Communication and Security in Machine-to-Machine Systems

Iva Bojic[1], Jorge Granjal[2], Edmundo Monteiro[2], Damjan Katusic[1], Pavle Skocir[1], Mario Kusek[1], and Gordan Jezic[1]

[1] University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia
[2] University of Coimbra, Department of Informatics Engineering, Polo 2, 3030-290 Coimbra, Portugal

Abstract. Machine-to-Machine (M2M) systems and technologies currently constitute a hot topic in the field of Information and Communication Technology (ICT), and reflect an increasing need for technologies enabling applications in diverse areas, as well as interactions between continuously increasing numbers of connected devices. Important participants in making M2M systems widely used and applicable in numerous real-life scenarios are standardization organizations. They try to develop technical specifications that address the need for a common M2M service layer, which can be realized through various hardware and software implementations. This chapter presents current standards and architecture of M2M systems with the focus on communication and security issues, while also discussing current and future research efforts addressing important open issues. One of the main problems in the area is correlated with heterogeneous devices, which are using different technologies for communication. Because of communication technology diversity, research challenges are to uniquely identify devices, and to enable them to communicate securely. To tackle the former, previously proposed, a unique identifying scheme that enables device identification regardless of used technology is explained. Regarding the latter, we analyze how current standards and architecture of M2M systems define basic processes for secure connection establishment, and also discuss open issues, both in respect to aspects not covered by current standards and in relation to research proposals which may integrate with M2M systems in future versions of the standards.

Keywords: M2M, communication identifiers, M2M security, 6LoWPAN security

1 Introduction

Abbreviation M2M has several different meanings: Mobile-to-Mobile, Machine-to-Machine, Machine-to-Man (or vice-versa), Machine-to-Mobile (or vice-versa) [1]. In this chapter we will use it in the context of Machine-to-Machine communication. M2M communication is established between two or more entities

without any need of direct human intervention [2] [3]. Actors in such an environment include broad range of communication capable devices: computers, mobile phones, tablets, but also a variety of sensors, actuators, pieces of industrial and medical equipment, and countless other everyday devices [4] [5]. Another important aspect of M2M communication, as it can be seen from its longer acronym M2(CN2)M that stands for Machine-to-(Communication-Network-to)-Machine [6], is the notion of the underlying communication network that allows bidirectional exchange of information between these devices. M2M systems find applications in different areas such as home and industry automation [7], connected consumer [8], smart metering [9], healthcare [10], smart traffic [11], and many others. Work done by Beecham Research is one of many attempts trying to systematize all application areas applicable to M2M systems paradigm [12]. Through this variety of possible uses, M2M communication helps to achieve the vision of connected things - Internet of Things (IoT) [13] [14], a world where ubiquitous and intelligent applications contribute to a better and safer world.

The number of connected devices is rapidly growing. International Data Corporation predicts there will be 15 billion devices communicating over the network by the year 2015 [15], while Cisco Internet Business Solutions Group (IBSG) forecasts 25 billion devices connected to the Internet by 2015 and 50 billion by 2020 [16]. Machina Research white paper states that by 2022 there will be 18 billion M2M connections globally, up from approximately 2 billion today [17]. Ericsson claims that their vision of more than 50 billion connected devices by 2020 may seem a bit ambitious today, but with the right approach, it is within reach [5]. Due to this rapid growth, the concept of M2M communication is gaining more and more significance. Interoperability, between devices based on different access network technologies (e.g. mobile (2G/3G/4G), Wi-Fi, Bluetooth), using different platforms and data models is still very limited or non-existent [18]. The idea is to connect a plethora of different devices that communicate through different technologies and thus create a heterogeneous environment. In order to enable connection of heterogeneous devices, globally accepted standards have to be developed to achieve ubiquitous connectivity and security. In addition, service platforms that will be reusable across different application areas and will unify isolated vertical "silo" solutions, based on common device capabilities, have to be developed [6]. Apart from standardization, policy and government incentives are also necessary to speed up the maturity of M2M systems [19]. On one hand, governments allocate funds for the development of certain technologies which are thought to increase the quality of everyday life (e.g. smart metering in households). On the other, regulation incentives provide precise directions for the development of the sets of standards applicable within a certain country or a region.

Standardization efforts in the area of M2M communication are very strong. One of the most influential standardization organizations involved in creating common standards for M2M communication is the European Telecommunications Standards Institute (ETSI) [20]. Recently it has joined six other standardization organizations from around the world in forming a global M2M initiative:

oneM2M [21]. These organizations are: Association of Radio Industries and Businesses (ARIB) [22] and Telecommunication Technology Committee (TTC) [23] from Japan, Alliance for Telecommunications Industry Solutions (ATIS) [24] and Telecommunications Industry Association (TIA) [25] from the USA, China Communications Standards Association (CCSA) [26], and Telecommunications Technology Association (TTA) [27] from South Korea. The goal of oneM2M is to develop technical specifications which address the need for a common M2M service layer, which can be realized through various hardware and software implementations, to connect diverse M2M devices with M2M servers [21]. Currently oneM2M have not published standards, but it is planned for 2014. Such specifications relate to how M2M devices may be identified, how they communicate, and also with how such interactions and communication between M2M systems may be supported with security in place. The M2M system in a very simplified aspect, as will become clear in the following sections when current considerations regarding architecture standardization by ETSI will be presented, consists of M2M devices, M2M gateways, and M2M servers.

Apart from heterogeneity in types of M2M devices, M2M systems should also allow communication between different M2M entities (i.e. M2M devices, M2M gateways, or M2M servers), ignoring the differences in the network technologies, including the underlying used addressing mechanism. For example, in an Internet Protocol (IP) based network, the communication establishment between M2M entities should be possible when either static or dynamic IP addressing is used regardless of the use of public or private IP address space. Moreover, it is very important to emphasize that IP connectivity is not the only option, i.e. M2M devices can be connected using different M2M area networks (e.g. Zigbee, Bluetooth, M-BUS). Interactions with security technologies developed for such M2M area networks must also be considered. Therefore, a common identification scheme for different entities within M2M system has to be developed. In ETSI standards, the need for identification of a single M2M entity is addressed in [28]. Single M2M device can support several different communication technologies at once. Therefore, this identifier has to be independent of access networks that M2M devices use for communication. ETSI, among several other identifiers, recognizes pre-provisioned identifier used for the bootstrapping procedures of M2M devices, and in this chapter we will explain how it was proposed in [29] it should look like.

One of the main problems in any information system is security, and M2M systems are certainly no exception. With a huge market expected for M2M devices and networks, M2M systems need to be properly developed and deployed. We also realize that many of the applications envisioned for M2M will only be realizable if security is properly addressed from the start. Despite an urgent need for proper security mechanisms and procedures, various characteristics of M2M systems and applications may pose challenges to the design of appropriate security mechanisms. Among such difficulties we may identify the support of heterogeneous communication technologies and protocols, of autonomous communication between M2M devices, the limitations on hardware capabilities of

many M2M sensing and actuating platforms, and expectations from users regarding security, in particular privacy and liability [30]. Although many lessons and technical solutions have been learned from research in areas such as mobile ad hoc networks [31] [32] or wireless sensor networks [33], M2M systems may also require new approaches to security.

The employment of different wired and wireless communication technologies motivated by the usage of a common service platform determines the careful evaluation of the adopted cryptographic algorithms, or on the other hand the design of new ones. The support of autonomous communication requires also appropriate universal identification techniques, such as we discuss in this chapter. The characteristics and resource constraints of M2M systems also pose challenges to the design of appropriate security technologies that are able to deal with heterogeneous sensing and actuating M2M devices. Regarding expectations on security of the users of M2M systems and applications, privacy and liability appear as important factors, as users will require that systems allow the control of how much personal information is exposed, while on the other end certain applications will require that a certain degree of personal information is guaranteed to be available [34]. In the light of such challenges, this chapter analyses how security is addressed in the context of the M2M architecture defined by ETSI [28] that we describe in the next section. Given the significance of this architecture for the future of M2M communication and security technologies, this analysis sheds some light on how security is currently being addressed in the area of M2M.

This chapter is organized as follows. In Section 2, we give an overview of M2M high-level and functional architectures. Also, we provide a high level description of one of many possible sets of specific service capabilities (SCs) that will allow efficient deployment of M2M applications. Section 3 describes M2M communication scenarios with focus on identification, authorization, trust, and security. A brief overview of proposed pre-provisioned identifier that can uniquely identify M2M devices is given. Section 4 presents current research opportunities on communication and security in M2M systems, and discusses standardization challenges that will enable further development of M2M systems as they evolve and gradually replace proprietary technologies. Section 5 concludes the chapter.

2 M2M architecture defined by ETSI

ETSI's work regarding the M2M communication has been so far mostly focused on different use case scenarios (e.g. smart metering [9], eHealth [10], connected consumer [8], automotive applications [11], city automation [7]), M2M communication service requirements [35], its high-level and functional architecture [28], as well as defining M2M interfaces [36]. ETSI closely co-operates with other standardization organizations such as 3rd Generation Partnership Project (3GPP) [37], 3GPP2 [38], Open Mobile Alliance (OMA) [39], and Broadband Forum (BBF) [40] in integration efforts of their respective technologies into M2M systems. Their work in the M2M architecture domain is twofold. First, they define

a high-level architecture view that identifies all constituents of M2M systems, their roles, and relationships. Second, they also define a functional architecture view together with reference points between different entities in M2M systems, as well as M2M service capabilities, and common functions that are being shared among different applications.

2.1 High-level Architecture

A high-level architecture of M2M system consists of a Device and Gateway Domain, and a Network Domain (c.f. Figure 1) [28]. The device and gateway domain is composed of the following elements:

- **M2M Device** - runs **M2M Device Applications (DA)** using M2M Device Service Capabilities Layer (DSCL).
- **M2M Gateway** - runs **M2M Gateway Applications (GA)** using M2M Gateway Service Capabilities Layer (GSCL).
- **M2M Area Network** - provides connectivity based on Personal or Local Area Network technologies (e.g. Zigbee, Bluetooth) between M2M devices and M2M gateways. The case of device-to-device communication is out of the scope of ETSI's efforts (denoted in dashed line in Figure 1).

The network domain is composed of the following elements:

- **M2M Access Network** - allows M2M devices and M2M gateways to communicate with the Core Network. It can be based on any of the following existing access network solutions: Digital Subscriber Line (DSL), satellite, GSM EDGE Radio Access Network (GERAN), Universal Terrestrial Radio Access Network (UTRAN), evolved UTRAN (eUTRAN), Wi-Fi (IEEE 802.11), and Worldwide Interoperability for Microwave Access (WiMAX), that can be optimized for M2M communication if needed.
- **M2M Core Network** - enables interconnection with other networks, provides IP connectivity or other connectivity options, service and control functions, and roaming. Similarly to access network, it can be based on varied existing core networking (CN) solutions (3GPP CN, ETSI Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) CN, and 3GPP2 CN) that ought to be optimized for specific M2M communication needs if necessary.
- **M2M Network Service Capabilities Layer (NSCL)** - provides M2M functions that are shared by different M2M applications.
- **M2M Applications** - run the service logic and use M2M service capabilities available via open interfaces.
- **M2M Network Management Functions** - consist of all the functions (e.g. provisioning, supervision, and fault management) required to manage access and core networks.
- **M2M Management Functions** - consist of all the functions (e.g. M2M Service Bootstrap Function (MSBF)) used to facilitate the bootstrapping of permanent M2M service layer security credentials) required to manage M2M service capabilities in the network domain.

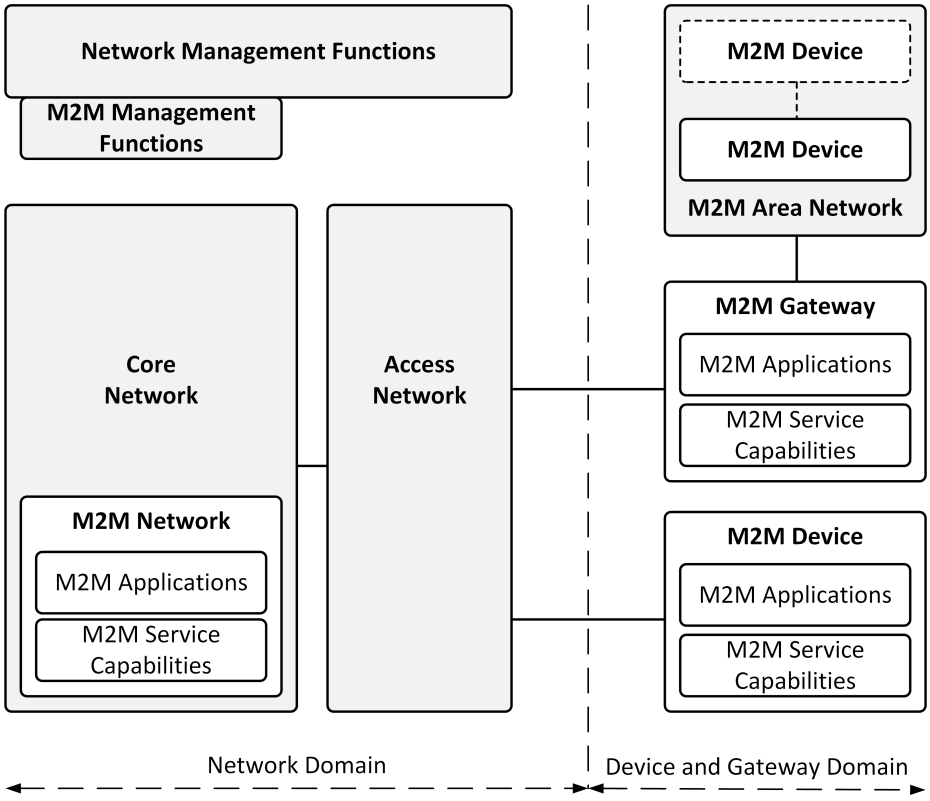

Fig. 1. High-level architecture of M2M system [28]

2.2 Functional Architecture

Each M2M domain has its own service capabilities layer (i.e. Network SCL, Gateway SCL, and Device SCL), which provides functions that are exposed on the mIa, dIa, mId, and mIm reference points [36] (c.f. Figure 2). The **mIa** reference point enables a Network Application (NA) access to the M2M service capabilities in the network domain. It supports possibility for NA to register to the NSCL, to subscribe for notifications for specific events, with a proper authorization to read or write information in N/G/DSCLs, and to conduct device management actions. The **dIa** reference point enables a Device Application residing in a non-legacy M2M device an access to different M2M service capabilities in that same M2M device or in an M2M gateway. Furthermore, this reference point enables a Gateway Application residing in an M2M gateway to access the different M2M service capabilities in the same M2M gateway, and supports the ability of DA/GA to register to the GSCL or DA to register to the DSCL. Through this reference point, DA and GA should also be able with a proper authorization to read or write information in N/G/DSCLs. The **mId** reference point enables an M2M SCL residing in a non-legacy M2M device or

M2M gateway to communicate with the M2M SCL in the network domain and vice versa. It supports the ability of G/DSCLs to register to the NSCL. It should also give support for information exchange between N/G/DSCLs, subscription to specific events, device management, and provide security related features. Finally, the **mIm** reference point extends the reachability of services offered over mId reference point. It is an inter-domain reference point, used for communication between NSCLs of different M2M service providers, that relies on public core network connectivity functions. Functionalities of all four reference points are in more details explained in [36].

One of main M2M standardization objectives is the development of functionalities that will allow efficient deployment for M2M applications. Service Capabilities are logical groupings of functions that can be shared by different applications. With standardized vertical interfaces (i.e. Application Programming Interfaces (APIs)) that allow applications to use service capabilities, and standardized horizontal interfaces between SCs on the service level, this objective is within a reach. The remainder of the section provides a high level description of one of many possible sets of specific SCs [28]. These service capabilities can be, with little differences, instantiated for each of the network or gateway and device domains (x in each of the below stands for either N for network, G for gateway, or D for device). The only exception is a Telco operator exposure that is specific for the network domain.

- **Application enablement (xAE)** - is the single contact point to M2M applications. It exposes functionalities implemented in each of the SCLs via a single reference point: mIa/dIa (depending on the SCL in question).
- **Generic communication (xGC)** - is the single point of contact for communication with each of the SCLs. This capability provides transport session establishment and teardown along with security key negotiation, encryption and integrity protection on data exchanged with the SCLs. Key material for the latter is derived upon secure session establishment.
- **Reachability, addressing, and repository (xRAR)** - provides a mapping between the name of an M2M entity or a group of M2M entities and its/their reachability status. It also manages subscriptions and notifications pertaining to events and allows creating, deleting, and listing of a group of M2M entities. It stores M2M application (NA/DA/GA) and SCL (DSCL, NSCL) data, and makes it available on request or based on subscriptions.
- **Communication selection (xCS)** - provides network selection, based on policies, when each of the available M2M entities can be reached through several networks or several bearers. It also includes alternative network or communication service selection after a communication failure.
- **Remote entity management (xREM)** - acts as a remote management client to perform the device remote entity management functionalities (e.g. software and firmware upgrades, fault (FM), performance (PM), and configuration management (CM)) for the M2M entities. It supports several management protocols, such as OMA-DM [41] and BBF TR-069 [42].
- **Security (xSEC)** - supports M2M service bootstrap and key hierarchy for authentication and authorization procedures. It also initiates mutual authen-

tication and key agreement, and is responsible for the storage and handling of M2M connection keys.
- **History and data retention (xHDR)** - is an optional capability deployed when required by policies. It archives relevant information referring to messages exchanged over the reference points and also internally to each of the SCLs based on policies.
- **Transaction management (xTM)** - is an optional capability that deals with transactions. Transaction is an operation that involves several atomic operations. This capability triggers a roll-back if any individual operation fails, aggregates the results of the individual operations, and commits the transaction when all individual operations have completed successfully.
- **Interworking proxy (xIP)** - is also an optional capability that enables interworking between non-ETSI compliant devices and the SCLs. It can be implemented either as an internal capability of DSCL/GSCL, or an application communicating via reference point dIa with DSCL/GSCL.
- **Compensation brokerage (xCB)** - is another optional capability deployed only when needed. It submits compensation tokens (i.e. electronic money) to requesting customers, bills the customer of compensation tokens after the validity of compensation tokens is verified, and finally refunds service providers for tokens acquired as compensation for services provided to customers.
- **Telco operator exposure (xTOE)** - enables interworking and using of core network services exposed by the network operator.

Regarding security, ETSI defines a set of functions and procedures to support security-related capabilities, with the main goal of supporting and providing fundamental security mechanisms and properties in the context of M2M systems, applications, and devices. Later in the chapter we analyze in greater detail such mechanisms, namely in the context of the various procedures defined for M2M communication establishment. For example, key negotiation, and encryption and integrity protection are supported in the context of the various communication between M2M entities, together with the required key bootstrap and negotiation functionalities. Cryptographic keys are used to support authentication and authorization of the M2M devices participating in a given M2M domain, in the context of the various procedures such as M2M device registration, management, and provisioning. As previously identified, the various security procedures are also visible in the context of the reference points defined for the architecture, particularly as regards to identification, authorization, registration, and interactions (reading or writing) with service capabilities layers.

3 Communication establishment

ETSI distinguishes two types of **M2M devices**: those that implement ETSI M2M service capabilities (marks them as D), and those that do not (marks them as D'). Both types of devices are still considered ETSI compliant, and there are two ways how they can connect to the network domain: directly or indirectly (through **M2M gateway**) (c.f. Figure 2). M2M gateway serves as a proxy for the network domain, which means that all the procedures mentioned in the previous section (registration, authentication, authorization, management, and provisioning) are performed through it. In the first case, M2M devices connect to the network domain via the **M2M access network**. In the second case, M2M devices connect to the M2M gateway using the **M2M area network**. To summarize, M2M devices can be connected to M2M network domain:

- directly through mId interface to NSCL (e.g. M2M device 1 (type D) in Figure 2),
- indirectly through dIa interface to GSCL (e.g. M2M device 2 (type D') in Figure 2),
- directly through dIa interface to NSCL (e.g. M2M device 3 (type D') in Figure 2).

However, an M2M device may not support IP protocol for communication[3], in which case it is called a **legacy M2M device** and is marked as *d*. A legacy M2M device can be connected to M2M network domain:

- indirectly through Gateway Interworking Proxy (GIP) on M2M gateway (e.g. M2M device 4 in Figure 2),
- indirectly through Device Interworking Proxy (DIP) on a non-legacy M2M device (type D) (e.g. M2M device 5 in Figure 2),
- directly through Network Interworking Proxy (NIP) (e.g. M2M device 6 in Figure 2).

M2M devices in both cases, if connected to IP network (e.g. Internet), communicate using IP protocol. In the first case (direct connectivity), an M2M device usually has a public IP address, while in the second case (indirect connectivity) it has a private one. The 3GPP standard *System Improvements for Machine-Type Communications* [43] analyses three possible addressing scenarios for communication between M2M server in the network domain and M2M device in the device and gateway domain. First scenario involves M2M server and M2M device both located in the IPv6 address space, in the second scenario M2M server is located in the public IPv4 address space and M2M device is located in the private IPv4 address space, and the third scenario involves both M2M server and M2M device located in the same private IPv4 address space. Using a taxonomy proposed in our previous work [44] we can classify aforementioned M2M communication types for two M2M devices into three categories. When both M2M

[3] If M2M device uses connectivity protocols such as ZigBee, Z-Wave, or Bluetooth that do not natively support IP.

devices are directly connected to a network domain via an M2M access network, then this type of communication can be classified as *direct* and *external*. In Figure 2, an example would be communication between M2M device 6 and M2M device 3 (denoted in dashed line), as both are directly connected to an M2M access network. Furthermore, when one M2M device is connected directly to a network domain, while the other one is connected to an M2M gateway using an M2M area network, then this type of communication is *indirect* and *external*. An example in Figure 2 are M2M device 6 which is connected directly to M2M access network, and M2M device 2 which is connected indirectly through an M2M gateway (denoted in dot dashed line). Finally, when both devices are connected to an M2M gateway using an M2M area network, then this type of communication is classified as *indirect* and *internal*. In Figure 2, an example of internal communication is between M2M devices 2 and 4 which are connected through an M2M gateway (denoted in dotted line). Figure 2 also features device 7 (denoted in transparent shade) which is an example of potential device-to-device communication with device 2. Such communication scenario is out of the scope of ETSI's standardization efforts and therefore is not featured in any of the analyses.

The process of communication establishment in M2M systems defined in [28] consists of the following six procedures: application registration, network bootstrap, network registration, M2M service bootstrap, M2M service connection, and SCL registration of D/GSCL with NSCL.

Application Registration procedure involves registration of an application (DA, GA, or NA) with local SCL. It allows interactions between local applications (i.e. those connected via the local SCL), while enabling M2M communication between applications connected on other SCLs requires involvement of several other procedures.

Network Bootstrap procedure defines initial configuration settings that allow an M2M device/gateway to connect and register to its access network, if it is based on fixed or mobile technologies. One example of these procedures is a bootstrap procedure from Universal Integrated Circuit Card (UICC).

Network Registration procedure consists of registration of an M2M device/ gateway with its access network, taking into account the characteristics of corresponding access network technologies. For example, registration of an M2M device in a 3GPP Network such as Universal Mobile Telecommunications System (UMTS) involves IP address assignment, mutual authentication as two sides agree on a set of security keys, authorization for using specific access network services, as well as initiation of potential accounting operations.

M2M Service Bootstrap procedure, together with M2M service connection procedure, defines basic prerequisites for the communication establishment and registration of an M2M D/GSCL with the NSCL. It involves, apart from the usual actors, the M2M Service Bootstrap Function and the M2M Authentication Server (MAS). Former facilitates the bootstrapping of permanent M2M Service Layer Security Credentials (**M2M Root Key**) between the M2M D/G/NSCL entities, as well as MAS, while the latter serves as a safe location for storage. If

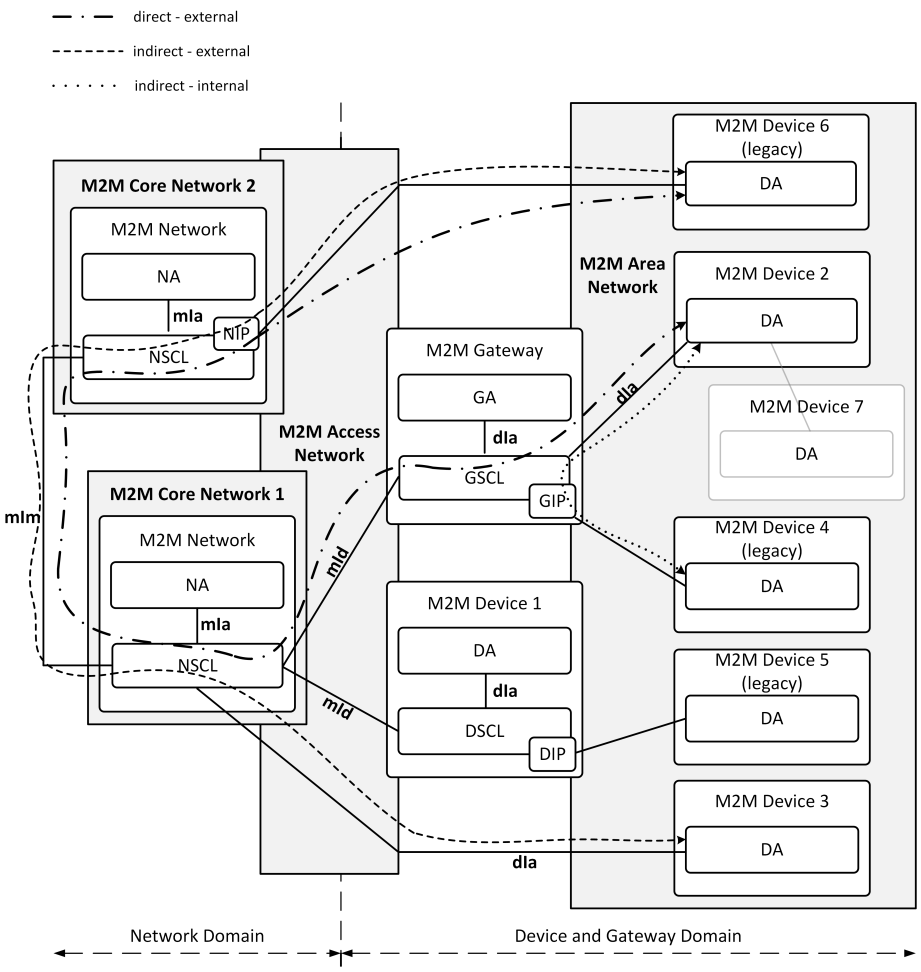

Fig. 2. Functional architecture of M2M system

the M2M service credentials have been pre-provisioned (e.g. in UICC), the M2M service bootstrap procedure is not needed. Otherwise, it is conducted with or without the assistance of an associated access network layer.

M2M service connection procedure includes a mutual authentication of mId end points (D/GSCL and NSCL), an optional agreement on an **M2M Connection Key** derived from an M2M root key, as well as an optional establishment of a secure encrypted session via mId.

SCL Registration procedure, as its name suggests, is involved in a D/GSCL registration with NSCL. Successful completion of M2M service connection procedure between D/G/N M2M entities is a prerequisite for performing SCL registration. This procedure occurs either periodically or on demand. When this

procedure is used periodically, the frequency of registration updates is decided by the M2M service provider. Successful registration, among others, results in an exchange of context information between D/GSCL and NSCL.

In other words, when a new M2M device enters the M2M system, it needs to establish an initial contact with a corresponding SCL and an access network, and then conduct all relevant credential creations and exchanges to establish a secure connection with its communication peer (e.g. M2M server in a network layer) within the M2M system. Only when this process is successfully completed, an M2M device is able to communicate with other entities in the M2M system (e.g. report measured sensor data to the server or update new version of software).

As for most security approaches, one fundamental security-related aspect of the ETSI M2M architecture [28] is how cryptographic keys are employed to support security mechanisms. The standard defines two types of keys, an M2M root key (Kmr) and an M2M connection key (Kmc). Kmr is used to support mutual authentication between the D/G M2M node and the M2M service provider, while a Kmc key is derived to protect communication in the context of the specific connection. Kmr must be set up for each specific D/G M2M node, in the context of a particular M2M service provider, for example during manufacturing or the deployment phase, and is also related to a particular M2M node identifier, as previously discussed. After successful authentication, Kmc is derived and delivered from the MAS to the M2M node where it is stored in a local secure environment domain (e.g. using UICC).

In general, security-sensitive functions and data (credentials and key material) shall be protected using a secure domain, which may be optionally integrity-protected using asymmetric cryptography for the provisioning and validation of trusted reference values. Integrity-validation extends to both devices and gateways, and allows verifying if a device is authorized to connect to the network, and enables validation of executable code on M2M devices, according to different security policies. Successful integrity validation may also be defined as a precondition for successful M2M service bootstrap, although we must note that all integrity-related steps are currently defined as optional in the ETSI M2M architecture [28]. Two further aspects deserve our attention in relation to how security keys are handled in this architecture. One is that, for M2M applications where the M2M service provider and access network provider have a trust relationship, access network credentials may be used to obtain the Kmr key, which consequently opens the door for the adoption of new authentication mechanisms designed at diverse layers of the network stack. The other is that the expiration of Kmc is left to particular policies from M2M service providers, which may constitute a problem if policies allow the same Kmc to be used to support symmetric encryption during unacceptably long periods of time.

3.1 Application Registration

Application Registration involves local registration of an M2M application with the local SCL, and the purpose of this procedure is to allow the M2M application to use M2M services offered by the local SCL. As a result, the local SCL obtains

context information on the registered applications. Two applications registered to a common local SCL can communicate via that local SCL. For the purpose of an application-level authentication and encryption, application specific keys can be generated optionally as previously discussed, thus a Kmc obtained from the Kmr root key after mutual authentication may be used also to protect application registration. Application registration depends on other procedures and thus also on the security procedures described in the following phases of connection establishment, as we proceed to discuss.

3.2 Network Bootstrap & Network Registration

The purpose of network bootstrap is to configure an M2M device or gateway with the initial configuration data required to connect and register to the access network. Bootstrap can occur from a secured environment domain (e.g. UICC), resulting in the data required to perform access network registration. Alternatively, bootstrap can also employ an over-the-air mechanism to provision the access credentials (including key material) required for registration operations. As for network registration, it involves the registration of the M2M device/gateway with the access network, based on the corresponding access network standards. This may involve (mutual) authentication with the access network, with the two ends agreeing on a set of security keys to protect the access network session. Registration may also involve IP address assignment, authorization approval for using specific access network services, and the initiation of access network accounting operations. Both the network bootstrap and network registration specific procedures are currently defined to be out of scope of the current ETSI specification [28].

3.3 M2M Service Bootstrap & M2M Service Connection

In order to successfully complete procedures of the connection establishment process and establish communication, each M2M entity has to have a proper unique identifier, and eventually address based on the used communication technology so it can be reached by other M2M entities. ETSI proposes several identifiers which are used during M2M service bootstrap and M2M service connection procedures regarding successful connection setup in M2M systems [28]:

- **Pre-provisioned Identifier** - represents an ID and needs to be pre-provisioned by the M2M device/gateway manufacturer and is considered out of scope for this document.
- **M2M Node Identifier (M2M-Node-ID)** - represents globally unique logical representation of the M2M components in the M2M device, M2M gateway, or M2M network. Such components include one SCL, M2M service bootstrap function if any, and an M2M service connection function. On a global level, M2M-Node-ID uniquely identifies a particular M2M entity.
- **M2M Service Connection Identifier (M2M-Connection-ID)** - identifies an M2M service connection, which is instantiated upon M2M D/GSCL getting authenticated and authorized by an NSCL for connectivity.

The creation of an M2M root key during the M2M service bootstrap procedure requires a pre-provisioned identifier that is typically assigned during the manufacturing process of an M2M device. Figure 3 shows a simplified version of the main activities that occur between M2M entities during either the service bootstrap or service connection setup procedures. Identifiers that are associated with M2M devices/gateways are considered inside the scope of current standards [36], except for the pre-provisioned identifier. Therefore, in this chapter we will present related work where it is explained how this identifier is formed in order for it to be globally known and unique in the whole M2M system. We will also discuss its format and its (dis)advantages.

In [29] authors propose how to construct a pre-provisioned identifier that can uniquely identify an M2M device regardless of used communication technologies. Namely, the same M2M device can establish communication using more than one communication technology. However, sent data has to be associated with the M2M device no matter which technology is used.

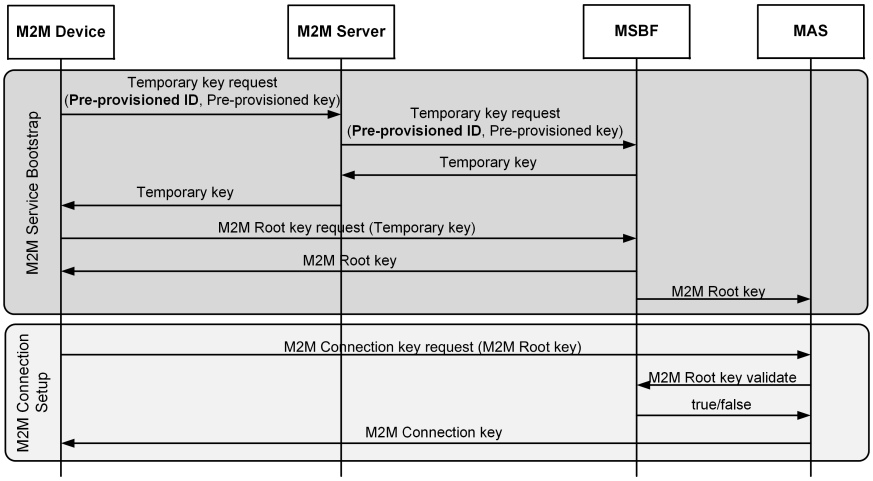

Fig. 3. Simplified view on the M2M connection establishment [29]

Because of all aforementioned, the identifier proposed in [29] is of a variable length where its header denotes the list of communication technologies that some M2M device supports, and bytes that follow after the header denote corresponding communication technologies identifiers/addresses (c.f. Figure 4). The authors specified the order of only the first four bits of the identifier header (i.e. Bluetooth on bit 0, Wi-Fi on bit 1, Wireless M-Bus on bit 2, and ZigBee on bit 3) leaving out the meaning of the other bits[4]. If the value of the bit in the header is equal to zero, then it means that this M2M device does not support that com-

[4] The authors said that the order of other bits should be standardized by standardization organizations active in the field of M2M systems.

munication technology. Opposite to that, if the value of the bit in the header is one, the M2M device supports the corresponding communication technology.

The last bit of every byte in the identifier header is reserved for denoting whether it is necessary to expand the header with other bytes. Namely, if device supports communication technologies that are not specified within the first byte, then the last bit of the first byte has to be equal to one, otherwise it should be zero. Bytes that come after the identifier header consist of addresses of communication technologies that M2M device supports. For communication technologies specified within the first four bits (e.g. Bluetooth, Wi-Fi, Wireless M-Bus, and ZigBee), those identifiers/addresses are of different lengths: six bytes for Bluetooth and Wi-Fi, less than a byte for Wireless M-Bus, and eight bytes for ZigBee.

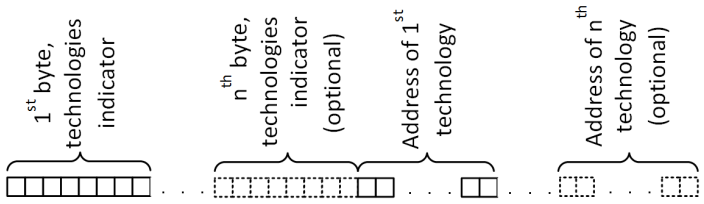

Fig. 4. Proposed identification scheme [29]

By this proposal identifier is tightly coupled with the hardware. In the proposed identifier there must be at least one communication technology while the upper limit does not exist. This is one of its advantages, together with the fact that because of the variable length, it is no longer than it should be. However, on the other hand the question is what if communication module of some M2M device gets broken and has to be replaced with another one. Since every communication module has its unique identifier, the new module will not have the same identifier as the old one had. Consequently, the M2M device will change it. The advantage of this approach is that tempering communication on device can be detected. The disadvantage is that if device is broken and needs to be changed by other device, the M2M system will detect that as new device and some extra management effort that would link the old identifier with the new one, thus preserving information about this change of the M2M device identifier, is needed. If the device identifier would be layer 3 identifier then this additional effort is not needed, but the startup procedure needs to deal with that. Also, detecting the device tempering after defining identifier is not possible.

We may note that, if the M2M device credentials have been pre-provisioned (e.g. in UICC), in practice no M2M service bootstrap is needed. Otherwise, depending on relationship between the access network provider and the M2M service provider, the M2M service bootstrap may by either assisted or without assistance from the network layer.

In the former, the access network involves security operations, and access network security credentials may be used for service layer bootstrapping. In particular, the ETSI standard defines procedures for M2M service bootstrap assisted by various network access mechanisms, from which Kmr and (possibly) M2M node and SCL identifiers result and are provisioned to the MAS. Mechanisms are defined for the support of Generic Bootstrapping Architecture (GBA) capable M2M gateways and devices, for bootstrap based on Extensible Authentication Protocol (EAP) [45] using EAP for GSM Subscriber Identity Module (EAP-SIM) [46] or EAP method for UMTS Authentication and Key Agreement (EAP-AKA) [47] credentials and for EAP-based network access authentication, in which network access authentication is utilized for the generation of the Kmr key, which is applicable to networks using EAP-based mutual authentication and key agreement (e.g. Wi-Fi, WiMAX).

As for M2M service bootstrap without access network assistance, the access network layer only transports M2M traffic but does not provide any security for such traffic. In this context, the standard currently defines an access network independent M2M service bootstrap mechanism. This mechanism is aligned with the M2M architecture, ensures mutual authentication between the D/G M2M node and the M2M service bootstrap server with perfect forward and backward secrecy in respect to the negotiated Kmr root key. The mechanism supports authentication based on Identify-Based Authentication Key Exchange (IBAKE) [48], EAP, or TLS [49] to protect the exchanged authentication messages. For certificates-based M2M service bootstrap, bootstrap credentials shall include globally unique identifiers, and the device must also be configured with the information related to certificate validation in the context of a root trust anchor (Certification Authority, CA). Certificate validation also supports validation of certificates using certificate revocation lists (CRL) or the Online Certificate Status Protocol (OCSP) [50].

Permanent security credentials bootstrapped using the M2M service bootstrap procedures are stored in a safe location, M2M MAS. This server can be an Authentication, Authorization and Accounting (AAA) server. On the other end, security credentials established for D/G M2M nodes during the same procedure are stored in a secured environment domain at the node itself. The credentials negotiated in the context of the bootstrap procedure are used for mutual authentication and secure communication between the D/GSCL on the D/G M2M node and M2M service capability layer in the network (NSCL), as well as for authorization of access to specific M2M services, and related accounting/billing functionality.

The M2M service connection takes place between a D/G M2M node and the network domain of the M2M service provider, for which an M2M root key (Kmr) has been established as previously discussed. The M2M service connection enables mutual authentication of the mId end points and key agreement, in the context of an M2M service connection session established between the two mId end points. This connection is optionally encrypted using a Kmc key agreed between both communicating parties, which enables the establishment

of a secure session (encrypted communication) via mId. This includes the protection of subsequent SCL registration messages, as we describe next. Other than mutual authentication and M2M connection key agreement, M2M service connection also enables reporting of integrity validation security attributes for those M2M service providers that support integrity validation, and the optional establishment of secure sessions using encrypted communication over mId. After successful establishment of M2M service connection, SCL registration and subsequent M2M (secure) communication can take place. The architecture also defines a mechanism for the usage of TLS-PSK for establishing Kmc between a D/G M2M node and network M2M node with the assistance of MAS (similarly to EAP), with whom the D/G M2M node has already established Kmr. As for M2M service connection based on GBA, it applies to scenarios where the access network and M2M service providers have a trust relationship (also possibly being the same entity). GBA may thus also support mutual authentication and key agreement between the D/G M2M node and the network M2M node.

3.4 SCL Registration

The SCL Registration procedure enables a D/GSCL to register with an M2M service capability layer in the network (NSCL), in order to be able to use M2M services offered by the network. A pre-requisite for this registration is an M2M service connection that has been established by the M2M service connection procedure previously discussed, with all the related security requirements verified. After SCL registration, information exchanged between M2M devices, gateways, and the network may be protected with data origin authentication, integrity, replay protection, confidentiality, and privacy. The architecture defines three distinct ways the mId may be secured: via access network layer security, via channel security, or via object security. Access network layer security is viable if the underlying access network is already physically or cryptographically secured, and in this case a careful study must be conducted in order to properly align end-points at the access network layer with the M2M network layer. As for channel security, a secure communication channel may be established between the D/G M2M node and the network M2M node, to protect all the exchanged information. This channel can be established after the M2M service connection procedure takes place. Finally, an M2M implementation may also rely on object security by applying security at the protocol payload level. We may also note that more than one security approach may be combined in a given M2M deployment.

4 Research opportunities and standardization challenges in M2M systems

Regarding the previously discussed characteristics and functionalities of M2M systems, while also considering the ETSI M2M architecture, we are able to identify various open issues that may motivate future research and standardization efforts in M2M communication and security, as we proceed to discuss.

4.1 Research opportunities: Communication and identification

Current standards have proposed a hierarchical organization of entities in three layers: M2M servers, M2M gateways, and M2M devices, and they have defined M2M architecture only for those devices that support IP. In real-world systems these assumptions are not always true. Namely, IP protocol, in a way, may be too complex for small devices such as sensors due to their energy constraints. On the other end, one can also consider existing optimizations enabling IP for particular classes of sensing devices, such as with IPv6 over Low power Wireless Personal Area Networks (6LowPAN) [51]. Moreover, many applications in distributed systems rely on flat, i.e. peer-to-peer architecture between devices that can communicate using different communication technologies (e.g. sensor networks, ad-hoc networks). There are two possible directions of how this problem could be solved. One possible solution is to achieve communication between devices without an M2M gateway regardless of communication technology, and other would be to modify current applications in such a way that they work even when devices do not communicate directly, but through an M2M gateway. Both approaches have their advantages and disadvantages that are discussed in the following sections.

Finally, in a wider context, research opportunities in M2M systems include achieving management functionalities. Namely, due to a huge number of interconnected and heterogeneous M2M entities, it is challenging to manage M2M systems. Some of the functionalities that need to be supported are: fault management, configuration management, and software upgrade, then mobility management, account management, and security management [52]. Fault management should support periodic, on demand and/or event driven reporting of faults that occurred in M2M systems. Configuration management and software upgrade should enable changing of M2M device states remotely (e.g. configuring M2M devices for reporting with specific parameters). Mobility management should provide support for both vertical and horizontal handoff, regardless of the used communication technology. Account management should include charging schemes for M2M service usage for both prepaid and postpaid. Finally, security management should provide end-to-end security.

Non-IP based protocols. Standards dominantly use IP based communication between M2M devices. The problem with IP based communication is that some M2M devices (e.g. small sensors) are resource constrained, so they cannot implement Transmission Control Protocol/Internet Protocol (TCP/IP) stack (standards call them legacy devices). M2M gateways serve as proxies between M2M devices and the rest of the network, regardless of the communication technologies these devices support. Therefore, gateways implement interworking proxy functions (e.g. Gateway Interworking Proxy, GIP) that allow communication between IP and non-IP M2M devices by providing interfaces for otherwise incompatible protocol stacks. Each of the currently available M2M gateway implementations (e.g. Actility Cocoon [53], OpenMTC [54]) implement these functionalities, as it was initially conceived in ETSI standards [28], and are constantly developing

new solutions. The second problem is that for some low level technologies there are no existing standards that provide TCP/IP stack over that technology. There are numerous initiatives that are trying to overcome this deficiency, so they are developing simplified IP stacks over existing low energy protocol suites, many of which are still in draft phase: Routing Over Low power and Lossy networks (ROLL) [55], 6LoWPAN [56], ZigBee over IP [57], Bluetooth Low Energy (BLE) over IP [58]. These initiatives are accompanied by the appropriate extensions on the application layer (e.g. Constrained Application Protocol (CoAP) [59] as a low energy replacement for Hypertext Transfer Protocol (HTTP)). In this setting main challenges are how to identify such devices and how to incorporate them into the existing standards. Pre-provision identifier proposed in [29] is an effort in that direction, because its variable header length supports any number of the existing communication technologies an M2M device can currently use (e.g. Bluetooth, ZigBee, Wireless M-Bus), but also provides room for future expansions to new communication technologies. It is important to emphasize that proposed identifier does not change anything in the original communication standard, because it only takes over the existing addressing/identification schemes and incorporates them in its fields.

Peer-to-Peer M2M device communication. Applications in distributed systems can be generally classified into two categories: *specific purpose* applications (e.g. collecting information about electricity consumption) and *general purpose* applications (e.g. identification of new M2M devices), where the specific purpose applications usually rely on services provided by the general purpose applications. Up till now, general purpose applications have been developed for homogeneous and peer-to-peer like distributed systems. Without a doubt, these applications will need to be adjusted according to M2M specificities. Namely, most of these applications rely on the fact that devices can communicate among themselves directly without gateways. However, this scenario is not covered in current M2M standards. Introducing this kind of communication brings different problems such as how to discover nearby M2M device (e.g. by broadcasting or by contacting gateway), how to define parameters of communication (e.g. protocols on top of lower layer communication), how to identify and achieve trust among different M2M devices without M2M gateways, etc. Moreover, a prerequisite of most of the applications is that devices are the same, but in M2M systems that is usually not the case. Consequently, applications need to be changed in such a way they would work in M2M systems where devices are diverse.

M2M devices always accessible. In some cases M2M devices need to be always accessible from M2M gateway. If such M2M device (e.g. M2M device is smart meter measuring gas consumption at home) is powered by battery, then continuous energy consumption is unacceptable because battery needs to be charged or replaced in a short period, which is costly. In this setting, M2M devices usually go to sleep mode, sleep for some time, wake up, do the job, and go back to sleep mode. To be able to switch devices between sleep and operating

modes, the information about its context which characterizes the situation of an entity, has to be stored somehow. One of the possible solutions is by using Rich Presence Information (RPI) which indicates the willingness or the ability of a user (or device in an M2M network) to communicate with other users (devices) in a communication network [60]. In this proposal, RPI is stored on an M2M gateway, and all negotiations are carried out by agents. Server Agent initiates communication with devices through M2M gateways. M2M gateway forwards requests only to those devices which are, according to RPI data, available. The problem is what if M2M gateway needs to communicate to that M2M device while it is in sleep mode. In that case, M2M gateway needs somehow to wake up the device from sleep mode. The research challenge is how to wake up the M2M device and that M2M device in sleep mode consumes as small amounts of energy as possible. There are also problems with security in such settings, as data obtained from sleeping devices (and possibly cached at the M2M gateway) should be properly authenticated and protected in regard to its confidentiality and integrity, among other security-related requirements.

4.2 Research opportunities: Security and privacy

The various challenges posed to the addressing of security in M2M may benefit from a paradigm shift in how the various security requirements are guaranteed. For example, scenarios without a security infrastructure in place (contrary to the previously analyzed ETSI M2M architecture) may consider classic security solutions side-by-side with new decentralized and distributed approaches. As in some scenarios M2M systems may be unable to derive definitive conclusions about the identity or intents of other devices, security mechanisms may need to consider compromises between the enforcement of definitive security controls and the acceptance of controlled risks [30].

Other aspects are trust and privacy, which may motivate the design of new security mechanisms and approaches. Distributed and autonomous trust management and verification mechanisms will be required to support autonomous M2M device-to-device identification and authorization [34]. M2M applications may also require the control of privacy and liability, as previously discussed. For some M2M applications (in the context of the IoT) the user will require to be able to control the amount of personal information exposed to third parties, for instance in maintaining privacy while exposing personal records in healthcare applications. On the other end, other M2M applications may require that some of that information is available in case of necessity, for instance with M2M vehicular applications in case of traffic accidents. Challenges also exist in the usage of M2M architectures such as the one from ETSI, side-by-side with emerging communication and security solutions.

Heterogeneity and resource constraints of M2M systems. Given the limitations on the computational capabilities of many sensing and actuating platforms, security technologies must be developed to cope with heterogeneous

devices, some of which may be very limited. In this context, further mechanisms are required to integrate such devices in M2M environments supported by architectures with the characteristics of the ETSI M2M architecture. For example, applications using passive Radio-Frequency IDentification (RFID) tags are unable to support security mechanisms requiring the exchange of many messages and communication with servers on a network domain. Lightweight solutions for symmetric and asymmetric cryptography [61][62], which have been proposed in recent years, provide a useful guidance in this context. The heterogeneity of sensing/actuating M2M devices may also be addressed by security approaches at higher layers of the protocol stack or at the middleware, in line with the approach previously discussed.

Identification, authorization, and trust. Identification and authorization of M2M devices in a dynamic and autonomous world will pose serious research challenges. Authentication mechanisms should work side-by-side with distributed trust management and verification mechanisms. Any two M2M devices should be able to build and verify a trust relationship with each other, and this problem is certainly more challenging in environments without a security infrastructure in place. Trust will be an important requirement for designing new identification and authentication systems for M2M. As authentication is related with identification, M2M systems will probably need to incorporate some type of secure identifier, tying information identifying the device or application with secret cryptographic material. Current proposals point to the usage of X.509-based certified secure identifiers, for example using IEEE 802.1AR [63], or on the other end of self-generated uncertified secure identifiers, also called cryptographically generated identifiers [64] [65]. As M2M systems require that privacy is balanced against disclosure of information, new authentication mechanisms relying on appropriate secure identifiers and incorporating privacy-preserving mechanisms are required. This aspect may also be incorporated in new trust computation mechanisms, as the evaluation of the risk in accepting communication with a partially unknown device may also consider the level of privacy accepted for an M2M application.

As distributed and autonomous trust mechanisms will be required for M2M environments, trust must be established on an M2M device from the start. Local state control via secure boot (local trust validation) may be enforced for M2M devices, similarly to the mechanisms previously analyzed in the context of the ETSI M2M architecture. This secure boot may allow the establishment of a trusted environment providing a hardware security anchor and a root of trust, from which different models for trust computation may be adopted. In this context, the Trusted Computing Group (TCG) [66] has proposed autonomous and remote validation models. Autonomous validation (using for example smart cards storing authentication secrets) presents the problem of requiring costly in-field replacements of compromised devices. Remote validation presents problems related to scalability and complexity, regarding limitations of M2M devices.

A promising avenue for research in this field may be that of semiautonomous validation [34]. Semiautonomous validation combines local validation with remote validation, meaning that a device is able to validate trust for another device and communicate with a trusted third-party in situations of absolute necessity (in many environments such third party may not be available at all). Distributed semiautonomous trust verification mechanisms are therefore necessary for M2M environments. The previously described M2M architecture from ETSI also incorporates the usage of secured and trusted environment domains, controlled by the M2M service, as a cornerstone for the (secure) usage of security credentials on M2M devices and gateways.

Anonymity and liability. As previously discussed, anonymity and liability are two interrelated security requirements for M2M applications. Such requirements are not only related with security, but they are also vital for the social acceptance of many applications envisioned for M2M. Anonymity is necessary as applications may only be accepted if the user is guaranteed to have a certain degree of protection of its personal (or other) information. Liability is a deeply related requirement, as other applications may require access to private information in case of necessity, for example for legal purposes. As anonymity will be required in M2M, research can target the applicability of light weighted formal anonymity models such as k-anonymity [67] to M2M environments. Possible alternative approaches are the development of mechanisms for data transformation and randomization. Intrusion detection will also be relevant for autonomous M2M environments. Autonomous and cooperative methods allowing the early detection of node compromises may be the path to follow in this domain [68].

4.3 Standardization challenges

As previously discussed, standardization on communication and security mechanisms and architectures to support M2M environments are essential for the evolution of M2M as a fundamental cornerstone of the IoT. Thus, research and standardization must symbiotically address security as a fundamental enabling aspect of future M2M applications. Research challenges must consider the efforts of standardization on M2M, and technologies developed by standardization bodies need to address security from the start. Standardization is also important because M2M can replace proprietary technologies such as Supervisory Control And Data Acquisition (SCADA) [69][70] in the future. Unlike SCADA, M2M devices are able to push data to a server and M2M also works with standardized technologies. Such factors will push towards the replacement of proprietary technologies with M2M solutions in the long term. This will open a huge market for M2M, but also many security and management challenges.

In the context of standardization, it is reasonable to expect that as the technology matures new opportunities and bridges between work being developed at different working groups may appear. For example, current efforts at the Internet Engineering Task Force (IETF) [71] include the work being developed

at 6LoWPAN [72] [51], Routing Over Low power and Lossy networks (ROLL) [55], and Constrained RESTful Environments (CoRE) [73] working groups. Work developed at these groups seeks to define a stack for the usage of Internet communication protocols on low-energy area networks, which qualify as M2M area networks in the context of the previously discussed ETSI M2M architecture. Considering the previously described security procedures and mechanisms defined in this architecture, one can investigate how security mechanisms being developed for 6LoWPAN-based communication protocol may fit in the ETSI M2M architecture described in this chapter. Possible approaches are mechanisms proposed to integrate security at the network layer [74] using 6LoWPAN communication technologies. This integration is also related with the evolution of sensing devices to adopt the usage of Representational state transfer (REST) web-services approaches, such as the IETF CoAP [59], which currently lacks security mechanisms. Also in the context of end-to-end communication between Internet and constrained M2M devices using CoAP-based REST communication, techniques have been proposed to support security in an efficient manner at the transport [75] and application layers [76]. Such mechanisms thus may provide the support for end-to-end or indirect (via an M2M gateway) communication with 6LoWPAN-based sensing devices, in the context of M2M systems enabled by architectures with the characteristics of the ETSI architecture.

Finally, we may refer that engineering and research challenges also reside in the design of new sensing platforms for M2M devices. A security co-processor may enable efficient cryptographic operations in low-end sensing and actuating platforms, and more complete hardware-based security solutions can also be used, such as the one currently proposed with Trustchip [77]. New platforms may be designed to allow efficient computation of security algorithms appropriate to M2M applications, and security-related data may be stored (as defined in the ETSI M2M architecture) using secure hardware modules with the characteristics of the Trusted Platform Module (TPM) proposed by the TCG [66] group. The usage of such a module allows the secure binding of the device identification and secret cryptographic information. As the usage of such hardware modules may not be economically feasible, research should address the design of alternative software secure-storage solutions and its impact on the overall security of M2M devices and applications.

In conclusion, various characteristics of M2M devices and applications will demand a new approach on how security and management is addressed. The ubiquity and autonomous nature of many M2M applications will dictate that many security-related decisions are performed in the absence of a centralized and trusted security infrastructure. In other contexts, such an infrastructure may be available as defined by ETSI. Considering the autonomous nature that is expected for many M2M applications, aspects such as autonomous communication, privacy and liability (among others) will pose major challenges to engineering and research. Many of the required security mechanisms will operate autonomously and in a distributed fashion.

Concluding remarks

As previously discussed, efforts from organizations such as ETSI, as well as from other organizations and researchers, are currently enabling architectures and communication technologies capable of supporting future complex M2M systems accommodating various application scenarios. M2M systems are primarily characterized by heterogeneity, i.e. their reliance on great variety of standardized communication technologies. Therefore, we propose a new pre-provisioned device identifier, transparent of the underlying communication technology. Nonetheless, there are still many open issues in the area of achieving end-to-end communication interoperability between various technologies, added by the fact that M2M systems integration into IoT environment will require further convergence on communication, data, and service levels. As in the current Internet architecture, security will remain of prime importance and will in fact represent a fundamental enabling factor of most of the current applications of M2M communication. In this chapter we have analyzed how communication establishment and security are addressed in the context of the ETSI M2M architecture [28], and also what are the main open issues that will require further efforts from both research and standardization. As we have observed, M2M architectures support valuable technologies and rules for the implementation of M2M systems, but in the context of the IoT we may expect that not all applications will be ruled by a single reference architecture. For example, applications may be unattended and M2M devices may require autonomous secure communication, and in this scenario a well-defined security infrastructure may be absent. Also in this context, important issues of security such as trust, privacy, and anonymity demand for immediate efforts from research and standardization bodies.

Acknowledgments

This work was supported by two projects: "Machine-to-Machine Communication challenges" funded by Ericsson Nikola Tesla, Croatia, and iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER.

References

1. V. Galetic, I. Bojic, M. Kusek, G. Jezic, S. Desic, and D. Huljenic, "Basic Principles of Machine-to-Machine Communication and its Impact on Telecommunications Industry," in *Proceedings of the 34th International Convention MIPRO*, 2011, pp. 89–94.
2. "SingTel M2M, http://info.singtel.com/large-enterprise/about-m2m," visited on March 30, 2014.
3. 3GPP, *TR 22.868 Study on Facilitating Machine-to-Machine Communication in 3GPP Systems*, 2008.

4. D. S. Watson, M. A. Piette, O. Sezgen, and N. Motegi, "Machine-to-Machine (M2M) Technology in Demand Responsive Commercial Buildings," in *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings*, 2004, pp. 1–14.
5. B. Emmerson, "M2M: The Internet of 50 Billion Devices," *Win-Win*, pp. 19–22, 2010.
6. D. Boswarthick, O. Hersent, and O. Elloumi, *M2M Communications: A Systems Approach*. Wiley-Blackwell, 2012.
7. ETSI, *TR 102 897 Use Cases of M2M Applications for City Automation*, 2012.
8. ——, *TR 102 857 Use Cases of M2M Applications for Connected Consumer*, 2013.
9. ——, *TR 102 691 Smart Metering Use Cases*, 2010.
10. ——, *TR 102 732 Use Cases of M2M Applications for eHealth*, 2013.
11. ——, *TR 102 898 Use Cases of Automotive Applications in M2M Capable Networks*, 2013.
12. "M2M World of Connected Services - Beecham, www.m2m.com/docs/DOC-1221," visited on March 30, 2014.
13. L. Atzoria and A. I. nad Giacomo Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
14. D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of Things: Vision, Applications and Research Challenges," *Ad-hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.
15. J. Gantz, *The Embedded Internet: Methodology and Findings*, 2009.
16. D. Evans, *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*, 2011.
17. M. Hatton, *The Global M2M Market in 2013*, 2013.
18. M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From Today's INTRAnet of Things to a Future INTERnet of Things: A Wireless- and Mobility-Related View," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.
19. D. Katusic, M. Weber, I. Bojic, G. Jezic, and M. Kusek, "Market, Standardization, and Regulation Development in Machine-to-Machine Communications," in *Proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks*, 2012, pp. 1–7.
20. "ETSI M2M, www.etsi.org/technologies-clusters/technologies/m2m," visited on March 30, 2014.
21. "oneM2M, www.onem2m.org," visited on March 30, 2014.
22. "Association of Radio Industries and Businesses, www.arib.or.jp/english," visited on March 30, 2014.
23. "Telecommunication Technology Committee, www.ttc.or.jp/e," visited on March 30, 2014.
24. "Alliance for Telecommunications Industry Solutions, www.atis.org," visited on March 30, 2014.
25. "Telecommunications Industry Association, www.tiaonline.org," visited on March 30, 2014.
26. "China Communications Standards Association, www.ccsa.org.cn/english," visited on March 30, 2014.
27. "Telecommunications Technology Association, www.tta.or.kr/English," visited on March 30, 2014.
28. ETSI, *TS 102 690 M2M Functional Architecture*, 2011.
29. D. Katusic, P. Skocir, I. Bojic, M. Kusek, G. Jezic, S. Desic, and D. Huljenic, "Universal Identification Scheme in Machine-to-Machine Systems," in *Proceedings of the 12th International Conference on Telecommunications*, 2013, pp. 71–78.

30. D. Jiang and C. ShiWei, "A Study of Information Security for M2M of IOT," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering*, 2010, pp. 576–579.
31. D. Djenouri, L. Khelladi, and N. Badache, "A Survey of Security Issues in Mobile Ad-hoc Networks and Sensor Networks," *IEEE Communications Surveys*, vol. 7, no. 4, pp. 2–28, 2005.
32. J.-H. Cho, A. Swami, and R. Chen, "A Survey on Trust Management for Mobile Ad-hoc Networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 562–583, 2011.
33. Y. Wang, G. Attebury, and B. Ramamurthy, "A Survey of Security Issues in Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 8, no. 2, pp. 2–23, 2006.
34. I. Cha, Y. Shah, A. U. Schmidt, A. Leicher, and M. V. Meyerstein, "Trust in M2M Communication," *IEEE Vehicular Technology Magazine*, vol. 4, no. 3, pp. 69–75, 2009.
35. ETSI, *TS 102 689 M2M Service Requirements*, 2010.
36. ——, *TS 102 921 mIa, dIa and mId Interfaces*, 2012.
37. "3rd Generation Partnership Project, www.3gpp.org," visited on March 30, 2014.
38. "3rd Generation Partnership Project 2, www.3gpp2.org," visited on March 30, 2014.
39. "Open Mobile Alliance, www.openmobilealliance.org," visited on March 30, 2014.
40. "Broadband Forum, www.broadband-forum.org," visited on March 30, 2014.
41. Open Mobile Alliance, *OMA Device Management Protocol*, 2008.
42. Broadband Forum, *TR-069: CPE WAN Management Protocol*, 2011.
43. 3GPP, *TR 23.888 System Improvements for Machine-Type Communications*, 2012.
44. I. Bojic, G. Jezic, D. Katusic, S. Desic, M. Kusek, and D. Huljenic, "Communication in Machine-to-Machine Environments," in *Proceedings of the 5th Balkan Conference in Informatics*, 2012, pp. 283–286.
45. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, *Extensible Authentication Protocol*, 2004.
46. H. Haverinen and J. Salowey, *Extensible Authentication Protocol Method for Global System for Mobile Communications Subscriber Identity Modules*, 2006.
47. J. Arkko and H. Haverinen, *Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement*, 2006.
48. V. Cakulev, G. Sundaram, and I. Broustis, *IBAKE: Identity-Based Authenticated Key Exchange*, 2012.
49. T. Dierks and E. Rescorla, *The Transport Layer Security Protocol Version 1.2*, 2008.
50. S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol*, 2013.
51. "IPv6 over Low power WPAN, https://datatracker.ietf.org/wg/6lowpan," visited on March 30, 2014.
52. S. Pandey, M.-J. Choi, M.-S. Kim, and J. Hong, "Towards Management of Machine-to-Machine Networks," in *Proceedings of the 13th Asia-Pacific Network Operations and Management Symposium*, 2011, pp. 1–7.
53. "Actility Cocoon, http://cocoon.actility.com," visited on March 30, 2014.
54. "The OpenMTC Vision, www.open-mtc.org/index.html," visited on March 30, 2014.
55. "Routing Over Low power and Lossy networks, http://datatracker.ietf.org/wg/roll/charter," visited on March 30, 2014.

56. J. Hui and P. Thubert, *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, 2011.
57. "ZigBee IP Specification Overview, www.zigbee.org/Specifications/ZigBeeIP/Overview.aspx," visited on March 30, 2014.
58. J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, *Transmission of IPv6 Packets over Bluetooth Low Energy*, 2013.
59. Z. Shelby, K. Hartke, and C. Bormann, *Constrained Application Protocol*, 2013.
60. M. Kusek, I. Lovrek, and H. Maracic, "Rich Presence Information in Agent based Machine-to-Machine Communication," *Proceedings of the 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems*, pp. 321–329, 2013.
61. X. Xiong, D. S. Wong, and X. Deng, "TinyPairing: A Fast and Lightweight Pairing-Based Cryptographic Library for Wireless Sensor Networks," in *Proceedings of the IEEW Wireless Communications and Networking Conference*, 2010, pp. 1–6.
62. O. Delgado-Mohatar, A. Fúster-Sabater, and J. M. Sierra, "A Light-Weight Authentication Scheme for Wireless Sensor Networks," *Ad-hoc Networks*, vol. 9, no. 5, pp. 727–735, 2011.
63. "IEEE 802.1AR, Secure Device Identity, www.ieee802.org/1/pages/802.1ar.html," visited on March 30, 2014.
64. R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, *Host Identity Protocol*, 2008.
65. T. Heer and S. Varjonen, *Host Identity Protocol Certificates*, 2011.
66. "Trusted Computing Group, www.trustedcomputinggroup.org," visited on March 30, 2014.
67. L. Sweeney, "k-Anonymity: A model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.
68. R. Lu, X. Li, X. Liang, X. Shen, and X. Lin, "GRS: The Green, Reliability, and Security of Emerging Machine-to-Machine Communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 28–35, 2011.
69. S. A. Boyer, *SCADA: Supervisory Control and Data Acquisition*. International Society of Automation, 2009.
70. V. M. Igure, S. A. Laughter, and R. D. Williams, "Security Issues in SCADA Networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006.
71. "Internet Engineering Task Force, www.ietf.org," visited on March 30, 2014.
72. J. Granjal, J. Sa Silva, E. Monteiro, J. Sa Silva, and F. Boavida, "Why is IPSec a Viable Option for Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 802–807.
73. "Constrained RESTful Environments, https://datatracker.ietf.org/wg/core," visited on March 30, 2014.
74. J. Granjal, E. Monteiro, and J. S. Silva, "Network-Layer Security for the Internet of Things using TinyOS and BLIP," *International Journal of Communication Systems*, pp. 1–14, 2012.
75. J. Granjal, E. Monteiro, and J. Sa Silva, "End-to-End Transport-Layer Security for Internet-Integrated Sensing Applications with Mutual and Delegated ECC Public-Key Authentication," in *Proceedings of the IFIP Networking Conference*, 2013, pp. 1–9.
76. J. Granjal, E. Monteiro, and J. S. Silva, "On the Feasibility of Secure Application-Layer Communications on the Web of Things," in *Proceedings of the IEEE 37th Conference on Local Computer Networks*, 2012, pp. 228–231.

77. “Trust Chip Mobile Device Security, www.koolspan.com/trustchip,” visited on March 30, 2014.