An Approach for Secure Edge Computing in the Internet of Things

Markus Endler, Anderson Silva and Rafael A.M.S. Cruz
Department of Informatics, PUC-Rio
Rio de Janeiro, Brazil
Email: {endler, anderson}@inf.puc-rio.br and ramscrz@gmail.com

***Abstract*—In this paper, we discuss the singular characteristics of the Internet of Things (IoT) in regard to security issues and present a security architecture with special focus on Edge networks and Smart Things. The architecture is based on the assumption that data integrity, authentication, and service/device access control are the most important aspects that have to be guaranteed for safe operation of highly decentralized and heterogeneous IoT systems. The architecture is made concrete as an extension of the ContextNet middleware for IoT. This middleware has a cloud and a mobile component and advocates the use of smartphones - or devices that execute smartphone platforms - as the universal gateways between cloud-hosted IoT services and the Smart Things. In addition to general architecture, the other major contribution is a protocol for establishing a secure connection between the Smart Things and these mobile gateways.**

***Index Terms*—Internet of Things (IoT); Edge Computing, Mobile Internet of Things, Data Integrity, Data**

I. Introduction

The growing density and size of networks formed by heterogeneous devices and many users with different profiles poses new challenges to cyber-security. One notable example of this is the Internet of Things (IoT), where cyber-offenses, data theft and manipulation and DDoS attacks take place in very dynamic, polymorphic and very distributed scenarios. Currently, IoT technology already supports connection of millions of smart devices and meters, and by 2025 it shall support more than 50 billion connected devices. Thus, there are too many smart devices to be controlled, and any device with minimal computing and communicating capabilities can initiate or participate in cyber-attacks without leaving an easily detectable trace. For example, on October 10th 2016 occurred the first DDoS attack on DNS severs that was carried out by surveillance cameras and home routers. Since nearly all internet services depend on DNS, this attack affected major websites such as Twitter, Pinterest, Reddit, GitHub, Etsy, Tumblr, Spotify, PayPal, Verizon, Comcast, and the Playstation network. The attackers employed hundred thousands of such devices infected with malicious code to form a massive botnet. Even though some of these devices have limited computing capacity, they can generate massive amounts of bogus traffic to swamp targeted servers. And as more of our daily objects and vehicles become interconnected other cyber-attacks that disable or change the original function of smart objects may become more frequent in next future.

The following section motivates the Internet of Mobile Things and our approach to enable Smart Things connectivity. In the next section we present related work on security for IoT, with special focus on security of Edge Networks. Then, in Section IV we classify the types of security threats to IoT systems and in Section V we present our general architecture for common IoT system topologies. In Section VI we then present the main elements of the ContextNet middleware, and in Section VII we show how we will instantiate our general architecture in the components of the ContextNet middleware. We close the paper with our conclusion in Section VIII.

II. Internet of Mobile Things

The Internet of Things is evolving towards a heterogeneous network including a mix of IP-based connectivity and an array of short-range, low-power wireless technologies (e.g., Bluetooth, NFC, LoRa), the latter used by peripheral devices in the Edge networks, the Smart Objects. Moreover, according to Francis daCosta [2], IPv6 does not solve all IoT problems because management, rather than addressing and routing, are the biggest challenge of IoT. In fact, IP-based protocols will neither be supported by the vast majority of

Smart Objects, nor will their over-provisioned and reliable services be suited to most IoT applications.

Considering that personal mobile devices (smart phones and tablets) and mobile Internet are becoming increasingly ubiquitous, more affordable and powerful, and that opportunistic and intermittent connectivity will become common place in a world filled with mobile, wearable and embedded technology, such personal mobile devices become good candidates for propagator nodes (i.e. gateways) to the Internet to the simpler and energy-constrained smart objects. This led us to propose the concepts of Internet of Mobile Things (IoMT) and *Mobile Hub (M-Hub)*[15], a general mobile middleware service responsible for discovering and opportunistically connecting to Smart Objects (with sensors and/or actuators) accessible only through short-range, low power WPAN technologies to the Internet. Moreover, the M-Hub can provide context information about the objects that are in its vicinity, such as their approximate location, if they are in movement, which is the luminosity of the space, using its embedded sensors.

In order to support IoMT applications, we developed the ContextNet middleware[4], which consists of a scalable software infrastructure for enabling cloud-based communication with mobile devices (SDDL), and parallel data stream processing facilities in the cloud/cluster. Stream processing is also supported in the Mobile-Hub, which runs as background services on Android smart phones, Beaglebone kits, and will soon be available also for iPhones.

III. Related Work

Many discussions about the security threats and solutions for IoT, and in particular for Edge Networks, are appearing at technology forums and industry white papers, but only to a lesser extend in academic publications. One reason for this may be that ITC industry is the one that is first facing the critical security concerns when designing their products or IoT services. Nevertheless, some authors describe interesting edge network issues and solutions which we summarize in the following.

Work [11] focuses on smart things and study the IoT network security issues in the smart home, health care and transportation domain, and then present a taxonomy of security attacks. Their goal is to make the IoT application developer more aware of the risks of security flaws so that better protection and safeguard mechanisms are incorporated. Work by Kulkarni et al [5] also addresses security of smart things but focuses on how data about/from smart things can be made secure using cryptography. On the other hand, [13] presents a general threat model that can be used to develop a security protection methodology for IoT services against cyber-attacks and shows that an Anomaly Behavior Analysis (ABA) Intrusion Detection System (ABA-IDS) can detect and classify a wide range of attacks against IoT sensors. Work [9] presents an analysis on the current status and concerns of IoT security, and proposes some countermeasures such as authentication measure, trust establishment, federated architecture and security awareness to reinforce security.

In [7] the vulnerabilities of Gateways and Edge networks are presented, and in this scope the author discusses three specific security threats: Network exposure of the Smart Objects, Man-in-the-Middle and Impostor Attacks. In this paper we present a security architecture suited to a general, common topology of IoT systems that comprises smart things/objects, (mobile and stationary) gateways and IoT services executing in a cloud. To this end, we address the threats to each component of the general topology and we propose the use of common security mechanisms for each component of the system and for the communication links and protocols between them.

IV. Security Threats to IoT Systems

According to [6], although the threats in the IoT environment might be similar to those in the traditional IT environments, the overall impact could be significantly different because the targets are abundant and cover many different industry segments. The potential impact could span from minor irritant to grave and significant damage to the infrastructure and loss of life. Herein, we propose to group the threats that can compromise the security of IoT systems into two distinct groups: (GT1) threats to the operation of the entities of the IoT system; and (GT2) threats to the communication between the entities of the IoT system. The main entities in a generic IoT topology are the networked Smart things/objects (i.e. Sensors and Actuators), the gateways, which can be mobile or stationary, and the cloud-based IoT services, as suggested by Figure 1.

Among the threats of the first group are:

(GT1.1) Threats to the operation of the cloud services The processing system hosted on the cloud (or data-centers), in general is responsible for analyzing the data acquired by the smart things and stored in the cloud which, in turn,

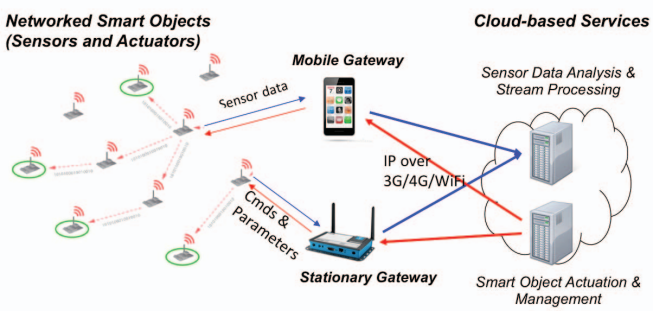

Fig. 1. Generic Topology of an IoT System

is formed by real and virtual machines that run operating systems, network services, web applications and data base management systems. These elements are usually exposed on external shared networks and, as such, are subject to well-known targeted attacks that aim to gain privileged or unprivileged access, steal information or disrupt systems, among the achievement of others threats.

(GT1.2) Threats to the operation of the gateways The operating system of the gateway is, in general, a network service exposed on external shared networks and, as such, is also subject to well known targeted attacks against such systems that aim to gain privileged or unprivileged access, tamper control information or disrupt systems, among the achievement of others threats. **(GT1.3) Threats to the operation of the Smart Things** The operating/control system of smart things is, in general, quite simple and stored in firmware. This simplicity often makes these entities vulnerable to targeted attacks that aim to gain privileged or unprivileged access, tamper the firmware or produce false data, among the achievement of others threats.

Among the threats of the second group are:

(GT2.1) Threats to the communication between the cloud and the gateways The communication between the cloud and gateway, in general, takes place over an external shared network and is subject to well-known targeted attacks that aim to monitor the content of the messages (passive attack), intercept and tamper messages, masquerade or block messages,

(GT2.2) Threats to the communication between the gateways and smart things The communication between the gateway and smart things, in general, may take place over a point-to-point dedicated wired network (more secure) or over a shared wired or wireless network (less secure). Both communication forms are also subject to well-known targeted attacks that aim to monitor the content of the messages (passive attack), intercept and tamper messages, masquerade or block messages.

V. Security Architecture for the General Topology of IoT Systems

The main insight regarding security of IoT systems is to realize that there is nothing essentially new or different in those networked systems when it comes to security threats that can compromise its operation. According to [14], in order to determine the need of a security control, we first have to analyze the security risks, which means to evaluate their likelihood, technical impact and harm to the business or organization. As an example, in a scenario where we need to acquire data about the soil moisture or the environment temperature in order to keep the well-being of the plantation, we can accept the risk to use low-cost smart things with simple security controls. On the other hand, in another scenario, where we have to monitor the same kind of data related to the reactor of a nuclear power plant, we will eventually need special smart things with the necessary processing capabilities to implement classic and well-known high security standards.

This section discusses classic security solutions to reduce the impact of the threats addressed in section IV. We do not exhaust the analysis of all risks and all threats, but rather point out the classic security solutions that are applicable in the general topology of IoT systems, typically composed by three main actors: cloud, gateway and smart things. Targeted attacks on the cloud (or a computer network and its systems, or a datacenter) that aim to achieve the threats pointed out at GT1.1, are commonly defeated using a concept of protection known as defense in depth [12], [10]. This concept consists of multiple layers of security controls composed of a series of defensive mechanisms such that if one mechanism fails, another will already be in place to thwart an attack: Briefly, it creates a perimeter through defensive mechanisms such as network firewall, proxy firewall, intrusion detection system (IDS) or intrusion prevention system (IPS), among others [12]. The defense in depth concept can reduce the risks of successful attacks to an acceptable security level.

Targeted attacks on the gateway cause the threats pointed out in GT1.2 and are commonly defeated using, whenever possible, the strategy of defense in depth, considering the gateways as the elements of the internal network that require

protection. If we consider the gateway directly connected to an external shared network, it should support the execution of some components of the perimeter defined by the defense in depth, such as, firewall and IDS/IPS (Figure 2), to defeat targeted attacks on the application services. These components run as local processes on the gateway and act to reduce the risks of successful attacks to an acceptable security level. Nevertheless, like any other network element accessible by the external shared network, the gateway is subject to denial of service (DoS) attacks that aim to overload the target and disrupt its service. We claim that no classical solution can eliminate this threat at all and we have to assume this risk to provide service to the external shared network. Regardless of the position of the gateway, this entity shall provide two basic application services to reinforce the security of IoT systems (Figure 2): (i) smart thing control service; and (ii) access control service. The first one aims to provide a control service that operates as a firewall proxy by intermediating the communication between the cloud and the smart things, and, thus, it can inspect the protocol messages in order to detect and block malformed ones that could harm the smart things. The latter one aims to offer a robust access control service to validate authentication credentials and restrict access to authorized users.

This smart thing control service shall implement a smart thing control protocol to provide monitoring and management of the smart things. This protocol shall enable the cloud to collect data, receive notifications and change configuration of the smart things. The basic operation of this service is to map the control protocol messages to the appropriate messages of the whole bunch of smart thing protocols and Low-Power Network/Personal Area Network Protocols used by the smart things. The access control service shall provide one or more authentication schemas to validate authentication credentials in order to restrict access to authorized users. This service may use a local credential database (e.g. a login and password file), an external authentication service (e.g. RADIUS, LDAP or any other robust authentication methods. The authentication schema is inherent to the IoT application and is beyond the scope of this paper to indicate the best of them for each case.

Targeted attacks on the smart things that aim to achieve the threats pointed out at GT1.3, are very difficult to be addressed because of the low-load processing capacity of the smart things. Even so, must support some sort of minimum-security mechanisms, as follows.

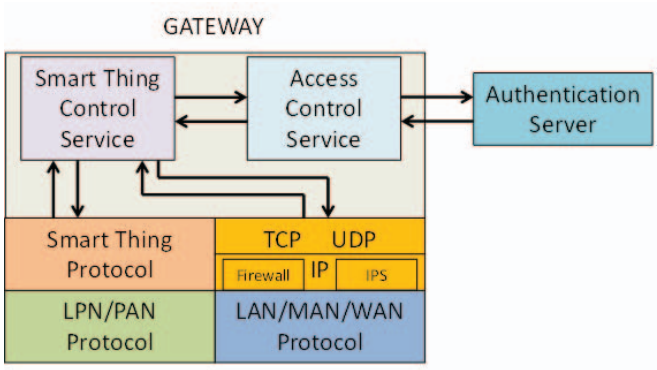

Fig. 2. Proposed gateway architecture for IoT security

Because of the high risk of connecting a smart thing to an external shared network, one important security policy to adopt, in any circumstance, is to restrict the communication of the smart things to the gateway. This security measure will avoid direct attacks from the outside network to the smart things. In the other way round, as an access control measure it is important to restrict the access of smart things to the control service that runs on the gateway. In turn, this service shall be responsible to validate users and computers that need to communicate with the smart things by way of a robust access control services. This security measure intends to prevent attempts of unauthorized users to gain access to the smart things. Another important security measure is to use a security mechanism on the smart things to check the integrity and authenticity of the firmware before booting the system to prevent running a false or invalid code. This feature requires the smart things to support cryptographic algorithms in order to process the validation of the firmware code.

In order to reinforce the security in the smart things, we propose that these sensors/devices shall provide two distinct operating modes: (i) configuration mode; and (ii) service mode. The first one allows configuration actions such as the modification of operating parameters (e.g. signal strength, cryptographic keys, network address, authentication method) and updating of the firmware, among others. The latter one is the common operating mode in which the smart thing do what it is intended to do and allows data to be collected. As a security measure, the smart thing shall use an access control method before switching modes, such as validating a PIN (Personal Identification Number). Targeted attacks on the communication between the cloud and gateway that aim

to achieve the threats pointed out at GT2.1, are commonly defeated if we establish a secure association between these parties and use a secure communication to protect the messages of the control protocol. Herein, we can use common standards to address this requirement, such as IPsec (IP Security) [8] and TLS (Transport Layer Security) [3]. Both standards can reduce the risks of successful attacks to an acceptable security level.

Targeted attacks on the communication between the gateway and the smart things that aim to achieve the threats pointed out at GT2.2, could be commonly defeated if we consider that these smart things support the necessary processing load to execute the security mechanisms to control the integrity, authenticity and confidentiality of the messages of the communication protocol. However, smart things used for non-critical application must support some sort of minimum-security mechanisms; at least, they shall be able to check the integrity and authenticity of the messages of the communication protocol. Independently of the cryptographic algorithm adopted for any of the security mechanisms, the setting of the keys shall take place when the smart thing is operating in the configuration mode and inside a trusted environment. This security measure intends to avoid the capturing of the keys by a malicious person or thing during the setting process.

VI. Current Middleware for IoMT

The ContextNet middleware has a scalable mobile-cloud communication layer, *SDDL*, plus the mobile component *Mobile Hub*, which is responsible for discovering and connecting Smart Objects to the Internet.

A. SDDL

The Scalable Data Distribution Layer (SDDL) [16] is a communication infrastructure that connects mobile nodes to stationary nodes (i.e. the *SDDL Core*), executing in a cloud or a cluster. The stationary nodes may be application-specific servers, gateways for connection with the mobile nodes, or monitoring and control nodes for displaying the mobile nodes' current position on maps, managing and communicating with groups of mobiles, and sending message to individual mobile nodes.

The interested reader can download a VM with pre-installed SDDL, as well as find examples and tutorials for implementing SDDL-based applications in Java, Android and Lua[1].

B. The Mobile Hub

The Mobile Hub (M-Hub) [15] is a general-purpose mobile middleware that discovers Smart objects/things (S-OBJs), connects with them and enables communication and remote access to them through the SDDL Core. Thus, the M-Hub acts as a "proxy" bridging the gap between the Internet connection with the SDDL Core and low power WPAN connections to the S-OBJs. To enable the uniform discovery and connection with nearby S-OBJs using different WPAN technologies, we designed the *Short-range Sensing, Presence & Actuation (S2PA)* API, a generic and technology independent service at the M-Hub that defines a common API for the different low range wireless technologies (WPAN).

The *S2PA* defines some general interface methods which the plug-ins (for each WPAN technology) must implement: 1) Discovery of, and connection with S-OBJs; 2) Discovery of services provided by each S-OBJ; 3) Read and write of service attributes (e.g., sensor values, and actuator commands) and 4) Notifications about disconnection of S-OBJs.

VII. Security Architecture for the ContextNet Middleware

ContextNet middleware is an implementation of the classic general topology of IoT systems, where the Mobile Hub (M-Hub) plays the role of the generic IoT gateway. Thus, we can apply the security mechanisms proposed in section V to secure the main nodes of an IoT system and the communication between them.

A. Securing the Smart Object Service Broker

The first security approach aims to assure the security of the Smart Object Service Broker. This entity shall be hosted in the cloud (SSDL Core), which, in turn, is secured by a strong security perimeter between the external shared network and the internal network. The broker will benefit from the security provided by the components of the perimeter, such as firewall, IPS/IDS and VPN gateway. As a security measure, the Mobile Hub can only access the broker via the VPN gateway of the perimeter. This measure aims to secure the communication between the Mobile Hub and the broker. We consider using IPsec [8] because this is a legacy standard available in common mobile device (e.g. smart phones in general) and, besides that, it is transparent for the upper

[1]http://www.lac.inf.puc-rio.br/dokuwiki/doku.php?id=tutorial

layers, thus, there is no impact to application services, such as the ones provided by SDDL.

B. Securing the Mobile Hub

The second security approach is to assure the security of the ContextNet IoT gateway, the Mobile Hub. Herein, we consider implementing the two services proposed in section V to enforce access control, integrity and authentication: the *Smart Thing Control Service* and the *Access Control Service*. These are two background services that work in tandem with Mobile Hub's S2PA and the Connection Service [15], as shown in Figure 3).

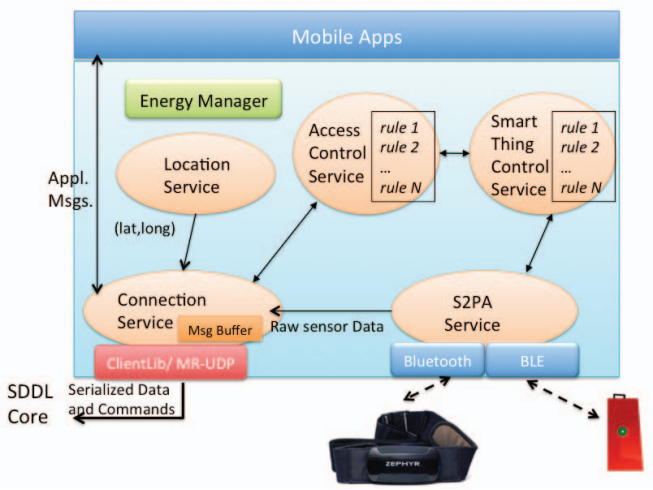

Fig. 3. Security Services in the Mobile Hub implementation

The Smart Thing Control Service operates as a firewall proxy, inspecting all control protocol messages and blocking malformed messages that could harm the Mobile Hub and, consequently, the Smart Things. We consider using the SNMP (Simple Network Management Protocol) standard as the smart thing control protocol mainly because it provides an architecture capable of monitoring and management of very diverse/heterogeneous devices, which is precisely the case for Smart Things. The control service operates as an SNMP Proxy Agent and provides a MIB (Management Information Base) with management objects that represents data that can be collected from manageable devices and actions that can be executed on them, in this context, the smart things. In short, the Smart Thing Control Service creates a security layer to avoid the misuse of the smart things.

The Access Control Service provides one or more authentication schemas to validate authentication credentials in order to restrict access to authorized users. This service may use a local credential database (e.g. a login and password file), an external authentication service (e.g. RADIUS or LDAP) or any other robust authentication methods. And only authorized users (or managers) can send SNMP messages to the smart thing control service.

C. Securing the M-Hub Communications

To secure the communication between the SDDL core and the Mobile Hub we are using TLS version 1.2 [3]. This legacy secure communication standard enforces the use of controls to check the integrity and authenticity of the messages, and, when necessary, guarantee also communication confidentiality.

To secure the communication between the Mobile Hub (M-Hub) and the Smart Things/Objects (S-OBJs/ST), we assume that the latter have at least the power to support the necessary processing load to control the integrity, autenticity and confidenciality of the WPAN messages with HMAC and RC4. To this end, we are considering the use of some simple cryptographic algorithms, such as HMAC or other basic encryption with some legacy symmetric cryptographic algorithms to ensure confidentiality control of sensitive data, such as RC4 [1]. Thus, a protocol must be designated to ensure an efficient secure communication. A strategy for access control must also be addressed for a consistent IoT security protocol for S-OBjs access, considering not every M-Hub Client might have the permission to access a given S-OBJ to read data and/or write data. For the proposed protocol being presented, the following elements must be acknowledged: the entities, namely the Smart Thing/Object(S-OBJ) and the M-HUB Clients and the SDDL Core (SDDL-C); the keys for the S-OBJs stored in the SDDL-C, namely K_auth_st and K_cipher_st; and the keys for the SDDL-C, namely K_auth_sddl-c. (as shown in 4 and **??**).

After running this protocol, a M-Hub and discovered S-OBJ will have established a secure association and will exchange messages that are signed using HMAC (OTP being the seed) and encrypted using Ksession. To avoid Man-in-the-Middle replay attacks, sequence and timestamp control of the message exchange must be implemented as well.

Step 1	Discovery Request message
The M-Hub advertises a message for detecting and discovering the IDs of the available in range S-OBJs.	
Step 2	**Discovery Response message**
The available in range S-OBJs share their respective IDs with the M-Hub responsible for the respective Discovery Request message.	
Step 3	**TLS Connection**
The M-Hub establishes a TLS connection with the SDDL-C through digital certification to ensure a secure communication.	
Step 4	**Get Authorization message through TLS**
The M-Hub provides the ID of the S-OBJ, and then the SDDL-C:	
4.1	executes checkAuthorization(ST ID, MHub ID) to verify if the M-Hub is authorized to access the S-OBJ.
4.2	executes getSTKeys(STID) to get the respective keys Kauth_st and Kcipher_st.
4.3	executes generateOTPChallenge(nounce) to generate the OTP challenge which must be a positive integer, bigger than 0 (zero) corresponding to the number of times a hash operation must be executed on the Kauth_st to generate the OTP (One-Time Password).
4.4	executes generateOTP(ST ID, M-Hub ID, OTPChallenge, Kauth_st) to generate the OTP (One-Time Password) which must be used to authenticate the M-Hub.
4.5	executes generateKSession(nounce) to generate the session secret key Ksession which must be used for the communication between the S-OBJ and the M-Hub.
4.6	executes generateST_Package_K(OTPChallenge, Ksession, Kcipher_st) to generate an encrypted (Kcipher_st key) package containing the OTPChallenge and the Ksession.
4.7	executes signST_Package_K(Kauth_sddlc) to sign the package ST_Package_K through HMAC. This package will be sent by the M-Hub to the S-OBJ.
Step 5	**Authorization Response message through TLS**
The SDDL-C provides the OTP, the Ksession, the ST ID and the ST_Package_K to the M-Hub that:	

Fig. 4. Secure Communication Protocol M-Hub/ S-OBJ (part 1)

5.1	executes storeKey(Ksession, OTP, ST ID) to store the pair <Ksession, OTP, ST ID> in its cache (timeout: 60s, if not used).
5.2	executes signHelloMessage(OTP, ST_ Package_K, M-Hub ID) to sign the M-Hub Hello message. The OTP is the seed for the HMAC.
Step 6	**Hello message**
The M-Hub sends the ST_Package_K and the M-Hub ID to the S-OBJ. It then:	
6.1	executes checkSignForPackage(ST_Package _K, Kauth_sddl-c) to verify the SDDL-C HMAC signature of the package ST_Package_K.
6.2	executes decryptPackage(ST_Package_K, Kcipher_st) to get the OPTChallenge and the Ksession.
6.3	executes generateOTP(ST ID, M-Hub ID, OTPChallenge, Kauth_st) to generate the OTP (One-Time Password) which must be used to authenticate the M-Hub.
6.4	executes checkSignForHello(OTP, ST_Package_K, M-HUB ID) to verify the HMAC signature of the M-Hub Hello message.
6.5	executes storeKey(Ksession, OTP, M-HUB ID), to store the pair <Ksession, M-HUB ID> in its cache.
6.6	executes signHelloAcceptedMessage(OTP, M-HUB ID), to sign the Hello Accepted message. This message will be sent by the S-OBJ to the M-Hub and the OTP is the seed of the HMAC.
Step 7	**Hello Accepted message**
The S-OBJ sends the signed message to M-Hub. It then:	
7.1	executes checkSignForHelloAccepetedMessage (ST ID) to verify the S-OBJ HMAC signature of the Hello Accepted Message.

Fig. 5. Secure Communication Protocol M-Hub/ S-OBJ (part 2)

VIII. Conclusion

In this paper we discussed the main security threats in a typical IoT system, considering each major component of an IoT system and the WWAN and the WPAN connections among them. We presented some of the related works to demonstrate a collective concern about the importance of security on IoT systems mainly because the threats herein might be similar to those in the traditional IT environments, but the overall impact could be significantly different because the targets are abundant and cover many different segments of our social and economic life. The threats that can compromise the security of IoT systems based on a general topology to acquire, aggregate and analyze data (smart thing-gateway-cloud) were settled into two distinct groups: (i) threats to the operation of the entities of the IoT system; and (ii) threats to the communication between the entities of the IoT system. Then, we demonstrated that classic security solutions can be directly applied to the cloud and gateway entities, and other classic solutions can be

adapted to the smart things or, when relevant, the smart things shall be adapted to the classic security solutions. Based on these thoughts, we used the proposed security mechanisms to design security mechanisms for ContextNet, our middleware for IoMT. This architecture requires the implementation of an *Access Control Service* and an *Smart Thing Control Service* on the gateways. We then we showed how this architecture will materialize in the Mobile Hub, the mobile middleware component of ContextNet. Although we are still in the process of developing the control services, we are confident that our security solution will not only be effective and efficient for our middleware, but that it can be easily adapted to other IoT systems and architectures.

REFERENCES

[1] Steve Burnett and Stephen Paine. *The RSA Security's Official Guide to Cryptography*. McGraw-Hill, Inc., New York, NY, USA, 2001.

[2] Francis daCosta. *Rethinking the Internet of Things: A Scalable Approach to Connecting Everything*. Apress Open, 2013.

[3] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2, 2008. Updated by RFCs 5746, 5878, 6176.

[4] M. Endler, G. Baptista, L. D. Silva, R. Vasconcelos, M. Malcher, V. Pantoja, V. Pinheiro, and J. Viterbo. Contextnet: Context reasoning and sharing middleware for large-scale pervasive collaboration and social networking. In *Workshop on Posters and Demos Track*, pages 2:1–2:2. ACM, 2011.

[5] S. Kulkarni et al. Internet of Things (IoT) Security. In *3rd IEEE Conference on Computing for Sustainable Global Development (INDIACom)*, March 2016.

[6] J. Frahim, C. Pignataro, J. Apcar, and M. Morrow. Securing the Internet of Things: A Proposed Framework, 2015.

[7] K. Holbrook. IoT Security in the Real World Part 1: Securing the Edge, April 2016. (Last access: January 14th, 2017).

[8] S. Kent and K. Seo. Security Architecture for the Internet Protocol, 2005.

[9] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of things (iot) security: Current status, challenges and prospective measures. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 336–341, Dec 2015.

[10] T. Mavroeidakos, A. Michalas, and D.D. Vergados. Security architecture based on defense in depth for Cloud Computing environment. In *INFORM Workshops*, 2016.

[11] M. Nawir, A. Amir, N. Yaakob, and O.B. Lyn. Internet of Things: Taxonomy of Security Attacks. In *3rd International Conference on Electronic Design (ICED 2016), Phuket, Thailand*, 2016.

[12] Stephen Northcutt, Lenny Zeltser, Scott Winters, Karen Kent, and Ronald W. Ritchey. *Inside Network Perimeter Security (2Nd Edition) (Inside)*. SAMS, Indianapolis, IN, USA, 2005.

[13] J. Pacheco and S. Hariri. IoT Security Framework for Smart Cyber Infrastructures. In *1st IEEE International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, 2016.

[14] T.R. Peltier. *Information Security Policies, Procedures, and Standards: Guidelines for Effective Information Security Management*. CRC Press, 2001.

[15] L. Talavera Rios, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. Silva e Silva. The mobile hub concept: Enabling applications for the internet of mobile things. In *12th IEEE Workshop on Managing Ubiquitous Communications and Services (MUCS 2015)*, pages 123–128, 2015.

[16] L.D.N. Silva. *A Scalable Middleware for Structured Data Provision and Dissemination in Distributed Mobile Systems*. Tese de doutorado, Departamento de Informática, PUC-Rio, July 2014.