against their compromise.

In GSM/UMTS/LTE networks the key distribution is managed by an Authentication Centre (AuC), which is part of the HLR/HSS. The AuC is security critical, and is hardened (typically using specialized hardware) to ensure the keys stored within it are kept safe. The AuC stores a persistent key (K or Ki) associated with a persistent subscriber identifier (IMSI) and uses this for authenticating the user, as well as generating and distributing session keys.

2.3.2 Man-in-the-Middle Attacks

Man-in-the-Middle Attacks (MITM) represent a kind of unauthorized access to data, and are a particular problem when considering electronic session key distribution.

A MITM attack intercepts communications between two systems by relaying messages between them. In this type of attack, the attacker makes an independent connection with both of the endpoint sides to relay messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

In general, the attacker attempts to insert himself in the middle of a communication with the purpose of intercepting a source's data and potentially modifying or discarding them before sending them out to the real destination. The attacker can than create new connections and kill existing connections, as well as view and replay anything that is private between the target endpoints.

MITM attack is also known as Fire brigade attack, Bucket-brigade attack, Session hijacking, TCP hijacking, Monkey-in-the-middle attack or TCP session hijacking.

Man-in-the-middle attacks take two forms:

1. In "passive" eavesdropping, the attacker simply listens to a set of transmissions to and from different hosts even though the attacker's computer isn't party to the transaction. Many relate this type of attack to a leak, in which sensitive information could be disclosed to a third party without the legitimate user's knowledge.
2. In "active" manipulation, the attacker builds on the capability of eavesdropping by taking this unauthorized receipt of a data stream and changing its contents to suit a certain purpose of the attacker-perhaps spoofing an IP address, changing a MAC address to emulate another host, or some other type of modification.

Passive eavesdropping attacks will work against unencrypted conversations, or conversations where a session key is passed in the clear, or else is encrypted to a master key that the attacker has already compromised.

However, passive attacks do not work in environments where a session key is negotiated via some form of public key method. To break the security in that context requires an active adversary, who is able to manipulate the conversation and pass "fake" public keys to the end-points. The MITM therefore has to be more powerful, and generally needs to be active at exactly the right time. Further, there is much more risk of the MITM being detected.

Authenticated Key Exchange (AKE) is one of the essential cryptographic primitives which allows a pair (or a group) of parties not only to authenticate each over an insecure network like the Internet but also to establish secure channels. Several techniques have been proposed for the distribution of public keys to achieve AKE and eliminate the residual risk of the active MITM. We discuss them here because they have a significant impact on provisioning processes.

Virtually all these proposals can be grouped into the following general schemes:

- Public announcement, meaning that the public key is just made public. One obvious drawback here is that the attacker (MITM say) may be able to forge such a public announcement, at least for long enough to execute the attack.
- Publicly available directory – maintenance and distribution of the public dynamic directory would have to be the responsibility of some trusted entity or organization.
- Public-key authority – maintenance and distribution of the public dynamic directory would have to be the responsibility of some trusted entity or organization with improvement that each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.
- Out-of-band distribution of the public-keys over a secure channel - The exchange takes place via another channel (for example, telephone or regular mail) or over a secure, already protected channel that requires the establishment of an additional secured channel between the two entities.
- Public-key certificates – certificates can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority. A certificate consists of a public key, an identifier of the key owner, and the whole block signed by a trusted third party. Typically, the third party is a certificate authority, such as a government agency or a financial institution that is trusted by the user community. A user can present his or her public key to the authority in a secure manner and obtain a certificate. The user can then publish the certificate. Anyone who needs this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature. A participant can also transfer its key information to another by transmitting its certificate. Other participants can verify that the certificate was created by the authority. The X.509 standard is a scheme that has become universally accepted for formatting public-key certificates. X.509 certificates are used in most network security applications, including IP security, transport layer security (TLS), and S/MIME. One drawback with certificate based schemes is that the parties have to agree to trust a common certification authority (CA), and to obtain certificates in advance from that authority. This usually involves an up-front cost, and prevents "ad hoc" security between parties that have not planned their communication in advance. The process of authenticating *to* the Certification Authority (to obtain a certificate) is also a weak-point in the system, and the MITM may try to impersonate an applicant to obtain a certificate. If the CA takes strong defences against impersonation, then the cost (and delay) in obtaining a certificate will increase, and just encourage participants to use a different CA. CAs may implement mechanisms to "revoke" (cancel) fake or mis-issued certificates, but then the participants must check up to date revocation lists or other central directories, which raises the problem of the "public dynamic directory" again. Often there is no effective revocation checking.

2.4 Security Trade-offs

Security in real-world systems always involves trade-offs between different goals, some of which are competing and potentially in direct conflict. The most relevant goals for EXALTED are: High overall security (resistance to attacks); Low costs of implementation (hardware, software, management); High usability (ease of user interface on install or configuration; low user effort involved in maintaining a device).

The conflicts are fairly straightforward to understand. High attack resistance generally involves a cost in secure hardware (see above) and a cost of provisioning and maintaining credentials and security associations (keys in the hardware; matching keys in a gateway or back-end/server system). There will also typically be costs of certification of the hardware (to ensure that attacks are thoroughly analysed and successfully resisted) and/or accreditation of the provisioning system and back-end server system. High usability requires considerable

effort by an Original Equipment Manufacturer (OEM), in terms of user interface design and testing, and the ability to support remote management protocols or secure local key agreement protocols. The OEM also incurs cost in provisioning and maintaining security credentials, since the usual alternative is forcing the user to provision security credentials (via manual entry of passwords, PINs, keys etc.). If the provisioning cost is minimized by the OEM, then security may be compromised as well (users are likely to provision weak PINs and re-use them across multiple devices).

One way of summarizing this is: “Secure, Usable, Cheap: Pick two”. However, the conflict represents a very hard choice for EXALTED, because *all* the goals are high priority for M2M:

- Can’t sacrifice usability (since many devices will be unattended, and hence can’t support complex user interfaces or user entry or PINs, passwords etc. “Usability” therefore means something much more dramatic than the usual sense of “user-friendliness”; poor usability translates into an inability to use the device at all.)
- Can’t sacrifice security (M2M devices often represent critical infrastructure with control systems for power supply, water supply; or they may be embedded in automobiles where security weaknesses are life-threatening)
- Can’t make expensive (too many devices; too much competition on price)

This situation calls for very careful risk assessment and risk management, as well as innovative thinking to achieve the best trade-offs between goals.

For example, the project members are convinced that the need for secure hardware will not go away: previous experience with mobile networks shows enormous fraud risks if network authentication keys are not protected by a SIM card (UICC) or equivalent secure element. These risks are compounded by new denial of service risks against critical infrastructure: such as persistent attacks on a network from unauthenticated (or weakly authenticated) devices, or attack against genuine devices by fake network infrastructure. Such denial of service attacks may not even be malicious, but simply result from poor device design (such as multiple devices all sending traffic to the network at exactly the same time).

However, we cannot expect users to acquire SIM cards from LTE-M operators and insert them into devices; nor can we expect LTE-M operators to carefully vet which devices the SIM card can be inserted into. The logistics and cost constraints call for the secure element to be embedded in a known device at manufacture, or shortly afterwards: this is the “eUICC”. Since users will still want the flexibility to select network provider, or change it over time (and this is likely to be a regulatory requirement in some vertical markets) there arises a further need to provision network authentication credentials remotely to the eUICC, and change them as needed. Nevertheless, LTE-M operators will want assurance that their credentials are being sent to a real eUICC, and cannot be extracted or compromised en route, which calls for the eUICC to ***already*** be provisioned with a unique initial security credential (probably a private key and certificate) so that it can be authenticated and verified as genuine. Care must also be taken that the remaining hardware of the device (that part which is not a secure element) boots into a secure state, and maintains that state during operation, so that it cannot easily be re-purposed by malicious parties or infectious malware. That way, authentication of the eUICC also implies authentication of an entire “well-behaved” device.

The need to optimize provisioning logistics while maintaining security demands this be done by a dedicated personalization centre, such as one operated by an existing SIM card supplier, before the eUICC is shipped to a (relatively insecure) OEM production line, which has been optimized to produce large numbers of (essentially identical, un-personalized) devices, but not optimized to manage security credentials. To keep the costs of the eUICC down, these must be manufactured and personalized in very large numbers (many billions of units), and heavily standardized so that they become a commodity component.

However, this is still not looking cheap, and we cannot seriously expect all devices to have such an eUICC or other SE any time soon. Mechanisms to allow devices to share a secure

element (for instance via capillary networks and gateway solutions) are needed, and we must also carefully consider mechanisms to set up strong credentials within such a local network, without forcing a high degree of user effort on installation.

We must also look at ways to ensure that a single eUICC can be used for multiple types of security credential: authentication to an M2M service provider and M2M application, as well as to an LTE-M network operator. This will avoid the cost of single devices needing several secure elements, and avoid the risk of application-layer credentials being stored insecurely.

None of this is an "obvious" solution to the dilemma, and the various components are likely to be controversial, since different organizations will have a different understanding and prioritization of the cost/security/usability elements. Accordingly, we must expect difficulties in standardizing the EXALTED approach.

2.5 Control Points

Securing a device also means securing an **authorization** infrastructure: device owners and users must be able not only to authenticate a device remotely, but control what it does. Suppliers or re-sellers of devices must be able to control who a device is handed over to i.e. to ensure that the correct purchaser of a device gets that control, rather than someone else.

Further, various service providers (such as LTE-M operators or M2M Service Providers) will not in general be the owners or users of M2M devices, and so will not necessarily trust that they are used safely, rather than in a careless or deliberately malicious way which can damage their service. (For instance the accidental or deliberate denial of service risks discussed above). This requires service providers to also have some control over how a device behaves.

Potentially whoever controls the security also controls the device; this can lead to severe conflict over who gets to set the security since other players don't want to cede this control point. Each stakeholder has particular interest in getting "their" credentials into the device, and ideally keeping them there. Each stakeholder may have different ideas about what credentials are needed (PIN, password, symmetric key, public/private key) as well as what algorithms and protocols should be used with those credentials. Each stakeholder will want to ensure interoperability with their existing systems and solutions to minimize their own costs. Stakeholders who have IPR around particular approaches will want to standardize them, either allowing a revenue stream from licence fees, or protecting themselves against costs from rival IPR. A high IPR load (e.g. from supporting multiple solutions) will of course increase the overall costs of the device, leading other stakeholders to be unhappy.

The initial provisioning process is likely to be seen as a particularly strong control point, as it includes the right to "sell" the security to other parties for their own use or for further provisioning. The embedded UICC is going to face this problem, and there is a particular challenge to design a security solution that doesn't create undue "lock-in" to the initial provisioner (so it won't be blocked in standards, or become a barrier to adoption), but still raises enough revenue to fund its own deployment.

This is unfortunately a recipe for standards conflict: either key ideas can't be standardized, or they are standardized in different ways by different groups. The mobile industry has seen previous such standards conflicts over SIM toolkit, JSR177, the High Speed Interface, NFC, and the Smart Card Web Server, as well as over WAP push, WAP client provisioning and OMA Device Management. Solutions may be standardized but not deployed because key stakeholders don't like them. Or there may be large numbers of options to try to keep everyone happy, but then there is no interoperability between different vendors' products: this is the status of ETSI M2M Release 1. Or all the security decisions may be delegated to the "user" (which won't work for unattended devices). Or there is no security at all, which is disturbingly common (consider WLAN infrastructure for instance).

2.6 Security and Key Usage in Cellular Networks

Cellular Mobile Networks (PLMNs) need a higher level of protection than traditional telecommunication networks. There are a few types of security that are especially relevant:

- Subscriber Authentication - ensures that no unauthorized users can access the network
- Radio Information Ciphering - information sent between the network and device cannot be eavesdropped.
- Mobile Equipment Identification – identifying stolen or other rogue devices

These security requirements have existed since the earliest days of GSM, and the overall architecture to deliver them is well understood.

KA (Key Agreement) is a mechanism which performs session key distribution in GSM and successor networks. AKA (Authentication and Key Agreement) is a challenge-response based mechanism that uses symmetric cryptography to combine the authentication step with the session key distribution step. In GSM, it runs on a special SIM application, hosted on a dedicated smart card. This provides associated robust and non-invasive key storage and symmetric key authentication using 128-bit secret keys. The smart card is also referred to conventionally as a "SIM" though this is strictly inaccurate, and has become out of date in successor networks.

The SIM card is ideal for computing "challenge-response" authentication, based on protected storage of an authentication key (see above), but cannot cipher data over the air interface, since the data processing and transmission capacity of a smart card are not adequate for real-time encryption of voice data. Instead, the SIM card computes a derived temporary key for transmission encryption and passes it to the mobile equipment itself.

2.6.1 UICC

AKA is used in 3G and LTE networks as well as GSM networks. However 3G and LTE provide enhanced security through the implementation of a Universal Integrated Circuit Card (UICC), which is more advanced than a basic SIM card, and supports an improved USIM application, which provides mutual authentication and allows derivation of longer key material for session keys.

Existing USIM-capable UICCs (designed for 3G) can be used for LTE access as well. LTE additionally incorporates stronger mutual authentication, stricter identity confidentiality and integrity protection of all signalling messages between UE and Mobility Management Entity (MME) and optional bearer data encryption. Mutual authentication is performed between the UE and MME using authentication vectors provided by the HSS. Compared to UMTS, in LTE the authentication protocol is enhanced in order to provide higher level of protection against attacks on other systems leaking into LTE. Also, the key hierarchy is more complex due to fact that cryptographic protection terminates in both the eNodeB and in the MME

Voice is another important issue, and it is handled as a "media" service on top of an LTE data bearer. The recommended approach is IMS (IP Multimedia Subsystem) in an LTE network. One of the evolutions of the UICC in LTE is the potential to have an IMS secure element within the UICC itself, called ISIM, which will uniquely identify the IMS end user.

It is important to note that the UICC is able to run multiple applications, not just the SIM, USIM and its variants (such as the SIM, ISIM). It is truly a general-purpose smart card and can support applications designed for bank and credit cards, Pay-TV cards, transportation passes, employee ID cards, passports and so on, as discussed above. For example, the UICC is an ideal location to store payment credentials, including NFC (Near Field Communication) credentials. Each day more and more applications are developed for use on end devices. Such applications can require higher security. Some of those can even require access to some secure functions in the UICC so that secure communication between the

terminal and the UICC is becoming more and more critical. 3GPP R8 brings some additional security with the implementation of secure channels between the terminal and the UICC.

The UICC is becoming even more critical for mobile network operators than in legacy technologies. UICC is the only high security component that identifies and authenticates the end user throughout this end of the IP network. This is an asset that is independent from the device and OS (Operating System) of the device, and that the end user can bring with him and transport easily from device to device.

2.6.2 EPS/LTE Overview

EPS stands for Enhanced Packet System, and is defined starting from 3GPP Rel 8 as a data ONLY network, meaning all the services are supposed to be delivered over IP (including voice and data). The intention of EPS/LTE is to provide the user with a true mobile internet experience, at least equivalent to the experience provided over fixed internet access. To achieve this, at least some main characteristics are:

- High data speed in terms of bit rate;
- Improved end-to-end latency
- Always-on IP connectivity; and
- Simple and efficient End-to-end QoS.

In a typical mobile network the radio access represents the bottleneck to achieve high bit rates. EPS defines a next generation Radio technology (LTE) that is an evolution of existing 3GPP technologies but uses an OFDM-based air interface (OFDMA in the downlink and SC-FDMA in the uplink) to deliver much higher data rates.

The end-to-end latency is dramatically improved with the introduction of LTE, the expected average latency from idle to active mode is approximately 140ms and the end-to-end Round trip Time delay for the 32byte ping is 20ms. The performance can be improved if a guaranteed Quality of Service bearer to be used.

LTE provides always-on IP connectivity where it is not necessary to wait for the session setup any time a subscriber is using an application from the device or from the PC. The always-on IP connectivity for users of LTE is enabled by establishing a default EPS bearer during network attachment. The attach procedure is not completed until the default EPS bearer is successfully established, therefore the attach and the bearer establishment (PDP Context) are not anymore decoupled as it was the case for the GPRS network.

The figure below represents the EPS architecture as deployed by a reference operator (Vodafone).

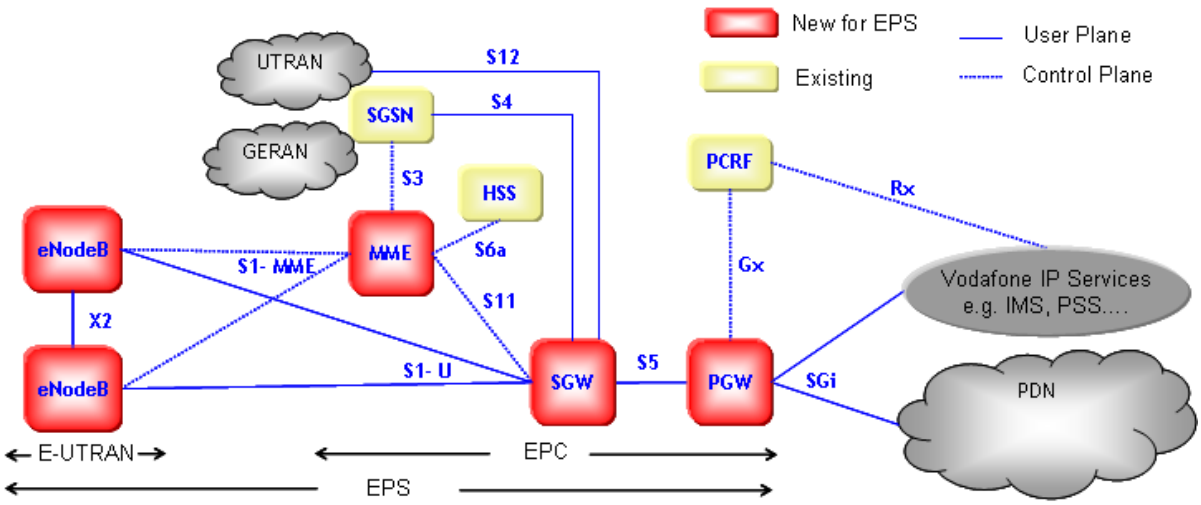

Figure 2-3 : EPS Architecture

An Evolved UTRAN (E-UTRAN) architecture is defined for the LTE access (see Figure 2-3). This architecture is introduced to enhance user performance in terms of latency, throughput and state transition (i.e. improved session setup time). This is achieved by flattening the architecture which moves the processing capability (e.g. formerly carried out by the RNC in 3G) to the eNodeB (eNB) (which also makes it easier to add capacity), and by minimizing the Transmission Time Interval. The result of such a "Flat Architecture" is that eNBs are interconnected via an X2 interface to support full mobility, minimizing the interruption during handover between cells.

Interconnection: Whilst adjacent eNBs are connected by X2 interfaces, each eNB is connected to the Evolved Packet Core (EPC) network by the S1 interface. On the User Plane the S1 interface terminates at the Serving Gateway (S-GW), on the Control Plane the S1 interface terminates at the Mobility Management Entity (MME).The eNBs are terminating points for Control and User Plane towards the UEs in the Evolved UTRA.

The Evolved Packet Core (EPC) is the evolution of the GPRS Core Network. The key aspects of the architecture are fewer nodes with a packet only flat architecture with standardised and open interfaces, separation of the user and control planes, improved latency, maintaining QoS and mobility support between multiple access systems (2G, 3G and non-3GPP systems).

In addition to other functions, the Core Network contains a Mobility Management Entity (MME), Serving Gateway (S-GW) and PDN-Gateway (P-GW), the Home Subscriber Subsystem (HSS) and the Policy and Charging Rule Function (PCRF).

Functionality (of entities in EPS): The eNB manages the new air interface that is based on OFDMA/SC-FDMA with a new set of Radio Resource Management protocols. The MME manages the control plane functionality whilst the S-GW acts as the user plane mobility anchor for 3GPP accesses and supports routing and forwarding of user plane. Both MME and S-GW may operate in a pool, defined by a set of Tracking Areas (TA). The PDN Gateway (P-GW) is the UE's IP point of attachment (i.e. UE IP address allocation) and is the gateway that terminates the SGi interface towards the PDN. The HSS stores EPS subscription data and supports the Authorisation and Authentication functions. The PCRF provides QoS and Charging policy information to the PDN Gateway (P-GW).

2.6.3 Overview of LTE security architecture

LTE incorporates multiple elements of security, as described in 3GPP TS 33.401 [1]:

- Symmetric key mutual authentication using 128-bit private keys and the EAP-AKA scheme. (EAP is the abbreviation for Extensible Authentication Protocol)
- Subscriber Identity Module (SIM) well-known methods of robust and non-invasive key storage.
- RAN encryption of bearer data using derived keys post-full-authentication to prevent eavesdropping.
- Integrity protection of all messages between UE and MME using covering codes to prevent alteration/snooping of sensitive subscriber ID (IMSI/IMEI) information in the signalling messages.
- Additional identity protection/confidentiality to prevent snooping/tracking of specific users (by use of aliases and minimal unencrypted use of real user/device identifiers).
- Internet Protocol Multimedia Subsystem (IMS) granular authentication/authorization per service.
- Use of Internet Protocol Security (IPsec) (mandatory in IPv6) for secure tunnelled mode between the IP communication endpoints.

The LTE security architecture can be subdivided into the following main areas:

- **Network access security**: This concerns protection of the vulnerable radio interface.
- **Backhaul security**: Architectural changes compared to UMTS mean that the backhaul link is more vulnerable to attack and needs dedicated protection. Notice for instance that the LTE network can potentially be accessed via any IP connection.
- **eNodeB security**: Architectural changes compared to UMTS combined with a trend towards smaller cells deployed in more vulnerable locations mean that more attention needs to be given to securing the eNodeB against physical and logical attack.
- **Core network security**: While the access network deserves special attention with respect to security, it is important that the security requirements concerning the core network are not neglected.

Each of these areas is discussed in the following subsections.

2.6.4 Network access security

2.6.4.1 Overview

The 3GPP EPS/LTE standard provides the following features to mitigate against network access security threats:

- Mutual authentication and key agreement
- Confidentiality and integrity protection of radio resource control (RRC) signalling
- Confidentiality and integrity protection of non-access stratum (NAS) signalling
- Confidentiality protection of user traffic
- Identity and location confidentiality

These features are based on corresponding UMTS security features, but there are some important differences:

- The authentication protocol is enhanced compared to UMTS to provide a better level of serving network authentication and a better degree of protection against attacks on other systems leaking into LTE.
- Cryptographic protection of user traffic and RRC signalling extends between and UE the eNodeB, whereas in UMTS this protection generally extends further back into the network to the RNC[1].
- Dedicated NAS level signalling protection is provided between the UE and MME. In UMTS all NAS level signalling is protected at an RRC level between the UE and RNC.
- The key hierarchy is more complex. This is due to the fact that cryptographic protection terminates in both the eNodeB and in the MME. The more complex key hierarchy also helps ensure that the effect of an eNodeB compromise has a limited impact on the rest of the network.
- The set of cryptographic algorithms used in LTE for providing encryption and integrity protection is different to the set of algorithms used in UMTS.

In the following sub-sections we describe each of the LTE network access security features in turn. (Further details can be found in 3GPP TS 33.401.)

2.6.4.2 Mutual authentication and key agreement

Mutual authentication is performed between the UE and MME using authentication vectors provided by the HSS. The authentication protocol is an enhanced version of the 3G AKA protocol used in UMTS.

[1] The exception to this is HSPA where the RNC function is collapsed into the Node B.

In 3G AKA, a shared secret key K is established between the USIM and Home Location Register / Home Subscriber System (HSS), and is stored in the UICC and AuC of the home network respectively. To authenticate subscribers, the HSS produces an authentication vector AV, based on K and sequence number. The AV vector consists of a random challenge RAND, authentication token AUTN, expected authentication result xRES, session key for integrity check IK, session key for encryption CK. The authentication vector is downloaded to a server. The server creates an authentication request which contains random challenge RAND, and authenticator token AUTN which is delivered to the client. Using the shared key and sequence number, the client checks AUTN with the USIM. If the verification is successful, the network authentication process is completed successfully. The client then generates an authentication response RES, using the shared secret key K and random challenge RAND, and it is delivered to the server. The server compares the authentication response RES with an expected response XRES and if those two matches then the user is successfully authenticated. IK and CK are then used for ensuring the secure communication between the client and server. Authentication synchronization parameter AUTS can be generated from the client and sent in authentication response message RES in order to re – synchronize the sequence numbers.

A high level view of the LTE authentication protocol is provided in the Figure 2-4 below, and explained in more detail in the following subsections.

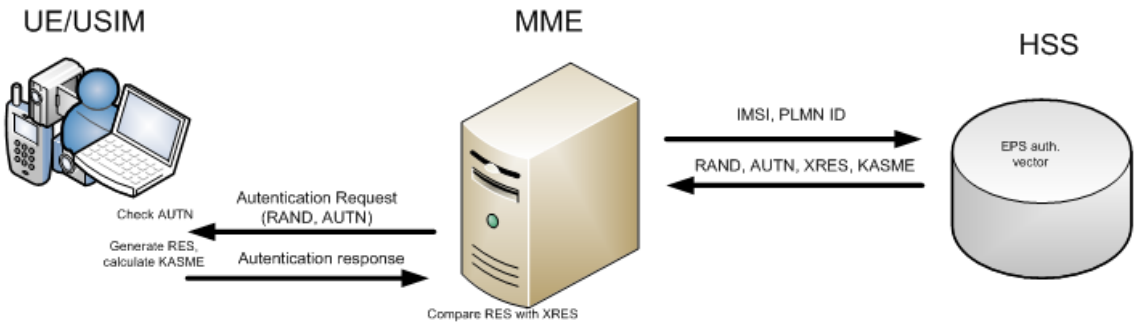

Figure 2-4: Mutual authentication procedure.

EPS AKA is the authentication and key agreement procedure that is used between UE and EPC Core Network. EPS AKA defines basic keying material for user plane (UP), RRC, and NAS ciphering keys as well as RRC and NAS integrity protection keys.

The purpose of the EPS authentication and key agreement (AKA) procedure is to provide mutual authentication between the user and the network and to agree on a key KASME.

EPS AKA procedure is always initiated and controlled by the network. However, in some cases it is possible that the UE reject the EPS authentication request sent by the network.

An EPS security context is established in the UE and the network when an EPS authentication is successfully performed. During a successful EPS authentication procedure, the CK and IK are computed by the USIM. CK and IK are then used by the ME as key material to compute a new key, KASME.

KASME is stored in the EPS security contexts of both the network and in the memory of the ME while attached to the network.

2.6.4.3 Authentication initiated by the network

If a NAS signalling connection already exists, the network can initiate an authentication procedure at any time. The network initiates the authentication procedure by sending an

AUTHENTICATION REQUEST message to the UE which contains the parameters necessary to calculate the authentication response.

2.6.4.4 Authentication response by the UE

The UE responds to an AUTHENTICATION REQUEST message. The UE calculates and determents the authentication challenge data and respond with an AUTHENTICATION RESPONSE message back to the network. If EPS authentication challenge is successful, the UE determines the PLMN identity to be used for the calculation of the new KASME. The new KASME is being calculated from the authentication data stored in EPS security context inside memory of the ME. The USIM determents the authentication response (RES) using the authentication challenge data received from the ME, and pass RES to the ME.

2.6.4.5 Authentication completion by the network

After receiving an AUTHENTICATION RESPONSE message, the network checks the correctness of RES. If the authentication procedure has been completed successfully then the related evolved Key Set Identifier (eKSI) is stored in the EPS security context of the network. Each time when the network initiates a new authentication procedure, it includes a different eKSI value in the AUTHENTICATION REQUEST message.

Although the authentication protocol is enhanced compared to UMTS (see [2]), no new security functionality is required on the USIM. This may mean that existing USIM-capable UICCs can be used for LTE access. However, some non-security functionality may need to be added to the USIM in order to allow or facilitate LTE access. Note that SIM-based access to LTE is not supported in the 3GPP standard.

At the network side, the LTE authentication protocol cannot make direct use of existing 3G authentication vectors. This means that existing HLR/HSS authentication functionality needs to be upgraded to support LTE. LTE authentication vectors are generated in much the same way as 3G authentication vectors with the following differences:

- Bit 0 of the Authentication Management Field (AMF) in AUTN must be set to 1. This is done to cryptographically separate LTE authentication vectors from authentication vectors used in other domains. It prevents security threats from those other domains leaking into LTE. E-UTRAN terminals are required to check that this so-called “separation bit” is set to 1 when running LTE authentication. If the bit is not set to 1 and authentication is being performed over E-UTRAN access, authentication is rejected by the terminal.
- The CK and IK session keys from the 3G authentication vector are not included in the LTE authentication vector. Instead, CK and IK are used to derive a session key called K_{ASME}, which is included in the LTE authentication vector. The Visited PLMN ID is used in the derivation of K_{ASME} to ensure that LTE authentication vectors can only be used within a specific PLMN. This provides an enhanced level of protection against network masquerade attacks compared to UMTS.

Authentication is performed at initial attach, and may also be performed in conjunction with subsequent NAS signalling events according to an authentication policy set in the MME. The MME should allow a suitable authentication policy to be configured by the operator.

The K_{ASME} session key derived during authentication is not used directly to protect user traffic and signalling information over the radio interface. Instead, K_{ASME} is used to derive lower level keys according to the key hierarchy shown in Figure 2-5.

3GPP E-UTRAN principles based on 3GPP specifications defines that the eNB keys are cryptographically separated from the EPC keys used for NAS protection thus making it impossible to use the eNB key to figure out an EPC key, and that they never leave a secure environment within the eNB, and user plane data ciphering/deciphering shall take place inside the secure environment where the related keys are stored. They also describe that AS

and NAS keys are derived from EPC/UE material generated by EPC/UE level AKA procedure (KASME) and they are identified by KSIASME.

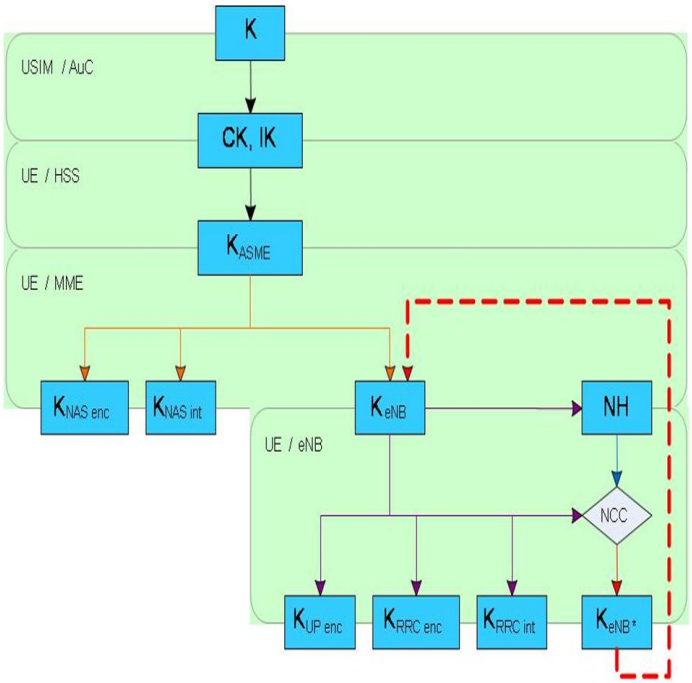

Figure 2-5: **The types of key assignment in LTE.**

The keys are used as follows:

- K_{NASenc} is used to encrypt NAS signalling between the UE and MME;
- K_{NASint} is used to integrity protect NAS signalling between the UE and MME;
- K_{UPenc} is used to encrypt user traffic between the UE and eNodeB;
- K_{RRCenc} is used to encrypt RRC signalling between the UE and eNodeB;
- K_{RRCint} is used to integrity protect RRC signalling between the UE and eNodeB.

Notice that the key derivation proceeds via intermediate keys (K_{eNB}, NH) to achieve the key separation of eNB from NAS keys as discussed above. Also notice that an NH ("next hop") intermediate key allows a fresh K_{eNB} to be derived when handing over between base stations; this ensures that even if the new eNodeB is compromised, then data protected by the previous eNodeB cannot be recovered.

2.6.4.6 Confidentiality and integrity protection of NAS signalling

NAS signalling security is established at initial attach using the NAS security mode command procedure. The procedure determines which cryptographic algorithms and keys should be used to encrypt and integrity protect subsequent NAS signalling messages. K_{NASenc} is used for encryption and K_{NASint} for integrity protection. NAS signalling protection is done at the NAS layer.

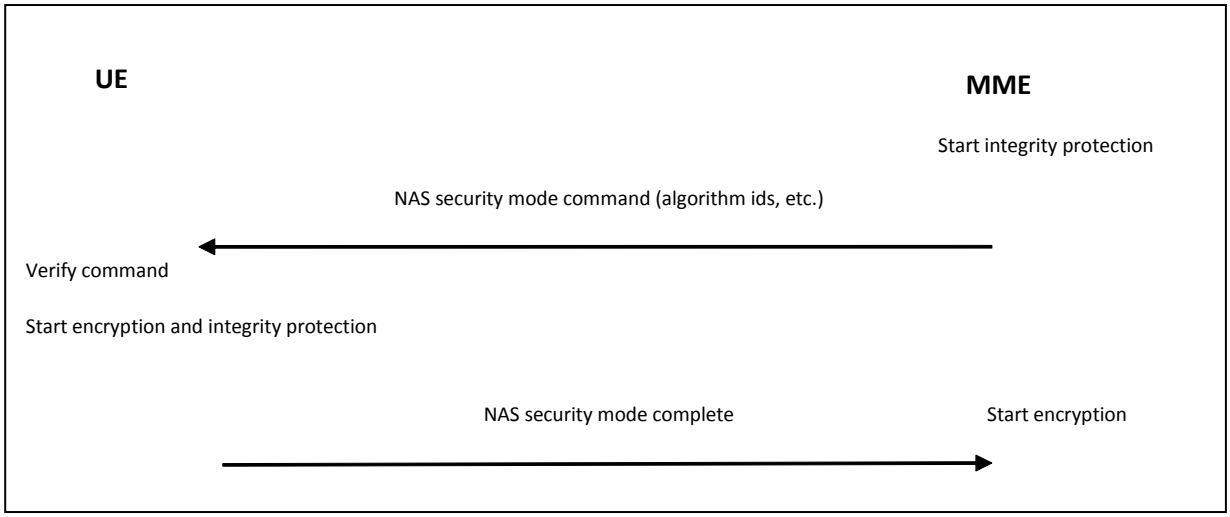

Figure 2-6: Protection of NAS signalling

EPS/LTE defines two sets of algorithms for NAS signalling protection:

Snow3G based algorithm set

- EEA1 EPS encryption algorithm 1
- EIA1 EPS integrity algorithm 1

AES based algorithm set

- EEA2 EPS encryption algorithm 2
- EIA2 EPS integrity algorithm 2

A null encryption mode called EEA0 is also defined.

According to the 3GPP standards, UEs and MMEs are required to support both sets of algorithms plus null encryption mode. Algorithm selection is controlled by the network based on preferences set in the MME. The MME should allow algorithm preferences to be configured by the operator.

The NAS security context established as a result of the NAS security mode command is stored in the UE and MME while the UE is attached to the MME. It may also be stored in the UE while the UE is detached or powered off and then re-used when the UE re-attaches. The NAS security mode command procedure can be used to refresh keys after authentication has been performed. It may also be used to change the encryption and integrity algorithms used for NAS signalling protection. Mechanisms are also defined in the standard to transfer the NAS security context between MME during relocation procedures, and to transfer and derive NAS security contexts when interworking with GSM and UMTS.

2.6.4.7 Confidentiality and integrity protection of RRC signalling

RRC signalling security is established at transition to active mode using the RRC security mode command procedure. The procedure determines which cryptographic algorithms and keys should be used to encrypt and integrity protect subsequent RRC signalling messages. K_{RRCenc} is used for encryption and K_{RRCint} for integrity protection. RRC signalling protection is done at the PDCP layer.

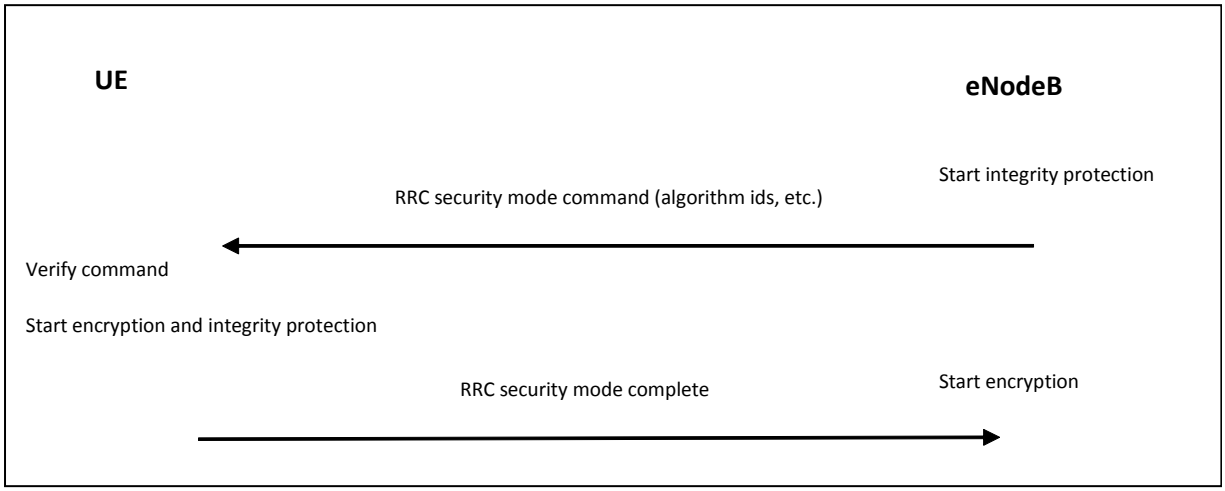

Figure 2-7: Protection of RRC signalling

The same sets of algorithms as for NAS signalling security may be used for RRC signalling security. According to the 3GPP standards, UEs and eNodeBs are required to support both sets of algorithms plus null encryption mode. Algorithm selection is controlled by the network based on preferences set in the eNodeB. The eNodeB should allow algorithm preferences to be configured by the operator.

The RRC security context established as a result of the RRC security mode command is stored in the UE and eNodeB while the UE is in active mode. The RRC security mode command procedure can be used to refresh keys after authentication has been performed. It may also be used to change the encryption and integrity algorithms used for RRC signalling protection. Mechanisms are also defined in the standard to transfer the RRC security context between eNodeB during handover.

2.6.4.8 Confidentiality protection of user traffic

LTE user traffic can be confidentiality protected, but not integrity protected, over the radio interface. User traffic encryption is established at transition to active mode using the RRC security mode command procedure. The procedure determines which cryptographic algorithms and keys should be used to encrypt user traffic. K_{UPenc} is used for encryption and the encryption is done at the PDCP layer.

The same encryption algorithms as for NAS and RRC signalling security may be used for user traffic encryption. According to the 3GPP standards, UEs and eNodeBs are required to support both encryption algorithms plus null encryption mode. Algorithm selection is controlled by the network based on preferences set in the eNodeB. The eNodeB should allow algorithm preferences to be configured by the operator.

The user traffic security context established as a result of the RRC security mode command is stored in the UE and eNode while the UE is in active mode. The RRC security mode command procedure can be used to refresh keys after authentication has been performed. It may also be used to change the encryption algorithm used for user traffic protection. Mechanisms are also defined in the standard to transfer the user traffic security context between eNodeB during handover.

2.6.4.9 Identity and location confidentiality

Identity and location confidentiality over the radio interface in LTE are achieved in a similar way to GSM and UMTS. In particular, temporary identifiers called GUTIs are generally used instead of the permanent IMSI when the user needs to be identified on the radio interface. The procedure works as follows:

- On initial attach to a new PLMN, the IMSI is used for identification.
- Immediately after encryption is established, the network allocates a GUTI and sends it to the UE over the encrypted channel.
- For subsequent communications, the GUTI is used by the UE and network to identify the user.
- The GUTI may be reallocated at regular intervals to avoid a user being tracked based on its GUTI. During the GUTI re-allocation procedure, the user identifies itself using the old GUTI and is then immediately assigned a new GUTI which is sent to the UE over the encrypted channel.

The MME should allow a suitable GUTI reallocation policy to be configured by the operator.

As in the case of GSM and UMTS, this mechanism provides good protection against passive eavesdropping, but it does not protect against active "IMSI catching" attacks using equipment which can masquerade as a false base station.

2.6.5 Backhaul security

There are several changes to the LTE system architecture and the transport network in particular, which mean that the backhaul link is generally more vulnerable to attack than in UMTS. These changes can be summarized as follows:

- There is no RNC in LTE so user traffic encryption extends between the UE and the eNodeB, whereas in UMTS the radio interface protection generally extends further back into the network to the RNC[2]. This means that user traffic is automatically protected on the backhaul link.
- The eNodeB has IP reachability into the operator core network, whereas in UMTS the Node B only has reachability towards the parenting RNC. This means that masquerading as an eNodeB on the backhaul network could have a more damaging effect.
- The backhaul transmission network will use IP/Ethernet which is generally more vulnerable to eavesdropping and spoofing attacks than the ATM/E1 transmission methods that are generally used in UMTS.
- Small cell LTE base stations may be quite common and may be deployed in vulnerable locations with a low level of physical protection. This will make the backhaul link towards such sites much more vulnerable to physical attack compared to traditional macro cell base station sites.

These changes mean that user traffic is more vulnerable to eavesdropping attacks on the backhaul link. The changes also mean that there is a greater risk of denial of service and penetration attacks against the core network. To mitigate against these threats, the 3GPP LTE standard specifies the use of IPsec. This can protect user plane, control plane and management plane on the backhaul network. IPsec can be used to protect both communications between the eNodeB and the operator core network over the S1 interface, and direct eNodeB to eNodeB communications on the X2 interface. At the core network side, there are several options for where the IPsec backhaul link protection should terminate.

There are several challenges associated with the deployment of IPsec on the backhaul network:

- Deployment of security gateways: The security gateways required to terminate the IPsec connections are new network elements that are not part of the existing operator core network or backhaul transmission network. The impact of deploying these gateways needs to be considered as part of the overall LTE end-to-end design.

[2] The exception to this is HSPA where the RNC function is collapsed into the Node B.

- Deployment of key management infrastructure: IPsec uses security keys to set up encrypted and authenticated connections. The 3GPP EPS standard specifies a key management approach based on public key certificates. This approach requires every eNodeB and security gateway to be issued with a public key certificate.
- Initial configuration of security parameters in the eNodeB: Depending on the details of the key management solution, various security parameters will need to be provided to the eNodeB as part of its initial configuration. The method for configuring this security sensitive information needs to be fully specified.
- Impact of IPsec on bandwidth efficiency: IPsec involves adding a security header to every protected IP packet. This will have an impact on bandwidth efficiency especially for traffic flows with small packet sizes. Measurements taken using NSN's backhaul IPsec solution for I-HSPA show a bandwidth increase of 5% for a constant traffic flow of IP packets with 1000 byte UDP payload, to an increase of 15% for a 400 byte UDP payload.
- Impact of IPsec on latency: IPsec will involve additional processing in the eNodeB which will add delay. It also requires a security gateway element to be added in the traffic path which will also add delay. Measurements taken using NSN's backhaul IPsec solution for I-HSPA in a test lab environment using a Checkpoint security gateway show an increase in round trip delay between an I-HSPA Node B and the core network of 0.4ms when IPsec is applied.
- Impact of IPsec on the monitoring of network interfaces: Clearly, if IPsec is used to encrypt traffic on a network interface, this will have an impact on the ability to monitor that interface for operational reasons. However, terminating IPsec in a dedicated security gateway ensures that monitoring can still be performed at the core network side of the gateway.
- Impact of IPsec on backhaul compression: It may be desirable to apply compression on the backhaul link. If IPsec is used on the backhaul link then the IPsec end point at the base station site should be integrated into the eNodeB itself. However, this would mean that any backhaul compression function must also be integrated into the eNodeB and cannot be deployed in a standalone box at the base station site unless a solution is developed to allow the compression function to remove and the re-apply IPsec protection without compromising overall system security.

In principle, the decision about whether to deploy IPsec on a particular backhaul connection could be taken on a case-by-case basis. However, this approach is labour intensive, complex and error prone, so it would be advantageous to enable IPsec for all backhaul connections providing that the overall cost and impact of deploying IPsec in this way is not too high.

2.6.6 eNodeB security

The changes to backhaul security described in section 2.6.5 above also mean that the eNodeB is generally more vulnerable to attack than in UMTS.

These changes mean that attention needs to be paid to protecting the eNodeB against physical and logical attack. This will include ensuring that radio interface security and backhaul link security are terminated inside the same secure environment within the eNodeB equipment. It will also ensure that the eNodeB operating system has been suitably hardened against attack, and that the eNodeB does not expose any weakly protected local interfaces which could be exploited by an attacker.

2.6.7 Core network security

The access network in LTE deserves special attention with respect to security due to the nature of the radio interface, the placement of eNodeB in untrusted locations, and the untrusted nature of the backhaul link. However, the core network should not be neglected as

far as security is concerned since this is where traffic is concentrated, so a successful attack on the core network is potentially much more devastating than an attack on the access network.

In general, the LTE core network should be given a similar or better level of protection as the existing UMTS PS core network. This should include suitably hardening network elements, and providing adequate protection on the transmission links between those elements.

While it may not be cost effective to protect all LTE core network interfaces, some particularly sensitive interfaces may need to be encrypted and authenticated using IPsec or TLS. In addition, interfaces with roaming partners, interconnect partners and other third parties may also need to be encrypted and authenticated. The 3GPP standard specifies profiles of IPsec and TLS for protecting core network interfaces that need protection. These profiles should be used where possible.

2.6.8 AuC vs. EIR

It is also worth mentioning the difference between AuC (Authentication Centre) and EIR (Equipment Identity Register). Both have existed since GSM networks. Both are used in the process of identifying the mobile station, although they involve rather different kinds of identification. While the AuC is used for the network access authentication process, based on the subscriber identity, the EIR is used for equipment identification, for instance it can be used to bar known stolen (or other rogue) equipment.

The main difference between AuC and EIR is not just which type of identifier parameter is used, but the level of authentication taking place. While the AuC uses IMSI, the EIR uses IMEI. It is known that IMEI can easily be spoofed, as it is not authenticated using any key and it is not well protected on the terminal side from re-write. By contrast, the IMSI is very hard to spoof, as it is authenticated using the corresponding K, and the K and IMSI are protected in secure hardware (on the UICC, and also typically inAuC). It is possible that if a UICC is embedded in a device, then authentication by a corresponding K will solve this problem, and make equipment identifiers hard to spoof. This would solve a number of outstanding problems in GSM, 3G and LTE security related to device theft and malfunctioning (or malicious devices).

2.7 Lessons from using Smart Cards

It is worth noting that while smart cards significantly reduce the risk of cloning or spoofing, they are not a fool proof solution. When using smart cards, different kind of problems may emerge. Frauds can still happen because the smart card, while itself secure, is embedded in an insecure card-reader environment. For example, it is possible that the account holder's computer is hosting to malware, so the security model may be broken. Malware can override and modify a transaction, without the knowledge of the user. Certain banks combine a smart card with an *unconnected* (stand-alone) card reader to avoid this problem.

A well-known example of fraud in the world of EMV (Europay, MasterCard and VISA) payment cards concerns bank ATMs. Sometimes a fake card reader, or skimmer, is placed over the real card-entry slot and a pinhole camera is used to record customers entering their personal identification numbers. The problem here is a combination of an insecure reader environment, and the interaction with legacy uses of the card magnetic stripe. It is sufficient for instance for the attacker to copy the magnetic stripe, and the PIN, and then use this information to make a “fake” card which is accepted in older ATMs (ones which do not yet support the EMV standard).

Other difficulties can arise because the smart card itself supports poor cryptographic algorithms. This typically arises where the crypto has been designed in a proprietary way, and not thoroughly reviewed by cryptographic experts. Or there may have been pressure to

economize on the cryptography to save money. These difficulties have been faced in multiple industries and sectors.

For example, the MIFARE Classic card used in physical access control systems (PACS) and contactless payment systems (including toll way and public transportation systems) has also been a target for attackers. A group of researchers was able to clone a MIFARE Classic card in less than two minutes. The impact of the MIFARE hack for those reliant on payment systems (and their consumers) is an increased cost to cover fraud. The cloning of the card does not require possession, only proximity. Basically a fraudster just by walking around a parking garage is able to clone toll way cards that are mounted in everyone's windshield. The only real solution is to replace the cards with better cryptography, and there are now a number of different MIFARE cards e.g. using Triple DES or AES. Since the introduction of biometric passports several attacks have been presented and demonstrated. For instance, it's possible to determine which country a passport chip is from without knowing the key required for reading it (by fingerprinting error messages of passport chips from different countries. Also, in 2005 it was shown that the document numbers of Dutch passports were predictable, allowing an attacker to guess / crack the key required for reading the chip. In 2006 it was demonstrated that it is trivial to copy passport data from a passport chip into a standard ISO/IEC 14443 smartcard using a standard contactless card interface and a simple file transfer tool.

Further on, the risk of SIM cloning is still a legacy problem for many operators. SIM cloning attacks occurred on SIM cards using the flawed COMP 128-1 "example" algorithm. The process of SIM card cloning is pretty simple if the card uses that algorithm, but it generally requires the attacker to have physical possession of the card. The attacker sends several thousand targeted challenges (RAND values) to the SIM card in order to extract IMSI and Ki (the 128 bit key) and then put the IMSI and Ki into card software and copy it to a new SIM card. When SIM cloning is completed, the cloned SIM card's identifying information is transferred onto a separate, secondary SIM card. The secondary card can then be used in a different phone while having all calls and associated charges attributed to the original SIM card. New security measures such as embedding security operations make obtaining a SIM clone more difficult. Now, if a SIM card detects that cloning has been engaged, the SIM card can render itself inoperable. However, the most effective solution is just not to use COMP 128-1; more recent standard algorithms like MILENAGE are much more secure.

It is clear that successive industries and sectors have had to learn their lesson on hard way, by deploying smart card solutions with cryptographic weaknesses that could have been (or were) noticed in advance by professional cryptographers. Clearly, there is a big risk that something similar will happen in the M2M space. We need to keep this in mind for the future.

One of the enablers for smart cards implementation was mature standardisation of physical form factor and the physical/logical interfaces. However, there is often still a lot of secrecy about exactly what is implemented on a smart card, and how.

The possible attackers can be divided into the following categories:

- **Class I (clever outsiders) -** This type of attackers is often intelligent but their knowledge of the system may be insufficient.
- **Class II (knowledgeable insiders) -** They have some specialized technical education and experience.
- **Class III (funded organizations) -** They may assembly teams of skilled specialists and they also have great funding resources. They are able to perform some in-depth analysis of the system, design powerful attacks.

Similar to the attacker categories, possible attacks can be divided in the following manner:

- **Attacks by the Terminal Against the Cardholder or Data Owner –** known as the terminal trusted problem. The cardholder must somehow trust the terminal that the terminal will do what it is asked for from the cardholder. This problem is often manifested when using old magnetic cards, in which case the terminal can copy the information from the card.
- **Attacks by the Cardholder Against the Terminal -** This type of attack is performed with fake or modified cards running some rogue software. The goal is to break the protocol between the card and the terminal.
- **Attacks by the Cardholder Against the Data Owner -** With eID cards, this type of attacks are relevant only if the card has been stolen. The data owner should be the only one who knows the PIN code for the data (secret key) so the new cardholder tries to get or modify the data some other way. These following techniques have already used with great success against some Pay-TV- and public telephone cards.
- **Attacks by the Issuer Against the Cardholder -** These types of attacks are typically some violations against the privacy of the cardholder.
- **Attacks by the Manufacturer Against the Data Owner -** When the data owner uses the card, how can he/she be sure what programs there are really running in the card? For example, when verifying a PIN code fails, how can we ensure that the card has actually tried to verify it? This question must be addressed properly through standards and specification.

The following technology and principles of storing keys in a hardware element are widely used (throughout the smart card world):

- Services which are provided by hardware should be basic key storage for chosen algorithms and crypto acceleration (symmetric and asymmetric).
- Keys should be kept in non-exportable parts of the memory.
- A secure *environment* (a place for key storage and cryptographic operations outside a dedicated piece of hardware) is harder to establish than a secure *element*. In the secure element case, just the physical secure element needs to be hardened and certified (similar to NFC concept), rather than an entire device.
- Depending on the use case, key lifecycle parameters should be defined – for example, can the key for some application be escrowed when provisioned to the element or should the key be created on the element and at what moment?

However keys are distributed, the important issue is that keys **must** be stored securely to maintain communications security. There are various techniques in use to do so. Likely the most common is that an encryption application manages keys for the user and depends on an access password (or PIN) to control use of the key.

3. Security Requirements

This section intends to provide an overview of all EXALTED security requirements, possibly originating from different sources:

- Use cases addressed by EXALTED.
- Requirements already identified by standard bodies.

3.1 Security requirements derived from EXALTED use cases

Some of the security requirements derived from EXALTED use cases are addressed in [25]. In this Section those requirements are redefined and presented in a form common with other requirements. The three use cases relevant throughout the EXALTED project are namely Intelligent Transport System (ITS), Smart Metering and Monitoring (SMM) and e-Health.

Subject	Security Requirement
Common security solution	R1. All types of end devices and secure elements should be treated the same way regarding security and provisioning, through consistent global processes and a common device management solution.
Data protection	R2. Security solution must ensure data protection (e.g. meter reading information in SMM, health information in e-Health) and protection of the system itself (hacker attack, quality of service). Prevention of theft and misusing of credentials and subscription information must be assured.
Subscription management	R3. Parameters related to subscription management, security credentials, cryptographic contexts, algorithms and methods, should be manageable remotely in a secure way.
Secure communication channels	R4. Data transfer in all use cases (ITS, SMM, e-Health) should be done via secured communication channels.
Device identification	R5. Each M2M device (UE) should be uniquely identified by the Application Server (used in particular use case) prior to any further actions.
Device authentication	R6. M2M devices (UE) should be authenticated by the Application Server.
Access control	R7. User access to application data must be controlled.
Automation	R8. Common security management solution needs to automate the provisioning and identity management of UE.

3.2 Requirements identified by standard bodies

A number of security requirements identified through the definition of exalted use cases have already been addressed by standard bodies. Here is a short list of security requirements addressed by 3GPP – see the summary in Section 6.1.1 for the references:

3.2.1 Known LTE-M security requirements, already identified in 3GPP

Subject	Security Requirement
General	R9. MTC optimizations shall not degrade security compared to non-MTC communications
MTC device triggering	R10. It may not be possible to totally prevent an MTC Device from receiving a trigger indication from a fake network. MTC trigger could be protected so that the impact of fake MTC triggers to the battery lifetime of the MTC device would be minimized.
Group-based optimization	R11. A MTC Group is a group of MTC devices that can be in the same area and/or have the same MTC Features attributed and/or belong to the same MTC user. MTC Group shall be identified uniquely across 3GPP networks
Secure connection	R12. The MTC Feature Secure Connection is intended for use with MTC Devices that require a secure connection between the MTC Device and MTC Server. Any 3GPP defined key management mechanisms for secure connection between the MTC Device and the MTC Server shall rely upon the UICC secure device to setup security
Security of Small Data Transmission	R13. The system shall support transmissions of small amounts of data with minimal network impact (e.g. signalling overhead, network resources, delay for reallocation). Also when the MTC Device is detached and no security context between the MTC Device and the core network is available.
Reject message	R14. If the Reject message is sent without integrity protection, any false base station could cause a denial of service attack to the MTC devices and the network. A security mechanism is needed to prevent the DoS attack.
MTC Monitoring	R15. It is required for the network to provide a location management mechanism for MTC Devices that should not move from an authorized location, or should only move in an authorized area to detect if the device has been moved to an unauthorized location.
Time controlled	R16. This is intended for use with MTC Applications that can tolerate to send or receive data only during defined time intervals and avoid unnecessary signaling outside these defined time intervals. Time interval and communication window shall be integrity-protected when sent to MTC device.
Congestion Control	R17. The aim of these solutions is when the network finds that the UE is a MTC device that will cause congestion or the UE is a low priority MTC device, it will reject the connection request. So the UE can use e.g. a low priority indicator. The low priority indicator shall be integrity-protected according to the rules in existing technical specifications.

ETSI is also addressing a number of security requirements defined below:

3.2.2 Known security requirements, already identified by ETSI

Subject	Security Requirement
Authentication	R18. Support of mutual authentication of the M2M core, M2M device and M2M gateway (single authentication of M2M device and M2M gateway) by the M2M core also supported)
Data confidentiality, integrity and privacy	R19. The system shall be able to protect the confidentiality of the data transferred, insure its integrity and protect the privacy of the data transmission.
Multiple actors	R20. The system should be able to maintain the security of end to end service even when several independent actors are involved in the delivery of the service
Device/gateway integrity check	R21. The system shall provide a mechanism to check the integrity of devices/gateways. The check may be requested by the system or initiated autonomously by the devices/gateways. R22. M2M devices requiring device integrity validation shall provide a trusted execution environment enabling trustworthy storage of sensitive data and execution of sensitive code.
Security credential and software upgrade	R23. When permitted by the security policy, M2M System shall be able to remotely provide the following features, at the Application level: • Secure updates of application security software and firmware of the M2M Device/Gateway. • Secure updates of application security context (security keys and algorithms) of the M2M Device/Gateway. This functionality should be provided by a tamper resistant environment e.g. TrE or security element in M2M Devices/Gateways supporting this functionality.

3.3 New requirements not covered by standards

A sensible target for an LTE-M device is to undercut the price of GSM, for which very low end radios and handsets are currently around 10-15 Euro; hence a target of 4-5 Euro for the whole device seems feasible (capillary-only devices would be cheaper still). Again, the costs of purchasing a conventional personalized UICC (~1 Euro) and distributing it to the customer (~10-20 Euro) look disproportionate.

We therefore can state some additional requirements:

Low-end devices	R24. The cost per device of provisioning LTE-M devices for network and service-level access and for maintaining the necessary security associations shall be a small fraction of the cost of the device itself.
Security Element	R25. The security element used to hold network access credentials for an individual LTE-M device or a group of LTE-M devices shall have an overall production and distribution cost which is a small fraction of the cost of the devices within that group.

SE re-use	R26. A single security element shall be re-usable for a variety of credentials, including service provider as well as network access credentials. Mechanisms shall be available to securely load additional credentials (and delete redundant credentials) during the lifetime of the device.

Some other security requirements, not related to EXALTED use cases, are addressed in [25], and hereby also summarized.

LTE-M architecture requirements	R27. LTE-M system (Network, Application and Device domain) should provide the communication which is secured, with guaranteed quality of service and optimized routing. R28. All architecture elements that are introduced in LTE-M architecture (gateways, relays and cluster heads) must properly handle secure communication and data exchange (e.g. by “relaying” or “proxying” information, authentication, identity management).
Diagnostic monitoring	R29. Analytic tools should offer the possibility to monitor diagnostics (in-depth real-time and historic data, i.e. event logs).
Management systems	R30. Service provisioning, management and billing should be done using proper management applications and systems.
Upper layer security	R31. There should be flexible IP layer and/or Application layer security mechanisms, allowing “any device to any application” and “any device to any other device” secured communication.

4. New Security Challenges

4.1 Low Energy/ Low Overhead

Some LTE-M devices are likely to be highly constrained in terms of energy consumption, even more so than typical mobile devices, as they may be unattended for long periods. This means they must either by powered by long-life batteries, or harvest and store energy from the local environment. Accordingly, there is a particular focus on “low overhead” security, where the “overhead” of energy consumption for security purposes is minimized. It turns out that minimizing energy consumption involves minimizing other sorts of overhead.

The impact of the security upon energy consumption of the device may be investigated by considering three factors:

- The power consumption linked to the execution of cryptographic algorithms inside the device.
- The power consumption linked to the need to data overhead associated to security
- The power consumption linked to factors related to infrastructure choices, e.g.: periods between re-authentications, number of simultaneous layers of security.

4.1.1 Power consumption linked to security related computations

The security of almost all cryptographic systems depends on the randomness, unpredictability and secrecy of the key. Many cryptographic protocols require random generator also for purposes other than the key meaning that there is a need for an internal source of random. However, real randomness is difficult to find and can be externally manipulated by the adversary.

Pseudo-randomness may require a large non-volatile memory and lots of computation.

A pseudo-random number generator can be built from block ciphers, hash functions, or linear feedback shift register.

The cryptographic primitive used to generate pseudo-random number could be the same as the cryptographic primitive which is used to protect the confidentiality and/or authenticity/integrity in order to minimize the cost overload.

ISO/IEC 29192 is a multi-part International Standard that specifies lightweight cryptography for the purposes of data confidentiality, authentication, identification, non-repudiation, and key exchange.

Lightweight block-ciphers and lightweight stream-ciphers are being standardized. Lightweight hash functions are likely to be standardized in the near future.

4.1.2 Power consumption linked to security related data transmission overhead

The protection of the confidentiality of short plaintext messages transmitted by low-power devices is a key issue. The most appropriate cryptographic primitives are stream-ciphers and block-ciphers; the use of public-key encryption does not seem appropriate due to the huge impact on the size of the cipher text in the case of small plaintext messages.

More significant overheads are likely to arise from message headers, and an inability to compress data if it has been already encrypted. For example, it is not possible to compress IP packet payloads if IPsec is in place, and so an entire IP packet may be needed to send a single byte. This motivates keeping encryption at a very low layer of the protocol stack, and performing data and header compression prior to encryption.

4.1.3 Power consumption linked to infrastructure related choices

The largest overheads are likely to come because a device has to interoperate with a particular network infrastructure. For instance, a device may need to interwork with protocols inherited from existing LTE, 3G or 2G networks, or interwork with the layered structure of IP, or use https to communicate with an M2M service provider requiring an http stack and TCP support. Alternatively, there may be complex security when binding to an unacknowledged (UDP) message protocol such as CoAP.

One particular concern is if there are multiple layers of encryption and integrity protection, then this creates overhead in all the ways discussed above. This motivates a question about whether certain security layers can be omitted or collapsed.

4.2 Cost

With a target device cost of 4-5 Euros, this is approaching the current cost of SIM cards / secure elements. How can we stop this security element becoming a disproportionate cost? In particular, the costs associated with a conventional SIM card (or UICC), or even with maintaining the security association for a "soft" SIM (network residency and HLR [Home Location Register] costs) become prohibitive.

To illustrate these points, we provide an estimate for the current costs of supporting M2M devices on a cellular network. At present it is difficult to provide an accurate measure of LTE M2M costs and therefore figure have been based on GSM/UMTS figures. Every end device will incur an initial network cost, and a capacity cost when it is used, even if the usage is not generating any data.

NOTE: The cost estimation shown assumes the use of a conventional SIM card but with a simplified distribution model. The SIM is sent direct from SIM vendor to M2M manufacturer for incorporation into an M2M device. More complex, distribution models where the SIM card is sent to an MNO, then to retail channels, and then to an end customer, to be finally inserted in a handset incur much larger additional costs, and this model is typically avoided by telcos for M2M uses.

Cost element	***Incurred by***	***Cost estimate***
SIM (hardware)	Procure and supply SIM to device manufacturer. Note M2M SIM may need extended temperature range and non-removable fitting. Incurred regardless of subsequent usage.	€1-€2 (per SIM)
IMSI and MSISDN allocation	Each SIM must have unique IMSI and MSISDN allocated from ranges allocated to the Mobile network operator.	Negligible e.g. there may be no MSISDN
HSS/AuC capacity	Every SIM must have matching HSS & AUC entry even if never subsequently used.	€0.25 (per SIM)
MME/VLR Capacity	Every IMSI-attached SIM will need space in an MME/VLR even if it generates no	€1 (per SIM)

	mobility or user plane traffic. Most M2M devices are likely to be permanently IMSI attached so that wake up messages can be delivered.	
SGSN (VLR) Capacity	Every GPRS-attached SIM will need space in an SGSN even if it generates no mobility or user plane traffic. This is a driver to ensure that the majority of M2M devices GPRS attach only when required and detach when finished.	This applies for 2G/3G networks. For LTE this will be merged with MME costs.
SGW SAU Licence capacity	Every Simultaneously Attached User (SAU) requires licensed capacity on SGSN. This is a driver to deter GPRS sessions from being left established and to activate PDP-Contexts only briefly, when data transfer is required	€1 (per SAU) Note this may well disappear under future licensing models (e.g. competition from open source)
PGW SAU Licence capacity	See above however if SGW is combined with PGW then this is a shared cost.	€1 (per SAU)
IT element costs (e.g. account management costs & SIM distribution model)	Provisioning System, Billing Systems, CUR etc. FFS.	Very variable, will be minimised for M2M

It will be seen that every SIM supplied to an M2M customer incurs a network cost in region of €2 to €3, and for each allowed to be GPRS attached and active in a session a further €2-3

The traditional revenue model is largely traffic driven, which is clearly inappropriate for such low traffic devices. For on-net usage, that misalignment is solved by the commercial terms of the contract, however for roaming (whether national or international) the existing roaming agreements would apply by default and result in a disproportionate cost to the roaming network compared to the revenue generated.

To some extent, this can be addressed by the M2M End device manufacturers, working in partnership with operators. However, not all may do so, and they will be an interest in keeping the end device simple and basic and thus reduce production costs.

To protect the operator from such unhelpful device behaviour it will be necessary to have some form of compensation. This may be a contractual cost penalty (implying a measurement capability, KPI, SLA and enforcement process) or a core element capability to constrain a category of resource usage.

4.3 Scale

Part of the vision for EXALTED is derived from a Wireless World Research Forum (WWRF) view of trillions (million of million) of connected devices, and with such numbers (essentially two further orders of magnitude bigger than currently-known use cases), the requirements start to look different. Further, even with “only” 50 billion devices, the price point for managing an individual device and associated “account” is important.

In cellular networks, account management is the process used to associate a K, IMSI with a paying subscriber, and assign corresponding usable identifiers (such as an MSISDN). It involves the creation and maintenance of various subscriber records in different network nodes e.g. in the HSS (formerly HLR), the MME (formerly SGSN) and the Serving Gateway/PDN gateway (formerly GGSN). Some of these nodes and associated records are described in the table in Section 4.2 above together with the typical cost impacts of such processes.

As well as such core network functions, account management typically involves the process of creating and collecting charging records and sending them to billing and prepaid systems. This further extends into other customer management systems (like SubAdmin, online billing, prepay top-up and customer care).

Such systems are highly variable, with specific customizations for particular countries, regulatory concerns, tariff terms etc. While some interoperability and common features are required (e.g. to exchange charging records for roaming customers or devices), it is difficult to say much more in general about how MNOs handle their billing arrangements.

It is predicted that large numbers of machines will require access to wide-area mobile networks. Each of these machines may only require authentication very occasionally but may have all the basic equipment to allow connection to at least one access network when that is required. However, just requiring that each device be allowed to authenticate itself to the network from time to time, may undermine the benefits of certain mobile M2M services (particularly those services that are predicated on a low cost machine/service).

Consider the implications of providing 50 billions (or trillions) of devices with a separate, provisioned SIM card. While this inconvenience is significant when considering the conventional provision of mobile telephones and data card/modems with SIMs, SIM-enablement of “machines” present additional problems simply by virtue of the number of these devices and their typical (low and sporadic) frequency of use. M2M applications are expected to increase significantly the number of unused or infrequently used SIMs and to cause a consequently greater level of disruption to the network operator who wishes to enable such devices. All the additional costs in terms of provisioning, quarantining (or keeping minimally active) etc. of such machines can be relatively expensive and when compared with the potential market for the mobile M2M service may be found incompatible with low cost services.

In theory, devices could have a "soft SIM" (a SIM module in software or firmware) instead, but storage of the authentication key (K) outside secure hardware would present major security issues (see key storage basics in section 2.2.1), and there is still significant cost to the network operator (requiring heavy usage of the core network components in particular the home location register (HLR) and the authentication centre (AuC)) and arranging provisioning/creating subscriptions. Or it could be possible for devices to have some other form of authentication technology. However such a solution would require major network re-design, and could potentially prevents connection onto existing 3G and GSM networks.

Therefore it can be seen that cost presents a number of challenges to the way that M2M devices are secured & provisioned, especially if the number of devices scales up dramatically. A number of clear optimizations for M2M traffic (such as single bills covering large numbers of devices, not individual bills per device) can however be expected.

The following elements for further cost reduction in scale could also be envisaged:

- Reduce SIM cost in the provisioning process by :
  - greater efficiency
  - improved scalability
  - reducing distribution costs (especially by use of embedded SIMs)
- Reuse the SIM card and the security it provides for multiple devices in capillary networks.
- Reduce the number of devices that require SIM's to reduce the HSS capacity & processing requirements
- Reduce network side costs by:
  - Leveraging typical I.T database technologies in the HSS and/or MME e.g. LDAP, UDC etc.
  - Improve MME licensing model so that storage capacity is not the main cost element but processing power becomes the limiting factor instead.
  - Reducing processing load by having fewer overall attach/detach events and minimise total number of separate device entries in the MME e.g. use device groups.

4.4 Unattended Devices / Services

M2M devices connect to themselves over various PAN and LAN technologies (e.g., ZigBee, Bluetooth, Wi-Fi), thus forming multiple capillary networks. The capillary networks provide connectivity within M2M devices and also between M2M devices and the backend LTE infrastructure through M2M gateways.

When deployed in capillary networks, M2M devices should be configured, prior to M2M application running. M2M may involve communication without or with only very limited human intervention. Furthermore, some of these devices may be deployed in remote and hostile places and left unattended. Without human intervention, M2M device configuration can be performed remotely using an infrastructure entity after automated bootstrapping operation relying on that entity. An AAA-based infrastructure (as in cellular networks like LTE) can be used to support the security of the M2M network. Re-configuration and other security update operations (e.g., security algorithms, software update) are also carried out using the remote infrastructure entity.

During application running, devices may need to access or to deliver data to other network devices or infrastructure entities. Remote authorizations need to be obtained, thus, allowing accessing devices to be provisioned with credentials that are used to provide security protections (e.g., confidentiality, integrity, non-repudiation, audit) for data access and/or delivery.

From the M2M device side, reducing the communication overhead is the main challenge associated with these remote procedures carried out by resource-constrained M2M devices. Compared to manual operations, remote and automated operations require communication resources. Communication resources are provided not only by the communicating device, but also within the M2M network itself if messages are relayed by intermediary devices. Connectivity is another important challenge. Unreliable communication is costly to the M2M device, because it requires multiple retransmissions of sent messages. Moreover, the device should be able to access the network in order to reach the service infrastructure, whenever it moves within the same network or from a network to a different one. For instance, it should be able to access the network through other devices in the network. M2M devices should be configured to relay other devices' messages. Additionally, M2M devices should be enabled to

support network access through network entities (e.g., gateways, access points) subscribed to different administrative domains.

M2M devices are generally small electronic devices that are deployed in large number in the environment. The infrastructure that handles the remote management of this type of devices should be able to scale to their large number. From the infrastructure side, managing a large number of M2M devices is the main challenge. Some of the security management mechanisms can be delegated to other network entities down to the M2M devices (e.g., gateway, other M2M devices), so as to offload the infrastructure. Such delegation mechanisms could be supported, in particular for frequent security operations (e.g., re-authentication, authorization). The challenge is to provide these delegation services in a secure way down to the devices with fewer changes to the already deployed infrastructure.

Communication between the M2M device and the management infrastructure involves multiple entities in-between (e.g., routers, gateways, other devices); therefore, the remote management infrastructure should provide end-to-end security mechanisms. It should also support secure inter-domain security mechanisms, if network entities from different administrative domains are used, in particular for network access.

4.5 Network Complexity

M2M networks are highly dynamic networks that frequently change of topology as a result of node mobility, frequent node join and leave, and possible long distances between nodes. Not only M2M devices are mobile, other network entities like gateways or routers (e.g., vehicular networks) can be, as well, mobile.

For a mobile device moving from a gateway where it has authenticated to another one, it should perform again the authentication procedure with the new gateway in order to get access to the network. The main challenge is how to allow the device to perform authentication while it can continue to seamlessly communicate. Fast re-authentication can be provided through proactive approaches, whereby the authentication entity will send keys to potential gateways before the mobile device hands over to them. Or the new gateway may interrogate the old gateway for the key material in order to accelerate the authentication procedure (e.g., S1-handover and X2-handover in LTE EPS [11]). For this, the authentication infrastructure should keep track of device location and their mobility patterns.

In a multi-domain setting, the M2M device may not be able to directly reach the network of its administrative domain. It may connect through a network from another domain to which its own domain has some business relationship. Network access should be protected based on authentication procedures within the multi-domain infrastructure.

M2M devices are limited in terms of communication range and may not be able to reach the gateway directly. They may rely on other devices to relay their messages, for instance their authentication request messages. Intermediary devices should relay such request messages toward the gateway. After successful authentication, all cryptographic keying material needed for secure packet forwarding should be provided to devices to build link-layer security associations.

M2M devices may move separately or in bulk (e.g., devices attached to passengers in a bus). Re-authentication procedures of the bulk of devices may result in large communication overhead at the network access entity. A key challenge is to propose re-authentication and network access control procedures to handle such peak overhead. For instance, organizing the bulk of devices into groups could be supported to reduce management costs. The movement of the bulk of devices may be coupled with the movement of the gateway. In this case, delegation mechanisms could be supported to handle the re-authentication procedure of the bulk of devices.

4.6 Application Layer Complexity

It is expected that multiple M2M service providers may soon be offering their service to enable the deployment of ubiquitous machine to machine applications. In anticipation of that, standardization bodies like ETSI are defining architectures suitable to accommodate a wide range of M2M application needs.

At the same time however those needs are evolving, creating new challenges. The internet of things, with the grand vision of objects talking to objects, applications, servers is leading to applications requiring more dynamic enrolment, security and communication mechanisms.

M2M service providers will provide the necessary mechanisms for data protection. However, they do not operate in a closed world. It should be possible for a device affiliated to one M2M service provider to be able to communicate with another party affiliated to another M2M service provider, pretty much as a subscriber affiliated to one telco can call another subscriber affiliated to another telco.

Notification is also an essential feature, enabling multiple authorized parties to subscribe to the data stream of a single device, and be notified whenever new data becomes available. Furthermore those parties must also be provided with the security keys to decipher the data received, if that is needed.

Enabling M2M data communication really involves providing two types of enablers:

- Enablers related to the routing, distribution, and advertising of the M2M data transmitted.
- Enablers related to the data security and trust

In some cases the need can arise for two distinct business parties to provide those enablers.

For example, the selection of the M2M service provider for an application may result from factors such as coverage, cost, reliability, existing business agreements already in place, which are all independent from the trust level associated to the M2M service provider. When this is so, the M2M service provider may not have the trust level required to handle sensitive data. In some other scenarios, the M2M service provider itself may be unwilling to assume the liability or the cost associated to dealing with confidential data.

This separation of roles may be achieved by creating mechanisms enabling end to end data encryption, in which the M2M service provider does not know the relevant decryption keys, as opposed to piecewise or hop based data protection (data source → M2M service provider1 → M2M service provider2 → data destination). End to end data encryption can lead the M2M service provider to handle opaquely data streams which have been secured using secrets distributed by a third party business entity.

We will come back on this topic in section 5.7 where we will describe how end to end data encryption could be achieved within the ETSI architecture.

5. Proposed Security Features and Solutions

5.1 Low Overhead Security

5.1.1 Minimizing Energy for Cryptographic operations

The primary purpose of block ciphers is to protect the confidentiality of stored or transmitted data. Block ciphers can also be used to ensure integrity and origin of data. It is possible to construct a lightweight message authentication code (MAC) from a lightweight block cipher.

Two lightweight block-ciphers are being standardized: PRESENT [15] and CLEFIA.[16]. The size of the block is an important parameter to select the most appropriate block-cipher for low-power devices since it is often directly related to the size of the internal state, and thus also related to the execution time.

The PRESENT algorithm is a symmetric block cipher that can process data blocks of 64 bits, using a key of length 80 or 128 bits. The lightweight block-cipher CLEFIA has a block size of 128 bits and a key size of 128, 192 or 256 bits.

These two lightweight block-ciphers should be considered as potential candidates, in addition to 3DES and AES.

As for block-cipher, the size of the internal state of a stream-cipher may be one criterion to be selected for low-power devices. 5 stream-ciphers are defined in ISO/IEC 18033-4:

- MUGI [17] is a stream cipher which uses a 128-bit secret key K, a 128-bit initialization vector IV, and a state variable of 19 64-bit blocks. It outputs 64-bit keystream blocks.
- SNOW 2.0[18] is a stream cipher which uses as input a 128 or 256-bit secret key K, and a 128-bit initialization vector IV, and a state variable 18 32-bit blocks. It outputs 32-bit keystream blocks.
- Rabbit [19] is a stream cipher which uses a 128-bit secret key K, a 64-bit initialization vector IV, and a 513-bit internal state variable. It outputs a 128-bit keystream block.
- DECIMv2 [20] is a stream cipher which uses an 80-bit secret key K and a 64-bit initialization vector IV, and a 192-bit internal state.
- KCipher-2 (K2)[21] is a stream cipher which uses as input a 128-bit secret key K and a 128-bit initial vector IV, and an internal state of 20 32-bit blocks

MUGI, SNOW2.0, Rabbit and KCipher-2 have quite large internal state which makes those stream ciphers a priori not well suited for low-power devices.

Two lightweight stream-ciphers seem to be well adapted for low-power devices due to the small size of the internal state: Enocoro-128v2 and Trivium.

- Enocoro-v2[22] is a stream cipher which uses a 80-bit or 128-bit secret key K, a 64-bit initialization vector IV, and a state variable of 34 bytes, and outputs one-byte keystream block.
- TRIVIUM[23] is a stream cipher which takes as input an 80-bit secret key K = (K0, ... , K79), an 80-bit initialization value IV = (IV0, ... , IV79), and generates up to 264 bits of keystream

Standard hash functions such as SHA-1/SHA-2 are primarily designed to be collision resistant in order to prevent forgery of digitally signed documents. This strong security requirement adds a lot of complexity to the design. Several lightweight hash functions have been proposed recently. The maturity of these new hash function primitives is maybe insufficient from a security point of view.

Table 5-1 extracted from [24] provides an overview of the performance of some current compact algorithms where block ciphers are ordered by block and key size while hash functions are ordered by the size of the output.

	Key size	Block size	Cycles per block	Throughput at 100KHz (Kbps)	Logic process	Area GE	Area rel.
Block ciphers							
PRESENT-80 [6]	80	64	32	200	0.18µm	1570	1
PRESENT-80 [7]	80	64	563	11.4	0.18µm	1075	0.68
DES [42]	56	64	144	44.4	0.18µm	2309	1.47
mCrypton [32]	96	64	13	492.3	0.13µm	2681	1.71
PRESENT-128 [6]	128	64	32	200	0.18µm	1886	1.20
TEA [54]	128	64	64	100	0.18µm	2355	1.50
HIGHT [24]	128	64	34	188.2	0.25µm	3048	1.65
DESXL [42]	184	64	144	44.4	0.18µm	2168	1.38
AES-128 [16]	128	128	1032	12.4	0.35µm	3400	2.17
Stream ciphers							
Grain [15]	80	1	1	100	0.13µm	1294	0.82
Trivium [15]	80	1	1	100	0.13µm	2599	1.66
Hash functions							
	Hash output size		Cycles per block	Throughput at 100KHz (Kbps)	Logic process	Area GE	Area rel.
MD4 [17]	128		456	112.28	0.13µm	7350	4.68
MD5 [17]	128		612	83.66	0.13µm	8400	5.35
SHA-1 [17]	160		1274	40.18	0.35µm	8120	5.17
SHA-256 [17]	256		1128	45.39	0.35µm	10868	6.92
MAME [53]	256		96	266.67	0.18µm	8100	5.16

Table 5-1 : current state of the art for new compact block ciphers and stream ciphers

Considerations of key-size are important when recommending any of these ciphers: GSM used short key lengths when the standards were being developed (54 or 64 bit cipher keys), and a significant security improvement was realized in 3G and LTE when moving to 128 bit keys. It will be virtually impossible for 3GPP to accept a shorter key length again.

Accordingly, the most plausible lightweight block-ciphers look like **PRESENT-128** and **CLEFIA**; and among stream-ciphers, **Encoro-128v2**. It would be premature to recommend a lightweight hash function.

Generally speaking, when using public key, mechanisms, Elliptic Curve algorithms are to be preferred to the use of 2048 bit RSA as they are more economical in term of computing power. An elliptic curve size of 256 bits (primary field curve) is considered equivalent to the security of a 128 bit symmetric cipher like AES. Nevertheless, when power consumption is an issue, asymmetric cryptography should be used sparingly, for example to derive and store symmetric keys for long-lived sessions, rather than repeatedly deriving them.

5.1.2 Minimizing Energy for extra data (transmission)

The choice of a stream cipher seems to be the best choice for the protection of the confidentiality of messages that are transmitted from low-power devices. Indeed, the use of a stream cipher should enable to transmit a cipher text whose length is equal or a few bit longer than the original message of small size.

However, if the stream-cipher is a synchronous stream cipher (i.e. a pseudo random number generator is used to generate a keystream sequence which is XORed with the data stream), then a potential adversary can modify a cipher text by controlling precisely the impact of the illegal modification on the plaintext (this is due to the bitwise operation between the keystream sequence and the data stream).

This security issue is specific to stream ciphers, and block-ciphers do not have this weakness.

The use of a lightweight block-cipher could also be appropriate for protecting the confidentiality of messages transmitted from low-power devices. In this case, it would be better to use a block-cipher with 8-byte block in order to reduce the power consumption related to data transmission.

However, in this case the small size of the message brings an additional security issue. Indeed, if the message space is too small then the renewal of secret keys should be done often. Otherwise, it would be practically possible to identify on the cipher texts whether the associated plaintexts are equal or not.

The choice of a small size for the MAC value raises a security issue, independently from the size of the cipher text. The level of security related to a cryptographic primitive used for protecting the confidentiality or integrity/authenticity of data depends on the size of the secret key but not solely. If both messages and MAC values are very small, e.g. 2-byte message and 2-byte mac value, then the choice of a very small size decreases significantly the security level of the solution, even if long secret keys are used for both the protection of the confidentiality and the integrity/authenticity.

As an illustration, we consider the use of a stream-cipher (with a k-byte secret key) for protecting the confidentiality of an n-byte message and the use of an m-byte MAC value (with an l-byte secret key) for protecting the integrity/authenticity of the cipher text.

The complexities of brute force attacks to recover secret keys are (as usual):

- The complexity of the exhaustive key search for the encryption algorithm is 2^(8k)
- The complexity of the exhaustive key search for the MAC algorithm is 2^(8l)

The probability that an adversary attempt to forge a valid MAC directly related to the size of the MAC value. Here, it 2^(-8l).

Then, given a valid pair (cipher text, MAC), an adversary can modify the cipher text by knowing exactly the impact on the plaintext and choose at random a MAC value. If the MAC value is 2-byte long, the attacker can impersonate the low-power device with a success probability 1/(2^16).

The minimal size of the MAC value has to be set with respect to the targeted security level. This requirement is independent from the choice of the cryptographic primitive for the computation of the MAC value. Indeed, it is always possible to apply a truncation.

There is potentially scope in LTE-M to negotiate MAC length and frequency of use (the above m and n values) when setting up a security association – but can we do this easily? Also, there is scope in LTE-M to keep encryption at a very low layer of the protocol stack, and perform data and header compression prior to encryption. Again, can this be done easily?

We must consider infrastructure issues to answer these questions.

5.1.3 Infrastructure (Signalling, Layer Collapse)

One possibility might be to use security purely at the M2M service layer, and “switch off” network access layer security (as defined by 3GPP). However, this is likely to be problematic for a number of reasons, the most basic one being that M2M “service layer” security is not at

all well-defined in the ETSI M2M security architecture, and may itself be switched off if access layer security is deemed sufficient.

Consider for instance the following extract from TS102690 [8], in section 8.5 “mID security”

> *The mId Reference Point between the D/GSCL and the NSCL shall support data origin authentication, integrity and replay protection, confidentiality, and privacy. There are three distinct ways the mId may be secured:*
>
> *1) Access network layer security...*
>
> *Alignment between the end-points at the access network layer and the M2M network layer is necessary for enabling such a substitution. A careful study shall be conducted before choosing this option.*
>
> *2) Channel security*
>
> *A secure communication channel may be built between the D/GSCL and the NSCL, for example using TLS or IPsec....*
>
> *3) Object security*
>
> *An M2M implementation may rely on securing data at the object (i.e. protocol payload) level.*

Also, within the detailed stage 3 specifications, there isn’t a unique “service layer” solution currently defined, and different vendors of M2M equipment may well use quite different security solutions.

Note that reliance on bearer or “access network” layer security is considered acceptable if the end-points match. Reliance on the “access network” security layer would be desirable for other reasons, related to the order of compression and encryption (see above), as well as minimization of fraudulent use/Denial of Service against the access network.

However, there is a clear issue in that for **LTE** access security, the “end-point” of user plane traffic is the LTE eNodeB (base station), whereas for M2M service layer security, the end-point would need to be a set of “Service Capabilities” existing either deep in the mobile carrier’s network, or in another provider’s network. These end-points **don’t** match, and so within current LTE architecture it is not realistic to collapse mID security to the access layer.

Still, this is not necessarily a problem for LTE-M. The end-point has shifted in various generations of mobile technology (GSM, 3G, LTE) and LTE itself supports several end-points for traffic. The “user plane” encryption terminates at the eNodeB, and some signalling traffic (RRC) also has a security end-point at the eNodeB; whereas other signalling traffic (NAS) has a security end-point much deeper in the network at the MME.

Thus there seems potential for defining an additional end-point for “machine-plane” traffic within 3GPP and aligning it with an M2M service capabilities - NSCL - node (which would retrieve corresponding keysets from the HLR, or possibly the MME). This “machine plane” will have characteristics quite different from conventional user plane traffic (e.g. sporadic, low volume data transmissions); it is in many ways more like signalling data than conventional user data, so should be optimized accordingly. The machine plane should not *replace* the RRC security between eNodeB and device (or NAS security between MME and device) since the device will anyway need to be authenticated before allowing connection to the eNodeB or EPS core. However, the “machine plane” could presumably run alongside the existing security associations/signalling planes. If this machine plane exists, then consideration also needs to be given to minimizing the security overhead on existing signalling traffic (especially on network attach or other authentication events during location updates).

5.1.4 Proposal for Machine Plane Security

Some specific proposals arise by considering the following diagram in draft TR 33.868, "Security aspects of Machine-Type Communications" [10]. The security architecture involves interfaces labelled A1-A3, B1-B2 and C1-C2. The box labelled "MTC Server" is closely related to (and may be identified) with the M2M service capabilities – NSCL – node in the ETSI M2M architecture.

A1 is the existing LTE RAN interface (UE to eNodeB) and A2 is the existing NAS signalling interface (UE to MME). A3 is an interface for non-3GPP access (convergence between 3GPP and WLAN for instance). The arrows are slightly misleading here, in that they should really start at the "UE" side of the left-hand box (the device) rather than at the "MTC Application" side.

The interfaces labelled C1 and C2 are particularly interesting given the above discussion: C1 matches the mId interface, and C2 would be a true end to end interface between the (device-side) MTC application and (network-side) MTC application. The end-points of C2 are unambiguous, but there is some debate over whether C1 should start at the "MTC application" or the "UE" side of the device. (See the "Editor's note" which was attached to the original figure in the draft TR, and has been copied here verbatim.)

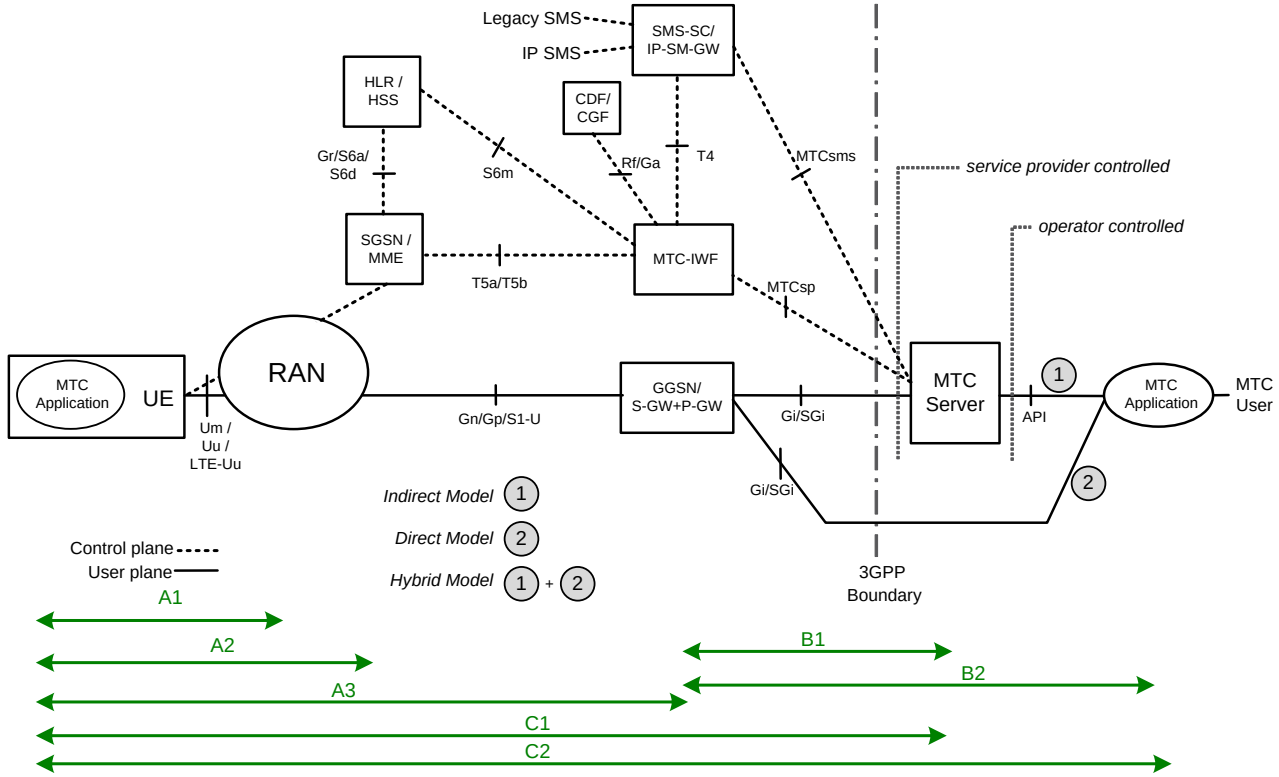

Figure 5-1: Potential high level security architecture for MTC Non-Roaming Architecture for 3GPP Architecture for Machine-Type Communication

Editor's Note: The termination point of security in the terminal side is FFS, i.e. whether it will be in the UE or in the MTC application.

Notice that if there is no specific security on the C1 interface, then there will be a security "gap" between the A1 and B1 interfaces; this is the existing backhaul interface between the eNodeB and PGW. While it is possible to switch on IPsec on this backhaul, there is no guarantee that it will actually be enforced by the network operator for "user plane" traffic. Typically this traffic consists of large volumes of non-secret data (web pages, streamed video

from YouTube etc.); high throughput and low latency are the main goals for delivering such traffic, and confidentiality is not, so that applying an IPsec layer overhead to all this traffic is possibly undesirable. Unfortunately, there is no way to discriminate between different user-plane traffic streams and apply IPsec selectively.

Thus an M2M Service Provider should probably make the (pessimistic) assumption that this backhaul traffic will not generally be encrypted in LTE. Or even if it is, there are still two encryption gaps (at the eNodeB and at the PGW) and either of these may be a concern.

Accordingly, there is a clear driver for 3GPP to provide a security solution over the C1 and/or C2 interface, and this is reflected in the identification of a "Secure Connection" as "Key Issue 2". However, there is an ambiguity in the statement of the "Secure Connection" requirement by SA1 and SA2 – the requirements document [1] states that this is at "application layer and out of scope of 3GPP specification" which suggests they are either thinking of C2, or wish to start the C1 arrow on the "MTC application" side. The architecture document [9] states that the operator shall provide "network security for connection between MTC Server and MTC Device/UE" which suggests they are thinking of C1, and wish to start the C1 arrow on the "UE" side.

SA3 have sent liaison statements to SA1 and SA2 asking to clarify this ambiguity, and there is an open opportunity to influence the decision in favour of the "network security/UE" side as discussed above.

If the security for the C1 interface is applied as "network security" then there are several related proposals and considerations:

1. Assuming encryption is enforced on the C1 interface, then further "user plane" encryption on the A1 interface is redundant, and can be switched off. (This is possible in LTE specifications already, though usually not desirable). Encryption and integrity protection over the signalling plane (A1 and A2 interface) can't and shouldn't be switched off, since it is an essential protection for the signalling traffic to the eNodeB and MME (RRC and NAS).

2. Existing 3GPP encryption and integrity algorithms can be applied over this C1 interface, or there may be scope for introducing further algorithms optimized for machine-type traffic: see the discussion on crypto-algorithms in section 5.1.1 above. The exact algorithm set to be used can be negotiated during network attachments and switches of cipher-mode.

3. There are two obvious candidates for deriving the cipher key and integrity key for use over the C1 interface, but not an obvious choice between them. The most efficient solution at the device side is probably to derive the key at network attachments (and re-authentication events) at the same time as deriving the keys for RRC and NAS signalling: essentially all derivation would happen during a single authentication challenge. However, this raises the problem of how the MTC server then receives the corresponding keys, and may require the lifecycle of keys to be synchronized. (A new keyset is produced at each authentication event). An alternative is to derive the C1 keyset in a separate specific authentication event, which would almost certainly use GBA, as this provides an existing defined interface for the MTC server to receive the corresponding keyset. Unfortunately, this approach may require additional signalling overhead and processing overhead on the device side (it consumes additional authentication vectors).

4. A separate security association over the C1 interface (one which persists even after network detaches) would address one of the other key security issues (Key Issue 1 – MTC Device/UE triggering). This may favour the use of discrete (and rather

infrequent) authentication events for the C1 interface, rather than coupling the lifecycle of C1 security associations to network attachment security associations.

5. We should still address the goal of minimizing signalling overhead when securing the RRC (A1) and NAS (A2) interfaces. We could reduce signalling overhead by only negotiating sessions sparingly; session keys could last for a rather long time-period provided they are only used for limited data. This suggests a longer lifetime for these security associations than is common for existing LTE devices (days or weeks vs. hours) and a mechanism to set key life-time by bytes, rather than by seconds etc. One difficulty is that eNodeBs and MMEs are not designed to store such persistent security association, and tend to "flush" them aggressively when timers are exceeded, to save storage space. Consideration needs to be given to exporting the session keys (in a protected form) rather than flushing, and re-importing as needed. A similar issue exists on the device side, where mobile equipment is typically designed to flush session keys on power-down events. It would be possible to keep session keys on the secure element (eUICC) in non-volatile memory to avoid loss during power-down cycles. This might even allow diversify the session key further on each power-up event (so even if one diversified key is compromised, no on-going risk). The standardization impacts of this are for further study.

6. Draft TR 33.868 discusses a number of other related "Key Issues" to allow optimization of traffic for machines. There is a specific "Key Issue" on "Security of Small Data Transmission which relates to the discussions in points 2 and 5 above. A further key issue on "Group Based Optimization" could be achieved for instance by broadcasting security challenges (rather than retrieving a separate RAND and AUTN for each individual authenticated UE), so reducing the number of passes and round-trip time on authentication events. The difficulty here is how to achieve authentication of the network (mutual authentication is a key functionality in 3G and LTE as opposed to GSM); presumably the broadcast challenges would need to be digitally signed by public key means to allow all UEs to recognize them, which implies a redesign of 3GPP AKA to use public key (at least on the network side). This is for further study in a future WP5 deliverable on broadcast security. Both these Key Issues have been de-prioritized from Release 11, since a lot of work is expected to resolve them, rather than small CRs to existing specifications. Still, one advantage of a capacity for the eNodeB to digitally sign things would be that it could digitally sign rejection events as well, solving another "Key Issue 3: Reject message without integrity protection".

5.2 Embedded Secure Elements and Remote Provisioning

5.2.1 Background on Embedded SIM

In November 2010, the GSMA announced the creation of a task force to "explore the specification of a remotely activated embedded SIM" (see [12]). This means that SIMs in embedded devices, which in most cases cannot be removed, can be securely updated with operator credentials up to and even after the point of sale, and will also allow the secure re-provisioning of alternative operators during its lifespan.

This initiative is motivated by a number of reasons:

- Various types of appliances are now becoming connected, some of them featuring a very small form factor where the accommodation of a traditional SIM connector may be challenging.
- M2M applications with specific environmental constraints (heat, vibrations, humidity, …) or usages (e.g. alarming) may impose the UICC to be soldered

- Some devices need to be provided "ready to connect" to the end user, requiring to fit the SIM in the device at production or distribution. Distribution and operating costs must be very low because the average revenue is currently very much smaller than in the mobile consumer market.
- It is may be necessary to change the operator within the life time of the device, whereas the SIM is not accessible.

The timing of the initial effort was aimed at delivering requirements to ETSI SCP (in March 2011) and to facilitate the creation in ETSI of both REQ and TEC work-items.

5.2.2 Relevance to EXALTED

The EXALTED project has reviewed the initial GSMA requirements document and considered what elements are needed so that the specifications of the embedded secure element are indeed applicable to LTE-M. These include (to name a few):

i) A realistic Over-The-Air (OTA) scaling model to reduce costs,

ii) Use of eUICC within Gateways,

iii) Commoditization of the eUICC to dilute the up-front certification costs

iv) Model for sharing the eUICC with M2M Service Providers and M2M Application Providers, who may wish to use the resources of the eUICC as a general-purpose secure element

Vodafone was a key driver of the GSMA work and ensured that the requirements document SCP(11)0088 [28] was delivered in a timely fashion, while achieving high consensus amongst GSMA members. There have been a number of additional liaisons since the start of the work item, including responses from 3GPP and 3GPP2 and reciprocal liaisons between 3GPP and GSMA.

A set of requirements discussed in the *GSMA Embedded SIM Task Force* for an embedded Subscriber Identity Module (SIM) which can be remotely activated were submitted to ETSI SCP in March 2011. In particular:

- The document SCP(11)0146r1 [27] was submitted at REQ meeting #29 in Sophia Antipolis proposing the creation of a Work Item on "Use cases and requirements related to Embedded UICCs" with GEMALTO being one of the supporting companies[3]. The Work Item request was approved at the subsequent SCP plenary meeting in March 2011.
- A liaison statement from GSMA, SCP(11)0088 [28], accompanied the above submission, described "Embedded SIM Task Force Requirements and Use Cases".
- A further liaison statement SCP(11)0147r1 [29] was sent from ETSI SCP to 3GPP and 3GPP2 notifying them of the Work Item and requesting further input on Use Cases and Requirements.

Since March 2011, Vodafone have co-authored several contributions to SCP REQ, with an aim at clarifying definitions and formalizing the requirement language. These include:

- SCPREQ(11)0043 CR against_TS_102_412, "Addition of requirements for the eUICC" [30];
- SCPREQ(11)0044 Discussion document on definitions pertaining to Embedded UICC [31]; and
- SCPREQ(11)0064 "High Level Components in eUICC First Provisioning" [32].

[3] Supporting companies: Gemalto, AT&T, Deutsche Telekom, France Telecom / Orange, Giesecke & Devrient, Incard ST Microelectronics, Infineon Technologies, Oberthur Tech., Oracle, Sagem Orga.

The following submissions to SCP REQ#32 (San Diego) were co-authored by Vodafone and submitted on behalf of the whole GSMA:

- SCPREQ(11)0113 “Embedded UICC – A high level remote provisioning architecture” [33]; and
- SCPREQ(11)0118 “GSMA and SIMalliance Collaboration on eUICC Protection Profile” [34].

The following contributions to SCP REQ were submitted or co-authored by Gemalto:

- SCP REQ AdHoc#113 London April 2011:
  - SCPREQ(11)0048: eUICC role definition [35]
- SCP REQ#30 Caserat, IT May 2011:
  - SCPREQ(11)0068: pCR on eUICC definition of profile manager and related requirements [36]
- SCP REQ#31 Marseille, June 2011:
  - SCPREQ(11)0079: pCR on eUICC system architecture [37]

A clear EXALTED requirement is that the solution can scale cost-effectively to tens of billions of eUICCs – many more than are currently deployed. This is difficult, since much of the initial pressure for eUICC is coming from rather high-end devices (consumer electronics or automobiles) and is aiming for the most expensive of the current standardised M2M Form Factors (MFF1/MFF2), mainly for reasons of small size and environmental durability. Further, that initial pressure could in principle be accommodated by ad hoc, non-standard solutions, but those will never reach the necessary scale to support LTE-M.

To address these points, Vodafone have ensured that the scope is not restricted to the existing Machine Form Factors but could apply to other UICC form factors. Even if a UICC is technically removable, it may well be impracticable for cost, design, or security reasons to remove and replace it: in which case the requirement for remote update of subscription still applies. Finally, we have supported interim solutions which will allow a new (even smaller) removable form factor, and so address some of the immediate market requirements (consumer electronics/smart-phones), which are expected to arrive earlier than LTE-M. This provides some breathing space to allow a more thorough eUICC solution to be developed.

Given that the TEC work item has not yet started, the GSMA Task Force has in recent months continued with its own technical work-stream considering the overall architecture and what technical solutions are possible to meet the requirements (the GSMA Task force is addressing questions such as: Is the solution architected around a few “Subscription Managers” as trust points, or a single Subscription Manager as a necessary shared cost, or is there direct MNO-MNO trust? Is there a requirement for a Public Key Infrastructure (PKI), with Embedded UICCs (eUICC) issued card certificates that can be trusted by MNOs?) This is accompanied by a commercial work-stream on the role of the “Subscription Manager” and what parties could be best placed to fill it: MNO, SIM Vendor, Original Equipment Manufacturer (OEM), GSMA as a whole, etc.

The latest analysis on these points was submitted as a LS to SCP REQ#32 in San Diego: SCPREQ(11)0113 “Embedded UICC – A high level remote provisioning architecture” [33].

The contribution was co-authored by Vodafone (who had submitted the original architectural model to GSMA – see figure below) and other GSMA task force members.

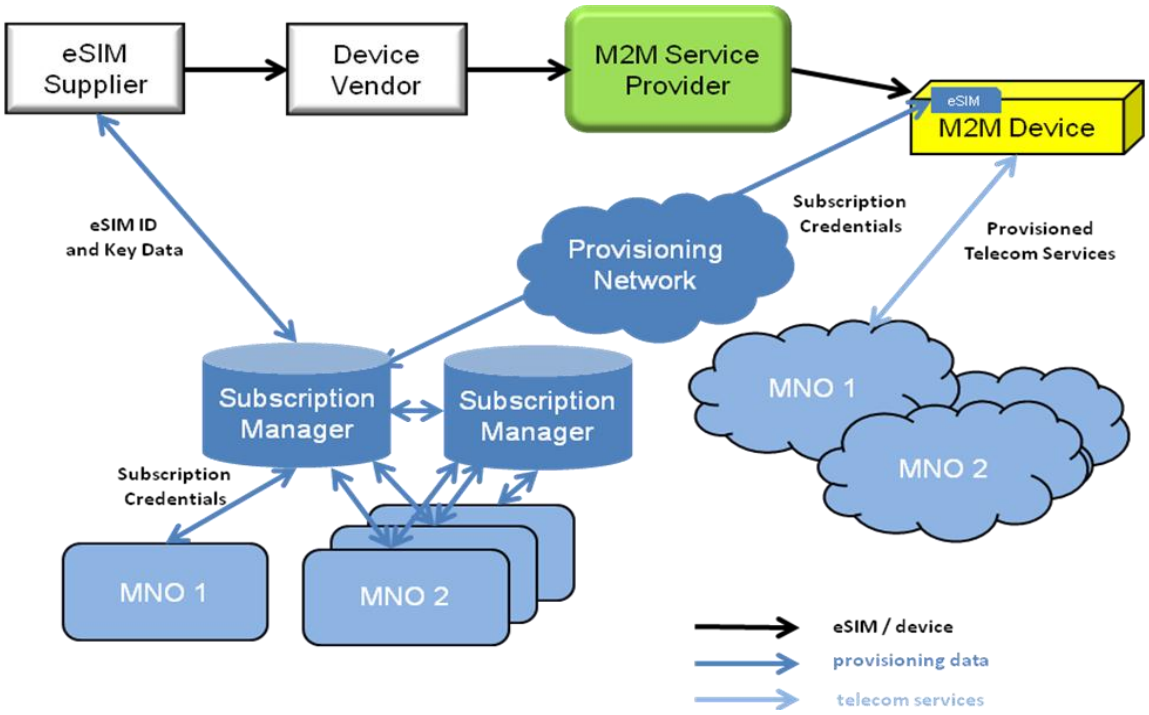

Figure 5-2: High Level Architecture for embedded SIM.

Further GSMA work-streams have considered new legal and contractual models for the eUICC, extensions to the GSMA's SAS, and have responded to further liaisons from 3GPP clarifying the GSMA's understanding of the requirements. Since March 2011, Vodafone have continued to be key drivers and contributors to all these work streams, as well as to the standardisation process in ETSI SCP.

5.2.3 Cost Considerations

In terms of BOM, the introduction of the embedded SIM is not likely to introduce a significant disruption with regard to the traditional removable models. But anyway the BOM represents only a tiny fraction (1/10th to 1/20th) of the overall subscription delivery & activation costs; here **large savings can be expected from eUICC introduction**:

- Reduced distribution cost for the SIM, due to the fact that the SIM will be distributed with the device. It eliminates the need for stocks of cards. Note that this is **also** true in the case of single-MNO UICCs embedded in devices at manufacture; however such an approach creates a new stock management problem. (The M2M manufacturer must maintain and distribute separate stocks of devices for all the relevant MNOs, and this rapidly becomes impracticable.)
- Flexible provisioning solutions will also eliminate the need for cards inventory management
- The type of subscription can be determined at the last minute, or even changed over time in accordance with MNOs' approved schemes, introducing greater flexibility.
- As the SIM will be provisioned/activated only when the device will be installed, subscription costs will be deferred in time. (And subscription costs for a given MNO will only be incurred when a device is actually in use with that operator.)

There are several reasons why the BOM itself for embedded SIM is expected to remain comparable to traditional SIM cards, or even higher, at least until high volumes are met

- New form factor (MFF1, MFF2) necessitating new packaging and new industrial processes

- Environmental conditions necessitating hardened components, with support for extreme environmental conditions,
- Extended life management (as the card replacement cost becomes very high) : specific memory technology, enhanced memory management, audit procedures
- High (indeed increased) level of security: due to specific use cases (tracking, security cameras, …), new anti-theft mechanisms,
- New provisioning schemes are likely to require specific certification processes and components (e.g. PK crypto-processor)

The solution to this problem of higher BOM is higher volume; if the number of eUICCs exceeds conventional UICCs then the cost equation will likely be reversed, even though the product is technically more complicated and has more demanding requirements. This models the evolution path by which UICC eventually became lower in cost than conventional GSM SIM cards, despite being a more complex product (the costs became absorbed by scale).

Note however that there will be a “transition” hurdle, with eUICC being a more expensive component for a time. The EXALTED project have considered whether there are motives to avoid that hurdle via a “low cost” version of eUICC one which sacrifices some security features in order to keep the cost down. For instance, the “low cost” alternative might not be a discrete certified component (instead it might be embedded in a conventional processor) or it might not be personalized with unique keys in the same way that conventional SIM cards receive unique individual keys.

The general analysis and stakeholder view is that these are unacceptable compromises: there are a number of good reasons why eUICC security must be higher than conventional UICC, not lower, and the cost barrier just has to be absorbed.

Even given a higher BOM, we are still nevertheless likely to see deployment because of the overall reduction in logistic costs as described above. Also, there are further ways of optimizing the cost by using eUICCs shared between devices: the eUICC may be included in a gateway or hub for a local capillary network, as described in section 5.5 below.

As a summary: embedded SIM will target different type of markets from consumer electronics to industrial applications. Embedded SIM is expected to allow cost optimization on the subscription management, because of a more flexible & dynamic account provisioning process. It will however require the deployment of high availability subscription management servers (either purchased and operated by operators, or offered as a service by third party providers). On the SIM itself, new technologies which need to be implemented won't allow cost reduction, until high volumes are met. Embedded SIM with flexible provisioning will break a barrier to M2M growth and therefore high volumes will be reached quicker: this will benefit to the ecosystem meaning all stakeholders in this market.

5.2.4 Multi-application eUICC

EXALTED project members (notably Vodafone and GEMALTO) have contributed to the scope of work in GSMA and SCP with the aim of supporting the EXALTED concept. In particular, ensuring that the scope covers multiple smart card applications and application security credentials not just connectivity applications and credentials (a SIM, Universal SIM (USIM) or Code Division Multiple Access (CDMA) SIM (CSIM) application with Ki/K etc.)

This is important because EXALTED envisages that a single secure element would be able to provide (or bootstrap) application layer security for machines; hence there is a need to port this functionality if the MNO changes.

Consider, for instance, what will happen if an eUICC is shared with the M2M Service Provider and/or an M2M Application Provider, as discussed in sections 5.5 and 5.7. These stakeholders may use an API offered by the eUICC, possibly together with remote provisioning facilities offered by the Subscription Manager, to bootstrap their own credentials.

But now, what happens if the owner of the device wishes to change subscription to a different Mobile Network Operator?

There are several possible models for handling this. Some models involve the MNO acting as a trust provider for the M2M Service Provider (and M2M Application Provider), since bootstrapping is performed based on MNO credentials. Other models require a different sort of trust provider (and different set of credentials) to support bootstrapping. It is unlikely that everyone will want to use the same trust provider, so different models need to be considered.

As well as the question of how the M2M Service Provider and Application Provider credentials are bootstrapped, the models also need to consider how they are stored.

1. Change of MNO requires change of M2M Service Provider. This model is likely in cases where the M2M Service Provider functions (NSCL) are tightly integrated with the Network Operator systems.

   Problems: Support for M2M Service Providers who address a mixture of cellular and non-cellular traffic, or who target particular market verticals is likely to be weak. Change of M2M Service Provider would also in general force a change of Application Provider root keys. This whole approach represents a large cost to changing operator, and if it becomes established, may require regulatory intervention to force operators to allow easy change, and bear the changeover costs themselves.

2. The bootstrapped credentials (Kmr, Kma etc.) are stored *persistently* outside the eUICC within the memory of the device. Any changes to the eUICC have no impact on these credentials so are essentially invisible to the M2M Service Provider or Application Provider.

   Problems: An obvious difficulty with this model is the persistent storage of sensitive keys outside the protected memory of the eUICC. One mitigation is to store them in a separate protected area or secure element, but that is just (rather pointlessly) increasing the cost of the device.

3. The bootstrapped credentials (Kmr, Kma etc.) are stored for a *period* outside the eUICC within the memory of the device, but are then re-bootstrapped every so often to mitigate the risks of long-term storage outside the eUICC. In this model, if the bootstrapping is based on Network Access credentials stored in the eUICC (in the USIM application), then a change of MNO would require bootstrapping with the new MNO when the Kmr or Kma next expires.

   Problems: The bootstrapping procedure is expected to require 3GPP's GBA (Generic Bootstrap Architecture) and the incoming MNO may not support this. Even if the incoming MNO does, it may not (yet) have a technical interconnect or commercial agreement to allow use of GBA with the M2M Service Provider or M2M Application Provider. This would be another barrier to changing operator.

4. The bootstrapped credentials are stored persistently *inside* the eUICC within a dedicated M2M Service Provider or M2M Application Provider applet. This applet persists on the eUICC even if the MNO changes.

   Problems: This model assumes that the "incoming" MNO will be happy to share the eUICC with whatever applications are already present. This may not always be the case for commercial and security reasons which are discussed below.

5. The bootstrapped credentials are stored for a period *outside* the eUICC and are then re-bootstrapped as needed (as in 3 above). However, in this case the "bootstrapping" application on the eUICC is one which exists *independently* of the MNO, and persists on the card (together with long-term credentials) even if the MNO changes.

   Problems: This model again assumes that the "incoming" MNO will be happy to share the eUICC with the "bootstrapping" application already present.

5.2.5 Commercial and Security Considerations with multi-application eUICC

It will be seen that there are real difficulties with all these options. Options 4 and 5 require the incoming MNO to potentially share the eUICC with parties that may already be installed on the eUICC. Including in some cases with a direct competitor and this may create business conflicts. This model, pushed to the extreme may even lead to have 2 network access applications such as the SIM or USIM installed on the same UICC. With the eUICC, the secure element owner may not be the application issuer and may simply be the end user or the M2M service provider.

Sharing a single secure element between business competitors has not been done in the past (the idea of having a single credit card shared between 2 payment institutions for example may seem odd). However it is possible that such model may become popular. In this case a new trust model, based on device owner responsibility needs to be developed. It should be noted, additionally, that these concepts totally change the concept of "branding" of the card.

From a security standpoint there may be a concern that the applications on the eUICC may interact in undesirable ways, possibly causing card fault or loss of information. However secure element technology has evolved and proper isolation and independence of applications is now addressed using robust card architecture of isolated GlobalPlatform “Security Domains” and a Java Card OS, with applets having separate memory and sandboxed execution threads. Certification (at a high level of assurance such as Common Criteria EAL 4+) that the eUICC truly remains secure in the presence of applets installed in other Security Domains is addressed by using a concept developed by GlobalPlatform of certification by composition. This certification scheme allows independent certification of the platform (the UICC) and the applications. This certification model is used for UICC implementing contactless applications (NFC).

Apart from this, there may be some business concerns (e.g. fraud prevention) if a eUICC supports two Network Access Applications that are simultaneously “selectable” (visible to the terminal). In that case, the terminal would be able to select between them, and there may be significant problems in demonstrating that the end user actually **intended** to use the NAA for MNO A rather than the one for MNO B. Furthermore, this potential issue is not confined to Telcos: very similar concerns have been noted for payment cards and NFC. Suppose for instance that a secure element (or terminal) hosting competitor payment applications is swiped to make a payment; one of the applications will be selected for payment, but the customer could potentially dispute which of them he **intended** to use. One solution to these problems lies in the implementation of suitable user confirmation mechanism such as the validation of a PIN code every time there could be an ambiguity in the selection of the proper application. (However, this would not work with unattended M2M devices.)

One way of interpreting the “profile” is as an overall selection of applications and files (files within applications) which can exist on the card and be simultaneously visible (selectable) to the terminal. There are a few “candidate” profiles for the eUICC, and each of these candidates contains a consistent set of applications whose owners are all willing to co-exist with each other on the eUICC. The application owners will interact with a new role of “profile owner” to ensure that these consistency and co-existence requirements are all met: an application can be part of a profile only through the agreement of both the profile owner and the application owner. The “profile owner” may itself be a particular application owner (such as an MNO), but it could in principle be anyone, including a Subscription Manager. This is the “many-few” stage.

Once a set of consistent “candidate” profiles have been identified, it remains to decide which candidate is then chosen or “enabled” on the eUICC. This is the concept of an “active” profile as described in SCP(11)0088 [28], and represents a “few to one” stage.

It must fundamentally be the choice of the subscriber (in general the owner of the device) which profile is enabled (active), though there may of course be commercial restrictions on how often that choice can be exercised (contractual periods, device subsidies to be paid back and so on). To facilitate this choice of profile, and any change of profile, there will also need to be stages where the applications of a replacement profile are loaded to the eUICC but are not yet in the "selectable" state. This is the concept of a "dormant" profile as described in SCP(11)0088 [28]. Notice that in the case where an application is part of **both** a currently-active (outgoing MNO) profile **and** a soon-to-be-active (incoming MNO) profile, then no addition, removal or re-keying of the application will be required, nor is there any need to move it between Security Domains.

Within this model, it can be quickly seen why the GSMA requirement of only one "active" profile at a time is imposed. A single "active" profile is a known consistent set of files and applications, whereas two distinct profiles will in general contain inconsistent sets (from application providers who may be unwilling to share the eUICC with each other). So there may be no way of making several profiles active at once while keeping everyone happy. An exception to this principle could occur if we consider different domains of application for profiles. For example, profiles containing NAA should be exclusive and activated one at a time, whereas profiles containing different domain of applications could be activated simultaneously (Payment application profile and MNO profile). Unattended devices shall have a policy clearly defined and endorsed by the device owner on how to use the different applications available on the eUICC.

In some cases, commercial conflicts may be easier to resolve. Consider Model 5 for instance. Suppose that the "bootstrapping" application consists of a simple private key and certificate installed at eUICC manufacture, and this is used for TLS or EAP-TLS exchanges in bootstrapping (as discussed in 5.7.1 below). Such a key is not in any obvious sense a "competitor" application to the MNO's NAA (or to other third parties), and further the private key and certificate may actually be essential functionality to allow the eUICC to work at all and install credentials from multiple operators.

If a eUICC is installed with such a "general purpose" private key and certificate, the eUICC manufacturer may be able to charge for using it, and so may be able to secure a revenue stream from bootstrapping M2M Service Providers or Application Providers. Another difficulty is ecosystem support: if the ecosystem is generally one of M2M Service Providers that are used to doing GBA bootstraps with their associated MNOs, then there may not in practice be much interest in using alternative certificate-based bootstrap methods.

There is a possible solution here based on the "provisioning profile" concept of SCP(11)0088 [28]; the provisioning profile is one which is initially installed on the eUICC to facilitate the remote loading of real operational MNO profiles, which are used for commercial service. For a number of reasons, the provisioning profile will probably need to stay on the eUICC through its lifetime (though it will often be dormant: see above). However, notably the provisioning profile will contain a USIM, and so could in principle be used for GBA-bootstrapping of M2M Service Providers, regardless of which MNO profile then provides Network Access. Also, it is a rather basic assumption of the eUICC model that MNOs are willing to co-exist (to some extent) with the provisioning profile; further, the revenue streams from bootstrapping may become part of the business model for someone offering a provisioning profile and maintaining it over time. This approach could be explored further

5.2.6 Activation Process

A clear aim of the eUICC is to reduce the cost of provisioning a device for use with a particular mobile network. Bearing in mind the sharing possibilities discussed in section 5.2.4, it should also reduce the cost of provisioning for use with a particular M2M service provider, and of provisioning for use with a particular application or service.

Some thought should be given to the workflow, since ideally all of these provisioning processes could be automated at device end with no human attendance. But how exactly does a human give permission to start it off? Does the M2M customer just request bulk provisioning through a web interface? There are a number of security issues of doing that, based on error and/or possible deliberate fraud.

Consider for instance an M2M customer who has devices they wish to provision. Presumably, the customer has a list of device identifiers which it provides to an MNO, requesting that these be provisioned with the MNO's Ks and IMSIs. The MNO then passes the same list to the Subscription Manager, requesting that corresponding K/IMSI pairs are delivered Over The Air to the devices. Some major questions arise here:

Question 1: In what format are the device identifiers, and how exactly are they matched up to identifiers that the MNO and Subscription Manager can process? For instance, if the customer just provides the MNO with a list of IMEIs, then the MNO may not easily be able to tell whether these devices contain an eUICC at all, or if they do, who the appropriate Subscription Manager is.

One possibility may be that the device (at power-on) interrogates its eUICC to find the corresponding eUICC id, and then itself reports the pair (IMEI, eUICCid). But to whom – how does the device know which customer or MNO or Subscription Manager is supposed to know this paired information? And how is the information transmitted anyway: presumably it has to use a pre-loaded provisioning subscription.

Alternatively, the interrogation could work the other way: the eUICC interrogates the device to find IMEI, and reports the combination of eUICC id and IMEI, presumably to the Subscription Manager. At that point, if an SM is asked by an MNO if it "knows" whether an IMEI has an eUICC (and if so, which one) it could in principle respond; but it still leaves a rather inelegant procedure whereby the MNO may have to ask around several SMs to find out who currently is able to update that device. And if the device hasn't yet powered on and reported in, it is possible that none of them will know.

Perhaps the best solution is that the M2M customer needs to provide eUICC identifiers as well as (or instead of) device identifiers, but then how does the M2M customer learn these? Is it part of the ordering process that the device manufacturer sends the customer a list of eUICC identifiers paired with IMEIs. Are the eUICC identifiers printed (or bar-coded) on the side of the equipment?

Question 2: How does the M2M customer demonstrate that they **own** the device being targeted, or at least have the permission of its owner to update it with a designated MNO's credentials? One advantage is that the fraud risk seems small: an M2M customer doesn't obviously benefit from putting a subscription that it is paying for on someone ***else's*** device; nevertheless Denial of Service and spying attacks must be considered, and "mistakes" in the provisioning process could represent a devious attack method.

If the M2M customer is entirely trusted by the MNO (and so won't claim ownership of devices it doesn't own) this is not a problem, but even then there could still be mistakes in listing device identifiers. The process of requiring the M2M customer to provide both eUICC id and IMEI could help here, since only a genuine customer (or someone else in the supply chain from the device manufacturer) is likely to know both. Plus the Subscription Manager can check the pairing matches by using reporting information from the device (as discussed above) so that simple errors will be spotted quickly.

Question 3: When the Subscription Manager sends an MNO's credentials to the eUICC and instructs the eUICC to make them "active", are there any confirmation checks at the device side? Does someone near the device have to push a special button, or enter a special boot cycle to confirm that they will accept the new subscription? If there is no explicit confirmation under the customer's control, at each install of a subscription, then there is potential for bill dispute for reasons similar to those discussed in section 5.2.5 above.

Clearly a user-controlled acceptance is going be difficult/impossible with unattended devices, though in theory the device could be installed by the user in a “ready to accept” mode, so that at first power up it will accept any IMSI/K provided, or any IMSI/K from a relevant MNO (while rejecting others). The eUICC could then report the “ready to accept” mode to the Subscription Manager, so that the SM does not waste time and resources providing credentials that the destination device will not subsequently accept. Then, after installing the new subscription to the eUICC, and selecting the new network, the device would signal “accepted” and leave the “ready” mode.

However, notice that the sending of this “accepted” signal should also be under the customer’s control, and authenticated as such. This means that it may not be possible to send the accept from the device until a full security bootstrap is complete, at least up to the level of M2M Service Provider keys. Once this bootstrap is done though, it becomes easier to authenticate a further change of subscription request: the customer could instruct the device remotely via the bootstrapped keyset (or via Device Management) to re-enter the “ready” mode, and so start again.

Question 4: What happens if the installation of new credentials fails, or the “accept” signal is otherwise not sent from the device? It seems there needs to be some sort of recovery or fall back to a previous subscription to try again. Fall back to the “provisioning subscription” is the most obvious solution here, but that requires the provisioning subscription to exist, and still be workable (e.g. the corresponding K still exists in an MNO’s HLR/AuC somewhere). As discussed above, it is not obvious why an MNO would maintain the provisioning subscription on a long-term basis just on the “off-chance” that it needs to be used for recovery purposes.

While all the above have been discussed in the context of an M2M customer selecting an MNO subscription, similar issues apply in the case of selecting an M2M service provider or Application provider. If any of these selection events require new credentials pushed to the card remotely, then a similar process flow will be needed.

5.2.7 List of EXALTED contributions to Standards

Bearing in mind that EXALTED was still in the first year of activity in 2011 and the output derivable from the actual project work is limited, especially in WP5, a significant number of security contributions have already been submitted to 3GPP and ETSI. These are summarised in table 6-1, below.

Note that 3GPP, ETSI and GSMA are the bodies most relevant to EXALTED. However GSMA contributions are not publically available, so this list focuses solely on contributions to 3GPP and ETSI.

Date	Meeting	Document number	Document Title	Source companies	EXALTED Partner(s)	Comments	earlier versions	Link
Apr-11	3GPP SA3#63, Chengdu	S3-110558	LS to ETSI M2M on potential co-operation between 3GPP work on MTC security and ETSI M2M	Vodafone	VGSL	Work was progressed on security aspects of Machine Type Communications. Based on a Vodafone proposal, a liaison statement was sent to ETSI M2M to identify areas of co-operation and avoid unnecessary overlap.		ftp://ftp.3gpp.org/TSG_SA/WG3_Security/TSGS3_63_Chengdu/Docs/S3-110558.zip
Mar-11	ETSI SCP REQ #29, Sophia Antipolis	SCP (11)0146r1	Proposed WID: Use cases and requirements related to Embedded UICCs	AT&T, Gemalto, Sagem Orga, et al	VGSL, GTO	Work Item accepted: see also LS from GSMA. Requirements for embedded UICC for M2M devices (scope of EXALTED) were submitted as part of GSMA use cases and requirements documents.		http://portal.etsi.org/portal/server.pt/community/SCP/333
Mar-11	ETSI SCP REQ #29, Sophia Antipolis	SCP (11)0088	Embedded SIM Task Force Requirements and Use Cases	GSMA	VGSL	LS from GSMA to instigate creation of above work item in ETSI SCP REQ		http://portal.etsi.org/portal/server.pt/community/SCP/333
Mar-11	ETSI SCP REQ #29, Sophia Antipolis	SCP (11)0147r1	Liaison Statement on new Work Item for eUICC - to 3GPP and 3GPP/2	ETSI TC SCP	VGSL, GTO	See above. This outgoing LS to 3GPP was created when setting up the Work Item in ETSI		http://portal.etsi.org/portal/server.pt/community/SCP/333
Apr-11	ETSI SCP REC ad hoc #113, London, UK	SCPREQ (11)0043	CR against TS_102_412, "Addition of requirements for the eUICC	Deutsche Telekom, Giesecke & Devrient, Telefonica O2, Vodafone	VGSL			http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

Apr-11	ETSI SCP REC ad hoc #113, London, UK	SCPREQ (11)0044	Discussion document on definitions pertaining to Embedded UICC	Deutsche Telekom, Telefonica O2, Vodafone, Giesecke & Devrient	VGSL			http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639
Apr-11	ETSI SCP REC ad hoc #113, London, UK	SCPREQ (11)0048	Additional role definition for eUICC remote provisioning	Gemalto	GTO			http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639
May-11	ETSI SCP REQ #30, Caserta, Italy	SCPREQ (11)0064	High Level Components in eUICC First_provisioning	Vodafone Group	VGSL			http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639
May-11	ETSI SCP REQ #30, Caserta, Italy	SCPREQ (11)0068	pCR on eUICC definition of profile manager and related requirements	Gemalto	GTO			http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639
Jun-11	ETSI SCP REQ #31, Marseille, France	SCPREQ (11)0079	pCR on eUICC: System architecture description	Gemalto	GTO			http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639
Jul-11	ETSI SCP REQ #32, San Diego, USA	SCPREQ (11)0113	Embedded UICC – A high level remote provisioning architecture	GSMA Embedded SIM Task Force: Technical Stream	VGSL	co-authored by Vodafone and submitted on behalf of the whole GSMA		http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639
Jul-11	ETSI SCP REQ #32, San Diego, USA	SCPREQ (11)0118	GSMA and SIMalliance Collaboration on eUICC Protection Profile	GSMA	VGSL	co-authored by Vodafone and submitted on behalf of the whole GSMA		http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

Table 5-2: EXALTED contributions to 3GPP and ETSI

5.3 Self-organization and Pairing in Capillary Networks

Section 5.7.1.3 below summarizes the M2M service bootstrap procedures defined by ETSI M2M working group. M2M service bootstrap serves the purpose of establishing initial shared secrets between devices/gateways and the Network Service Capability Layer (M2M service provider). The shared secrets defined in the initial run of the service bootstrap procedure may be used afterwards on a regular basis to derive shorter lived keys which are used to secure M2M communications.

The same need to define initial secrets arises also in capillary networks located possibly behind an M2M gateway. However, if the ETSI architecture assumes Wide area networks connectivity and the possibility to rely upon infrastructure components for the bootstrapping operation, that hypothesis is not always valid within capillary networks which may be isolated from the net.

The methods used for bootstrapping security within capillary networks may be classified in 3 categories:

- Methods suitable when capillary networks do have IP connectivity to a wide area network, and relying upon an infrastructure to perform security bootstrap.
- Methods relying upon the gateway to help perform security bootstrap for capillary devices located behind the gateway.
- Self-organized security bootstrap between capillary devices

Security bootstrap in the capillary networks may target the establishment of pairwise keys or a group key among the communicating devices Security bootstrap may also target the publication of a public key associated to each device to other devices. The public key is then used in conjunction with the private key of each device to establish secure shared key and/or shared group key through the authenticated key establishment mechanisms or to directly secure communication between devices using asymmetric cryptographic techniques.

In self-organized capillary networks where devices do not have a priori trust relationship, the key establishment can be done through the pairing mechanisms. Pairing is the process of establishing shared key between two or more devices or authenticating devices' public keys without using pre-shared secrets or Public Key Infrastructure (PKI). Based on the application and scenario, the desirable outcome of a pairing mechanism can be pairwise keys, group key, or authenticated public keys.

Whitfield Diffie and Martin Hellman were the first to study the issue of establishing shared keys between two parties that do not have a priori trust relationship and only communicate over insecure communications channels. They laid the foundations of public key cryptography in their 1976 seminal paper [12] and proposed a new key agreement protocol known as Diffie-Hellman (DH) protocol. The DH protocol has been extended to group key agreement protocols such as Burmester and Desmedt (BD) protocol [14] which is a two-round contributory Diffie-Hellman based group key agreement protocol.

The DH and its group based key establishment protocols are secured against passive adversaries (eavesdroppers), but they are vulnerable to active adversaries (Man-In-The-Middle attack). As a result, a variety of cryptographic authentication solutions using Public Key Infrastructure (PKI) or shared passwords have been proposed in the literature to provide authenticated Diffie-Hellman based key (and group key) agreement protocols.

A different approach is using Out-of-Band (OoB) channels in authentication process. The aim of the out-of-band channel is to exchange some limited amount of confidential or authenticated information between the pairing devices which is then used to authenticate the established key over the main insecure wireless channel. The out-of-band channels provide demonstrative authentication (identification based on physical context), data origin

authenticity (giving assurance of the source that the data came from), data integrity (providing assurance that data wasn't tampered with by a MITM attacker), and in some cases data confidentiality.

In our work on secure group device pairing for self-organized capillary networks, we studied group device pairing mechanisms and discussed their application in constructing authenticated group key agreement protocols. We divided the mechanisms into two categories of protocols with and without the trusted leader and showed that protocols with trusted leader are more communication and computation efficient. In our study, we considered both insider and outsider adversaries and presented protocols that provide secure device pairing for uncompromised node even in presence of corrupted group members. This work in an on-going research and the final result will be published in the future EXALTED deliverable D5.3.

5.3.1 Using Secure Elements in the Capillary Network

Security protocols use cryptographic keying materials that are stored at M2M devices. The robustness of these protocols depends on the security of the storage of these keys. For example, the leakage of secrets used for authentication leads to the impersonation of the device. An SE provides secure storage and secure execution, whereas devices without a SE may use client-based software protection (e.g., password/PIN protection). Otherwise, keys are stored without protection and can be leaked with software or physical attacks against the device.

There are three cases to be considered here, depending on which devices are communicating in the capillary network, and how many of them have an SE.

- Between two devices with SEs: Communication between M2M devices with independent or embedded SE (e.g., UICC, eUICC) relies on the secure environment provided by the SE. In the network, M2M devices may not have the same security level (e.g., UICCs with different security levels). Therefore, communication between two devices will be at their lowest security level.
- Between one device with SE and one without: In a heterogeneous network composed of devices with SEs and others without SEs, devices with SE may authenticate devices without SEs opportunistically based on their identities provided securely by the authentication server. But, their session keys may be compromised if shared with devices without SEs. Keys should be regularly refreshed to limit (but, not completely prevent) the damage caused by their disclosure.

  Devices with SEs can monitor the behaviour of devices without SEs, evaluate their trustworthiness, and propagate this information to other devices in the network; thus, helping in reducing the threats caused by the absence of SEs.
- Between two devices where neither has SE: For communication between devices without SEs, collaboration between sets of devices (e.g., secret sharing schemes) can be used to build security associations. For example, based on recommendations sent by a sufficient number of devices in the network, a device will accept to communicate with another device. Even though, collaboration between devices exposes the system to new types of attacks (e.g., Sybil attack, collusions), it still can help in increasing the trustworthiness of communication.

It will be seen that, unless carefully managed, secure storage of keys drops to the "lowest common denominator" in a capillary network, which may discourage parties who wish to supply devices with SEs from doing so, or it may deter customers from buying capillary devices with SEs (Why bother, if there is no security gain?)

However, one important way to add value through an SE is to observe that it typically comes personalized with a unique credential, whereas devices without SEs tend to come unpersonalized; or they are not personalized with any reliable credential (compare IMSI and

Ki versus IMEI). While the credential cannot be used to "bootstrap" the pairing process (since the bootstrap server is not yet reachable), it **can** be used to securely report, after pairing, what has happened. For instance:

- The SE could contain a private key/certificate, and report a signed log of pairing events to an M2M Service provider or M2M application. So it could report how many devices were joined to the capillary network, what device identifiers they claimed, what pairing method(s) were used, and cryptographic hashes of the agreed keys and so on. The application provider could then check whether this matched his expectations, and if not, instruct the device with SE to void the pairings.
- The SE could contain a shared key (such as in a USIM application) ; this is used to bootstrap a further shared keyset with the M2M Service Provider or Application provider, and then the shared keyset is used for reporting what has happened during the capillary pairing process. This could also include confidential reporting (e.g. of keys that have been agreed, or of confidential data shared within the capillary)

These methods are likely to be particularly useful where the SE is in a gateway, since the gateway could then receive secured instructions (protected by the SE) to disallow some connections to capillary devices, or to attempt to re-authenticate questionable connections. Better still, if there are several different SEs in the capillary network, then the M2M service provider or Application provider may receive several different "views" of what the capillary network looks like, and could more easily detect Man In The Middle and impersonation attacks (since in many such attacks, different devices will end up with a different "view" on what the agreed keys were).

5.4 Device Management for Low Cost Devices

Device Management (DM) includes a set of procedures, activities and data models which must be supported by both centralized Device Management Servers (DMS) and End Devices (User Equipment, Customer Premises Equipment), in order to allow device management operation. Independently of the DM protocol used, a DMS generally performs (but is not limited to) the following functions:

- Initial Device Provisioning,
- Firmware Update,
- Configuration Backup/Restore,
- Alarm Collection,
- Service Activation/Deactivation,
- Troubleshooting (Customer Support),
- Device Monitoring, Reporting and Network Issues Analysis.

For a successful DM operation, there are several mechanisms that must be implemented in communication and relationship between DMS and Devices, such as:

- Bootstrapping (i.e. an initial configuration to start managing the device),
- Session initiation
- Device Polling mechanism,
- Server Notification,
- Security
- Server and device authentication,
- Confidentiality,
- Data and Configuration Modelling (i.e. implementation of the proper data model for managing various parameters on devices),

- Resources requirements
- Memory and CPU capacity in device,
- Bandwidth on the wire/air.

These mechanisms introduce various challenges when a DM solution is implemented for low cost and low power consuming devices, which operate with rather simple protocol stack. There are several standardization areas which design DM protocols for certain groups of devices, but further adjustments of the protocols for particular implementations might become inevitable in order to deal with these challenges.

Under the focus of security there are 2 aspects of device management which need to be investigated:

- Security mechanisms enabling to securely perform remote management of the devices.
- Bootstrapping of the security schemes that will enable secure communication with the devices.

With the emergence of new technologies as M2M, IPTV, femtocells, standardization bodies are currently discussing this remote management and are choosing the appropriate technology adapted to the new environment.

The selection of a protocol for use in M2M application has not been performed so far. A number of options are being discussed by the standard bodies. We will summarize here the main features of two such protocols, candidates for adoption for M2M device remote management:

- TR069 specified by the broadband forum
- OMA-DM, specified by OMA

5.4.1 TR-069

The protocol called CPE WAN Management Protocol (CWMP) is specified in the Technical Report TR-069 by the Broadband Forum (former DSL Forum). Current version of the protocol (v1.1) is described in Issue 1, Amendment 2, and dates from 2007 [43].

TR069 describes a protocol Customer Premises Equipment (CPE) WAN Management Protocol (CWMP) used to communicate between a (CPE) and an Auto-Configuration Server (ACS) that encompasses within a common framework secure auto-configuration and other CPE management functions.

Figure 5-3 shows the end-to-end architecture and reference of all documents involved with the TR069.

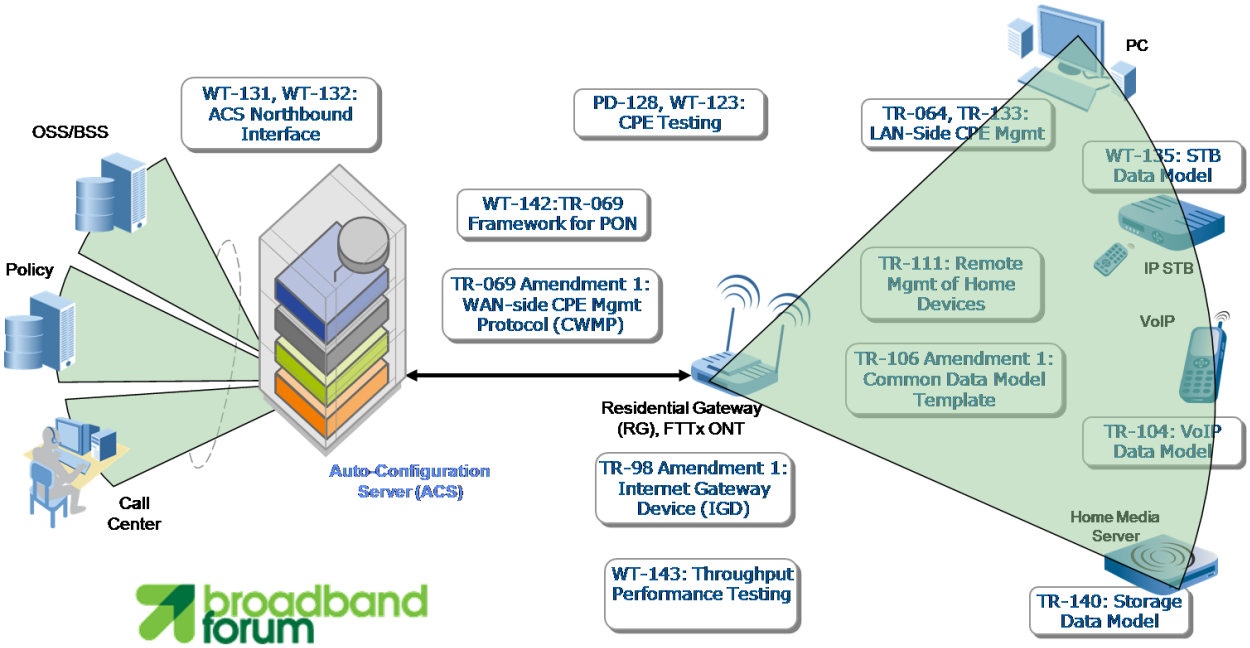

Figure 5-3 :**End to end Architecture and reference documents for TR069**

In TR-069 devices which offer services can be managed by Auto Configuration Servers (ACS) in a simple bi-directional client-server operating mode. The ACS is a server that resides in the network and manages devices at the subscriber premises. CWMP may be used to manage various types of CPE, including stand-alone routers and LAN-side client devices. It is agnostic to the specific access medium utilized by the service provider, although it does depend on IP-layer connectivity having been established by the device.

The CPE WAN Management Protocol is intended to support:

- Auto-configuration and dynamic service provisioning;
- Software/firmware image management;
- Software module management;
- Status and performance monitoring;
- Diagnostics.

The CWMP protocol is intended to be agnostic to the underlying access network. It is also designed to be extensible. It includes mechanisms to support future extensions to the standard, as well as explicit mechanisms for vendor-specific extensions. The CWMP is designed as RPC (Remote Procedures Calls) Methods transmitted in SOAP protocol.

5.4.1.1 Security Goals

These are described in [43] as follows:

- *Prevent tampering with the management functions of a CPE or ACS, or the transactions that take place between a CPE and ACS.*
- *Provide confidentiality for the transactions that take place between a CPE and ACS.*
- *Allow appropriate authentication for each type of transaction.*
- *Prevent theft of service.*

5.4.1.2 Security mechanisms

The CPE WAN Management Protocol is designed to protect the transactions taking place between a CPE and ACS, provide confidentiality for these transactions, and allow various levels of authentication.

The protocol supports the use of TLS for communications transport between CPE and ACS. This provides transaction confidentiality, data integrity, and allows certificate-based authentication between the CPE and ACS.

The HTTP layer provides an alternative means of CPE and ACS authentication based on shared secrets. However, the protocol does not specify how the shared secrets are learned by the CPE and ACS.

5.4.1.3 Service bootstrapping

The RPC Method Specification defines a mechanism that allows a CPE to inform a corresponding ACS of various conditions, and to ensure that CPE-to-ACS communication will occur with some minimum frequency. This includes mechanisms to establish communication upon initial CPE installation in order to 'bootstrap' initial customized Parameters into the CPE. It also includes a mechanism to establish periodic communication with the ACS on an on-going basis, or when events occur that must be reported to the ACS (such as when the broadband IP address of the CPE changes).

In each case, when communication is established the CPE identifies itself uniquely via manufacturer and serial number information (and optional product class identifier) so that the ACS knows which CPE it is communicating with and can respond in an appropriate way.

5.4.1.4 Server Initiated operations

An important aspect of service auto-configuration is the ability for the ACS to inform the CPE of a configuration change asynchronously. This allows the auto-configuration mechanism to be used for services that require near-real-time reconfiguration of the CPE. For example, this may be used to provide an end-user with immediate access to a service or feature they have subscribed to, without waiting for the next periodic contact.

The CPE WAN Management Protocol incorporates a mechanism for the ACS to issue a Connection Request to the CPE at any time, instructing it to establish a communication session with the ACS. While the CPE WAN Management Protocol also allows polling by the CPE in lieu of ACS-initiated connections, the CPE WAN Management Protocol does not rely on polling or establishment of persistent connections from the CPE to provide asynchronous notification. The basic mechanism defined in the CPE WAN Management Protocol to enable asynchronous ACS initiated communication assumes direct IP addressability of the CPE from the ACS. An alternative mechanism is defined which accommodates CPE operating behind a NAT gateway that are not directly addressable by the ACS.

5.4.1.5 Configuration server discovery

The CPE WAN Management Protocol defines a number of mechanisms that may be used by a CPE to discover the address of its associated ACS:

- The CPE MAY be configured locally with the URL of the ACS. For example, this may be done via a LAN-side CPE auto-configuration protocol. If necessary, the CPE would use DNS to resolve the IP address of the ACS from the host name component of the URL.
- As part of the IP layer auto-configuration, a DHCP server on the access network may be configured to include the ACS URL as well as a number of service parameters as a DHCP options.
- The CPE MAY have a default ACS URL that it MAY use if no other URL is provided to it.

5.4.2 OMA DM

OMA DM is a standard developed by the Open Mobile Alliance for remote management of mobile devices (clients). The latest "Candidate" version is 2.0 while the latest "Approved"

version is 1.2.1 [7]. The specifications follow on from an earlier WAP forum and OMA specification for "Client Provisioning" (CP). This allows uni-directional pushing of useful network settings such as APN, proxy address etc. to a mobile client (typically using SMS).

OMA-DM protocol is defined by the OMA-DM Enabler. It contains the following specifications:

- Device Management Bootstrap,
- Device Management Notification Initiated Session,
- Device Management Protocol,
- Device Management Representation Protocol,
- Device Management Security,
- Device Management Standardized Objects,
- Device Management Tree and Description Serialization
- Device Management Tree and Description.

"Tree and Description", "Standardized Objects", and "Tree and Description Serialization" define the data model part:

- Tree and Description defines the formation of the management tree, including nodes, properties, and the device description framework
- Standardized Objects defines the "DM Account", "DevInfo", "DevDetail", and "inbox" management objects.
- Tree and Description Serialization defines the way to convert a run-time management tree or sub-tree into an XML or WBXML structure.

"Protocol" and "Representation Protocol" define the main protocol based on the SyncML Representation Protocol and SyncML Synchronization Protocol. "Bootstrap", or client provisioning, is the first step for preparing a managed device. "Notification Initiated Session", on the other hand, allows management servers to tell the client to initiate a session back.

Finally, DM Security, along with some DM tree properties, constitutes the specification's policy part. Besides the principles of confidentiality, integrity, authentication, and so on, ACL is one example of policies that control whether you can manipulate the node.

OMA DM can be divided into three logical components, shown in Figure 5-4 below.

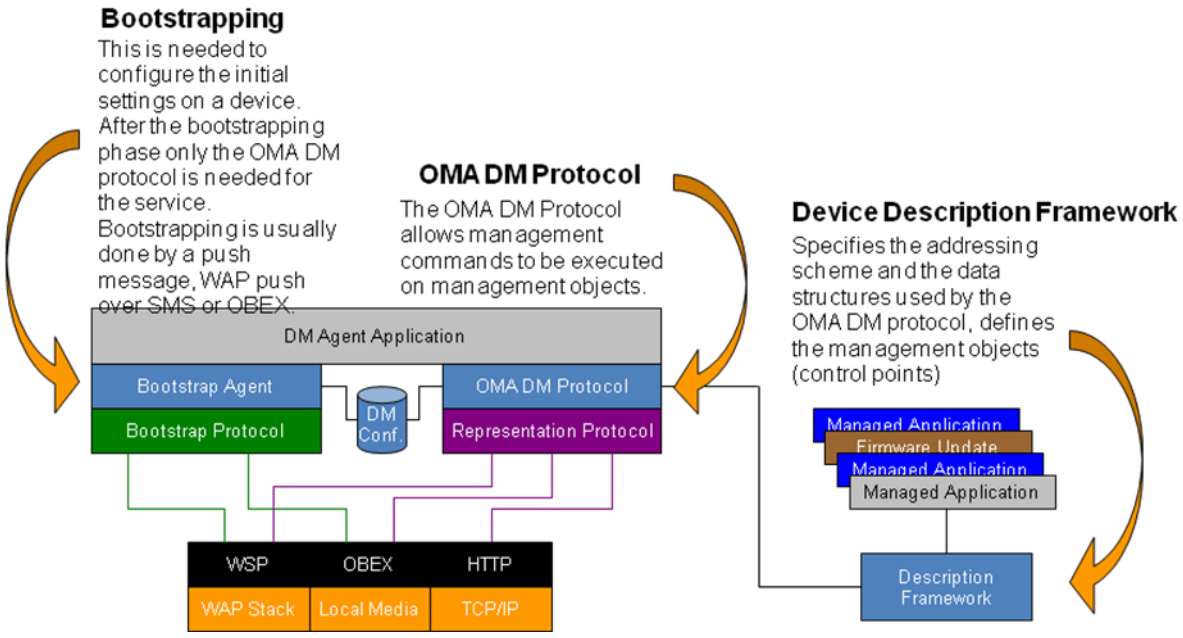

Figure 5-4: OMA Device Management, Client Architecture

The OMA DM protocol handles:

- Decoding and encoding of WBXML (WAP Binary XML),
- Authentication of device and server.

The protocol supports:

- Large objects (i.e. Firmware files),
- Control of OMA DM sequences (i.e. preserve sequence of commands contained in the DM document).

5.4.2.1 OMA-DM Security

There are two security-critical processes in OMA DM: the bootstrap process and the DM-Protocol itself. There are a number of ways to perform the bootstrap process:

- Customized bootstrap: Devices are loaded with OMA DM bootstrap information (e.g. including shared secrets) at manufacture
- Server-initiated bootstrap: Server sends out bootstrap information via some push mechanism, e.g. WAP Push or OBEX. Server must receive the device address/phone number beforehand
- Bootstrap from smartcard: The smartcard is inserted in the device and the DM client is bootstrapped from the smartcard

The bootstrap "information" is referred to as a Bootstrap *profile.* Typically the content is itself a standard OMA DM message. For the server-initiated bootstrap (which is the most common in practice), the bootstrap information is sent to the device protected by a MAC which is derived from a "SEC" parameter. This SEC parameter is a combination of the IMSI and/or a user-known PIN, and the process is vulnerable to spoofing and social engineering (tricking the user into entering a specific PIN). Further a user-entered PIN is not feasible for M2M devices. Alternatively, the bootstrap information is pre-loaded or retrieved from the SIM card for example. Bootstrapping from the smartcard takes precedence if available on the card.

The DM Protocol is then secured using credentials that are provisioned in the device using the Bootstrap *profile*. Types of credentials that can be used for DM protocol security:

- Username, password and nonce
- Server identity, password and nonce
- Certificates
- A network, transport or server specific mechanism, for example, WAP

Transport layer security: Servers must support both client and server authentication at the transport layer. Some clients may not support transport-layer client authentication. Note that the provisioning of credentials/certificates for transport layer authentication is beyond the scope of OMA DM Security

Application layer security: DM notification messages, used to trigger a client to initiate a connection to a server (e.g. over WAP push), must be secured at the application layer. Authentication is provided by including a Digest in Cred header of the DM message where:

- Digest = H(B64(H(username: password)):nonce)
- and H(X) is the result of the selected digest algorithm (MD-5) applied to octet stream X, and B64(Y) is the base64

Integrity is provided by including a MAC in x-syncml-hmac header of the DM message where

- MAC = H(B64(H(username: password)):nonce:B64(H(message body)))

- Both password and nonce are recommended to be at least 128 bits (16 random octets) in length
- MAC/Digest length not specified (except for notification messages, length 128 bits)

There is no application layer confidentiality. The use of a transport layer protocol that supports encryption (SSL 3.0 or TLS 1.0) is *recommended* where the exposure of the data to third party could have significantly negative consequences. (For example where securing the transport of credentials during bootstrapping.) Management objects may be transferred in encrypted or signed form; no restrictions are placed, since this is independent of the OMA DM protocol itself. However, OMA DM security specification describes the use of XML-encryption and XML-signature in an informative annex

5.4.2.2 OMA-DM to ETSI M2M Mapping

Several OMA standards provide building blocks that map into the ETSI M2M framework, described in section 5.7.1. Device Management can provide ETSI's Remote Entity Management service in the following manners:

1. Gateway Management Object fulfils some ETSI Gateway service requirements;
2. Management operations such as: firmware updates, software updates, provisioning, diagnostics and monitoring, are supported.

Converged Personal Network Services (OMA CPNS) maps into ETSI M2M Area Network, supporting: reachability, address mapping, inter/intra-area-network messaging, service publication and discovery. Some OMA enablers (e.g. Location) support services that can be used in M2M applications.

An overview of mapping between OMA Enablers and ETSI Service Capabilities is shown in Table 5-3.

Table 5-3: ETSI Service Capabilities – to – OMA Enablers mapping.

ETSI Service Capability	Function	OMA Supporting Enabler
Reachability, Addressing, and Repository (xRAR)	Name mapping, address translation, reachability	GwMO: device/network name and address translation CPNS: device/gateway reachability, address translation, service publication and discovery, service group management
Remote Entity Management (REM)	Configuration management, performance management, fault management, software and firmware update, diagnostics	Device Management: provides all needed functionality GwMO: provides same, but specific to gateway SCOMO: Software Component Management FUMO: Firmware Update
Network Interworking Proxy (NIP)	Interworking between gateway and devices not implementing ETSI standard	GwMO: protocol translation CPNS: Support CPNS service to non-CPNS enabled devices
Gateway Application Enablement (GAE)	Intra-M2M Area Network message transmission	CPNS: Inter/intra-personal-area-network communications
Telco Operator Exposure (TOE)	Mapping of Operator capabilities to M2M resources	Location: user-plane location services applied to M2M applications XDM: XML Document Management ParlayREST: Exposure of SMS, MMS, location capability

OMA has also identified areas where further standardization will enhance support of generic M2M implementations. The following are some of the areas:

- Device Management improvements:

- Lightweight DM protocol for M2M devices;
- DM gateway supporting M2M networks;
- DM security enhancements specific to M2M applications;
- Extend OMA DM to support M2M device;
- Network APIs addressing M2M Service Capabilities;
- Location services for mobile M2M applications;
- Messaging to M2M devices that are sleeping;
- Addressing of M2M devices lacking MSISDN.

To achieve this, it is necessary to ensure that OMA service capabilities map into ETSI M2M framework and ETSI M2M framework is consistent with relevant existing OMA enablers.

5.4.3 Device Management issues and challenges

The Device Management protocols described in previous Sections have a huge potential for implementation by operators, since they cover various groups of devices which are offered to customers. Their standardization is necessary for the interoperability purpose, which is the key to seamless maintenance and integration of devices, services and applications. DM helps operators and IT departments manage access capabilities, diagnose problems, fix and update devices over the network.

Being an important aspect of successful DM implementation, the security is particularly taken care of by the protocol standards. Standards allow the existence of a Secure Element (SE) on the device, or define alternatives for security implementation if a SE is not present. However, certain issues are obvious here. In case of cellular access network, devices usually need to be provided “ready to connect” to the end user, requiring to fit the xSIM (SIM, USIM, CSIM…) in the device at production or distribution. Production and distribution of SIM cards significantly increases the operating costs on one side, and on the other it may be necessary to change the operator credentials within the life time of the device, whereas the SIM is not accessible. This leads us to the option of implementing an embedded SIM in a device (in the device production phase), which cannot be removed, and can be securely updated with operator credentials, and securely re-provisioned by alternative operators during its life time. The research in this area is on-going, and it is discussed in more detail in Section 5.2.

Another challenge comes from the fact that a certain group of devices (e.g. sensors, vehicular network etc.) is quite simple by its architecture and operating modes. For simple devices the two functions of communicating and remotely managing the devices merge into a single one. This can be illustrated with the extreme example of a device implementing a single remote controlled switch. Both communicating with the device and remotely managing the device share the same unique goal: activating the switch. The DM can actually merge with device usage, and by working on the security of data transmission from/to devices, we also cover the subject of device management. Taking this approach into account, we can move from the centralized client-server operating mode to a more decentralized mode, which allows to any particular instance (person, application, another machine) that has security credentials to access to a device to actually manage that device.

If we assume that the environment in which a DM solution is implemented is analogous to the ETSI framework (and OMA strives to make this mapping possible), it is obvious that the role of an M2M Gateway in DM operation is significant. ETSI Gateway corresponds to the M2M Gateway in EXALTED. In an LTE-M system, a Gateway is a boundary between the LTE-M Network and capillary network(s) of various types, performing functions such as: protocol translation, network connectivity, data aggregation etc. to devices which it is “responsible” for. There are three possible communication types between Device Management Server and end devices, shown in Figure 5-5.

1. **Device Management Server ↔ LTE-M Device**

This is the simplest bidirectional communication type, but with the largest number of Device Management operations and features. We can assume that LTE-M devices (running LTE-M protocol stack) are more advanced than non-LTE-M devices (typically sensors, simple controllers etc.), and that DM server has a larger set of feasible management procedures. LTE-M interface allows a larger set of data to be sent from the DM server to the end device. These functionalities are applicable for all LTE-M enabled devices including LTE-M Relay and M2M Gateway.

2. **Device Management Server ↔ LTE-M Relay ↔ LTE-M Device**
   This type of communication is similar to the first type, with a remark that the LTE-M Relay is used for coverage extension to devices that cannot reach LTE-M network. It is assumed that a Relay is transparent for the information flow from the DM server to a Device.

3. **Device Management Server ↔ LTE-M M2M Gateway ↔ Non-LTE-M Device**
   This is the most complex type of communication, because it presumes DM protocol translation functionalities embedded in M2M Gateway. The DM server communicates with a Non-LTE-M device in two steps. In the first step, the LTE-M DM protocol is used to transfer information to the M2M Gateway. M2M Gateway has a Device Management Client that is responsible for conversion of DM protocol messages and commands to a non-LTE-M protocol that is used in capillary network behind the M2M Gateway. The non-LTE-M protocol is the second step of DM communication.

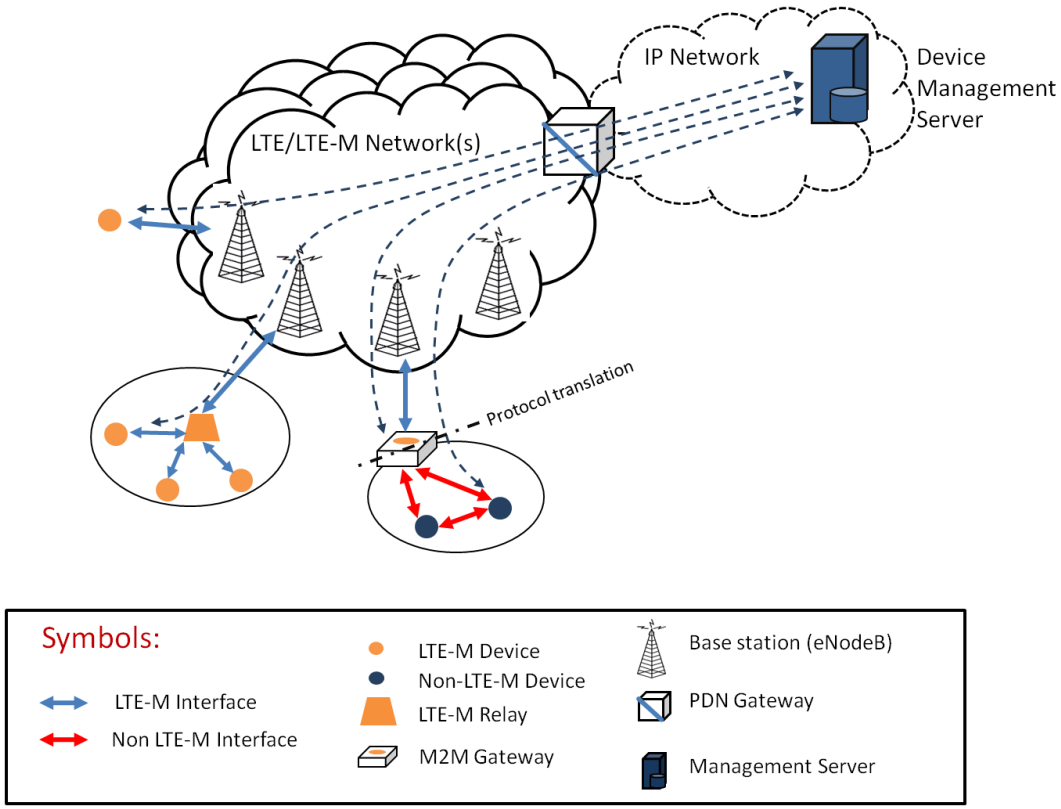

Figure 5-5: Device Management Communication Types

The third type of communication is the most interesting regarding the challenges of DM implementation. Depending on the type of devices in capillary network, set of functionalities of non-LTE-M protocol could (/should) be significantly reduced in comparison to the set of

functionalities described for LTE-M DM protocol. Sometimes the simplest commands are sufficient to change some on/off sensor state or to update sensor URI value.

In general, some of the potential gains of the Gateway implementation are the following:

- Reduced Access/Core Network Signalling Load – Device and network registration information needs to be stored in a repository, typically located in the M2M server/core. This information can be mirrored, shared, and/or coordinated with the Gateway alleviating network signalling load.
- Efficient Management of M2M Area Network devices by a Gateway allows:
  - More efficient scheduling of management of individual devices,
  - "Bulk" management of similar devices, reducing signalling in M2M area network and access/core networks,
  - Protocol translation if M2M Area Network management protocol is different from management protocol on the network side,
  - Legacy device support.
- Gateways will allow ETSI compliant service layer to interact with legacy devices through an interworking unit.
- Security:
  - Gateway permits group authentication, authorization, and registration, of M2M Area Network devices
  - Provides first level of "filtering" to prevent interaction with access and core network
- Network Selection:
  - Gateway allows selection of the optimum access network for communication to a network application.
  - Gateway allows selection of optimum M2M Area Network parameters for communication to M2M devices.

Within the scope of the EXALTED project, Device Management and its security aspects are important objectives. Many initiatives, some of which are described in previous sub-sections, are monitored and participated in, in order to obtain a full picture of the standardization process on the one side, and possibly to impact the future development with the EXALTED objectives on the other. The goal is to design a device management protocol which can deal with rather simple operating mode of devices, and to secure the communication channels between those devices and Device Management application servers, but with a certain consistency with the standard approach. Therefore, the EXALTED solutions should be backward compatible with the standard ones, but on the other hand they should provide some necessary innovations. The idea described above of merging the communication with a device and the device management is one of the examples of those novelties.

5.5 Sharing Secure Elements

One of the possibilities to lower the cost of security previously identified aims at lowering the operational cost of secure elements. To recap one of the new requirements listed above:

SE re-use	R32. A single security element shall be re-usable for a variety of credentials, including service provider as well as network access credentials. Mechanisms shall be available to securely load additional credentials (and delete redundant credentials) during the lifetime of the device.

Note in particular that this strongly suggests architecture with:

a) Embedded Security Elements, which are distributed with the device, and then remotely personalized (e.g. with a particular MNO or SP credentials, as needed)

b) Small, highly-optimized SEs, with highly-optimized provisioning processes (pre-personalized with a minimal identity + keyset, and cheaply post-personalized).

c) Re-used SEs that are shared amongst groups of low-cost devices, and can provide authentication credentials to all members of that group.

Several ways can be outlined to achieve this goal:

- Embed the secure elements inside the device, using an initial device personalization which can then be used to bootstrap the provisioning of new security keys once the device owner initiate a business relationship with an operator. This is the goal of the embedded SIM project initiated in November 2010 by The GSMA. The solution currently being outlined is described in Section 5.2 above.
- Reuse the secure elements inside the device to secure several communication layers. In addition to securing the network access layer, secure elements may be used to also secure the application traffic.
- Reuse secure elements between a group of devices.

5.5.1.1 Re-use SE to secure several communication layers

The ETSI M2M group has proposed an architecture relying upon a hierarchy of keys which are used for different levels of authentication and authorization. This hierarchy involves 3 levels of keys:

- Kmr – The M2M Root Key. This key is used for mutual authentication and key agreement between the M2M Node and the M2M Service Provider .The root key is initially provisioned in the devices and in the authentication server side (MAS)
- Kmc: the connection key is derived from Kmr, upon successful mutual authentication of the M2M Node. A different Kmc is generated for every new M2M Service Connection procedure of the M2M Node with the same or a different Network M2M Node.
- Kma – The M2M Application Key is used as symmetric shared secret for setting up secure application data sessions between the network and the device communication endpoints or between the gateway and the network communication endpoints. There may be several applications keys which are all derived from the connection key Kmc. Kma is used for authentication and authorization of M2M Applications at the M2M Device/Gateway and for protection of application data traffic. As an example, The Kma key may be used as a Preshared Key to secure the traffic of an http based application using a TLS data protection scheme.

The secure element will be involved in the definition of the Kma keys described above . In order to reach that goal, and considering that the application traffic is generated by applications executed on the device, those applications need an API to initiate cryptographic operations on the secure element. The example given above, explains that the Kma key may be used as a preshared key to secure the http connection of an HTTP based application. In this case, the communication is typically secured using a Diffie Hellmann key exchange, and the Kma key is used to authenticate this exchange and thus avoid possibilities of man in the middle type attacks. The Kma key can therefore be safely be stored in the secure element and used via a cryptographic API accessible to device applications.

However, a major question is exactly what API is exposed by the secure element to allow such application-layer operation? A number of crypto-APIs are well-established in the PC/smartcard world including the (fairly) standard PKCS#11, and the (proprietary) MS-CAPI. However, neither of these has received any noticeable traction/support in the mobile space. This means that even if a UICC comes provided with a PKCS#11 interface there will be no middleware/drivers in the terminal to expose it to applications. A specialized secure element card called the “WIM” (wireless identity module) was standardized by the WAP forum over a

decade ago, and was expected to be supported on UICC as one of its deployment modes, but this has seen essentially no usage.

A number of “lower level” interfaces are slightly more standardized, essentially channels to run arbitrary commands to the UICC (technically APDUs). It is clearly possible to build some sort of crypto interface on top of these low-layer channels. One approach is for the application layer of a terminal (or associated device) to treat the radio layer as a “modem” and send it AT commands. There are in fact particular AT command sets defined by 3GPP which allow arbitrary commands (AT+CSIM) or filtered commands (AT+CGLA) to be transmitted straight to the UICC. This model can also work where the radio module is on a device elsewhere in a capillary network, for instance AT commands can be run over a Bluetooth channel. Bluetooth also allows direct low-level usage of a UICC from a remote device through the “remote SIM access profile”. Another approach is to use Java JSR 177, which contains classes to support APDU-level access (as well as crypto-API access) to a UICC. Yet another approach is access to a UICC via a USB interface – the so-called “high speed interface” standardized by ETSI SCP.

While all these options are fairly well established in standards, actual deployment has been very spotty. AT+CSIM is fairly widely deployed in mobile terminals, though it tends not to be available to applications on the mobile terminal itself, only on tethered PCs etc. It is also not widely supported in data modems (those most likely to be used in M2M devices). High Speed Interface has essentially no support in mobile terminals or data modems, and JSR 177 only started to get some traction when Java MIDP was already going out of fashion in the mobile space. Newer execution environments like Android and iOS tend not to support similar APIs. One further difficulty is the security risks involved in exposing such low-layer functionality up to the application layer: with attack possibilities such as harvesting of authentication responses from the SIM/(U)SIM in real-time, or permanently crippling the UICC by blocking PINs and PUKs.

There is at least one “high level” API which would allow applications on a device (or on other connected devices) to talk to a UICC using http. This is the so-called Smart Card Web Server (SCWS) standardized by OMA. While attractive in concept, it is complex in deployment, involving a rather complicated UICC. Again, terminal support to allow browsers, applications etc. to actually access the UICC is very limited.

Finally (see section 6.1.6), GlobalPlatform, a cross industry, not-for-profit association identifies, develops and publishes specifications [34] which facilitate the secure and interoperable deployment and management of multiple embedded applications on secure chip. To that end, it proposes a set of technical specifications. which serve multiple actors and support several business models. In particular, GlobalPlatform enables remote management of security devices in a multi-application, multi-tenant environment, as well as defining a client side API, leveraging an Open Mobile API from the SIM alliance.

It will be seen that there is really no shortage of standards in this area, since APIs have been defined in 3GPP, ETSI SCP, WAP/OMA and JCP, the SIM alliance and GlobalPlatform. Historically, the real difficulty seems to have been the long lead-time to deployment of the standards, the general lack of interest of terminal vendors in supporting such APIs, and the fast-moving nature of the mobile application space, which tends to make APIs obsolete before they get serious usage. There is some hope that this is changing though, as we discuss in section 6.4.6 below.

5.5.1.2 Re-use SE to secure a group of devices

One approach to the scaling challenge on the network side is the “gateway” concept; the idea is that there are (relatively few) LTE-M devices acting as gateways and providing wide area access, and other devices connect to them locally via “capillary” networks. Thus the mobile network only sees one “big” device (the gateway) and can’t distinguish the many small “parts” of the device.

However, this concept has a number of problems, one of them being that forcing all traffic from the devices to be funnelled through the gateway could be very inefficient in radio terms. Also, it means that devices must be continually connected to the gateway via capillary links, something which may not be feasible in practice. Finally, it makes identifying and resolving problems with individual devices (device failed, software not updated, device compromised etc.,) much more challenging, since these individual devices are essentially invisible at a network level.

Thus it is predicted that large numbers of machines will require direct access to wide-area mobile networks (such as the GSM, GPRS and/or 3G cellular networks), and not exclusively via gateways. Each of these machines may only require authentication very occasionally but may have all the basic equipment to allow connection to at least one access network when that is required.

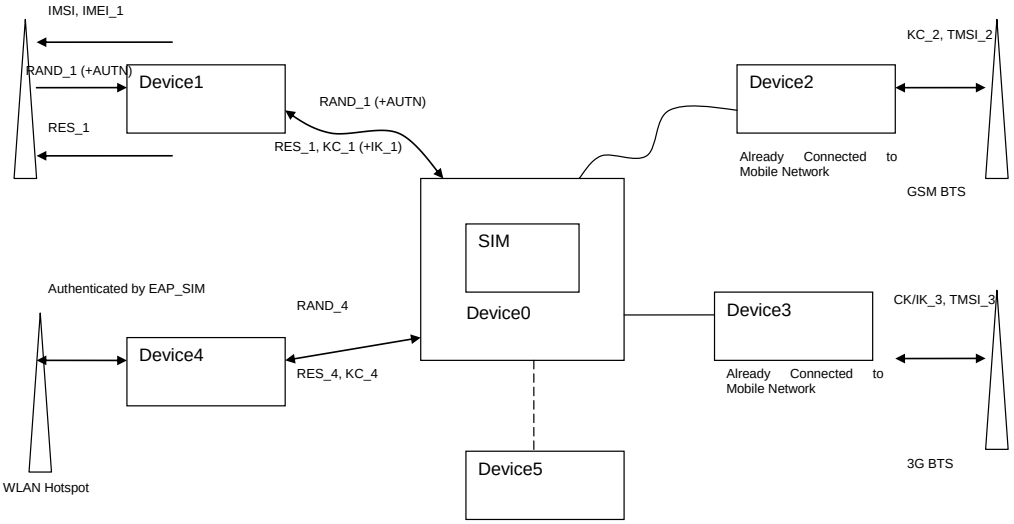

Figure 5-6: Multiple machines connected to SIM-containing device (“hub”)

Rather than provide each machine with its own (U)SIM and tolerate the level of account complexity that that would entail, we propose to facilitate authentication of multiple devices using the same (U)SIM.

Typically, as shown in the above figure, the devices are joined to a SIM-containing device (referred to hereafter as the “hub” device, to distinguish it from a “gateway” device) via a variety of short-range connections (USB, WLAN, ZigBee, NFC etc.) and/or long-range connections and secure channels. When each device needs to authenticate to a wide-area mobile network (or heterogeneous access network) it forwards a challenge to the (U)SIM and receives back a RES and key material (Kc or CK||IK).

Multiple devices can thus be connected substantially simultaneously, each with a distinct TMSI (or in LTE, a distinct GUTI, or in LTE-M a different identifier like an IP address) and key association (in LTE, K_ASME) but all related to the underlying IMSI, and billed against the same subscription.

To facilitate this behaviour in a cellular telecommunications access network (such as a GSM network, 3G network or LTE network), some changes to the HLR and other parts of the core network are required.

1. In a first approach, the HLR must track multiple devices at once, and single out a "master" device (for example, the hub device) to receive incoming pushes, SMS etc. In an alternative, the HLR may only track the "master" device, on the assumption that

the other devices never need to be routed to (i.e. they have data-only connections and there is no incoming traffic accepted). A number of mechanisms are available to indicate to the HLR which device is the "master", examples include: a special flag in the IMSI (dedicated bit) which indicates when connecting or doing location-updates with the master; or use of the IMEI which is presented at connection or location update (with a separate record indicating which device is the master).

2. The visitor location register (VLR), associated with a mobile switching centre (MSC) currently maintains only one record per IMSI, with associated TMSI and Kc (or CK||IK for UMTS). To support the above, VLR must maintain multiple records i.e. same IMSI may have multiple TMSIs (or other temporary identifiers) at once, and VLR must associate each TMSI with corresponding IMSI.

3. The HLR may maintain multiple records per IMSI, and associate each record with IMEI so it can track each device's location. This requires IMEI to be reported to HLR along with IMSI during Location Updates. This can be done using techniques such as the "Automatic Device Detection" facility standardised in 3GPP Release 6. Alternatively, where the HLR only tracks location of one device (e.g. "master" device for incoming calls, SMS etc.), location updates with the "master" device conveniently report a base IMSI (say IMSI_0) and other devices report an offset IMSI, say IMSI_0+1. The HLR then need only track updates reporting IMSI_0.

A number of example scenarios follow:

Example 1: Consider a vast array of sensors in a building or on a campus. A single SIM-holding device, to which sensors are locally connected, may be used to perform authentication on behalf of each sensor. Sensors have a low bandwidth radio (just to confirm that they are "OK" or "alert" every so often). To avoid the complexity of maintaining persistent local connections to the "hub" across the campus, the SIM-holding device is portable (e.g. it could be a security guard carrying a mobile phone), with devices only temporarily in range.

Example 2: Now consider a vast array of sensor devices in the environment outside a building, whereas the "hub" device is protected inside the building. The sensors can connect locally to the hub over short-range channels, however, using the hub as a full LTE-M "gateway" for traffic would be very inefficient from a radio perspective. All traffic from the sensors would need to be funnelled into the building (through the walls), and then re-transmitted using an LTE-M radio so that it travels out of the building again (again through the walls). It is more efficient to have some of the LTE-M radios in the sensor network outside the building, and allow them to transmit directly to the cellular network.

Example 3: In another scenario, consider sensors installed on parcels, delivery crates etc. travelling away from a depot, then back again, or between depots. They can only connect to the SIM-holding device when they are in the depot.

Example 4: In a fourth scenario, consider a home energy system with multiple devices reporting usage, adapting usage, sending alarms etc. In this case the SIM-holding device is the home owner's mobile phone; and the owner is only around in the evening.

Notice that in several of these examples, there may be many devices/machines having equipment suitable for establishing a connection with (one or more) wireless access networks but lacking a **continuous** connection with an authentication storage means (the "hub device") necessary for authenticating the device with the selected access network.

We propose therefore to allow a device to "pre-fetch" certain authentication information while (or whenever) a suitable connection to the SIM and access network is available.

As illustrated in the diagrams below, while the device does have a connection to both the SIM, and any access network, it "pre-fetches" a RAND from the HLR. (Or for a USIM it pre-fetches a RAND and corresponding AUTN). It then relays the RAND to the (U)SIM, and retrieves and stores the corresponding RES and key material (Kc or CK||IK).

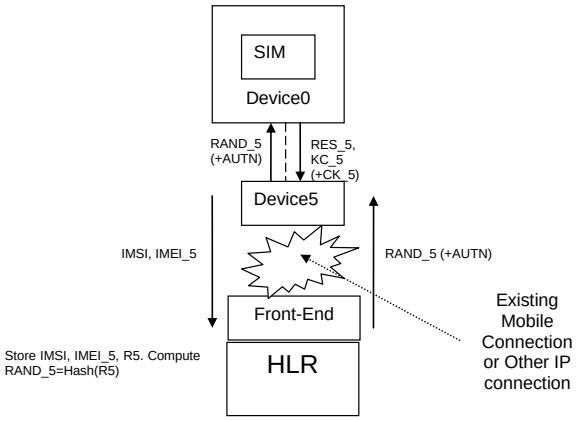

Figure 5-7: Pre-fetching an Authentication Vector

At an authentication challenge, the challenging node (base station or RNC or MME) obtains a matching authentication vector from the HLR (triplet, quintuplet or quadruplet), and the challenged device can immediately respond.

Notice that there is a risk here of false base station (e.g. since the RAND + AUTN has already been retrieved it might have been exposed). There is a further risk that the corresponding RES and key material may be hacked from the pre-fetching device before being used to authenticate with the access network.

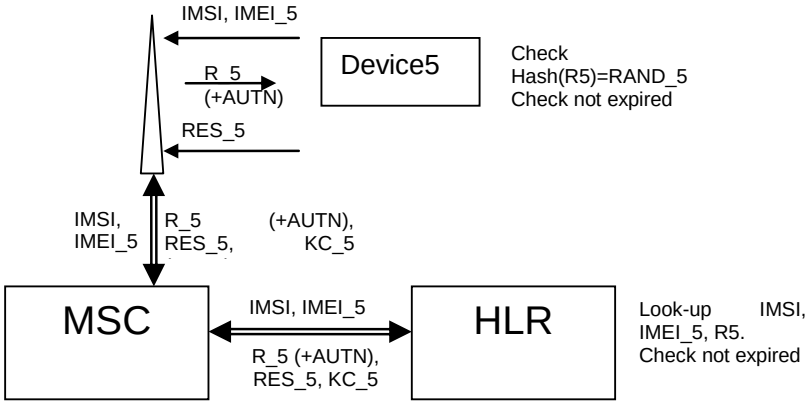

Figure 5-8: Use of pre-fetched Authentication Vector

There are several protections though:

1. The HLR may not actually provide the RAND to the challenging node (base station etc.) but rather a hash pre-image of RAND (e.g. something R with Hash(R)=RAND). The device can recognize the correctness of the pre-image by hashing it, and since the corresponding R is unknown outside the HLR, the challenging node must have fetched it from the HLR, so is probably a valid base station. While the preceding discussion relates to 3G, this also works for GSM.

2. The HLR imposes a "time out". If there is no matching request for an authentication vector within a time-out of the previous "pre-fetch" then the HLR discards the pre-fetch and the associated R.

3. The HLR uses a form of device authentication to ensure that the pre-fetching device is entitled to pre-fetch for the corresponding IMSI and is sufficiently robust. This can be achieved by establishing a suitable stack over the current access network, and running - say - an EAP method between the device and a "front-end" to the HLR. The device can then be authenticated by a device certificate, or a pre-shared key or single-use password (e.g. one set up by a previous authentication event), or by "contextual" information (e.g. declared IMEI and TMSI and some log of recent connection attempts that this device has made). The HLR can verify that the device is "entitled" to pre-fetch for the IMSI concerned e.g. because it is currently connected to the network with a matching TMSI.

There is one further variant. If the HLR is sufficiently convinced as to device robustness, or there is sufficiently low risk of fraud (data-only subscription, severe limits on volume/value of data) then the HLR can provide a complete pre-fetched authentication vector to the device, obviating the need for the device to establish a connection to a corresponding SIM card. In this case, the alternative authentication means is essentially replacing SIM-based authentication, and so would have to be used extremely carefully.

Where multiple devices use the same, common (U)SIM card for authentication, the same network adaptations as above will be needed to support “multiple devices per (U)SIM”. Either the HLR will need a pre-fetch record per IMEI (i.e. per unique terminal), or if IMEI is not presented between MSC and HLR then each device must present a different IMSI offset (e.g. IMSI_5 = IMSI_0 + 5).

5.6 Direct Modes and Local Breakout

One area of interest to EXALTED (and considered in WPs 3 and 4 for instance) is methods for using an LTE-M network which avoid traffic being routed into and then back out of a packet core. Clearly if this can be achieved, it can improve efficiency and latency, and reduce costs in the core network. Ideas along these lines have already been considered by 3GPP (under the terms “local breakout” or “direct mode”) and are at various stages of standardization. “Local Breakout” involves data traffic being routed via a base station (eNodeB in LTE) but then not into the Evolved Packet Core. “Direct Mode” involves direct transport of data from one piece of User Equipment (UE) to another, without even going through an eNodeB. This is reminiscent of “Ad hoc” mode in WLAN networks.

These approaches provide particular challenges to the security architecture, and it is not at all obvious that the challenges can be solved in an M2M context. Fundamentally, all of these solutions involve a transfer (or “collapse”) or core network functionality to the “edges” of the network, in particular the ability to route and assign a reachable address (e.g. IP address) to LTE-M devices. Assuming the security is not switched off, they also require a transfer of authentication and encryption functionality to the edges of the network: the encryption must terminate at the eNodeB or at the receiving device in case of Direct Mode.

The particular concern here is how does the “edge” device acquire sufficient credentials to authenticate the UE? If the core network has the (persistent) authentication credentials (a Ki, K or equivalent), then the eNodeB or direct-mode peer (such as a gateway) will need to fetch session credentials from the core, relay a challenge to the target UE, and check that the response matches the expected response. It must also be trusted to terminate encryption without “spying” on user traffic and to terminate integrity protection without manipulating user traffic. This need to fetch authentication vectors from the core implies a large number of devices having access to critical authentication infrastructure (essentially access to the HLR) and significantly increases the risk of fraud or denial of service. This is already a significant

risk with femtocells, and requires the eNodeB to be in a strong sense under the direct ownership or direct control of the network operator, with strong requirements on secure boot and key storage. Similar issues apply with WLAN when considering EAP-SIM authentication.

Clearly, the more devices have this access, the higher the risk, and extending access out to all LTE-M devices (including the ones which cannot be owned or controlled by a network operator) becomes infeasible.

An alternative is to move **all** the core network functionality (including HLR/HSS) out to the edge, so that the eNodeB or gateway becomes a miniature network in its own right. We call this "delegated authentication". It clearly has a cost impact (the node is becoming much more complicated) and raises a very difficult provisioning problem. How does the edge node ever acquire the persistent credentials that it will store in its own HLR? It can't issue its own SIM cards, and it probably won't have the ability to update embedded UICCs (if all edge nodes had those rights, the eUICC would become very unstable and insecure).

Presumably the edge node (eNodeB or gateway) has to do some sort of local registration or pairing on first connection of a device to the node, and successfully manage the risk of pairing with an incorrect or rogue device (see Section 5.3 above) This causes a problem though if the device ever has to move from one eNodeB to another, or from one gateway to another: mobility in this case requires "roaming agreements" between the different edge devices (each with its own HLR), and even if that is technically possible, it is commercially very difficult.

While it is not impossible that either approach (acquiring credentials from the core vs. delegated authentication) could be standardized, the challenges are very great. Some initial attempts to introduce "delegated authenticated" to 3GPP did not have a great reception.

A reasonable conclusion is that neither of these approaches is really feasible at the moment.

5.7 Application Layer Security Model

Section 6.1.3 has dealt with solution to achieve network access security. This section will deal with application level security, and will use extensively the security architecture proposed by ETSI.

5.7.1 Recapitulation of ETSI architecture

ETSI is defining 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.

Specific non IP data services (e.g. SMS, CSD, etc.) are used only in the ETSI perspective for out of band communications.

The whole architecture revolves around the notion of service capability. Service Capabilities provide functions that are to be shared by different applications. They expose those capabilities through a set of open interfaces. 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.

This approach focused on service capability is justified by the desire to come up with a restful [43] architecture that is seen as beneficial and simplifying implementations.

5.7.1.1 High level architecture

The high level architecture proposed by ETSI is described on Figure 5-9. This architecture aims at enabling a specific business actor: the "M2M service provider" to offer services facilitating M2M communications

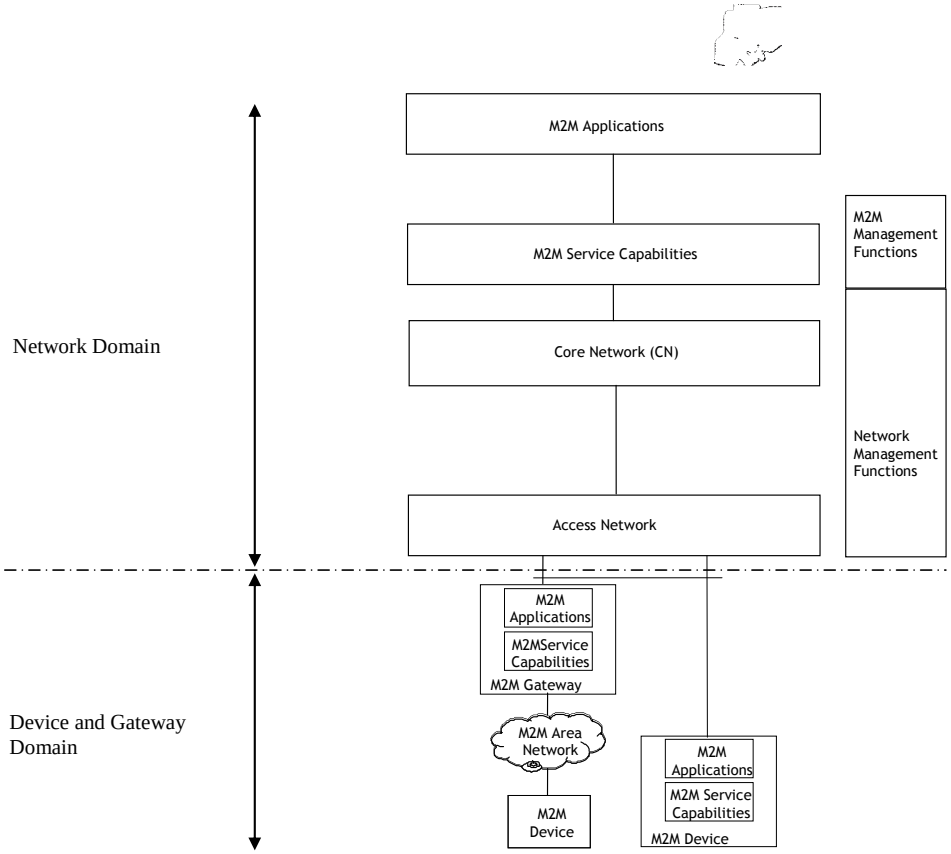

Figure 5-9 : ETSI high level architecture

The architecture includes a Device domain, a gateway Domain and a Network domain. Each domain implements generally distinct security mechanisms.

The ***Device and Gateway Domain*** is composed of the following elements:

- **The M2M Device:** A device that executing the M2M Application(s) (application may be as simple as just sending data) using M2M Service Capabilities. M2M Devices connect to Network Domain:
  - Directly**:** M2M Devices may connect to the Network Domain via the Access network. The M2M Device performs the procedures such as registration, authentication, authorization, management and provisioning with the Network Domain. The M2M Device may provide service to other devices (e.g. legacy) connected to it that are hidden from the Network Domain.
  - Using a Gateway as a Network Proxy**:** In this case the M2M Device connects to the Network Domain via an M2M Gateway. M2M Devices connect to the M2M Gateway using a local Area Network (M2M area network). The M2M Gateway acts as a proxy for the Network Domain towards the M2M Devices that are connected to it. Examples of procedures that are proxied include: authentication, authorization, management, and provisioning. Capillary networks fall in this category

- **The M2M area network**: a proximity network (local area network, personal area network, body area network) providing connectivity between M2M devices and M2M gateway (ex: ZigBee, Bluetooth, 802.11…)
- **The M2M Gateway:** A gateway that runs M2M Application(s) using M2M Service Capabilities. The Gateway acts as a proxy between M2M Devices and the Network Domain. The M2M Gateway may provide service to other devices (e.g. legacy) connected to it that are hidden from the Network Domain.

The ***Network Domain*** holds among others the following elements:

- **The Access Network:** Network which allows the M2M Device and Gateway Domain to communicate with the Core Network.
- The **core Network** which provides:
  - IP connectivity at a minimum and potentially other connectivity means.
  - Service and network control functions.
  - Interconnection (with other networks).
  - Roaming.
  - Different Core Networks offer different features sets.
- The **M2M Service Capabilities**:
  - Provide M2M functions that are to be shared by different Applications.
  - Expose functions through a set of open interfaces.
  - Use Core Network functionalities.
  - Simplify and optimize application development and deployment through hiding of network specificities.
  - Business wise, the M2M services capabilities may be offered by and M2M service provider, possibly but not necessarily distinct from the network access provider
- **The M2M applications:** Applications that run the service logic and use M2M Service Capabilities accessible via an open interface.
- **Network Management Functions:** They include all the functions required to manage the Access and Core networks: these include Provisioning, Supervision, Fault Management, etc.
- **M2M Management Functions**: They include all the functions required to manage M2M Service Capabilities in the Network Domain. The management of the M2M Devices and Gateways uses a specific M2M Service Capability. In particular the set of M2M Management Functions include a function for M2M Service Bootstrap function. This function is called MSBF (M2M Service Bootstrap Function) and is realized within an appropriate server. The role of MSBF is to facilitate the bootstrapping of permanent M2M service layer security credentials in the M2M Device (or M2M Gateway) and the M2M Service Capabilities in the Network Domain. Permanent security credentials that are bootstrapped using **MSBF** (such as the M2M Root Key,) are stored in a safe location, which is called **M2M Authentication Server (MAS).** Such a server can be a AAA server. MSBF can be included within MAS, or may communicate the bootstrapped security credentials to MAS, through an appropriate interface

5.7.1.2 Functional architecture

The functional architecture proposed by the ETSI is described on Figure 5-10.

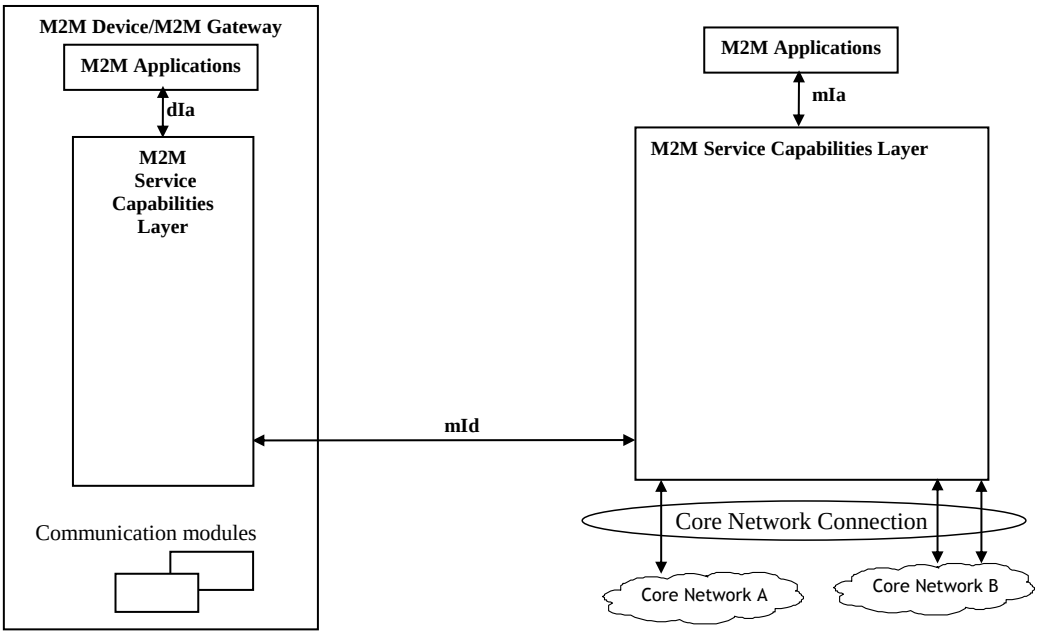

Figure 5-10: ETSI functional architecture

The M2M Service Capabilities (SCs) Layer provides functions that are to be exposed on the reference points shown on the figure.
Three distinct types of services capabilities are considered:

- **NSCL**: Network Service Capabilities Layer refers to M2M Service Capabilities in the Network Domain.
- **GSCL**: Gateway Service Capabilities Layer refers to M2M Service Capabilities in the M2M Gateway.
- **DSCL**: Device Service Capabilities Layer refers to M2M Service Capabilities in the M2M Device.

A number of procedures are defined and detailed in order to use those Services capabilities to establish M2M communications:

5.7.1.3 M2M security framework

The ETSI M2M architecture describes security at the “service capabilities layer” (SCL), as discussed above. This is **neither** strictly the network access level **nor** the application level, but rather some sort of middleware in between the network and application levels.

Furthermore the ETSI architecture allows multiple options about which ”layer” in the ISO/OSI stack is used for the SCL security. Layers 1 and 2 are considered acceptable if the end-points match (see Section 6.1). However, options to use Layers 3 to 6 (IPsec or TLS) are also defined. Finally there are options which use layer 7, such as SOAP and XML encryption/XML signature. All of this is below/ invisible to the actual M2M “application” though.

The M2M security framework describes a number of procedures needed in order to use the Services capabilities needed to establish M2M communications. Among those:

- **M2M Service bootstrapping**: This procedure involves the provisioning of permanent M2M service credentials (Kmr, as described below) which will be used for connecting and registering the device/gateway with the M2M network
- **M2M service connection**: This procedure involves mutual authentication and key agreement between an NSCL and a connecting DSCL/GSCL; the keys derived are used

to secure the communication between the 2 end points over the mid Interface as shown on Figure 5-11 .

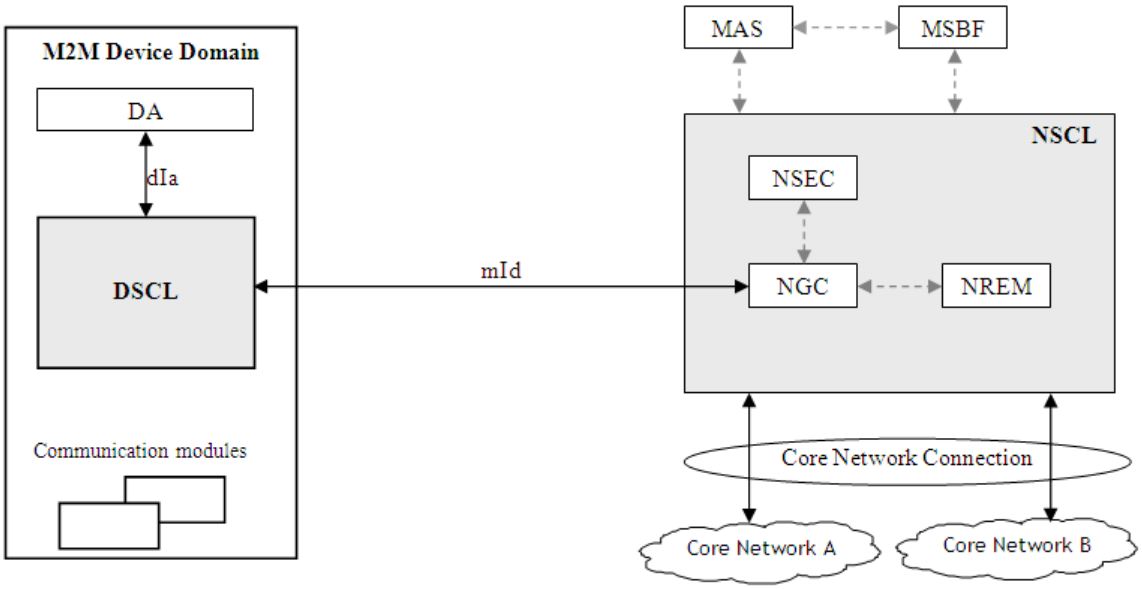

Figure 5-11: Functional architecture showing security related components

Figure 5-11 shows again the functional architecture, in the context of the security framework. In particular appear on this figure the following service capabilities and entities:

- **NSEC** : Network Security capability in the NSCL supporting in particular the following functions:
  - M2M Service Bootstrap.
  - Key hierarchy definition for authentication and authorization.
  - Mutual authentication and key agreement.

  NGC: Network Generic communications; NGC is the single point of contact for communication with the device or the gateways. It Provides transport session establishment, encryption/integrity protection on data exchanged with the M2M Devices/Gateways.
- **NREM**: Network Remote Entity Management, supporting in particular the configuration Management (CM) functions. Used to provision a set of Management Objects in an M2M Device, an M2M Gateway, a set of M2M Devices or a set of M2M Gateways. NREM covers also software and firmware upgrade of M2M Device or M2M Gateways:

The mechanisms of authentication and authorization rely upon the following hierarchy of keys

- **Kmr - M2M Root Key**. This key is used for mutual authentication and key agreement between the Device/Gateway(D/G) M2M Node and the M2M Service Provider, i.e. it is used by the D/G M2M Node to authenticate the M2M Service Provider, and by the M2M Service Provider for authenticating the D/G M2M Node. Kmr is also used for deriving an M2M Connection Key (Kmc), see below, through authentication and key agreement between the D/G M2M Node and the Network M2M Node. Kmr is coupled with a unique D/G M2M Node and M2M Service Provider through an M2M-Node-ID identifier, and ***may be bootstrapped in the D/G M2M Node in various different ways***, depending on the trust relationships between different stakeholders in the M2M ecosystem. At the Network M2M Node side, Kmr is stored in a Secured Environment within MAS.
- **Kmc - M2M Connection Key**. This key is derived from Kmr, upon successful mutual authentication of the Device/Gateway M2M Node, as explained above. Upon derivation,

Kmc is delivered from MAS Lifetime of Kmc is less than or equal to the lifetime of Kmr. A different Kmc is generated for every new M2M Service Connection procedure of the D/G M2M Node with the same or a different Network M2M Node.

- **Kma - M2M Application Key**. This optional key is used as symmetric shared secret for setting up secure application data sessions between NGC and DGC and/or between NGC and GGC, for authorized applications. Kma keys are derived from Kmc, after successful mutual authentication between D/G M2M Node and M2M Service Provider. Kma is shared between D/G M2M Node and NGC. Kma is used for authentication and authorization of M2M Applications at the M2M Device/Gateway and for protection of application data traffic.

5.7.1.3.1 M2M service bootstrap

The M2M service bootstrap procedure is used to provision the M2M root key Kmr in the Device/gateway node as well as in the MAS. The bootstrap procedure is made necessary by the fact that devices/gateways and M2M service providers do not initially share secrets.
2 types of bootstrap procedures are described in the ETSI M2M architecture:

- **Access Network Assisted M2M Service Bootstrap** where the credentials used to access the M2M network and shared with the M2M network access provider are used to create a new security relationship (materialized by the M2M root key) between the device/gateway and the NSCL (M2M service provider). Three procedures are identified in this category
  - GBA based service bootstrap. Using this method, the network access provider propose as a service to the M2M service provider to leverage network access credentials to bootstrap the generation of a shared secret (The Kmr) between the Device/gateway SCL and the Network SCL. It does not require the M2M service provider to be affiliated to the network access provider.
  - EAP based bootstrap reusing SIM/AKA credentials. This method aims at leveraging UICC credentials using EAP protocol to derive a new shared secret: The Kmr. However, it will probably require access to authentication triplets and is will be realistically useable only when the M2M service provider IS, or is directly affiliated to the network access provider.
  - Derive Kmr from EAP based network access authentication. This case differs from the previous one by the fact that there is no need to run another authentication session using network access credentials. The Kmr results from a mere key diversification operation. However, here again this scenario only apply when the M2M service provider is also the network access provider.
- **Access network independent service bootstrap:** This procedure is typically applicable in scenarios where M2M Service Bootstrap is not facilitated by the access network. This is typically the case when there is no business relationship between access network provider and M2M service provider; three methods are proposed by ETSI in this category.
  1. The first one is based upon a traditional PKI architecture and is using EAP-TLS as a bootstrapping procedure.
  2. The second is based on Identity based Encryption, a more recent development in public key cryptography. The bootstrap procedure is leveraging EAP-IBAKE.
  3. The third is a variant of the first one involving a TLS exchange over TCP

In cases one and two, PANA is used as the transport mechanism: the device/gateway and the MSBF run an EAP exchange using the NSCL as authenticator resulting in the derivation of the Kmr by the MSBF by the MSBF, from which the Kmr key is then derived. The general flow diagram of the bootstrapping exchange is shown on Figure 5-12.

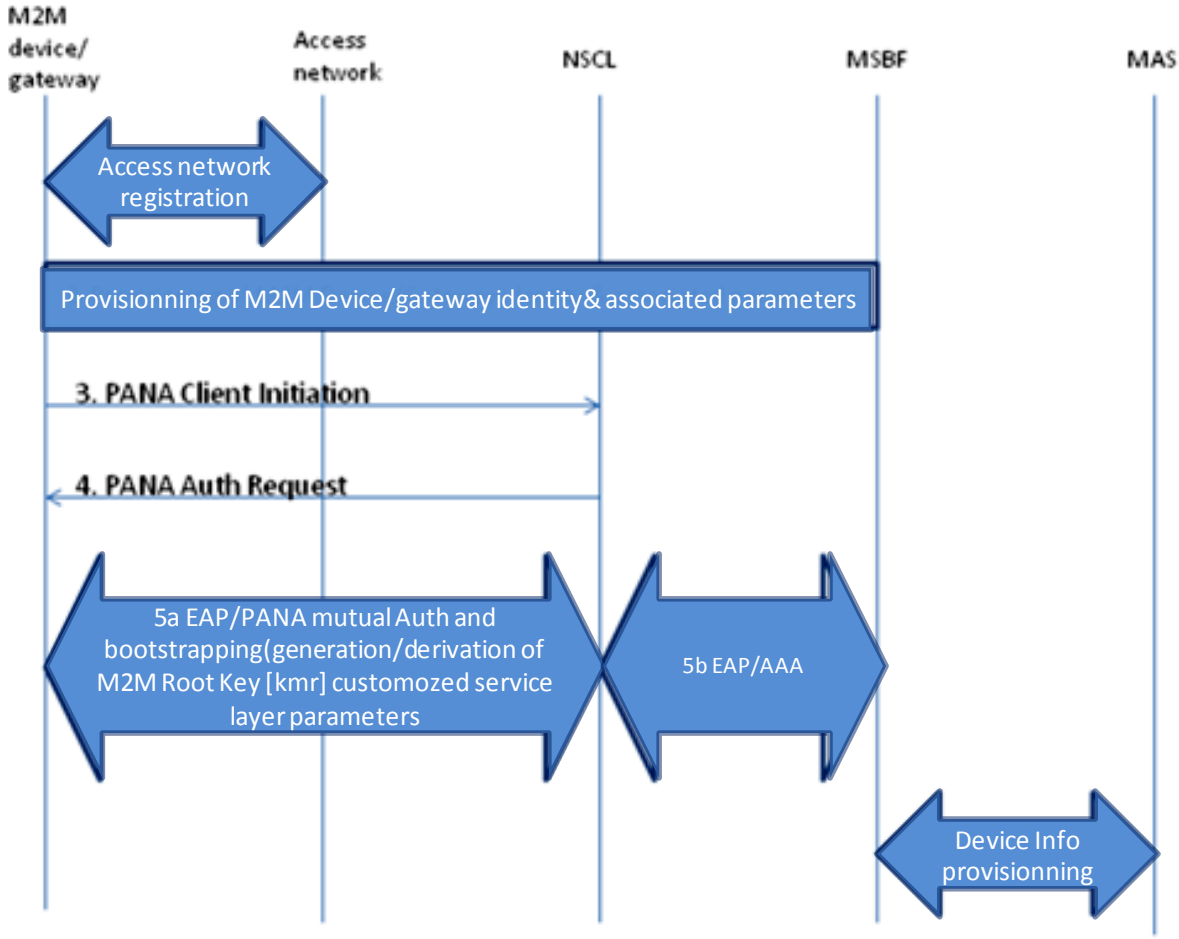

Figure 5-12 : M2M service bootstrap using EAP over PANA exchange

5.7.1.3.1.1 EAP-TLS service bootstrap

The overall exchange for an EAP/TLS service bootstrap is described on Figure 5-13. According to this workflow the device/gateway is pre-provisioned at manufacturing time with a set of private/public key as well as with the certificates of a number trusted certificates authorities enabling the verification of the certificates presented by the MSBF. The MSBF/MAS are also provisioned with suitable certificates.

The device/gateway connects to the NSCL and runs an EAP TLS exchange with the NSCL playing the role of the authenticator. The root key Kmr is computed at during this exchange independently by the device/gateway and by the MBSF/MAS. It is then securely stored in the MAS.

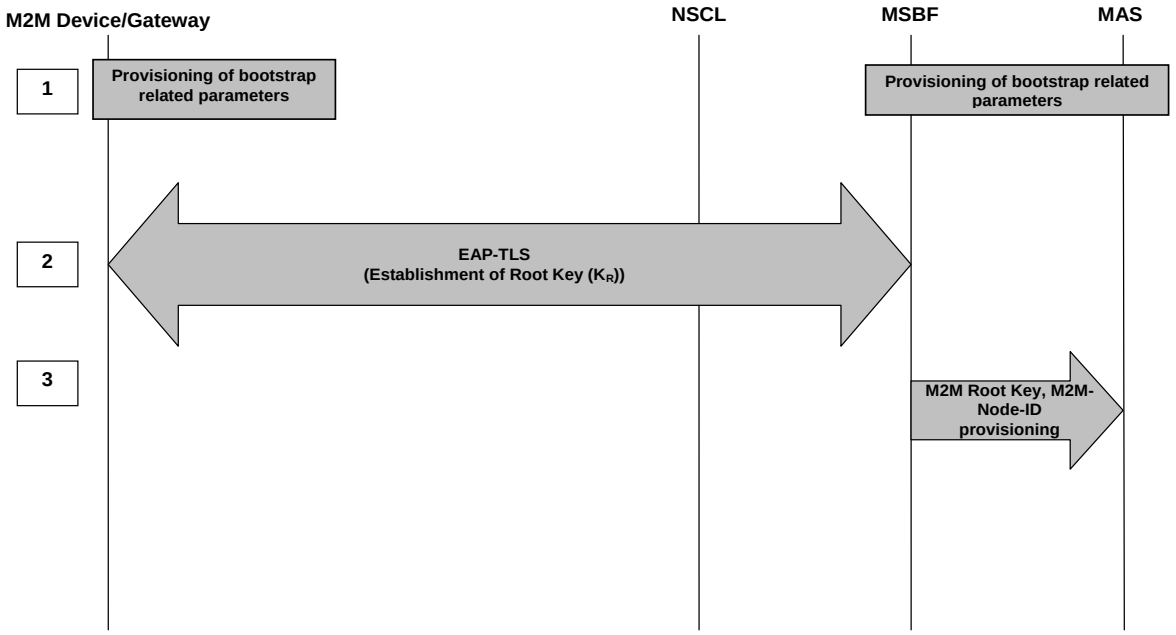

Figure 5-13 :EAP-TLS service bootstrap flow of operations

5.7.1.3.1.2 Overview of Identity Based Encryption (IBE) and IBAKE

IBE protocols have been proposed as alternative methods to public key protocols requiring the existence of PKI. The idea with IBE is that the public key is a mathematical function of the identity that is associated with this key. Therefore, there is no need for binding an entity's identity with a public key through the use of certificates, since the public key is inherently derived from the identity, using a known algorithm. IBE, messages are encrypted with the IBE public key of the recipient which can decrypt these messages, using its associated private key. The use of certificates that are managed by large public key infrastructures becomes obsolete. The only cryptographic material that is required by the message sender for IBE-encrypting a message is a set of publicly known cryptographic parameters that are used for generating IBE public keys.

The use of EAP-IBAKE to perform service bootstrap suffers however from 2 drawbacks:

- Intellectual properties rights associated to IBAKE which will probably increase deployment costs
- Relative security weakness of an IBE private key, compared to certificate-based TLS

5.7.1.3.1.3 Use of EAP IBAKE over PANA service bootstrap

The flow of operation when bootstrapping M2M service using EAP-IBAKE over PANA is shown on Figure 5-14 . This figure shows a particular way to implement the bootstrap exchange described in Figure 5-12 using EAP-IBAKE

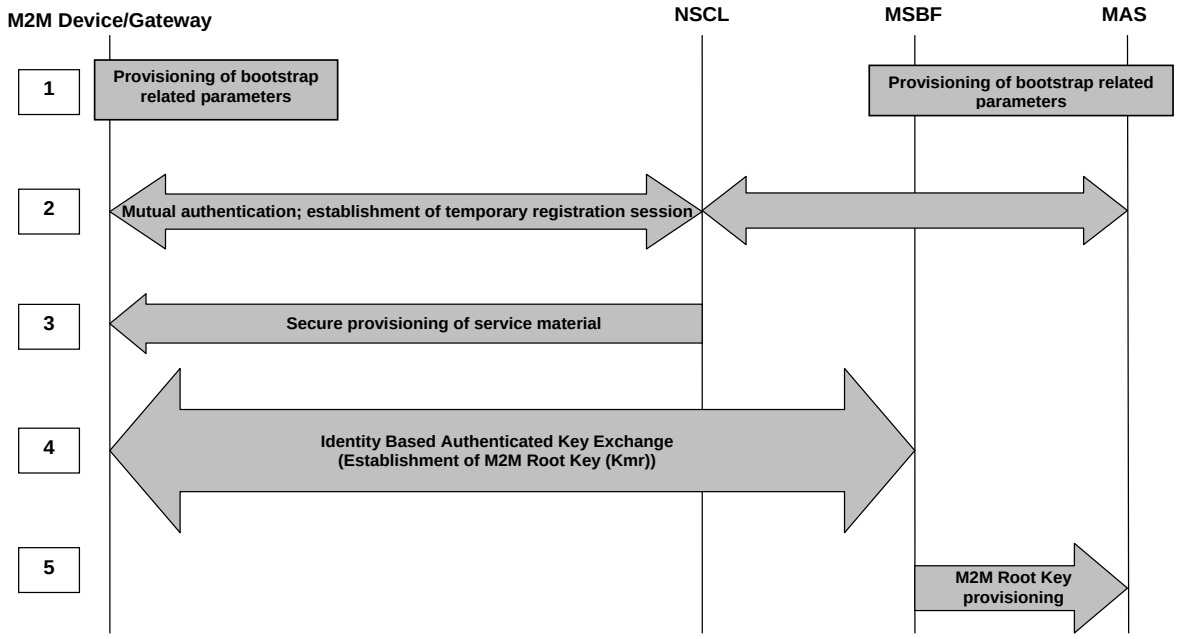

Figure 5-14 : M2M service bootstrap procedure using EAP-IBAKE over PANA

The different steps of this exchange shown in the figure may be described as follows:

Step 1: The M2M device is provisioned at manufacturing time with a temporary password and an Identity string. This password and the associated identity string is communicated to the M2M service provider in an initial step and provisioned to the MAS. This may be achieved for example via a web interface provider to the device owner through which after subscribing to the service he can register his device by entering the identity and the password.

Step 2: Using the temporary password, the device establishes an, initial connection with the NSCL, following a mutual authentication procedure using the credentials shared in step 1

Step 3: In order to perform the IBAKE procedure, the device needs to obtain a private key associated to its identity string and a set of parameters used in the IBAKE procedure. Following the initial and temporary connection to the M2M service both the private key and the bootstrapping parameters will be remotely provisioned to the device typically via the NREM (device management) service capability.

Step 4: both the device and the MSBF perform the IBAKE procedure with involves 3 exchanges. Resulting in the definition and sharing of the root key: Kmr

Step 5: the root key Kmr is kept in the device and safely stored in the MAS on the server side. The Kmr key will then be used as a permanent credential to enable subsequent connections of the device to the M2M network.

5.7.1.3.2 M2M service connection

Once the Kmr key has been defined and the M2M device registered with the M2M service, the key can be used to secure the connection to the NSCL. The service connection procedure will result in the definition and usage of the session key Kmc and possibly one or several application keys Kma.

Here again, several methods (3) have been specified by ETSI:

- M2M service connection based on EAP authentication over PANA transport. The principle of this method is summarized on Figure 5-15. The EAP messages between the M2M device/gateway are transported using the PANA protocol. The NSCL play the role of the

authenticator and encapsulate the EAP messages in the AAA protocol (Radius, Diameter...) used by the MSBF/MAS.

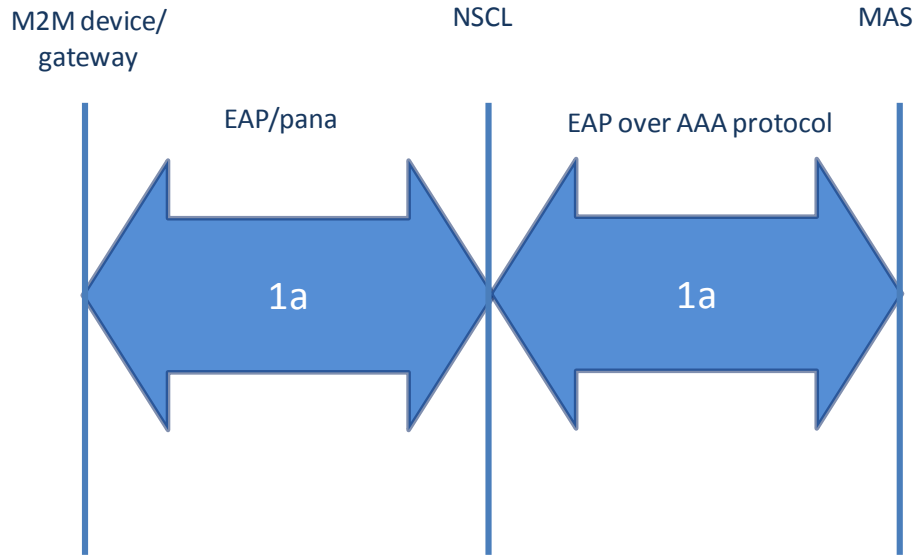

Figure 5-15 : M2M service connection using EAP

- M2M authentication using TLS-PSK. The device sets up a TLS-PSK session with the NSCL using the previously defined root key (Kmr) used as the preshared key
- Connection using the GBA. In this case, there is no need of a predefined Kmr. The definition of a shared secret between the device and the M2M service provider results from the GBA exchange. The principle of the connection is summarized on Figure 5-16. It can be seen in this figure that the NSCL implements the NAF function. In a first step, the M2M device/gateway sets up a long term shared secret with the BSF (Bootstrapping server function). This long term shared secret is used to generate shorter terms secrets which can be shared by the BSF with the NAF (in this case the NSCL).Using the GBA for service connection makes particular sense when the M2M service provider is a 3G telco provider.

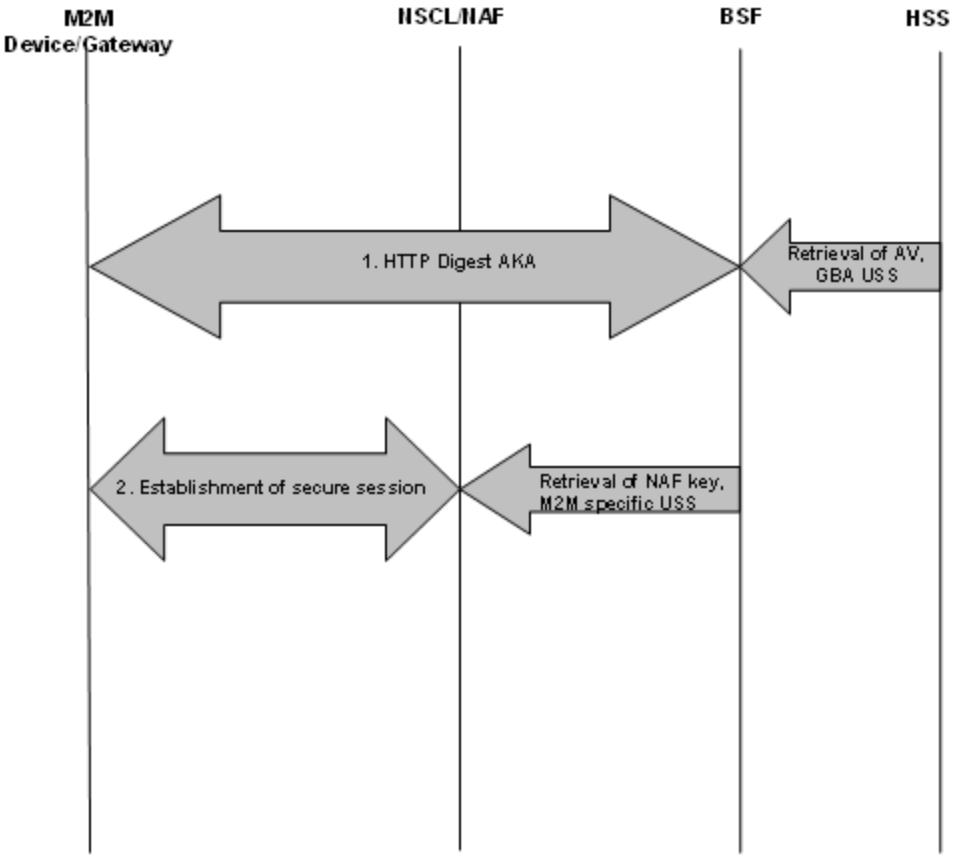

Figure 5-16 : M2M service connection using GBA

5.7.1.3.3 M2M application data transfer

ETSI has defined a comprehensive set of M2M resources management functions and procedure. Everything is defined in such a way to enable a restful [43] implementation.. This choice has deep implication on the way M2M devices, applications, Services capabilities are exchanging information with each other.

To illustrate this, imagine that one device (device1) needs to exchange data with another device (device2) via an NSCL.

Typically a bucket resource will be created in the Service capability layer. Bucket resources have specific properties and can be managed using the resource management functionalities described by ETSI.

Access rights to this bucket will need to be acquired in read/write mode by the emitting device (device1), and possibly in read-only mode by the receiving device (device2). Both devices will probably first bootstrap their security with the M2M service provider and then connect to the M2M service as described above.

The receiving device (device2) may request to be notified whenever some data is transmitted by device1. In order to transmit data, device1 will issue a write request in the bucket resource. This will trigger the NSCL to send a notification to device2 which in turn will issue a read request in the bucket resources.

This flow of operation is summarized in Figure 5-17. It can be seen that all functions involved can be provided in a restful way, while the process of setting up a data connection and using it for data exchange between devices is by nature a stateful process.

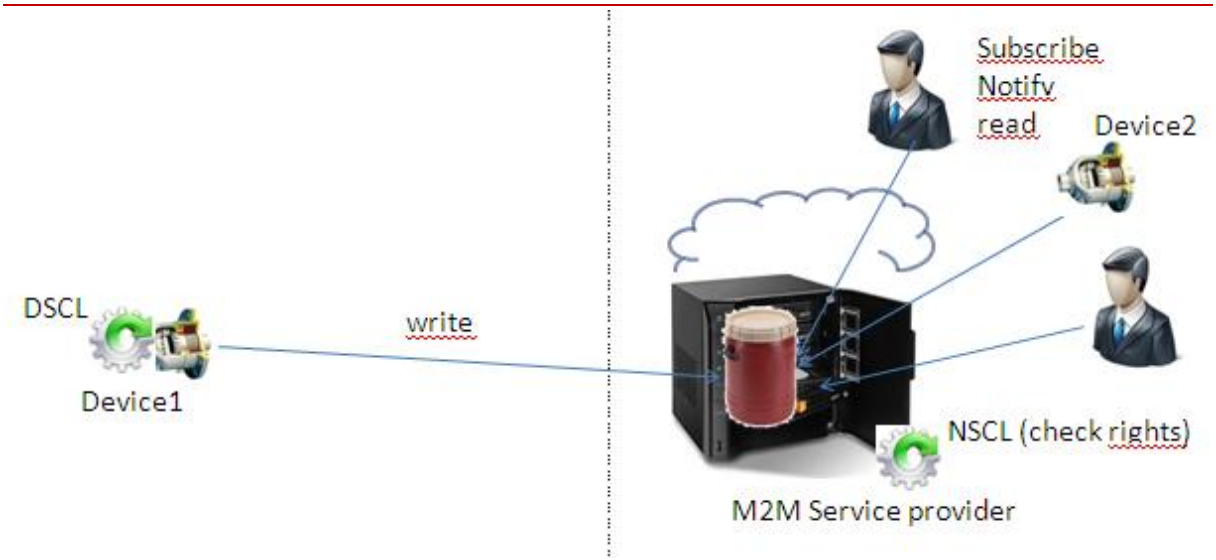

Figure 5-17 Illustration of M2M data exchange with ETSI architecture

5.7.1.3.4 Discussion of the ETSI architecture overall security model

It can be seen in Figure 5-17 that the security model endorsed in the ETSI architecture is a star hub security model. Each of the parties involved in the data communication scheme secure their connection to the NSCL. This is turn implies that each party sets up a business relationship with the M2M service provider. A more sophisticated model will imply the possibility for M2M services providers to relay requests from the M2M/device/gateways of their subscribers to other M2M providers for execution on behalf of their own subscribers.

This new “trapezoidal communication model” is illustrated on Figure 5-18. This involves a communication layer between M2M service providers which is not yet defined at the time of writing in the ETSI architecture.

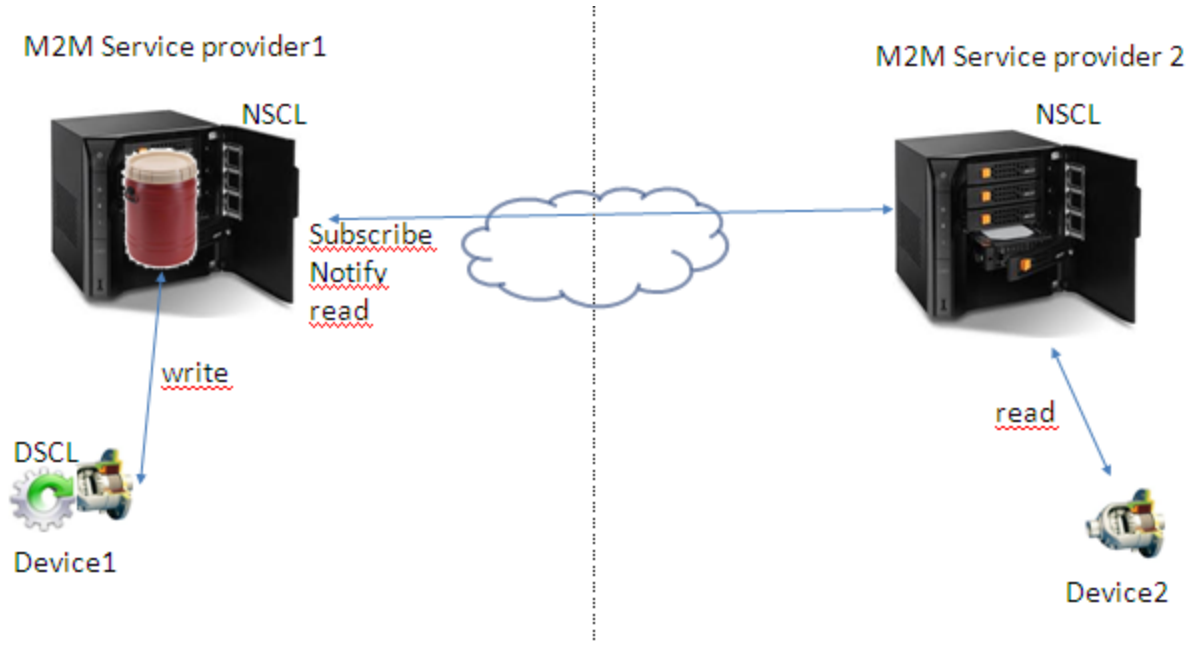

Figure 5-18: exchange between devices affiliated to different M2M service providers

Even with the trapezoidal communication model described on Figure 5-18. The security of the data transmitted is to be achieved piecewise: according to the currently defined security

mechanisms, device1 would secure its connection to the NSCL of service provider 1, while device2 would secure its communication with the NSCL of service provider 2. Service provider 1 and service provider 2 would need to secure their own communication.

This piecewise security model is fairly common in the communication world. However, in the particular case of M2M communications, the possibility to achieve an end to end data encryption may be beneficial for a number of reasons we will develop.

5.7.1.4 Business drivers for end to end data encryption

M2M service providers are supposed to provide the infrastructure components required to establish M2M communication, routing data from devices/gateways to applications, devices and gateways.

M2M service providers are not always involved in dealing with the semantics of the data they help transmitting. For example, an M2M service provider may carry data originating from health body sensors and route this data to a data processing centre for interpretation.

Trust may then be an issue when sensitive data is transmitted.

We can compare the landscape of M2M service providers with the one of 3G telco operators. While 3G telcos constitute a small population of actors generally associated to a high level of trust, the population of M2M service providers can grow much larger and entail a significantly lower level of trust. If this is so, customers may object in seeing sensitive M2M data such as vital statistics accessible to an M2M service provider not directly involved in their health management process.

Furthermore, whenever the M2M data carried over involves a privacy issue, the piecewise security model is a problem for the M2M service provider itself, who will have to prove that the security level of its infrastructure is adequate and fit for purpose. This without doubt will carry a price tag related to investments in security that would not be required if data was protected by an independent trusted party and transiting in an opaque way through the M2M service provider infrastructure.

We investigate now how the security mechanisms described in ETSI M2M infrastructure could be modified to enable end to end data encryption, with a special focus on two specific cases where M2M service bootstrap is performed using EAP-TLS or EAP-IBAKE over PANA. The idea is to be able to reuse the bootstrap mechanism and credentials that were used with the M2M Service Provider.

5.7.2 End To end data encryption

End to end data encryption involves securing data transmission at the source and up to the destination. This presupposes the availability of a secret definition mechanism which will provide all communicating parties with suitable keys to cipher and decipher transmitted data.

Methods leading to the definition of shared secrets to be used to secure end to end communications can be classified in 2 categories:

- Methods resulting from an exchange between the communicating parties using publicly available information (such as public keys), without resorting explicitly to an external trust provider. The Diffie-Hellman algorithm is a well known example of this type of method. They have been investigated in EXALTED in the scope of capillary networks communications (see section 5.3), with a possible applicability for infrastructure networks. Their main drawback lies in the fact that the communication activity related to the setup of the security is often dependent upon the number of parties involved in the communication, putting a computational and energy consumption burden on low end devices. Recall that these devices are possibly energy constrained in scenarios involving one-to-many data transmission patterns.

- Methods that will rely upon an external trust provider to help achieve “secrets” distribution.

In this later case, the “secret” distribution system may be centred on an authorisation server which has the capability to manage the rights to data access. This authorisation server can be controlled by the M2M service provider (which presupposes the trustability of this provider for identity services). Or it can also be controlled by an external actor offering identity services used by both data source and recipient.

It is also possible to combine these techniques to some extent. For example, each device could run an initial Diffie-Hellman exchange with the authorization server, and so establish a unique key shared with the authorization server. The authorization server could then use these shared keys to distribute session keys allowing individual devices (and other players) to send encrypted and integrity-protected information between each other.

When using such a hybrid method, it would clearly be desirable to minimize the number of parties which could perform an “active Man In The Middle” attack and hence attempt to recover a key shared with the Authorization Server. For instance it could be arranged that only the M2M Service Provider is able to perform such an attack. It is assumed by ETSI that it is reasonable to assume that the M2M service provider may possibly perform passive attacks upon data transmission, but that active attacks are unlikely.

The resulting communication architecture enabling end to end data encryption is shown in Figure 5-19. This architecture differs from the one shown Figure 5-17 by the presence of the authorization server which will provide both data senders and receivers with the secret keys necessary to protect the data transmitted.

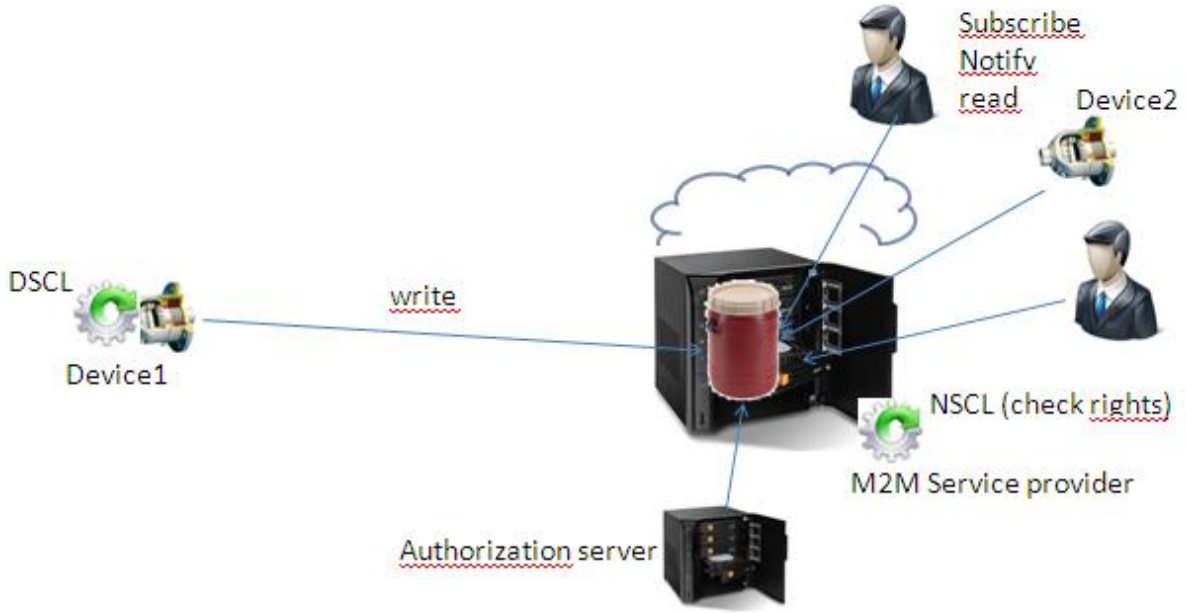

Figure 5-19 : Communication architecture for end to end data protection

Figure 5-20 describes a flow of operations leveraging the material used for M2M service bootstrap to bootstrap a new security relation between the device/gateway and the authorization server.

This security bootstrap exchange can be implemented in the ETSI architecture via the definition of a new Service Layer Capability enabling the M2M service provider to help bootstrapping an initial security association between the device/gateway and an external trust provider.

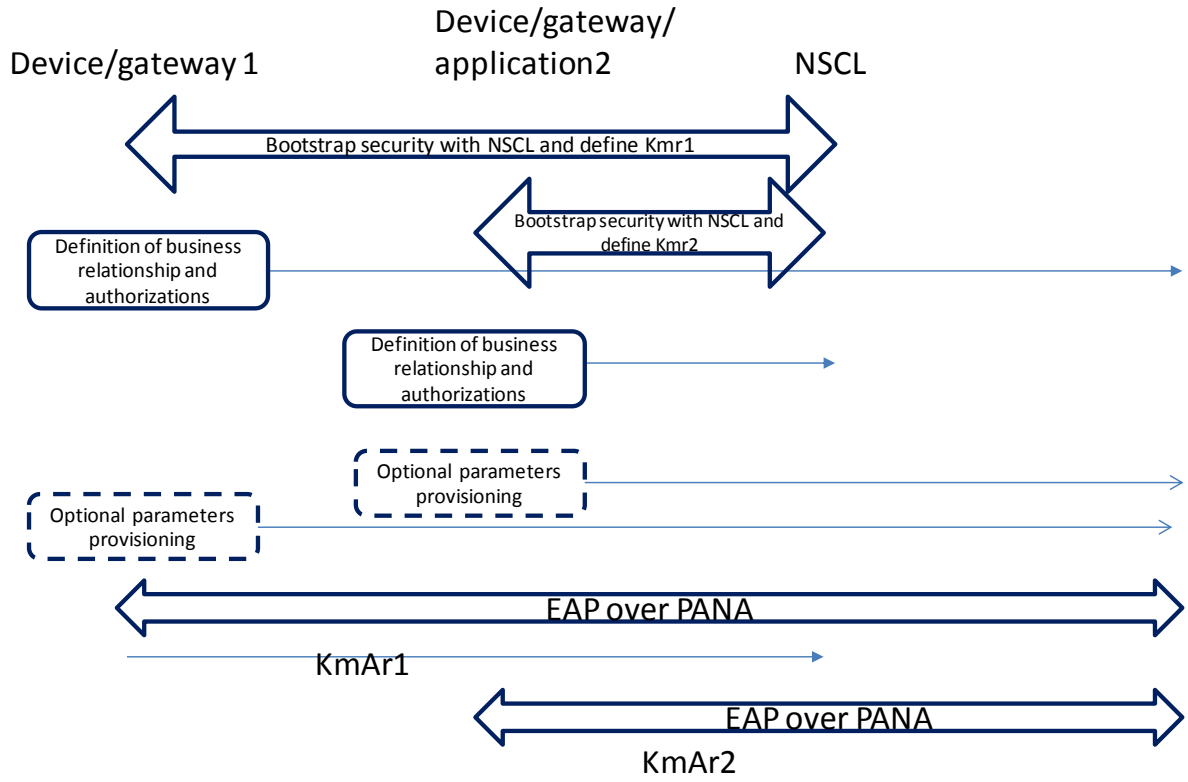

Figure 5-20 : Authorisation server service bootstrap overview

This figure depicts 2 M2M players: device/gateway1 (referred to as player1) and device/gateway/application 2 (referred as player2). For the sake of simplicity, it is assumed that the 2 players are affiliated to the same M2M service provider (triangular communication architecture).But the scenario can easily be extended to the case where player1 and player2 are affiliated to different M2M service providers. After detailing the general flow of operations, we will describe more specifically the cases where M2M service bootstrap results either from a TLA or an IBAKE exchange.

To simplify the argumentation we will also assume that player1is seeking the possibility to secure its data stream while player2 is seeking the possibility to receive player1 data.

The authorisation service will need to authenticate player1 and player2. To that end, it will either embed an AAA server functionality or communicate with an affiliated AAA server. We will assume for now that the AAA function is embedded in the authorisation server.

The flow of operation described in Figure 5-20 is as follows:

- In a first step player1 and player2 will perform and M2M service bootstrap with the NSCL using one of the methods described above. This will result in the definition of 2 root keys (kmr1 and Kmr2) shared between the devices and the NSCL (MAS)
- Player1 and player2 both setup a business relationship with the Authorisation server (which may or may not be a business entity distinct from the M2M service provider). This step may involve the owners of the associated devices or application to create a registration on the web site of the authorization server. The type of registration may be different as player1 is seeking the possibility to secure its data while player2 is seeking the possibility to access player1 data.
- Both player1 and player2 provision optional parameters on the authorization server by an out of band mechanism. It may as simple as the capture of a unique Identifier to be used in the authentication. Other parameters may be needed according to the bootstrapping method used. Again, the capture of this material may be performed by a human being

entering the information on a web interface provided by the authorization server. Alternatively, the out of band provisioning of the parameters may be achieved using an web service enabling the mass provisioning of devices for scenarios involving a large number of devices

- Using the M2M service bootstrap material, both player 1 and player 2 authenticate with the authorization server by communication through the NSCL, using EAP packets transported over PANA transport. The authentication exchange results in the definition of root keys KmrA1 and KmrA2 shared by the authorization server with player1 and player2. As the communication between the 2 players and the authorization server transit through the NSCL controlled by the M2M service provider, it is particularly important to select an EAP authentication scheme, which will insure the confidentiality of the generated root keys with respect to the M2M service provider.

The keys kmrA1 and KmrA2 are then used to distribute to player 1 and player2 a session keys of shorter lifetime KmcA which is itself possibly used to create one or more application keys kmaAx which are actually used to cipher/decipher the data exchanged between player1 and player2.

The revocation of either player can be done by voiding its root key, resulting in the impossibility for that player to obtain the next session key.

5.7.2.1 End to end data security with service bootstrap performed using MNO credentials (e.g. GBA)

A key assumption of EXALTED is that each device is able to use a cellular network, either directly (through its own radio) or indirectly (through a gateway). Either way, each device either contains a (U)SIM or has – by hypothesis – some connectivity channel towards a device containing a (U)SIM, such as another device in its capillary network.

This connectivity is sufficient **in principle** to allow **each** device (whether cellular or not) to do a 3GPP GBA bootstrap with the M2M Service Provider. Accordingly, no other pre-loaded credentials in devices are needed, and it should be assumed (in general) that none have been provided i.e. there are no pre-loaded private keys, certificates etc. installed by the device manufacturers of non-cellular devices. Still, a practical difficulty is that each non-cellular device would need to use an API to the USIM-holding device to allow it to run the GBA protocol and retrieve the derived key. Several such APIs are possible, though – for the reasons discussed above in 5.5 – it is quite difficult to use any particular one of them.

In this GBA-based model, we can assume that each device supporting an ETSI M2M mId interface has established a Kmr keyset with the M2M Service Provider using network access credentials. Thus each device is able to send encrypted/integrity-protected information over a channel to the M2M SP, with no intermediaries being able to spy on or manipulate that channel (except perhaps a very devious MNO – see below). Further, we can assume that the Authorization Server is able to use the mIa interface with the M2M Service Provider, and again this is a channel protected against spying or manipulation. We are therefore in exactly the situation described at the start of 5.7.2 above; if device and Authorization Server run a simple, anonymous, Diffie-Hellman exchange between each other, then they can agree a unique shared key, KmrAx, safe in the knowledge that the only possible attacks are from an active MITM at the M2M Service provider (or perhaps at the network access provider, if the MNO has already managed to insert an active MITM into the mId interface.)

By ETSI assumption, an active MITM at the M2M Service Provider is very unlikely. Further, if the M2M Service Provider is the same party as the MNO, or has a close association with the MNO (so allowing it to make use of GBA when bootstrapping), then the application provider should not expect an active MITM attack to be executed by the MNO against the M2M Service Provider. Alternatively, if the application provider *is* worried about the M2M Service Provider being undermined by the MNO (e.g. is worried that the MNO performs an active

MITM attack on SP traffic), then it is likely that the application provider will not use that M2M Service Provider in the first place.

Concretely, there are a number of ways that the DH exchange could be run: one option is to use TLS (RFC 2246 etc. [39],[40],[41]) with no certificates at either side (one of the TLS_DH_anon cipher suites). Another alternative is to use Preshared key TLS (RFC 4279 [42]) but with a mixed symmetric/public-key mode (one of the TLS_DHE_PSK cipher suites), and with the "pre-shared key" set to a weak shared value, such as a simple device identifier or default device password. Assuming some form of TLS is used, the actual bootstrap protocol will be very like the one defined in 5.7.2.2 below, but without any certificates.

The above solution is straightforward in connection with an M2M Service Provider who uses GBA bootstrapping, e.g. an M2M Service Provider who is closely associated with an MNO. However, we should recall that ETSI have defined other defined bootstrapping methods which never make use of network access credentials.

If the M2M Service Provider has used one of these alternative methods, then the device must have been provisioned with a different sort of credential. In that event, it turns out to be possible to re-use that alternative credential for bootstrapping with the Authentication Server, and also – in the process – even eliminate the (small) risk that the M2M Service provider hosts an active MITM. The simplest approach is as follows:

5.7.2.2 End to end data security with service bootstrap performed with EAP-TLS

We detail now the operation flow described above when service bootstrap has been performed using EAP-TLS as described in section 5.7.1.3.1.1.

In this case the M2M manufacturer may have provisioned in the device (player1) a pair of keys (private and public) as well the certificates of a number of trusted certification authorities.

Accordingly the device identity may be both indicated in the public certificate of the device and also printed on the device body. This identity will be communicated by an out of band mechanism to the authorisation server (i.e.: capture in a web page, or mass provisioning through an authenticated web service.)

Similarly, the authorisation server obtained a set of key (private and public) signed by a reputable certificate authority.

Under these conditions, the fact that player 1 and player 2 communicate through the infrastructure does not impact their ability to create a shared secret: KmrAx that will be unknown to the M2M service provider.

5.7.2.3 End to end data security with service bootstrap performed with EAP-IBAKE

The reuse of the EAP-IBAKE M2M service bootstrap material to perform a new service bootstrap with the authorisation server is more complex, and represents an additional barrier to using IBAKE.

It can be seen in step 3 of the flow described in Figure 5-13 that the M2M service provider (operating the IBE Key Generation Function(KGF)) will provision the IBE private key on the device. The knowledge of both IBE private and public key of the device by the M2M service provider will void the possibility to define a shared root key between the device and the authorisation server **unknown from the M2M service provider**.

The MSBF and the authorisation server may be operating their own KGF.

Figure 5-21 describes the IBAKE exchange taking place when performing M2M service bootstrap and authorisation service bootstrap. Before the exchange described on this figure, the device should advertise its manufacturer, and the M2M service provider as well as the authorisation server should retrieve publicly known IBE parameters associated with this

manufacturer and compute the public key from the device identity communicated via an out of band mechanism (i.e. web interface). Another option is for the device to communicate its public key provisioned in it by the device manufacturer. The IBAKE exchange itself involves 3 messages as indicated in Figure 5-21.

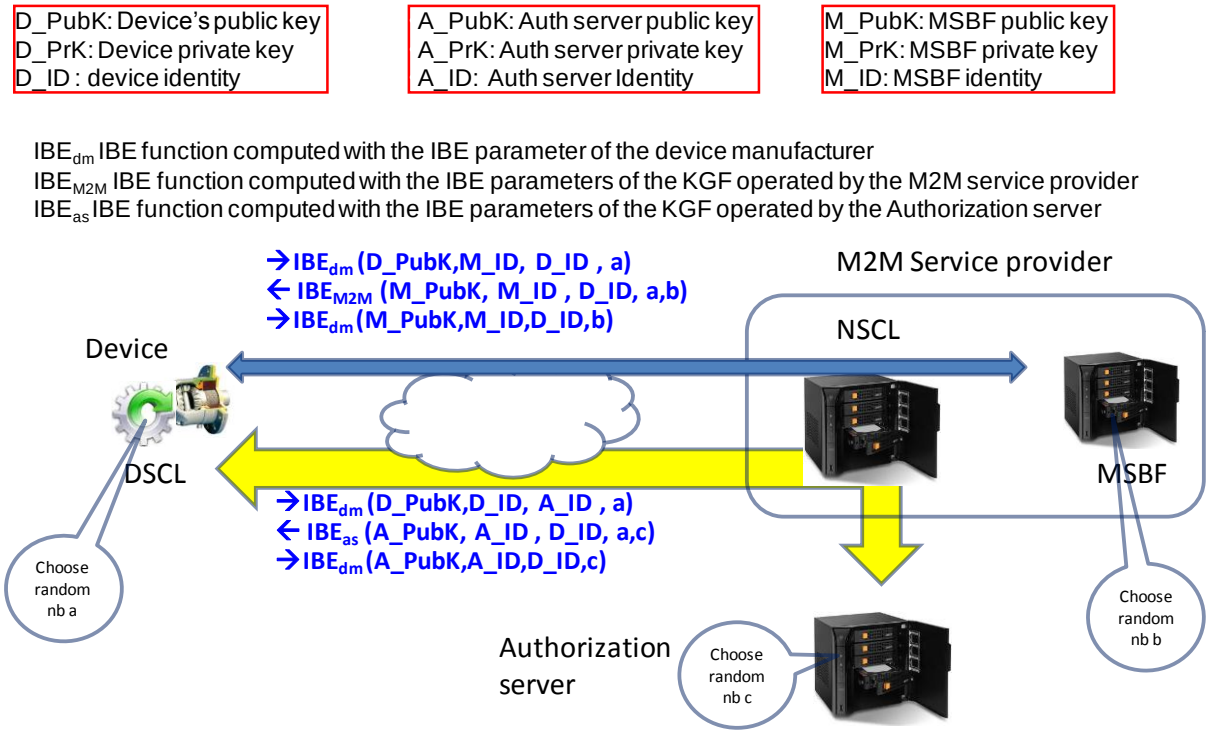

Figure 5-21: M2M and authorisation service bootstrap using IBAKE

The same IBAKE exchange is executed between the device and the MSBF and between the device and the Authorisation server. The device used the keys generated using the KGF of the device manufacturer, the M2M service provider and the Authorization server use keys generated using KGF that they operate.

6. Standards strategy

This section will provide an overview of the main standards specification related to EXALTED scope of work, particularly from a security standpoint.

6.1 Overview of Relevant Standards Groups for M2M communication

6.1.1 3GPP

3GPP has delivered a number of technical specifications (TS) as well as technical reports (TR) covering extensions to mobile networks in order better to support Machine-to-Machine traffic. In 3GPP terminology this is called “Machine Type Communication” (MTC).

3GPP Release 10 contains a document TS 22.368 “Service requirements for Machine-Type Communications” [1]. A second architecture document TR 23.888 “System Improvements for Machine-Type Communications” defines a number of Key Issues for MTC; this was presented for information in Release 10 but not approved; work is continuing on the document in Release 11 [9]. Some of the issues discussed in the document are “Group-based” optimizations (the situation when MTC devices may be managed on the network as a group) and “Potential overload issues caused by Roaming MTC Devices” (a large number of MTC devices might simultaneously change host network, depending on various factors).

The security working group (SA3) has developed a draft technical report TR 33.868 [10], which analyses the threats and security requirements associated with the Key Issues. One of the goals of the mentioned specification is to consider how possible technical solutions could address these threats. This report is defined as a “Release 11” document; however, following a prioritization exercise, a number of the Key Issues will not be addressed until Release 12.

Regarding security requirements for MTC, according to TS 22.368 these include that MTC optimizations shall not degrade security compared to non-MTC communications.

TS 22.368 also defines the MTC Feature Secure Connection which is intended for use with MTC Devices that require a secure connection between the MTC Device and MTC Server.

SA3 has identified various threats and security requirements derived from key point issues. One of those is MTC device triggering, where the main threat lies in a false network attack. This threat is more serious for MTC devices compared to non-MTC devices. The second key issue, identified by 3GPP SA3 TS 22.368 is Group Based Optimization. It allows grouping of MTC devices so that operators are saving resources. The threat here is that attacker can act as a MTC belonging to the group itself. Regarding the security requirements here it is essential that each MTC inside the group has the same attributes belonging to the MTC user.

SA3 also defines Secure Connection as important key issues although the actual encryption of data between the MTC Device and MTC Server is due to happen at application layer.

Further on, regarding Security of Small Data Transmission, defined in TR 22.368, it is required that in situations where the MTC Device is detached and no security context between the MTC Device and the core network is available, it has to be addressed.

The total signalling load from large numbers of Machine Type Communication (MTC) devices is a concern, especially in the security terms. This happens when an application requests many devices to do "something" at the same time; and/or when many MTC devices are roamers and in the situations where their serving network fails, then they can all move onto the local competing networks, and potentially overload the network(s). The way to prevent congestion is those networks nodes are able to reject attach requests. The aim is to prevent the MTC which causes the problem and not all MTCs. This can be dealt with low priority indicator, according to TS 33.102, TS33.401, TS 23.060 and TS23.401.

TR23.888 defines a solution for Time Controlling. It is essential that all information's regarding time interval and communication window must be integrity-protected when sent to MTC device in order to prevent possible abuses in terms of limiting or extending the given window or time.

An important issue raised by SA3 is also a question of MTC Monitoring. It reflects on situations when it is of a crucial interest to secure mechanisms which would enable tracking in terms of a location for MTC Devices (some MTC Devices must not be moved from curtain area, an example is MTC Device used for water metering in curtain home). In such cases it is required from the network to provide a location management mechanism for MTC Devices that should not be moved from an authorized location.

From the network perspective, in general, a highly important key issue is the question of Reject Messages without integrity protection. If the Reject message is sent without integrity protected, any false base station can fake the MM/GMM/EMM reject cause values such as "IMSI unknown in HSS", "illegal ME", or "PLMN not allowed" in the Reject message and this is treated as a denial of service attack.

6.1.2 GSMA Embedded SIM Task Force

The GSMA has formed a task force of mobile operators to explore further development of an embedded SIM that can be remotely activated. It is expected to enable the design of exciting new form factors for mobile communications and to speed the development of M2M services by making it easier to bring mobile broadband to non-traditional devices such as cameras, MP3 players, navigation devices and e-Readers, as well as smart meters.

Example use-cases for the Embedded SIM are: set-up subscription for a number of connected M2M devices (automated reading of utility meters, household security camera, automotive services etc.), set-up for consumer electronics devices (tablet PC, personal navigation device etc.).

The main security challenge is to securely provision the MNO unique key and authentication algorithm needed for chargeable telecommunication services.

A basic assumption is that the SIM card is the MNO network presence in the device, and this function should be preserved.

The overall security must be at least equivalent or higher compared to current removable SIM cards, processes and OTA management.

The preferred method is to encrypt the subscription key under a root key shared by the SM and the embedded SIM which allows MNO to choose the subscription key which can be installed using OTA.

Provisioning should be done by using secure packets in the OTA mechanism [TS 102 225] through Internet (wired or wireless) or local backhaul (NFC etc.). The provisioning of the first MNO key with IMSI can be done over the wire or over the air.

The GSMA-led task force comprises a group of leading technical experts drawn from operators including AT&T, China Mobile, Deutsche Telekom, France Telecom Orange, KT, NTT DOCOMO, SK Telecom, Telecom Italia, Telefónica, Verizon Wireless and Vodafone.

All of them are working closely with different SIM manufacturers.

The task force will analyse market trends and use it to define a technical solution as an evolution of the current SIM provisioning mechanisms. The technical solution must be built on the principles of openness and standardisation. Devices featuring the new SIM activation capability are expected to appear in 2012. Traditional SIM-supported devices will continue to work on existing networks.

6.1.3 ETSI

The job of the ETSI Machine to Machine communication group is to define an architecture which uses an IP capable underlying network including the IP network service provided by 3GPP, TISPAN, and 3GPP2 compliant systems to enable secure Machine to Machine communications and applications. ETSI defines security at a service capabilities level.

The Network SECurity (NSEC) capability in the M2MService Capabilities of the Network and Application Domain provides 3 functions:

- Authentication and Service Key Management
- Root Key Provisioning (KR)
- Device integrity validation

Authentication and service key management performs M2M service layer registration trough authentication and performs service key management between M2M Device/GW and NSEC.

In M2M world there is a hierarchy of keys which are used for different levels of authentication and authorization.

According to ETSI recommendations there are three basic types of keys: The root key Kr (derives the service trough authentication and key agreement between the M2M Device/Gateway and the M2M Service Capabilities at the M2M Core), The service Key Ks (derived from Kr, used for generating application keys), and The M2M Application Keys Ka (derived from previous key, unique key for each M2M application).

Kr is stored in M2M Authentication Server (MAS) and there is only one root key for each set of M2M device/gateway credentials to be provisioned. Ks is delivered by MAS and it is stored in NSEC on the core side, while it is stored in DSEC or GSEC on the device side. Application key Ka is used for authentication and authorization of M2M Applications at the M2M Device/Gateway. ETSI proposed key hierarchy in order to permit cryptographic separation between keys.

ETSI defines a few possible scenarios regarding Root Key Provisioning. It is possible that the M2M Device/M2M Gateway is already equipped with pre-provisioned keys inside a Universal Integrated Circuit Card (UICC) that is used to store security credentials for both the access layer registration, and the service layer registration (for both of this it is possible to have different root keys). The second scenario is that the M2M Device/M2M Gateway may use a UICC for access layer registration, but when the device treys to register on service layer for the first time it will use different key than key provisioned in the device itself. The third method of provisioning includes the scenario where the M2M Device and the M2M Gateway are different entities, which can use UICC or automated method for Kr device provisioning (M2M Gateway may use a UICC/UICC card or automated procedure for bootstrapping/provisioning the root key). Here, the M2M Gateway is responsible for access layer registration and the M2M Device is responsible for M2M service layer registration.

NSEC enables validation of the integrity of MEM Device or M2M Gateway. It triggers NREM to upgrade firmware or software of devices and gateways.

GSEC (GW Security Capability), according to ETSI, performs various functions such as key management (obtains a Ks, generates Ka), M2M GW Service Registration as well as management (service layer registration, integrity verification, specific security classes negotiation) and Application authentication.

DSEC (Device Security Capability) is performing following functions: Key management (obtains a Ks, generates Ka), M2M Device Service Registration and Device Application Detection.

ETSI security requirements are derived from ETSI TS 102 690[8].

6.1.4 Certification Standards

There are two essential standards for security certification:

- FIPS 140-2 (Federal Information Processing Standards), derived from US government computer security standards.
- CC EAL (Common Criteria for Information Technology Security Evaluation / Evaluation Assurance Level) is an international standard (ISO/IEC 15408) for computer security certification.

FIPS 140-2 defines four levels of security, simply named "Level 1" to "Level 4". It does not specify in detail what level of security is required by any particular application. For example, FIPS 140-2 Level 1 is the lowest and imposes very limited requirements and level 4 makes the physical security requirements more stringent.

Each EAL corresponds to a package of security assurance requirements which covers the complete development of a product, with a given level of strictness. For example EAL 1 is the most basic (and therefore cheapest to implement and evaluate), EAL 4 is used for smart cards with application, EAL 5 is used for smart cards and EAL 7 is the most stringent (and most expensive).

6.1.5 ISO/IEC 29192

ISO/IEC 29192 is a multi-part International Standard that specifies lightweight cryptography for the purposes of data confidentiality, authentication, identification, non-repudiation, and key exchange. Lightweight cryptography is suitable in particular for constrained environments. The constraints selected by ISO/IEC SC27 WG2 are: Chip area, energy consumption (cycles, bits per cycles, power, energy, energy per bit) , program code size and RAM size, communication bandwidth, and execution time.

The purpose of ISO/IEC 29192 is to specify standardized mechanisms which are suitable for lightweight cryptographic applications including RFID tags, smart cards (e.g. contactless applications), secure batteries, health-care systems (e.g. Body Area Networks), sensor networks, etc.

Part 1 (general) of ISO/IEC 29192 sets the security requirements, classification requirements and implementation requirements of mechanisms that are proposed for inclusion in subsequent parts of ISO/IEC 29192.

Part 2 (block ciphers) of ISO/IEC 29192 specifies block ciphers suitable for lightweight cryptography, which are tailored for implementation in constrained environments.

Part 3 (stream ciphers) of ISO/IEC 29192 specifies keystream generators for stream ciphers suitable for lightweight cryptography, which are tailored for implementation in constrained environments.

Part 4 (mechanisms using asymmetric techniques) of ISO/IEC 29192 specifies three lightweight mechanisms based on asymmetric cryptography: a lightweight asymmetric identification scheme, an asymmetric mechanism for authentication and key exchange, and is an identity-based signature scheme.

The WG2 of ISO/IEC SC27 is considering the possibility to launch a new standard dedicated to lightweight hash functions.

6.1.6 GlobalPlatform

GlobalPlatform identifies, develops and publishes technical specifications and market configurations which facilitate the secure and interoperable deployment and management of multiple embedded applications on secure elements. In particular, GlobalPlatform specifies the protocols used for remote provisioning of secure elements in order to enable a multi-

tenant application environment, as well as the protocols used to communicate remotely with the applications installed on the secure elements.

The SIM alliance released its Open Mobile API to standardize a controlled link between a mobile device application and a SE application. The specification creates a common API across multiple operating systems, which will facilitate the access to secure services hosted in a secure element, GlobalPlatform is building on this foundation to ensure only authorized applications can communicate with applications hosted in the SE in this manner.

6.2 Overview of standardization status for M2M device management

Device Management (DM) standardization activities is an important topic in telecommunication industry. It is responsible for the specification of protocols and mechanisms that manage device software versions and also the necessary configuration to access services. In addition, it defines some management features in devices such as setting initial configuration information, processing events and alarms, retrieval of management information and others. We present below an overview of the current activities in several standardization bodies about DM-related technologies for M2M scenarios.

6.2.1 ETSI M2M

ETSI M2M created a separated group (WG5) for dedicated discussions of how to adapt the current specification TS 102 690, responsible for defining the M2M Communications Functional Architecture, in order to reuse existing protocols for Device Management. The group main objective is to define and specify a generic management model for ETSI M2M Devices, Gateway and Core (D/G/C) that allows relevant management authorities to perform management tasks.

The selected protocols for re-usage in ETSI Architecture for device management were both OMA DM 1.3 and TR-069. The group is also finishing two specifications (ETSI TS 101 404 and ETS TS 101 405) responsible for the definition of 3 ETSI M2M resources in OMA and BBF formats, respectively. Publication of technical work is scheduled to Q1/2012.

6.2.2 OMA-DM

In OMA, the DM work group is responsible for the standardization of Device Management activities. The following enablers are important contributors to M2M technology:

6.2.2.1 OMA DM 1.3:

This enabler is the backward compatible evolution of OMA DM 1.X. In addition of editorial and bug-fix corrections, the group introduced new features to the specs, such as a new mechanism for management authority delegation, an extensible notification message structure, Bootstrap from Smartcard mandatory, Client Initiated Bootstrap and SHA256 authentication scheme.

Its final “approved as stable” specification (Candidate) is forecasted to be accomplished in Q1/2012.

6.2.2.2 OMA GwMO v1.0:

In some Device Management scenarios, a DM Gateway plays an important role in the management of the end devices by a DM Server. This Work Item aims to define a management object (MO) to be implemented in a DM Gateway. Special considerations in the GwMO will allow it to be implemented in M2M scenarios. The main reason for this is that some end devices may need to be managed via a Gateway in certain circumstances (e.g. Firewall, devices with non-routable addresses, etc.).

In order to manage devices via a DM Gateway some special considerations must be defined, such as the inventory of the devices behind the gateway, fan-out commands and response aggregation from devices to the gateway. The approved as stable" specification (Candidate) is on-going and must be accomplished in Q1/2012.

6.2.2.3 OMA Lightweight M2M v1.0:

This work item, proposed by China Mobile, ZTE, Huawei, China Unicom and Intel, attempts to simplify M2M service implementation by using a new "lightweight" protocol in service layer to address devices with low processing capabilities. In order to achieve this simplification, this protocol will use a Binary based addressing scheme.

In addition, it aims to support various secure models to fulfil different use cases and also to support a suite of transports to address different deployment environments. Although the enabler's development is progressing slowly and it is delayed according to initial schedule, its final "approved as stable" specification (Candidate) is still officially forecasted to be accomplished in Q4/2012.

6.2.2.4 M2M Device Classification v1.0:

The scope of the M2M Device Classification Whitepaper v1.0 is to provide a M2M classification framework based on the horizontal attributes (e.g., wide/local area communication interface, IP stack, human I/O capabilities, UICC) of interest to communication service providers (CSPs) and M2M service providers (MSPs), independent of vertical markets.

It can help not only to clarify the scope and requirements of M2M related work in OMA, but also articulate its linkage with all other M2M related standard development organizations (SDOs) to avoid overlapping and facilitate coordination related to M2M. In addition, it contributes to integrate M2M services and management across the boundaries of vertical markets. Its final "approved as stable" specification (Candidate) is forecasted to be accomplished in Q2/2012.

6.2.2.5 OMA DM NG v1.0:

OMA DM NG v1.0 (also called DM v2.0) has defined the clear scope to develop a non-backward compatible enabler to support all existing DM 1.X functionalities with some simplifications in the protocol and using RESTful transport approach. Targeting M2M scenarios was not originally mentioned or explicitly considered in DM NG. However, the usage of a RESTful based architecture may provide an easy usability of this protocol in M2M markets. This type of architecture, may allow a possible implementation of a DM Server to manage also M2M device applications (DA) due to the RESTful-based work in ETSI M2M (as previously mentioned) to manage applications on Devices. For this reason, this enabler must be closely monitored.

The activities in this group are moving slowly due to other priorities in OMA-DM, such DM 1.3 and GwMO v1.0. The realistic forecast for the "approved as stable" specification (Candidate) is set up to Q1/2013.

6.2.3 OMA CPNS

The goal of the OMA Converged Personal Network Services (CPNS, [4], [5], [6]) enabler is to provide application-layer support for converged-network Services, including end-to-end management of Service sessions, Service publication and discovery, remote management of user's Personal Network Equipment (PNE) configuration data, security and charging. This enabler considers the interfaces and interactions between the key entities of the CPNS Enabler:

- CPNS Server: an entity of CPNS Enabler that replies to requests from PN GW and ensures that the appropriate application is selected and appropriate content is provided to the PNE(s).
- PN GW: A Personal Network Gateway serves as an intermediary entity between the PNE(s) and other networks that forwards the requests from the PNE(s) to the other networks and the other way around. The user's handset can take the role of a PN GW.
- PNE(s): are PN entities that are connected to the PN GW and between each other and are used for rendering the content received from the PN GW or from each other.

The CPNS V1.0 specification is for public review and testing until December 2012. The new CPNS Version 1.1 objective is to extend the target usages, enhance the guidance for implementation, strengthen the interoperability. CPNS v1.1 will for instance:

- Support non-CPNS capable devices via a Non-CPNS Device Proxy,
- Allow interworking with ETSI M2M via Service profiles: The M2M device can provide a service or application profile to CPNS enabler as part of the registration process, so that PN GW and CPNS Server are aware of M2M Device needs and requests. Furthermore the CPNS v1.1 will facilitate the mapping of M2M GW functionality onto PN GW.
- Have the responsibility for providing the categories of supporting devices (e.g. differentiate a health sensor from a car navigation device etc…)
- Strengthen the capabilities handling the personal network and service group
- And provide implementation guidelines

The target date for a public release is expected for August 2013.

6.3 Critical Timelines and Influence Points

3GPP is currently progressing on Release 11, though the Stage 1 documents (requirements) have already been frozen (in September 2011). The Stage 2 (architecture) freeze is tentatively scheduled for March 2012, so this is the probable cut off for modifications to TR 23.888, "System improvements for Machine-Type Communications (MTC)", if we wish them to be approved in Release 11. However, note that there is only one further SA2 face to face meeting scheduled before the cut off (6-10 February 2012 in Vancouver).

The Stage 3 freeze (which will affect TR 33.868, "Security aspects of Machine-Type Communications") is tentatively scheduled for September 2012. However, this document is only a (non-normative) technical report, and any actual specification changes advised by the report will need to be incorporated as CRs into normative TS documents, which may well require liaisons to other 3GPP working groups. SA3 have face to face meetings scheduled for February, April, May, July and October 2012.

The full 3GPP schedule, including Release 12 dates (when agreed) is available at http://www.3gpp.org/Releases and http://www.3gpp.org/3GPP-Calendar.

ETSI M2M Release 1 has now closed, with its Stage 3 documents being completed in December 2011. These contain a large number of possible options, which mean that interoperability is likely to be a major problem, and "profiling" will be required by other standards bodies, or by ETSI itself for particular market segments (verticals etc.). Also Release 1 is missing critical functionality like charging and billing records.

Discussion has now started on Release 2 requirements, but with nothing agreed yet, not even timescales for closure of the requirements. Also, there is a new M2M global harmonisation (partnership project) initiative in 2012, involving co-ordination between standards' bodies in Europe (ETSI) and China (CCSA); this may see another forum take responsibility for this service layer and ETSI M2M will stop their work.

The ETSI SCP activity on Embedded UICC is still in the Requirements Phase, and so is some way behind its initial target of finalizing requirements in November 2011. A new, more realistic, target date is March 2012 for presenting the requirements to plenary.

There are two relevant standardization activities in OMA Device Management; both of which are expected to be completed in 2012. Both represent some significant changes to the existing OMA Device Management protocol. The activity OMA DM NG ("Next Generation") completed its requirements stage in December 2011 (approved as a "Candidate" document) and the Architecture Document is likely to be approved soon (Q1 2012). The TS (technical specification) completion is scheduled for **August 2012**. The activity OMA DM LW ("Lightweight") is in the late stage of Requirements; while the timescale for a TS is not fully stable; it is currently targeting completion by **December 2012.**

Although LW M2M work is focusing on requirements there have been a few proposals on the technical direction. Indeed, one proposal is to merge the LW and NG (e.g. develop a COAP binding for DM NG). But today it is open whether the group decides this path or another one (e.g. a WMMP-like protocol) with bindings to very constrained protocols like SMS and USSD.

6.4 Assessment of what can be Standardized

6.4.1 3GPP

Section 5.1.4 of this report describes a detailed proposal on "Machine Plane Security" which should be targeted at 3GPP. Initially, we propose to contribute these ideas to SA3, and seek review from SA2 and SA1 as well to the extent to which the proposal addresses the need for a "Secure Connection". It seems unlikely that these will result in a TS for Release 11 and perhaps not for Release 12 either, since the architectural changes look quite significant. This would not be a problem in connection with LTE-M, but a "quick patch" for LTE, 3G or GPRS security is not possible.

6.4.2 ISO/IEC 29192

Other parts of section 5.1 include discussions of optimized crypto-algorithms: lightweight block ciphers, stream ciphers, RNG-functions etc. It is not immediately obvious where these would be targeted, though standardization through IETF or ISO (and/or subsequent incorporation by 3GPP, perhaps by delegation to ETSI SAGE) looks like an attractive route.

EXALTED will be monitoring the activity of ISO/IEC 29192 working group. Contributions may be submitted to this group, most probably related to the choice of block or stream ciphers suitable for low energy M2M communications, where appropriate.

6.4.3 ETSI SCP

The material around a shared embedded UICC (section 5.2.4), and its importance to M2M (via possible M2M Service provider or M2M Application Provider applets, section 5.2.5) is highly relevant to SCP, and could be submitted as a discussion document to SCP, or a CR to existing requirements document. It would also be useful to submit these documents to GSMA to improve understanding of operators of the likely need to handle third-party applications, and explain why they are relevant in M2M (not just NFC).

The activation discussions (section 5.2.6) are also likely to be of interest to GSMA, and could motivate some particular technical requirements in ETSI SCP: such as the need to capture and report IMEI to the Subscription Manager; for the eUICC to report "ready" and "accepted" state; for eUICC to have a persistent identifier which is known by the M2M customer (and probably supplied in a matched IMEI-eUICCid list by the device manufacturer). GSMA will also be interested in these process issues.

6.4.4 Capillary networks pairing

The subject of pairing in capillary networks described in section 5.3 is currently the object of on-going work. Dissemination activity related to this subject will be likely related to paper publication. No standardization activity has been identified at this point.

6.4.5 Device Management

The security challenges related to M2M device management have been described in section 5.4. Progress in this area will be primarily driven by distinct standardization groups including OMA and the Broadband Forum.

EXALTED will be monitoring the activity of these groups and may submit contributions according to the developments, especially where they suggest a uniform solution suitable across the expected range of EXALTED devices, or where there are specific security problems which need resolving. However no specific contribution is planned at this time.

6.4.6 APIs to access secure elements

For the reasons discussed in 5.5, it is extremely difficult to provide an API from a Secure Element (UICC or eUICC) which allows its general use for application layer security.

The problem is not an absence of standards; by contrast, there are multiple standards, but the difficulty is giving key stakeholders an interest to deploy them in a reasonable time frame. Quite often, the standards have become obsolete before the API is put into significant use.

However, one thing EXALTED should watch for is the opportunity to elevate an existing standard to a “mandatory” deployment because of a particular compelling use-case in M2M; maybe in connection with eUICC or M2M service bootstrapping, or both. The fact that a secure element like eUICC is going to be pre-installed in the device by the manufacturer may give that manufacturer an incentive to support a mandatory API towards it, an incentive which was not necessarily present in the case of a (removable) UICC.

It is – for instance - expected that EXALTED could leverage the specifications of GlobalPlatform and SIM Alliance for the purpose of sharing a secure element between multiple M2M applications. There is an expectation that the GlobalPlatform API will be implemented by (at least some) manufacturers to interface with an embedded secure element. Contributions may be submitted to these groups if it turns out that currently available specifications are not fully adequate for our purpose.

6.4.7 Local breakout and direct modes

There is no standardization proposal here, except to watch developments closely, especially in 3GPP, and see if an opportunity opens up for any of the security solutions considered in section 5.6. We are not currently optimistic though.

6.4.8 ETSI M2M

As discussed above, the ETSI M2M group will be addressing in 2012 its phase 2 definition of the architecture. At this occasion, EXALTED WP5 may submit contributions along 2 lines

1. Define the security of the dialog between M2M service providers in order to insure the interoperability of M2M data communication across M2M data networks. The need for a communication link between M2M service providers in order to achieve a “trapezoidal” communication model has been discussed in section 5.7.1.3.4
2. Capability to achieve end to end data encryption by resorting to a trust provider distinct from the M2M service provider. This has been discussed in section 5.7.2

7. Conclusion

This report has addressed the security architecture needed to secure LTE-M communications.

In a first step we have provided a background on security basics, in particular as they affect existing mobile cellular networks: GSM, 3G and LTE. We have then listed the specific security requirements to be taken into account for LTE-M networks, including requirements already identified by the various working or standardization groups, as well as new requirements emerging from the EXALTED project.

Next we have analysed the major new security challenges facing LTE-M, arising from the need to achieve low cost security, but do so in a large-scale, complex environment, and without sacrificing overall security level, or impacting on device usability (since many devices will be unattended through much of their lives). This included giving an overview of the existing authentication and provisioning mechanisms used in LTE networks as well as a cost analysis explaining the difficulty to scale up using existing mechanisms to support an exponential growth of M2M communications.

We have also described our proposals to meet these challenges. The main technical proposals are as follows:

1. Optimizations to network access security, in particular via lightweight cryptography and a “machine plane” to collapse security layers for the LTE-M network;
2. Use of a an embedded SIM (eUICC) to reduce distribution and provisioning costs while maintaining a strong security level for authentication keys;
3. Secure pairing and group key agreement in capillary networks to allow low-cost capillary devices to connect through a gateway onto the LTE-M network;
4. Unification of various different device management approaches and with device usage for low-cost devices;
5. Sharing and re-use of secure elements, both between different devices, and at different layers: network access layer, M2M service layer and application layer.

To explain our plan to drive these proposals forward, we have provided an overview of the various industry standardization groups already dealing with similar subjects, in order to understand, their focus, roadmap and how they position with respect to each other. We list some proposals for future contributions, capturing the above points, in order to define solutions which take into account the security requirements specific to LTE-M networks. We have already engaged significantly with the standards on embedded SIM.

List of Acronyms

Acronym	Meaning
3G	3rd Generation of cellular wireless standards
3GPP	3rd Generation Partnership Project
3GPP SA	3GPP Service and System Aspects
4G	4th Generation of cellular wireless standards
ACS	Auto configuration server
AKA	Authentication and Key Agreement
AKE	Authenticated Key Exchange
ASME	Access Security Management Entity
AuC	Authentication Center
BOM	Bill of Materials
CH	Cluster Head
CPE	Customer Premise Equipment
CPNS	Converged Personal Network Services
CWMP	CPE WAN Management Protocol
CSD	Circuit Switched Data
DM	Device Management
DoS	Denial-of-Service
DSA	Digital Signature Algorithm
DSCL	Device Service Capability Layer
DSEC	Device Security Capability
DSL	Digital Subscriber Line
EAP	Extensible Authentication Protocol
EDGE	Enhanced Data rates for GSM Evolution
eID	Electronic Identity (card)
EMV	Europay, MasterCard and VISA (Smartcard payment system)
EPS	Enhanced Packet System
ETSI	European Telecommunications Standards Institute
eUICC	Embedded UICC
Femtocell	Small cellular base station
GBA	Generic Bootstrap Architecture
GERAN	GSM EDGE Radio Access Network
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSCL	Gateway Service Capability Layer
GSEC	Gateway Security Capability
GSM	Global System for Mobile Communications
GSMA	GSM Association
GUTI	Globally Unique Temporary Identifier
HLR	Home Location Register
HSS	Home Subscriber Server
ICC-ID	Integrated Circuit Card Identifier
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
ISIM	IP Multimedia Services Identity Module
IT	Information Technology
IV	Initialization Vector
KDC	Key Distribution Centre

LTE	Long Term Evolution
M2M	Machine-to-Machine
MAC	Medium Access Control
MAS	M2M Authentication Server
MFF	M2M Form Factor
MITM	Man In The Middle
MNO	Mobile Network Operator
MSBF	M2M Service Bootstrap Function
MSISDN	Mobile Subscriber ISDN (Integrated Services Digital Network)
MTC	Machine-Type Communication
NAS	Non-Access Stratum
NAT	Network Address Translation
NFC	Near Field Communication
NSCL	Network Service Capability Layer
NSEC	Network Security Capability
OEM	Original Equipment Manufacturer
OFDMA	Orthogonal Frequency-Division Multiple Access
OMA	Open Mobile Alliance
OTA	Over The Air
PACS	Physical Access Control Systems
PAN	Personal Area Network
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PDP	Packet Data Protocol
PHY	PHYsical layer
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
PMP	Personal Media Player
PnP	Plug and Play
PUF	Physically Unclonable Functions
RPC	Remote Procedure Call
RRC	Radio Resource Control
RSA	Security algorithm named after its inventors (Ron Rivest, Adi Shamir, and Len Adleman)
SC	Service Capability
SE	Security Element
SIM	Subscriber Identity Module
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SP	Service Provider
TISPAN	Telecoms and Internet converged Services and Protocols for Advanced Networks
TMSI	Temporary Mobile Subscriber Identity
TTI	Transmit Time Interval
UE	User Equipment
UI	User Interface
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
WAN	Wide Area Network
WAP	Wireless Application Protocol
WiMAX	Worldwide Interoperability for Microwave Access

WLAN	Wireless Local Area Network
WWRF	Wireless World Research Forum
xDSL	DSL technologies

References

[1] 3GPP TS 33.401 “3GPP System Architecture Evolution (SAE); Security architecture” http://www.3gpp.org/ftp/Specs/html-info/33401.htm
[2] 3GPP TS 33.102 “3G security; Security architecture” http://www.3gpp.org/ftp/Specs/html-info/33102.htm
[3] 3GPP TS 22.368, “Service requirements for Machine-Type Communications (MTC)”. Latest version at http://www.3gpp.org/ftp/Specs/html-info/22368.htm
[4] OMA : Converged Personal Network Service Architecture; Candidate Version 1.0 – 15 Jun 2010
[5] OMA: Enabler Release Definition for Converged Personal Network Service; Candidate Version 1.0 – 15 Jun 2010
[6] OMA: Converged Personal Network Service Requirements; Candidate Version 1.0 –17 November 2009
[7] OMA: Device Management V1.2; Approved Enabler-Release Date 17 June 2008; http://www.openmobilealliance.org/Technical/release_program/dm_v1_2.aspx
[8] ETSI TS 102 690 V 1.1.1 (2011-10-25) : Machine- to- Machine communications (M2M); Functional architecture. Latest version available for download from ETSI site at: http//webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=30459
[9] 3GPP TR 23.888, “System improvements for Machine-Type Communications (MTC)”. Latest version at http://www.3gpp.org/ftp/Specs/html-info/23888.htm
[10] 3GPP TR 33.868, “Security aspects of Machine-Type Communications” Draft TR. http://www.3gpp.org/ftp/Specs/html-info/33868.htm
[11] Rolf Blom, Karl Norr man, Mats Näslund, Stefan Rommer and Bengt Sahlin, “Security in the Evolved Packet System”, Ericsson Review, Feb 2010
[12] GSMA, “GSMA-Led Task Force Defines Market Requirements for a Standardised Embedded SIM” http://www.gsma.com/articles/gsma-led-task-force-defines-market-requirements-for-a-standardised-embedded-sim/17583/
[13] W. Diffie and M. Hellman, “New Directions in Cryptography,” IEEE Trans. on Information Theory, vol. 22, pp. 644-654, 1976.
[14] M. Burmester and Y. Desmedt, “A Secure and Efficient Conference Key Distribution System,” in Advances in Cryptology - EUROCRYPT 1994, LNCS, vol. 950, A. De Santis, Ed., Springer, 1995, pp. 275–286.
[15] - A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C.Vikkelsoe, PRESENT — An Ultra-Lightweight Block Cipher. In Proceedings of Workshop on Cryptographic Hardware and Embedded Systems — CHES 2007, Lecture Notes in Computer Science, volume 4727, pp. 450 - 466, Springer-Verlag, 2007
[16] - T. Shirai, K. Shibutani, T. Akishita, S. Moriai, T. Iwata, The 128-bit Blockcipher CLEFIA. In Proceedings of Workshop on Fast Software Encryption 2007 — FSE 2007, Lecture Notes in Computer Science volume 4593, pp. 181-195, Springer-Verlag, 2007
[17] - Watanabe, D., Furuya, S., Yoshida, H., Takaragi, K., and Preneel, B., "A New Key Stream Generator MUGI," Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers, eds. Daemen, J. and Rijmen, V., Lecture Notes in Computer Science vol.2365, Springer-Verlag, pp.179-194, 2002
[18] - Ekdahl, P. and Johansson, T., "A new version of the stream cipher SNOW", Selected Areas in Cryptography, 9th Annual Workshop, SAC 2002, St. John's, Newfoundland, Canada, Aug. 2002,Revised Papers, eds. Nyberg, K. and Heys, H., Lecture Notes in Computer Science vol. 2595, Springer-Verlag, pp.47-61, 2002
[19] - Boesgaard, M., Vesterager, M., Pedersen, T., Christiansen, J., and Scavenius, O., "Rabbit: A new highperformance stream cipher". In T. Johansson, editor, Proc. Fast

Software Encryption 2003, Lecture Notes in Computer Science vol.2887, Springer-Verlag, pp.307-329, 2003

[20] - Berbain, C., Billet, O., Canteaut, A., Courtois, N., Debraize, B., Gilbert, H., Goubin, L., Gouget, A., Granboulan, L., Lauradoux, C., Minier, M., Pornin, T. and Sibert, H., "DECIMv2, a compact hardware oriented stream cipher", SASC 2006 - Stream Ciphers revisited Workshop, Leuven, Belgium, 2006.

[21] - Kiyomoto, S., Tanaka, T., and Sakurai, K., "A Word-Oriented Stream Cipher Using Clock Control", In SASC 2007 Workshop Record, pp.260-274, January, 2007

[22] - D. Watanabe, K. Ideguchi, J. Kitahara, K. Muto, H. Furuichi, T. Kaneko, "Enocor 80: A Hardware Oriented Stream Cipher". ARES 2008: pages 1294-1300

[23] - C. De Cannière and B. Preneel, " Trivium ". The eSTREAM Finalists: pages 244-266, LNCS.

[24] Hash Functions and RFID Tags: Mind the Gap" by A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, and Y. Seurin: http://yannickseurin.free.fr/pubs/Bogdanov_et_al08_CHES.pdf

[25] FP7 EXALTED consortium: "D2.1 - Description of baseline reference systems, scenarios, technical requirements & evaluation methodology," project report, May 2011.

[26] S3-110558, "LS on potential co-operation between 3GPP work on MTC security and ETSI M2M"; 3GPP SA3, contact company Vodafone; http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_63_Chengdu/Docs/

[27] SCP(11)0146r1, "WID: Embedded SIM Use Cases and Requirements"; ETSI SCP; http://portal.etsi.org/portal/server.pt/community/SCP/333

[28] SCP(11)0088, "Embedded SIM Use Cases and Requirements v1.0"; GSM Association; http://portal.etsi.org/portal/server.pt/community/SCP/333

[29] SCP(11)0147r1, "Liaison Statement on new Work Item for eUICC"; ETSI SCP REQ; http://portal.etsi.org/portal/server.pt/community/SCP/333

[30] SCPREQ(11)0043, CR to ETSI TS 102 412 "Addition of requirements for the eUICC and its remote management"; Deutsche Telekom, Giesecke & Devrient, Telefonica O2, Vodafone; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[31] SCPREQ(11)0044, "Embedded UICC"; Deutsche Telekom, Telefonica O2, Vodafone, Giesecke & Devrient; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[32] SCPREQ(11)0064, "High Level Components in the eUICC – First provisioning"; Vodafone Group; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[33] SCPREQ(11)0113, "Embedded UICC – A high level remote provisioning architecture"; GSMA Embedded SIM Task Force: Technical Stream; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[34] SCPREQ(11)0118, "GSMA and SIMalliance Collaboration on eUICC Protection Profile"; GSMA; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[35] SCPREQ(11)0048 "Additonal role definition for eUICC remote provisioning", Gemalto; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[36] SCPREQ(11)0068: "pCR on eUICC definition of profile manager and related requirements", GTO; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[37] SCPREQ(11)0079: "pCR on eUICC: System architecture description", Gemalto; http://portal.etsi.org/portal/server.pt/community/SCP/333?tbId=639

[38] Global Platform Specifications http://www.globalplatform.org/specifications.asp

[39] RFC 2246 "The TLS Protocol" http://www.rfc-editor.org/info/rfc2246

[40] RFC 4346 "The Transport Layer Security (TLS) Protocol Version 1.1" http://www.rfc-editor.org/info/rfc4346

[41] RFC 5246 "The Transport Layer Security (TLS) Protocol Version 1.2" http://www.rfc-editor.org/info/rfc5246

[42] RFC 4279 "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)" http://www.rfc-editor.org/info/rfc4279

[43] REST, http://en.wikipedia.org/wiki/Representational_state_transfer

[44] TR-069, CPE WAN Management Protocol v1.1, Issue 1, Amendment 2

http://www.broadband-forum.org/technical/download/TR-069_Amendment-2.pdf