Towards Secure and Context-Aware Information Lookup for the Internet of Things

Michalis Giannikos, Korina Kokoli, Nikos Fotiou, Giannis F. Marias and George C. Polyzos
Mobile Multimedia Laboratory
Deaprtment of Informatics
Athens University of Economics and Business , Athens, Greece
Email: {giannikos,kokolh,fotiou,marias,polyzos}@aueb.gr

***Abstract*—The Internet of Things (IoT) is receiving more and more attention from the research community. The opportunities it creates for new services make it an intriguing technological step. Based on this paradigm, objects (things) become recognizable and gain intelligence by exchanging information with their environment or by having access to information aggregated by other objects. Huge amount of information is expected to be generated, raising at the same time significant concerns: the generated information has to be organized in a meaningful way that facilitates the development of a variety of applications, and it should be protected against unauthorized access. In this paper we design and implement an architecture for retrieving information in the IoT. In the proposed scheme, access control policies limit information retrieval and information is organized in such a way that facilitates the development of context aware services.**

I. Introduction

The Internet of Things (IoT) is a dynamic global network infrastructure with self-automatic configuration capabilities based on standard and interoperable communication protocols. In this network the objects or "things" have identities, physical attributes and use intelligent interfaces to connect and communicate within social, environmental, and user contexts [1]. This novel paradigm fuses the digital and physical world by bringing different concepts and technical components together: wired and wireless sensor and actuator networks, standard and improved communication protocols, ambient intelligence, tracking technologies, even new business models. The IoT is expected to have great impact on several aspects of our everyday-life, enabling the development of a wide range of applications, including: sensor data aggregation, intelligent transportation schemes, business management, environmental monitoring, e-health and many others.

Information in the IoT is expected to be generated in vast amounts, creating the same time two significant questions that may constitute an overwhelming barrier to the expansion of this paradigm: how can information be protected against unauthorized access and how can information be organized in an effective way. The majority of the information generated in the IoT is expected to originate from small, pervasive sensors, therefore this information is deemed private and should be protected. Moreover this information is expected to be personalized as well as application and context specific, therefore its dissemination should be limited only to the interested parties.

In this paper we describe and implement an information lookup architecture for the IoT which tries to achieve a dual goal: to provide access control for the available information items, as well as, to facilitate the development of context aware services. In a nutshell, our architecture tries to achieve its goal by associating both users and information items with attributes and by having a lookup service performing information lookup based on these attributes. In contrast to similar approaches, in our scheme objects are associated with many information items, but users are allowed to access only a subset of them: the items that are associated with the same attributes as the users. This way items can be protected against unauthorized access, e.g., by associating an item with a user role, and can be context-specific, e.g., by associating an item with a context-specific property–such as user language, time, weather etc. The implementation of our scheme demonstrates its feasibility.

The rest of this paper is organized as follows. Section II presents the main components of our architecture, its interactions as well as its security functions. In Section III we describe the implementation of our architecture and we evaluate its security properties. Related work in this area is presented in Section IV whereas our conclusions and plans for future work are presented in Section V

II. Architecture overview

A. Components

The main components of the proposed architecture are: the *user middleware* running on a user's (portable) device, the *proxy server*, the *authoritative server*, and the information *storage points*. The user middleware is a piece of software that allows applications to interact with the architecture by abstracting its operations. The role of the proxy is to handle any complex operation–such as encryption and decryption of messages–on behalf of the user device. Proxies are required when user devices have limited processing power or power consumption constrains, otherwise they can be omitted. The authoritative server is the most important part of the architecture: this is the component which decides which information items a user can access. The storage points are servers with adequate storage capacity in which information items are stored. Storage points can not be accessed directly, they are rather accessed through an authoritative server. The same authoritative server can be used in order to access multiple storage points.

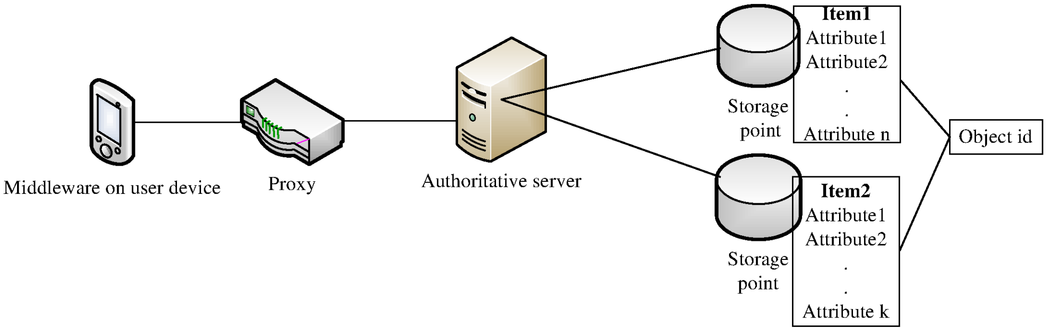

Fig. 1. Architecture components

All objects are uniquely identified by an identity, stored in a form that can be "sensed" by a user device (e.g., in a QR-code or in an RFID). Every object identity can be mapped to numerous information items, stored in various storage points. Finally information items are associated with *attributes*.

Figure 1 shows the main components of our architecture.

B. Operations

In order for a user to access an information item related to an object, he uses the middleware to retrieve the object identity, and sends a request to the appropriate authoritative server (through a proxy if necessary). Every user request includes the object identity, as well as, some user-specific attributes. These attributes define the user semantically, operationally and regionally, and are used by the authoritative server in order to perform access control, as well as, to provide personalized information to the user. These attributes are categorized in the following categories:

Static attributes: These are attributes stored encrypted in the form of a certificate, installed in the user's device. This certificate contains a unique user-id, a user role, as well as, various attributes that indicate personal information, the properties, the state, and the access rights of a user. To better understand these attributes consider the case of a system responsible for providing announcements required by a university during the exam period. These announcements can be accessed by scanning a QR-code, located in an announcement table. The static attributes of user may include his registration number (used as a user-id), his role–which can be for example *student*, *secretary*, or *professor*–his name, email, department, semester, and other personal information.

Dynamic attributes: These are attributes that are collected automatically by the middleware, running in the user's device, every time the user interacts with it. These attributes are provided by the device's sensors, and the device itself. They can be geo-location coordinates, time and date information, local and regional settings, display language, operator, available network interfaces information and statistics, available network connections, and many other.

Attributes are used in order to provide access control (e.g. in the previous case, a secretary may have the right to create, modify or delete an announcement, while students can only read it), as well as, in order to provide personalized experience to the user (e.g. students of the CS department will receive announcements only for their department).

Upon receiving a request, the authoritative server chooses the information item that should be responded back to the user. This decision is based on the object identifier and the attributes included in the user request. The item that eventually will be chosen by the authoritative server is the one to which maps to the object identifier included in the user request and is associated with the attributes provided by the user.

The association of an information item with attributes is achieved by defining a list of static and/or dynamic attributes, that a user request should include in order to access it. The purpose of this list is dual: to provide access control policy by preventing unauthorized access, and to provide context specific services. The former is achieved by including in the list attributes that are specific to a user, or to a group of users–such as user identity, user role etc–and the latter is achieved by including in the list attributes that are specific to the user context–such as date, location and others.

Figure 2 shows the message sequence of a typical transaction in our architecture

C. Security roles and mechanisms

In our architecture the following security roles are defined:

Information security director: This role is responsible for the system-wide security strategy. It creates the system information protection guidelines and strategies, and defines the static and dynamic attributes that will be used in the system. All the attributes created by this entity are consider to be well-known by all entities in the architecture.

Data custodian: This role is responsible for the authoritative server management and the enforcement of the access control policies that give access to information items. This role implements the security mechanisms that protect the confidentiality, integrity and availability of the stored informatio items, and it is responsible for creating the user

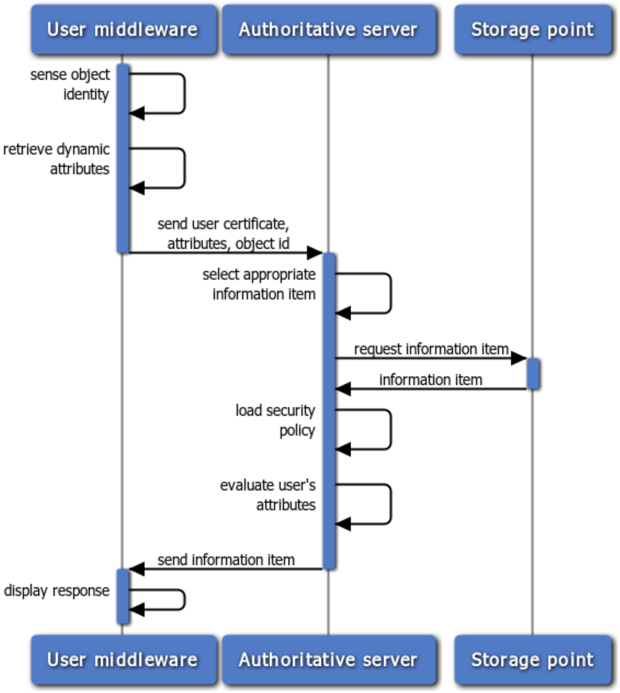

Fig. 2. Messages sequence diagram

certificates.

Authorized users: These are the end-users of the system that have been authorized, by data custodians, to access information items that are stored in the system. Through a registration process, that takes place in an authoritative server administrated by the custodians, users obtain a certificate and a personal password that enables them to use the system.

Moreover the following security related procedures are defined:

Certificate installation: For each user, identified by UId, data custodians a create a pair of public/private keys (PU_{UId}, PR_{UId}), as well as, a certificate $Cert$ that contains UId, PU_{UId}, and the user's static attributes ATT_s. Each certificate is signed by the private key of the data custodian. Moreover the public key of the data custodian, is considered well-known. In order for a user to obtain his certificate a straightforward procedure, whose security is ensured using the Pretty Good Privacy (PGP) mechanism [2], is followed: the user's middleware sends securely the UId and receives back from the authoritative server the certificate.

Secure communication: All messages should be securely transmitted. To fulfill this, we have designed the following protocol:

Let OI be the object identifier, as sensed or scanned by the middleware of the smartphone, $E_{PU}(m)$ the encryption of a message m with the public key PU, $D_{PR}(c)$ the decryption of the ciphertext c using the private key PR, $Sign_{PR}(m)$ the digital signature of a message m, $Ver_{PU}(m)$ its verification, and TS a timestamp that changes in every message. The user middleware initially sends to the authoritative server the following message:

$$\{Sign_{PR_{UId}}(E_{PU_{AS}}(Cert||OI||Att_d||TS))\}$$

Where PU_{AS} is the public key of the authoritative server and ATT_d the dynamic attributes of the user. After the verification of the user and retrieval of the appropriate item I, the authoritative server creates a symmetric key K, that is unique for each user, and responds with the following message:

$$\{Sign_{PR_{AS}}(E_{PU_{UId}}(I||K||TS))\}$$

Where PR_{AS} is the private key of the authoritative server. In any subsequent communication the symmetric key K can be used. Moreover since K is user specific, the authoritative server can cache the user certificate and associate it with K, therefore the user will not have to resend it in every request.

III. Implementation

A prototype of our architecture has also been implemented. Our prototype consists of a user middleware that can be run on windows phone 7.1 and android 2.1 based smart phones. Since both these platforms support a variety of symmetric and asymmetric encryption mechanisms, the proxy server is omitted. The asymmetric algorithm used is RSA whereas the symmetric algorithm is 3DES. Printed QR-codes are used as object-identities. A QR-code encodes information using black modules arranged in a square pattern on a white background. The QR-codes used in our implementation are able to encode a 160-characters string in a 184X184 pixels image. In order to enable QR-code scanning the ZXing library is used[1]. The produced QR-codes, in addition to the object identity, contain the IP-address of the authoritative server. The authoritative server is implemented as a Java application running on an Ubuntu 11.04 server and MySQL databases are used as storage points. In our implementation the information items are records in the databases. The association of an item with attributes is implemented using the eXtensible Access Control Markup Language (XACML) [3]. User's static attributes are encoded in an xml file that is stored in the user's mobile device. This file is encrypted, for security reasons, with a user-provided password. As dynamic attributes the mobile device

[1] http://www.ohloh.net/p/zxing

model, the screen size and resolution, the network operator, the OS language, the mobile phone location–as reported by its GPS sensor–as well as the wireless network SSID–are used.

In order for a user to access an information item, she has to decrypt her certificate, scan a QR-code with her middleware enabled device, connect to the authoritative server–defined in the QR-code, and send her certificate, the object identifier–provided by the QR-code–and the captured dynamic attributes. The authoritative server then, selects the appropriate item from the storage point (i.e., the database) and sends it back to the user.

A. Security evaluation

Our system is robust against the following security attacks:

Sniffing, Eavesdropping: Packet sniffing is the interception of data packets traversing a network. In this attack the attacker can learn the content of all packets. This attack is countered by having all packets encrypted using symmetric and asymmetric end-to-end encryption. The first packet that the user sends is encrypted with the authoritative server's public key (the middleware is pre-configured with the authoritative's server public key). The response that the user will receive may contain a secret key for any further communication. Also every message is signed in order to assure its integrity.

Hijacking, Man in the Middle attack: This kind of attack takes advantage of weaknesses of the TCP/IP protocol stack, and the way headers are constructed. Hijacking occurs when an attacker is between the two communication endpoints. The attacker in this case can monitor, control, change communication data and masquerade himself. For example an attacker in our system my pretend to be a user or the authoritative server. To counter this attack digital signatures are used. All packets are digitally signed therefore their integrity and provenance can be verified.

Unauthorized Access Attacks: The attacker in this case requests resources for which she is not authorized. For example the attacker may scan a QR-code and pretend to be another user. To counter that issue access control policies and certificates are used. In order for a user to access a resource she must provide a valid certificate. The data custodians create signed certificates for legitimate users so the attacker can pretend to be someone else only, if she has a valid certificate. But even in that case, every certificate's access rights are limited by specific access control policies: so even if an attacker steals a valid certificate she will be limited to the actions that correspond to the security level of that certificate.

Smartphone loss: In order to prevent unauthorized access to a user certificate (e.g., in case of mobile phone loss) certificates are encrypted in the device using a user-generated password. Only the person who knows that password can use the certificate, therefore a certificate without the knowledge of the password is useless.

Replay Attack: In this kind of network attack, the attacker retransmits a packet. In order to counter that kind of attack timestamps are used: in every packet timestamps are included in order detect packet retransmissions.

In our implementation we have not considered DoS attacks, as well as, fake dynamic attributes attacks (e.g., fake GPS coordinates). The former can be prevented using in-network mechanisms whereas, the latter, can be prevented using tamperproof hardware or trusted third parties.

IV. Related Work

Object Naming Service (ONS) [4] is a standardized system for information lookup in the IoT. ONS is a specification of how a Domain Name System (DNS) can be used to locate authoritative metadata and services associated with an Electronic Product Code (EPC). ONS is a discovery service that returns the location of the metadata assigned to an EPC. The proposed architecture allows richer queries, enhanced with user attributes, enabling the development of a wider range of services with less effort. Moreover, since ONS uses DNS services it faces DNS's vulnerabilities (e.g., as mentioned in [5]). The proposed architecture is not binded to a specific underlay technology for the information discovery and end-to-end security mechanisms safeguard the discovery process. However our architecture can be easily adjusted to be used over ONS for compatibility reasons.

Various information lookup architectures have been proposed in the literature. These architectures are either centralized–such as [6], [7]–or distributed–such as [8], [9]. In all cases these architectures enable the discovery of services associated with an object identity. This is achieved by informing the user about all the available service endpoints and by having him choosing the one he considers the most appropriate. In our scheme the selection of the information that the user will receive is done by a third party–namely the authoritative server–based on the user attributes and context.

Rouillard [10] has developed an architecture that enables the creation of contextual QR-codes. This architecture uses an xml file, stored in the user device, which contains information related to the user. This information enables the application running on the user device to display in a contextual way the information stored in the QR-code. Our architecture is more distributed. The information stored in the QR-codes is not the information in which the user is interested, but it is rather a "key" to a "database" of information items.

Covington et al.[11] have proposed a role-based access control framework for context-aware applications. In this framework user roles are stored in certificates and access control policies are defined based on these roles as well as on dynamic environmental parameters captured by sensors–such as turned on devices, time, location and other. This framework can be used in parallel to our architecture as an alternative access control mechanism.

V. Conclusions and Future Work

In this paper we proposed and implemented a secure and context aware information lookup architecture for the Internet of Things. By using attributes and tags our architecture is able to define access control policies, as well as, to semantically determine users and information items, facilitating this way the development of context aware applications. An indirection point in our architecture, namely the authoritative server, permits the association of a single object identity to multiple information items. Moreover this indirection point also acts as an abstraction of the complex storage infrastructure required in order to store the huge amounts of data, that will be generated in the Internet of Things.

Future work in this domain includes the development of a distributed system of authoritative servers and storage points. In its current form the authoritative server and the storage points reside in a single location. Storage points and the authoritative server can be implemented in a distributed manner to facilitate the great amount of data IoT eventually will handle. A DHT network of authoritative servers and a cloud based infrastructure for storage points will be investigated. Moreover as already mentioned dynamic attributes are prone to tampering by malicious entities. For example a malicious user is able to send fake GPS coordinates. In order to solve this problem, the use of trusted computing hardware–such as Trusted Platform Module(TPM) [12]–and the implementation of distant bounding protocols–such as in [13]–will be considered. Finally a version of our architecture compatible with the ONS service will be considered

References

[1] European Commission, "Internet of things in 2020. a roadmap for the future," 2008, Available: ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/enet/internet-of-things-in-2020-ec-eposs-workshop- report-2008-v3_en.pdf.

[2] S. Garfinkel, *PGP: pretty good privacy*. O Reilly Media, 1995.

[3] OASIS, "eXtensible Access Control Markup Language (xacml)," 2005, Available: https://www.oasis-open.org/standards#xacmlv2.0.

[4] GS1, "Object naming service (ons) standard," 2008, Available: http://www.gs1.org/gsmp/kc/epcglobal/ons/.

[5] D. Atkins and R. Austein, "Threat analysis of the domain name system," in *DNS). RFC 3833, Internet Engineering Task Force*, 2004.

[6] Bridge Project, "Bridge WP02 high level design discovery services," 2007, Available: http://www.bridge-project.eu/data/File/BRIDGE WP02 High level design Discovery Services.pdf.

[7] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, and V. Trifa, "Soa-based integration of the internet of things in enterprise services," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, july 2009, pp. 968 –975.

[8] B. Fabian and O. Gunther, "Distributed ons and its impact on privacy," in *Communications, 2007. ICC '07. IEEE International Conference on*, june 2007, pp. 1223 –1228.

[9] B. Fabian, "Implementing secure p2p-ons," in *Communications, 2009. ICC '09. IEEE International Conference on*, june 2009, pp. 1 –5.

[10] J. Rouillard, "Contextual qr codes," in *Computing in the Global Information Technology, 2008. ICCGI '08. The Third International Multi-Conference on*, 27 2008-aug. 1 2008, pp. 50 –55.

[11] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles," in *Proceedings of the sixth ACM symposium on Access control models and technologies*, ser. SACMAT '01. New York, NY, USA: ACM, 2001, pp. 10–20. [Online]. Available: http://doi.acm.org/10.1145/373256.373258

[12] S. Saroiu and A. Wolman, "I am a sensor, and i approve this message," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems Applications*, ser. HotMobile '10. New York, NY, USA: ACM, 2010, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/1734583.1734593

[13] J. T. Chiang, J. J. Haas, and Y.-C. Hu, "Secure and precise location verification using distance bounding and simultaneous multilateration," in *Proceedings of the second ACM conference on Wireless network security*, ser. WiSec '09. New York, NY, USA: ACM, 2009, pp. 181–192. [Online]. Available: http://doi.acm.org/10.1145/1514274.1514301