NOKIA

M2M security
Ensuring device security for the Internet of Things

Strategic White Paper

The Internet of Things (IoT) is fast transforming the way people live and work. New solutions are appearing in various industries, ranging from connected cars to connected homes, cities, and industries, all driven by advancements in machine-to-machine (M2M) communications. The billions of “things” that are connected in these M2M networks have the same security requirements as mobile phones, computing devices, and consumer electronics devices. However, due to the autonomous operation of M2M devices, there are additional security challenges that are not fully addressed by the current security management solutions used for mobile phones.

This paper outlines the unique challenges involved in providing robust security for M2M devices. It discusses the requirements and components of a security solution that can ensure secure management of these devices, regardless of the simplicity of the devices or the complexity of the architectures in which they operate.

NOKIA

Contents

NOKIA

Introduction

The Internet of Things (IoT) is quickly becoming a massive market. Wireless connections are moving beyond mobile phones and Internet-connected computing and consumer electronic devices to billions of everyday "things," from parking meters, thermostats, and traffic cameras to streetlights, wearable health monitors, and vehicles. Cisco Systems Inc. claims the market for these hyper-connected devices will top $19 trillion in the next eight years [1]. Other forecasts indicate that machine-to-machine (M2M) devices will account for more than 40 percent of connected devices in the United States by 2018 [2].

New devices are connecting each day with the help of rapid advancements in machine-to-machine (M2M) communications. These advancements are driving new solutions in a wide range of areas, from connected vehicles to connected homes, cities, and industries. Gartner analysts recently forecast that the IoT would connect almost five billion things by the end of 2015, and 25 billion – or three for every person on the planet – by the end of 2020 [1]. The growth in connected devices translates into an increased security risk. A study released by Mocana previously estimated that there would be 500 million hackable endpoints by 2015 [3].

The devices connected in M2M networks have many of the same security requirements as mobile phones, computing devices, and consumer electronics. Device connections need to be safe; data must be safeguarded; and privacy must be maintained to ensure that the devices are not subject to hacking, manipulation, and other network threats. However, the work involved in ensuring the security of M2M devices includes challenges that are not fully addressed by the security management solutions and methods used for mobile phones and consumer electronics.

After deployment, M2M devices require remote management. The vast majority of the sensors, cameras, meters, monitors, actuators, and controllers that make up M2M networks are unmanned and may not even have a conventional user interface. Many are meant to operate unattended for extended time periods, with no physical human interaction. Without a proper user interface, it may not be possible to support user confirmation or user alerts. In addition, many M2M devices are embedded into vehicles, homes, buildings, bridges, tunnels, roads, utility poles and pipelines. It is impractical or even impossible to return them for updates, and in-person maintenance is often prohibitively expensive.

For devices that are part of a mission-critical application – such as pipelines, medical monitoring systems, or building security applications – alerts or faults must be processed in real time to ensure seamless service. Just as important, corrective actions must be initiated automatically, either from or to the M2M

devices, based on security policies. Finally, the data transmitted to and from M2M devices needs to be auditable to enable accuracy, governance, and regulatory compliance.

Security management architecture for the Internet of Things

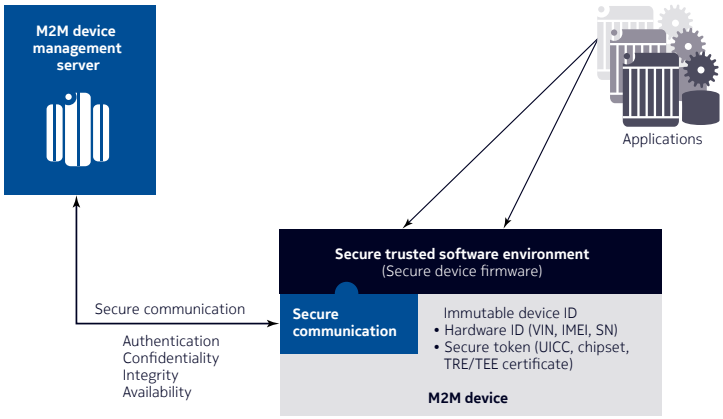
Security management architectures for M2M devices should include three key components. First, every device needs a secure "immutable identity." This identity is the key to establishing the device's credentials when it connects to the network. Managing device identity – authenticating or validating the identity of the device, authorizing or registering the device for access, and managing the specific privileges and services available to the device – is a critical first step to ensuring the security of that device in the field. Adding further complexity is the fact that the vast majority of M2M devices will not be connected directly. Instead, they will connect in a local-area PAN/LAN configuration, aggregating over a common gateway that provides an egress point to a mobile communication provider's network. The identity of each device, including the gateway devices themselves, needs to be managed. It is also necessary to be able to securely manage groups of devices connected via a gateway.

Figure 1. Three key components of M2M device security management

The second key component for M2M device security is establishing a secure communication channel for device management. This channel must provide a means to authenticate the device's identity, ensure the confidentiality of the data going to and from the device, protect the integrity of the data going to and from the device, and ensure device availability.

Finally, the device needs a trusted software environment to ensure the ongoing security of the M2M applications that use, drive, and run on the device. A secure firmware package running at the lowest level of the software stack is the mechanism used by nearly all phones, computing devices, and consumer electronics in the field today. Secure device firmware, signed by an immutable device identifier and securely delivered to the device over a secure communication channel, provides the most secure trusted software environment for M2M devices [4][5].

Figure 2. End-to-end M2M device security management



The combination of three key components – a secure device identity in the form of an immutable device ID, a secure communication channel that ensures the authentication, confidentiality, integrity and availability of the device during management actions, and a secure trusted software environment in the form of secure device firmware - creates an end-to-end security management architecture for M2M devices.

Managing device identity

Before M2M devices can be managed, they need an immutable identity that can be used to establish and ensure a secure communication channel for the device. One part of this identity is usually a hardware identifier, for example, the International Mobile Equipment Identity (IMEI) for mobile phones, the vehicle identification number (VIN) for vehicles, or the serial number for a host of other types of devices.

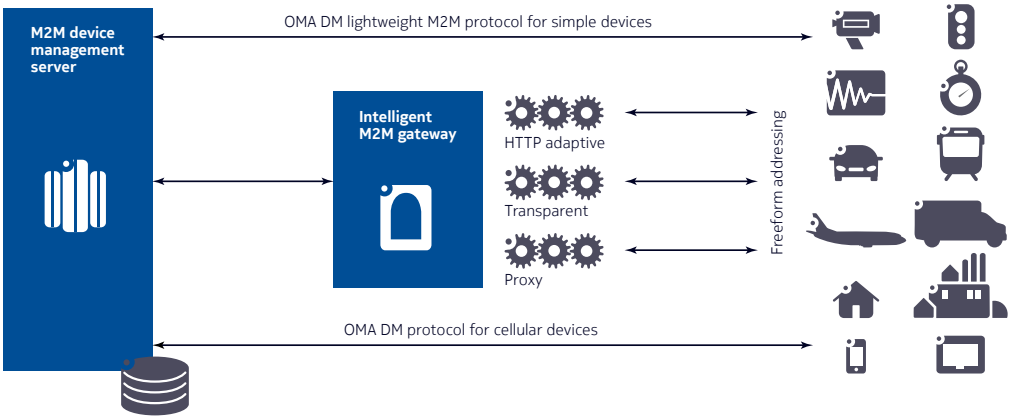
The other part of this identity is a secure hardware- or software-based token that stores keys, certificates, or other information that can be used to authenticate or verify the specific device's identity. These tokens can be:

- Smart-card-based, for example, a Universal Integrated Circuit Card (UICC) keystore
- Chipset-based, for example, Intel® Identity Protection Technology (IPT) or an ARM® keystore
- Software-based, for example, a keystore containing digital certificates in a logically separate
- Trusted Runtime Environment (TRE) or Trusted Execution Environment (TEE) on the device [6][7].

The tasks involved in managing device identity include authenticating the identity of the device, authorizing or registering the device for access to servers, and applying policies that determine specific privileges. One complexity of M2M architectures is the fact that there can be a hierarchy of devices to manage. While some devices will be managed directly, large numbers of devices – including sensors, actuators, and controllers – will be managed through a gateway device.

For example, in a connected home application, the contact sensors, keypad, motion sensors, smoke alarms, video cameras, door controllers, light controllers, thermostats, and water controllers all connect through a wireless control unit that serves as a gateway to the network. The identity of each device needs to be managed individually. In addition, groups of devices connected through the control unit that is acting as a gateway may also need to be managed as a group.

Figure 3. Managing M2M devices in complex architectures



Secure M2M device communication

Whether M2M devices connect through a gateway or directly, providing a secure communication channel for the management of these devices is critical to ensuring overall security. The basic principles of secure device communication are:

- Authentication: Ensuring that the device is legitimate
- Confidentiality: Ensuring that only addressed recipients can see data going to and from the device
- Integrity: Protecting data from being altered, intercepted, or manipulated at any point during communication or generation
- Availability: Protecting the device from malicious activity during communication that affects availability

Authentication

During authentication, the device is verified as a legitimate device and authorized or registered with specific management servers. Because they are largely unmanned, M2M devices must be authenticated using the device's immutable identity – a combination of the device's hardware identifier and a hardware or software security token on the device that contains security keys, certificates, or hash values. A wide range of cryptographic and non-cryptographic methods can be used during authentication, including:

- Random pre-shared key
- X.509 (ITU-T public key certificate)
- Raw public key
- Bootstrap International Mobile Subscriber Identity (IMSI) pre-shared key
- SMS authentication
- Transport Layer Security/Datagram Transport Layer Security (TLS/DTLS)
- Keyed-hash message authentication code (HMAC)

M2M devices can be bootstrapped at the factory with secure credentials. These credentials can also be established via key exchange. For M2M devices connected through a gateway, the gateway and the management server must collaborate to authenticate and authorize end devices using a secure mutual authentication process. In addition, the gateway can be used to bootstrap raw devices. It can also engage in periodic challenge-response messaging, where either the gateway requests a certificate from the device or the device requests a certificate from the gateway to validate identity during management communications.

Confidentiality

Encrypting the management communication channel ensures that only addressed recipients can see the data going to and from M2M devices during management actions. M2M devices that connect directly to the management server, including gateway devices, can use industry-standard mechanisms such as TLS and HTTPS to encrypt the data flow between the client on the device and the management server.

This encryption ensures that the contents of communications between the device and the management server cannot be intercepted, read, or forged by an imposter. Between non-cellular devices and a gateway, DTLS can use pre-shared keys or tokens, X.509 public keys, or raw public keys to ensure data confidentiality during communication. When a gateway is bootstrapping raw devices, it can use TLS or other standard mechanisms to ensure confidentiality while provisioning certificates on the end devices.

Integrity

M2M device data must be protected from being altered, intercepted, or manipulated by unauthorized entities at any point in the chain of communication. To uphold device-to-gateway integrity, a secure digest or hash algorithm (for example, SHA-256) can be used to ensure that the transmitted data has not been tampered with. For direct device-to-server or gateway-to-server communications, mechanisms such as HMAC (which can also be used for authentication) or nonce (a one-time random number code) can be used to ensure the integrity of the data during communication. For

critical devices in more sensitive applications, there are various additional cryptographic methods that can be used to further ensure that the data is not manipulated.

Availability

The M2M management architecture must include mechanisms to protect devices at all levels from malicious activity that affects their availability. This can be a particularly serious problem when the M2M system includes devices connected via a gateway. Individual devices could send malicious data that affects gateway availability, and thus the availability of all of the other devices connected via that gateway. They could even overwhelm the gateway by sending excessive data, resulting in a "data implosion" problem. In the extreme, a group of M2M devices could launch a denial-of-service (DoS) attack on the gateway or even the entire network.

To mitigate these issues, M2M management servers and M2M gateways can collaborate using capabilities such as:

- Continuous monitoring
- A robust system of alerts and alarms
- The ability to shut off ports on the gateway to prevent malicious devices from affecting the entire network
- The ability to trigger proactive actions to correct problems before they result in system-wide failures

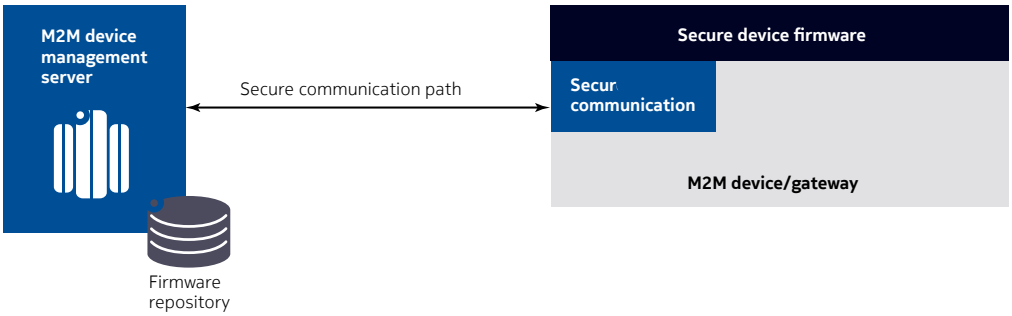
Moreover, M2M devices are not necessarily always on. Many connect with the server only reactively or periodically. The management server needs to be able to securely trigger communication with M2M devices at all levels of a potentially complex hierarchy of roles and capabilities.

Secure trusted device software environment

M2M devices need a secure, trusted software environment for running M2M applications. Nearly all mobile phones, computing devices, and consumer electronics in the field today use secure device firmware, running at the lowest level of the software stack, as this trusted software environment. In addition, the process of securely updating device firmware in the field is a well-understood and well-established process. It is used to provide everything from bug fixes to new capabilities, features, and services for a host of devices, including mobile phones, tablets, computing devices, TVs, DVD players, game consoles, cameras, navigation equipment, stereo equipment, printers, and networking equipment [4][5].

This same process can be used to securely update billions of M2M devices used in a wide range of systems, from connected home applications to connected vehicles, buildings, and cities. For many M2M devices, the ability to securely update firmware may be more critical than it is for mobile phones. There may be a large number of devices connected via a gateway, and some of them may be inaccessible, for example, utility meters on a rooftop or sensors on a pipeline. Some M2M devices may be manufactured or installed in a way that makes it impractical or prohibitively expensive to attempt a physical update or replacement. Examples include the components in a digital home solution or a vehicle infotainment system. The M2M system may also include components that must be updated quickly for regulatory or other reasons, such as the sensors or monitors in a health management system.

Figure 4. Secure firmware update for M2M devices



A secure firmware update package is generally signed by the manufacturer or supplier using a hardware ID or security token. The firmware updates on the files themselves vary considerably in terms of structure and format. For example, Microsoft® provides firmware updates for its devices as Portable Executable/Common Object File Format (PE/COFF) files [8]. Executable and Linkable Format (ELF) files are used in most UNIX® or Linux® devices [9]. Mach-O is used in Apple® OS X and iOS devices.

Many equipment vendors provide firmware updates as “delta” files, which are created as the output of a difference generator and contain only the changes to the file. Regardless of the format, the firmware update package is securely delivered to the device over the secure communication channel established between the management server and device. This ensures a secure, up-to-date, trusted software environment on the device.

NOKIA

OMA M2M enablers: Standards-based secure M2M device management

The Open Mobile Alliance™ (OMA) provides a number of capabilities that support the secure, standards-based management of M2M devices from end to end. The OMA Device Management (DM) V1.3 specification and related enablers provide protocols for creating a secure communication channel for managing M2M devices. These enablers include:

- OMA DM Gateway Management Object (GwMO), which enables the management of devices that are not directly accessible to the management server, for example devices connected through a gateway
- OMA M2M Device Classification, a common framework for classifying M2M devices independently of vertical markets
- OMA DM Lightweight M2M protocol, a protocol for managing low capability, lightweight devices, such as sensors, actuators and controllers.

Finally, the OMA DM Firmware Update Management Object (FUMO) defines the specification for managing tree objects, DM commands, messages, and the mechanism for securely updating device firmware. It provides a secure trusted device software environment on the device. This tried-and-true technology is more than proven in the marketplace. OMA estimates that more than 1.6 billion commercially deployed devices use the OMA DM FUMO enabler [10].

Conclusion

All of the things that connect to M2M networks and systems have the same security management requirements as mobile phones, computing devices, and consumer electronics. However, M2M devices present additional challenges because they are often unmanned or inaccessible, and because they function in increasingly complex architectures. To address these challenges, an end-to-end security management architecture for M2M devices must include three key components: secure identity management for each device, a secure communication channel between the management server and the devices, and a secure trusted software environment on each device.

Identity management ensures that each M2M device is authenticated and authorized or registered with the management server. A secure communication channel enables the authentication of M2M devices and uses appropriate cryptographic and non-cryptographic mechanisms to maintain the confidentiality and integrity of the data going to and from the devices. It also provides mechanisms to protect devices from malicious activity such as DoS attacks that could affect device availability. Secure device firmware provides

the most secure, trusted software environment for running software and applications on M2M devices — an environment that has been proven over the years on billions of mobile and consumer electronic devices. OMA provides standards-based capabilities that support the secure management of M2M devices from end to end through the OMA DM specification and related M2M enablers.

NOKIA

Abbreviations

COFF	Common Object File Format
DM	Device Management
DoS	denial of service
DTLS	Datagram Transport Layer Security
ELF	Executable and Linkable Format
FUMO	Firmware Update Management Object
GwMO	Gateway Management Object
HMAC	keyed-hash message authentication code
HTTPS	HTTP Secure
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
IPT	Identity Protection Technology
ITU-T	ITU Telecommunication Standardization Sector
LAN	local area network
M2M	machine-to-machine
OMA	Open Mobile Alliance
PAN	personal area network
PE	Portable Executable
SHA-256	Secure Hash Algorithm 256
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TRE	Trusted Runtime Environment
UICC	Universal Integrated Circuit Card
VIN	vehicle information number

NOKIA

Sources

1. “Gartner Says 4.9 Billion Connected ‘Things’ Will Be in Use in 2015”. Gartner press release, November 11, 2014. http://www.gartner.com/newsroom/id/2905717
2. “Mobile Cybersecurity and the Internet of Things”. CTIA white paper, 2014. http://www.ctia.org/docs/default-source/default-document-library/ctia-iot-white-paper.pdf
3. Scherf, K. “Security and Device Management for the Internet of Things” Parks Associates, February 19, 2014. http://www.parksassociates.com/blog/article/security-and-device-management-for-the-internet-of-things
4. “Safe and Secure Firmware Upgrade for AT91SAM Microcontrollers”. Atmel Corporation application note, September 2006. http://www.atmel.com/Images/doc6253.pdf
5. Shade, Loren K. “Implementing Secure Remote Firmware Upgrades” Allegro Software Development Corporation white paper, May 2011. http://www.allegrosoft.com/wp-content/uploads/Secure-Firmware-Updates-Paper.pdf
6. “Security in M2M Communication — What is secure enough?”. Gemalto M2M GmbH white paper, 2013. http://www.m2m-alliance.com/fileadmin/user_upload/pdf/2013/Whitepaper/gemalto_whitepaper_secure_M2M_communication_web.pdf
7. Cha, I., Shah, Y. et al. “Security and Trust for M2M Communications”. InterDigital, Inc.http://www.interdigital.com/wp-content/uploads/2012/08/WWRF_22_M2M_Security.pdf
8. Microsoft PE and COFF Specification, Microsoft. http://msdn.microsoft.com/en-us/gg463119.aspx
9. Executable and Linkable Format (ELF) Specification. http://www.skyfree.org/linux/references/ELF_Format.pdf
10. Pittampalli, E. “Management and Provisioning of M2M Devices and Applications”. Presentation delivered on behalf of Open Mobile Alliance at the M2M Evolution Conference, January 29–31, 2014. http://openmobilealliance.org/wp-content/uploads/2014/02/M2M_Evolution_Jan_2014_oneM2M.pdf
11. “Global M2M Network Security Market 2014–2018”. Infinity Research Limited report, August 2014. http://www.reportlinker.com/p02316647/Global-M2M-Network-Security-Market.html

Nokia Oyj
Karaportti 3
FI-02610 Espoo
Finland
Tel. +358 (0) 10 44 88 000

Product code: PR1510015183EN