# ARMY: Architecture for a Secure and Privacy-Aware Lifecycle of Smart Objects in the Internet of My Things

The authors propose a comprehensive architectural design to capture the main security and privacy requirements during the lifecycle of a smart object. The resulting architecture has been designed, instantiated, and implemented within the scope of different European IoT initiatives, in order to promote the design and development of secure and privacy-aware IoT-enabled services.

*José L. Hernández-Ramos, Jorge Bernal Bernabé, and Antonio Skarmeta*

## Abstract

The emergence of the Internet of Things paradigm promises a multi-disciplinary revolution covering different spheres of our daily lives. However, the ubiquitous nature of IoT requires inclusive approaches in order to agree on a common understanding about its implications. Particularly, in order to unlock its huge potential and maximize its benefits, it is necessary to minimize the risks that are associated with security and privacy concerns. In this work, we propose a comprehensive architectural design to capture the main security and privacy requirements during the lifecycle of a smart object. The resulting architecture has been designed, instantiated, and implemented within the scope of different European IoT initiatives, in order to promote the design and development of secure and privacy-aware IoT-enabled services.

COMMUNICATIONS STANDARDS

## Introduction

Since the birth of the Internet, security and privacy have represented recurring concerns in the design and development of new services and applications. With the advent of the so called Internet of Things (IoT) era [1], these issues take a more important position due to the inclusion of physical devices or *things* in the Internet infrastructure. On the one hand, basic security properties need to be ensured even in devices that can be physically accessed and deployed in uncontrolled environments. On the other hand, the incessant and massive data exchange among devices, which is promoted by the IoT, makes people's privacy more difficult to be preserved.

The IoT promotes global interconnectivity through the application of recent wireless communication technologies and pervasive computing, turning things into real *smart objects*. Therefore, we need to move from traditional security and privacy enterprise-centric approaches and user-centric solutions. In this sense, our objective must be oriented to a *user-managed smart object-centric* view, while interests from different IoT stakeholders (such as citizens, governments, companies, or regulatory bodies) are still reconciled. IoT security and privacy concerns demand cross-disciplinary and multidisciplinary approaches, which require efforts from different areas in order to bring citizens into the loop. Such requirements need to be tackled by holistic and all-encompassing approaches to support scenarios with a huge number of heterogeneous devices (e.g., sensors, actuators, gateways, or backend servers), while facing inherent challenges related to flexibility, scalability, interoperability, and lightness throughout the lifecycle of a smart object.

In recent years, a huge number of worldwide initiatives have been launched to provide a common understanding for promoting the design and development of IoT services. In Europe, the Alliance for Internet of Things Innovation (AIOTI) was initiated by the European Commission in 2015 as an ambitious effort to support the dialog and interaction among different IoT players in Europe. Specifically, the "IoT standardization" working group (WG03) provides a comprehensive list of IoT standards development organizations (SDOs) and alliances, as a first step toward defining a common high level IoT architecture.[1] However, in spite of these efforts, currently there is a lack of a unified vision on security and privacy considerations in the IoT paradigm, which embraces the whole lifecycle of the smart objects that are making up the digital landscape of the future.

Given the constant evolution of technologies and protocols that make up the IoT, in this work we propose a high-level architecture that abstracts from the underlying technology for managing the security and privacy concerns during the lifecycle of a smart object. The proposed design is based on the Architectural Reference Model (ARM) [2], derived from the IoT-A European project in order to give a comprehensive view of the inherent IoT security and privacy needs. Furthermore, we describe an instantiation of the proposed architecture (ARMY), including the main deployment components and technologies employed. This instantiation has been implemented, deployed, and tested under the umbrella of two European projects in the IoT area: SocIoTal[2] and SMARTIE.[3]

## Security and Privacy Considerations for Smart Objects' Lifecycle

The main purpose of this section is to motivate the need for a holistic IoT security and privacy architecture through an overview of the main requirements that must be addressed during the different stages of the smart objects' lifecycle. In this sense, ARMY's approach follows the smart object definition from [3], as "*autonomous physical/digital objects augmented with sensing, processing, and network capabilities*." Moreover, the interpretation of the lifecycle is based on the definition of different stages that are gone through by a smart object, from its manufacturing until its decommissioning. Furthermore, we discuss some of the major emerging approaches addressing these requirements, as well as the

*The authors are with the University of Murcia.*

[1] http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=11812

[2] http://sociotal.eu/

[3] http://www.smartie-project.eu/

specific technologies used for ARMY's instantiation to cope with such needs during each stage of the smart objects' lifecycle.

## Bootstrapping

Bootstrapping consists of a set of procedures by which a smart object joins a network. During bootstrapping, the cryptographic material statically configured in the *manufacturer domain* is used to derive dynamic credentials and keys to be used in the *deployment domain*. This stage represents an essential step before the smart object can operate, in which the static cryptographic material can be considered as the *root identity* to derive keys and credentials for secure and privacy-preserving operation. Indeed, operational security and privacy are jeopardized if bootstrapping is not carried out securely by using suitable and well known technologies. However, while currently there is a wide range of approaches to be used during bootstrapping, their application to IoT environments is not straightforward. In this sense, in addition to providing basic security properties, the approach for bootstrapping of smart objects should consider practical aspects of IoT devices, such as lack of a user interface, as well as a higher degree of scalability and flexibility, given the nature of the envisioned scenarios.

Under the IETF, in addition to Host Identity Protocol Diet EXchange (HIP-DEX), the Protocol for Carrying Authentication for Network Access (PANA) (RFC 5191) is widely accepted as the main candidate for security bootstrapping. Indeed, it is being employed by the ZigBee Alliance and ETSI TC SmartM2M, in conjunction with the Extensible Authentication Protocol (EAP) (RFC 5247) and transport layer security (TLS) (RFC 5246). However, despite being a mature technology, its applicability to address future open environments with millions of interconnected smart objects has not been demonstrated. An emerging alternative within the IETF uses the Constrained Application Protocol (CoAP) (RFC 7252) as the *EAP lower-layer* to transport EAP packets for IoT bootstrapping [4]. CoAP is an application layer protocol specifically designed to be used even in constrained devices and networks. By using this approach, smart objects could use the same protocol for bootstrapping and operation. The high level of flexibility and lightness makes this strategy a promising candidate for IoT environments. Therefore, the ARMY instantiation approach for bootstrapping considers the use of CoAP-EAP to address the security aspects of this stage.

## Registration and Discovery

An essential feature for realizing the IoT is to provide an infrastructure that allows smart objects to be addressable, named, and discovered by others. First, a smart object must be identifiable through the assignation and management of addresses/identifiers. This identifier could be associated with other attributes, such as manufacturer or hardware features. Second, such an infrastructure must provide a name resolution mechanism that allows smart objects to be organized according to taxonomies or hierarchical classifications. In addition, it should provide a registration/discovery process that allows the specification of security and privacy preferences to determine how an object wants to be discovered (for example, showing only a subset of its services) and by whom. This is an additional and necessary level of access control that should be considered for a controllable and privacy-aware discovery process.

X.500 is the OSI directory standard defined by the ISO and the ITU. It defines a hierarchical data model with a set of protocols to allow global name lookup and search. The Lightweight Directory Access Protocol (LDAP) (RFC 4510) was developed by the IETF as an alternative, while it also brings different issues related to its complexity to be implemented, and consequently, deployed in IoT scenarios where scalability is an essential feature. To address many of these concerns, the handle system (HS) (RFC 3650) was designed to offer efficient, extensible, and secure identifier and resolution services for the Internet. HS is part of the digital object architecture (DOA) and it is considered by the ITU under ITU-T Recommendation X.125. In HS, a digital object (DO) has a machine-independent and platform-independent structure that allows it to be identified, accessed, and protected. The syntax of the DO is a set of pairs (type, value) that can be hierarchic, providing descriptions and identifiers of other DOs in its parameters. The HS represents an alternative to well known resolution approaches, such as the domain name system (DNS) (RFC 1034), by providing a higher degree of flexibility to enrich the resolution infrastructure with security aspects. The deployment of HS in IoT scenarios has already been analyzed under the EU IoT6 project.[4] Consequently, ARMY instantiation for registration and discovery stages (including lookup and name resolution features) is based on the use of the HS.

## Operation

During the operation stage, security and privacy aspects can be considered at different levels depending on the layer of the IoT protocol stack [5]. However, given the high degree of flexibility required, the application of security and privacy mechanisms at higher layers is preferable, in order to abstract from the details of underlying lower layer technologies. In the IoT landscape, CoAP is considered as the standard application layer protocol, which defines a security binding through the use of datagram transport layer security (DTLS) (RFC 6347) for transport layer security. For authorization purposes, while OAuth 2.0 (RFC 6749) is widely deployed in Web environments, its applicability in IoT environments has not been demonstrated. In this sense, the approach followed by ARMY is based on the distributed capability-based access control (DCapBAC) model [6]. DCapBAC follows a SPKI certificate theory (RFC 2693) approach through the use of access tokens with similar JSON web token (JWT (RFC 7519) semantics, in which a set of access rights are bound to the smart object's public key. Additionally, the token provides simple semantics to specify access conditions to be verified locally by the smart object being accessed. These conditions have been used for the specification of a threshold trust value, as part of our IoT trust and reputation model [7] to cope with a broader range of security and privacy aspects of ARMY's functionality during operation. DCapBAC has been further integrated with

By using this approach, smart objects could use the same protocol for bootstrapping and operation. The high level of flexibility and lightness makes this strategy a promising candidate for IoT environments. Therefore, the ARMY instantiation approach for bootstrapping considers the use of CoAP-EAP to address the security aspects of this stage.

[4] http://iot6.eu/

The constant evolution of the IoT is resulting in a disharmonized and fragmented landscape of technologies and protocols. Consequently, it is necessary to define high-level architectures able to disengage from the technical details, thereby providing a common understanding of security and privacy needs.

| Lifecycle stage | Bootstrapping | Registration/discovery | Operation | | Management |
|---|---|---|---|---|---|
| | | | Pair communication | Group communication | |
| ARMY instantiation | CoAP-EAP | Handle system | CoAP-DTLS<br>DCapBAC<br>Idemix<br>Trust model based on fuzzy logic | NGSI 9/10<br>CP-ABE | LWM2M |

**Table 1.** ARMY instantiation technologies.

a policy-based access control mechanism based on eXtensible Access Control Markup Language (XACML) (OASIS standard). Besides the use of CoAP-DTLS to transport access tokens, privacy mechanisms have been instantiated through the integration of DCapBAC with privacy-preserving *proof-of-possession* techniques [8], such as identity-based encryption (IBE) [9] and anonymous credential systems (Idemix [10]).

In IoT scenarios with a huge number of devices, it is necessary to provide flexible mechanisms that allow communication among groups of smart objects that can be opportunistically created, as well as a scalable mechanism to share or outsource data, while end-to-end security is preserved. In this sense, the ciphertext-policy attribute-based encryption (CP-ABE) [11] cryptographic scheme provides the ability to dynamically define groups and subgroups of smart objects according to different combinations of identity attributes, without additional key management tasks. The application of CP-ABE for IoT scenarios has already been analyzed in the case of non-heavily constrained devices [8]. Furthermore, it has been integrated with the OMA next generation services interfaces 9/10 (OMA NGSI-9/10) specification under the EU SocIoTal project to outsource encrypted data for groups of devices.

### Management

The implicit requirements from the inclusion of constrained devices in the Internet infrastructure demand the redesign of traditional network management protocols, to support self-management and self-configuration capabilities in a broader spectrum of IoT scenarios. New management protocols and solutions for IoT are required to be scalable, extensible, distributed, and hierarchical, providing support for the entire range of smart objects. Moreover, they should support communication through lossy networks, by reducing message size to be tailored to such environments, as well as a high level of interoperability with already established mechanisms. As for the operation stage, security considerations are crucial to ensure that management tasks are effectively performed.

In this sense, well known protocols such as the Simple Network Management Protocol (SNMP) (RFC 1157) or the Network Configuration Protocol (NETCONF) (RFC 6241) do not provide sufficient flexibility, scalability, and lightness (e.g., regarding message size), and their data models are not adapted for IoT scenarios. The CoAP Management Interface (CoMI), which is an emerging initiative within the IETF, is an adaptation of the RESTCONF protocol for constrained devices and networks. It uses CoAP to access the management data resources that are specified in Yet Another Next Generation (YANG) (RFC 6020) and binary encoding. CoMI provides a lightweight design to reduce message complexity and size. CoMI security is based on mechanisms already available for CoAP, through the use of DTLS. Similarly, OMA Lightweight M2M (OMA LWM2M) from the Open Mobile Alliance, like CoMI, provides a RESTful device management service over CoAP. Although CoMI can be considered as a more interoperable approach since it reuses existing YANG data models, LWM2M is currently a more mature solution, for which there already exist open source implementations. Consequently, ARMY instantiation for the management stage is based on LWM2M, specifically through the use of the IoT-agent software that is developed in the scope of the FI-WARE EU initiative.

Finally, Table 1 summarizes, for each lifecycle stage, the set of technologies that are adopted to instantiate ARMY's functionality, which is described in the next section.

## Holistic Security and Privacy IoT Architecture

The constant evolution of the IoT is resulting in a disharmonized and fragmented landscape of technologies and protocols. Consequently, it is necessary to define high-level architectures able to disengage from the technical details, thereby providing a common understanding of security and privacy needs. Toward this end, IoT-A was a large-scale European project intended to define an architectural reference model (ARM) for a broader interoperability among IoT systems. The set of results from IoT-A include: a reference model (RM) to promote common understanding at high abstraction level; a reference architecture (RA) to describe essential building blocks and build compliant IoT architectures; and a set of *best practices/guidelines* to help in developing an architecture based on the RA.

In particular, the RA provides several views and perspectives focused on different architectural aspects. Among these views, the functional view, which is shown in Fig. 1, describes a set of functional components (FC), which are organized into nine functional groups (FG), as well as their responsibilities and interfaces. Specifically, the security FG is composed of five functional components: authentication, authorization, identity management (IdM), key exchange and management (KEM), and trust and reputation (T&R). However, while it provides basic security and privacy functionality of an IoT system, it does not define the interactions among these components or an exhaustive set of specific technologies to instantiate this functionality. Based on the functional view

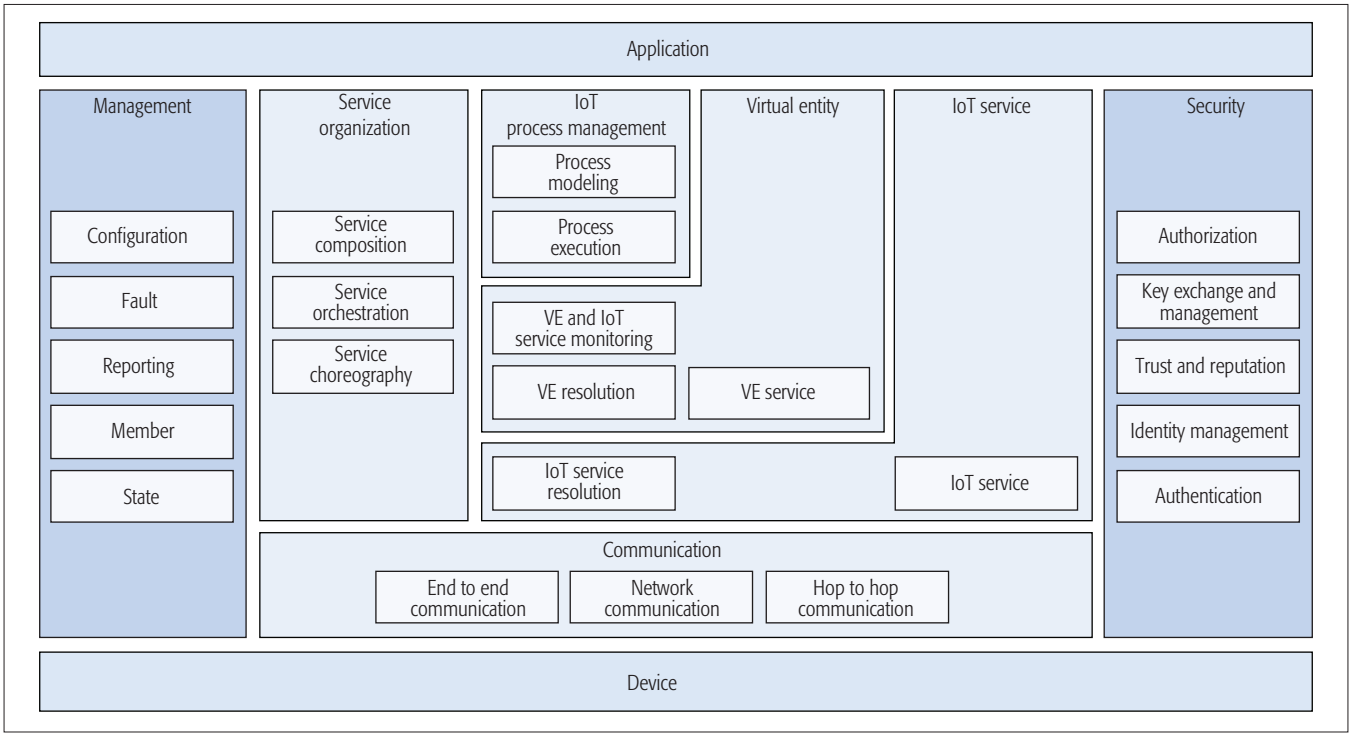


**Figure 1.** IoT-A functional view.

from RA and the different stages of the smart objects' lifecycle derived from [12], the proposed architecture (ARMY) represents an extension of the security FG, and an instantiation by defining the main interactions among the identified FCs.

This extension is based on the inclusion of two additional FCs: context manager and group manager, which complement the functionality of the other FCs that are already proposed by the security FG. The *context manager* aims to realize the vision of an adaptive security and privacy to the current context conditions in which the smart object operates [13]. Its main functionality is to reason about contextual information being perceived by a smart object from its surrounding environment, so other security FCs are able to adapt their behavior based on it. It is meant to be instantiated by data analysis techniques or simple rule-based mechanisms in case of more constrained smart objects. The *group manager* is designed to deal with security and privacy concerns when information needs to be shared or outsourced with a group of smart objects. It is intended to be implemented through the application of attribute-based cryptographic techniques, and deployed on smart objects participating in scenarios where publish/subscribe or multicast communications are required.

In order to describe the functionality of the architecture components, and for the sake of clarity, we adopt a *producer*/*consumer* approach, where smart objects act as information producers and consumers. While it is assumed that a smart object will act as data producer and consumer throughout its lifecycle, we consider these two roles to further the understanding of ARMY's functionality. The description of the relationship among components is based on the existence of an infrastructure level, assembling the set of elements (e.g., gateways or backend servers) that are required to support secure and privacy-aware interactions among smart objects. It should be noted that ARMY is intended to describe the functionality and interactions only among security FCs to address security and privacy requirements during a smart object's lifecycle. This is complementary to other interactions required among FGs to realize a particular use case or scenario. Furthermore, it is abstracted from underlying technologies, so the same FC can be instantiated by a different technology (or implementing different aspects of the same technology), depending on whether that FC is instantiated at the infrastructure (e.g., within a gateway) or smart object level.

Figure 2 shows the required interactions during the bootstrapping and registration/discovery stages. Indeed, the smart objects lifecycle begins when it is installed and then commissioned during bootstrapping. For this stage, we claim the need of statically configured cryptographic material to enable smart objects to join securely in a specific deployment domain. Such a credential could be embedded by the manufacturer, and considered as the *root identity* for bootstrapping. By using its root identity (1.1, 1.2), the smart object is commissioned and connected to the network, which implies an authentication (1.3) and authorization (1.4) process. As a result of a successful bootstrapping, it obtains some cryptographic material (denoted as *domain identity* (1.5)). This credential, along with the root identity, make up the *complete identity* of the smart object. The domain identity allows the smart object to be identified within the domain for subsequent processes, and it is associated with additional attributes that are specific to the deployment domain, such as the *owner*, which can be used for management tasks.

Afterward, the smart object is also registered to be discovered by other smart objects. This functionality is already considered by the IoT-A project through the IoT service resolu-

The notion of group in ARMY is realized by the association of identity attributes to cryptographic group keys, which are obtained during registration and provisioning stage. This functionality is carried out by the group manager FC, which is responsible for encrypting and decrypting outsourced data.

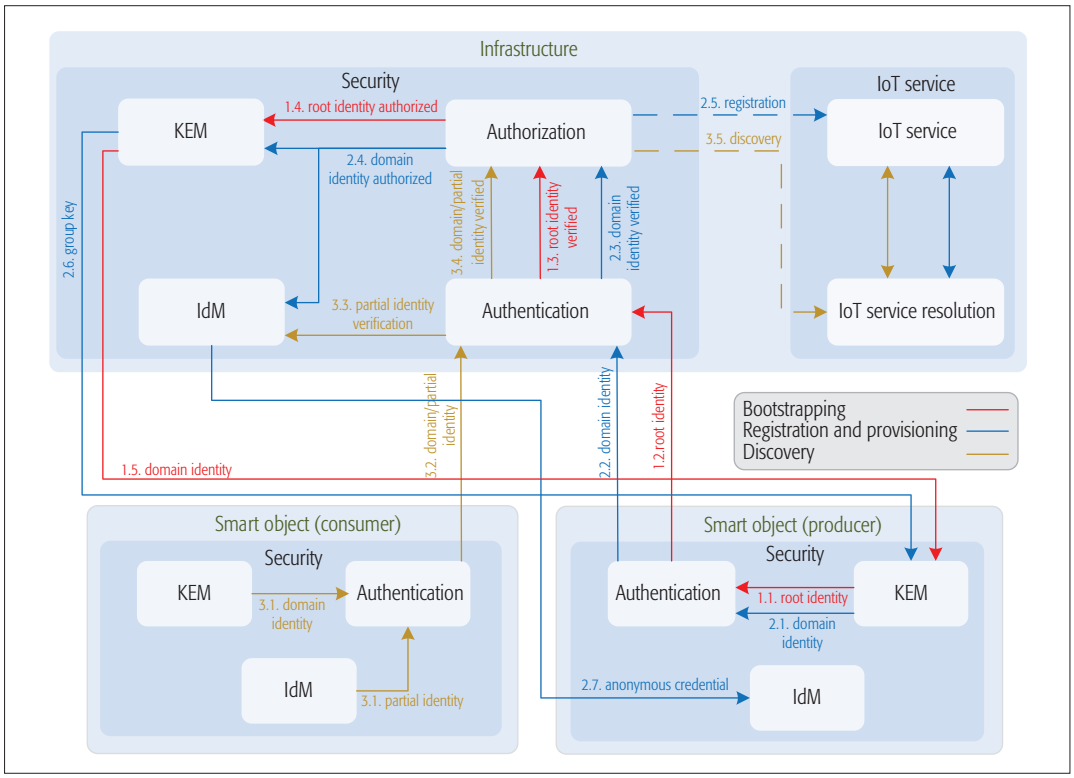


**Figure 2.** ARMY bootstrapping and registration/discovery interactions.

tion FC, within the IoT service FG, by providing lookup, resolution, and discovery functionality. For this purpose, the smart objects makes use of their domain identity (2.1, 2.2) to be authenticated (2.3) and authorized (2.4). If successful, the smart object is registered (2.5). Furthermore, during this stage, other cryptographic material is derived (*provisioning*) to be employed by the smart object when operating, such as *group keys* (2.6) and *anonymous credentials* (2.7) associated with the complete identity's attributes, which are previously demonstrated.

During the discovery stage, a smart object (consumer) tries to discover the services being provided by another device (producer). This stage also requires authentication (3.3) and authorization procedures (3.4) to determine whether a legitimate smart object is authorized to find that service. The authentication can be performed through the use of the domain identity, or by considering privacy concerns of the consumer through the use of a partial identity (3.1, 3.2) (as a subset of its identity attributes from the complete identity), derived from the anonymous credential.

A smart object can get into operation providing the services for which it was manufactured, or into the management stage. The required interactions among functional components are shown in Fig. 3. For operation, we consider two cases in which communication is either between two smart objects, or involving a group of them. The main reason for this distinction is the need to consider different encryption techniques depending on the case being contemplated. For the *operation-pair* case, a smart object (consumer) tries to get a credential to perform a specific action over the discovered smart object (producer). To this aim, it uses either its complete identity (by which it is unequivocally identified), or a partial identity (4.2, 4.3), which is selected according to its context information in order to preserve its privacy (4.1). Once it is authenticated (4.4, 4.5), the demonstrated identity attributes are used to launch an authorization process. If successful, an authorization token is generated (4.6) and given to the smart object (4.7).

Afterward, the consumer smart object is authenticated against the producer, which requires a process similar to the interactions previously described (4.8-4.10). If authentication is successful, the smart object consumer uses the authorization token (4.11, 4.12) to get access to a service being hosted by the producer. Then, the producer evaluates this token by considering additional information, such as context data (4.13) or trust and reputation scores (4.14, 4.15) associated with the requesting smart object for a more fine-grained access control. Furthermore, the consumer assesses the quality of the service provided by the producer (4.16), so later on it can get trust scores associated with the consumer when roles are interchanged.

The notion of group in ARMY is realized by the association of identity attributes to cryptographic group keys, which are obtained during the registration and provisioning stage. This functionality is carried out by the group manager FC, which is responsible for encrypting and decrypting outsourced data. Thus, for the *operation-group* case, a smart object (producer) makes a piece of data available to a group of (consumers) smart objects. For this purpose, it encrypts such information by using its group key (5.1), and selecting the set of identity attributes that must be satisfied by the consumer smart objects in order to access the outsourced information. These attributes are selected depending on the context being

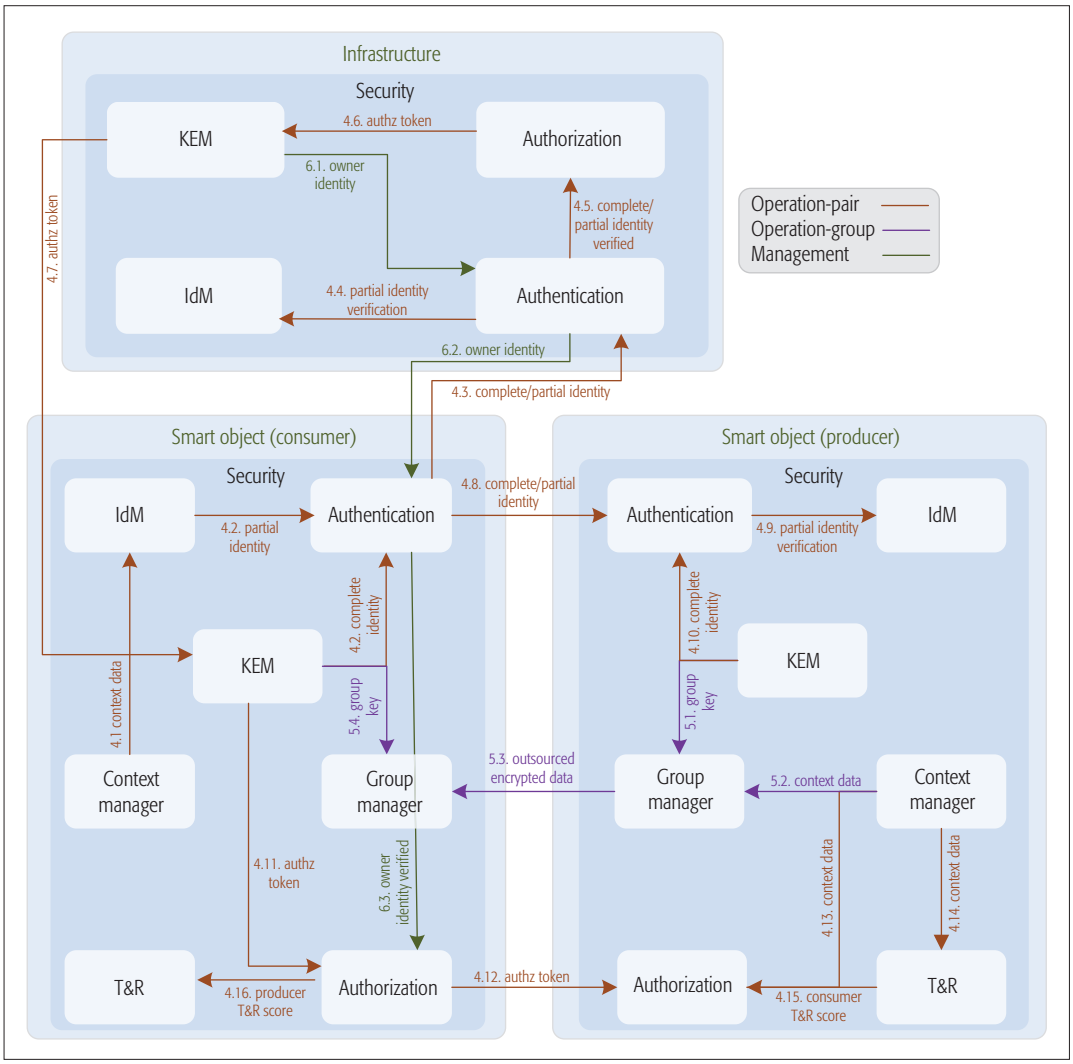


**Figure 3.** ARMY operation and management interactions.

detected (5.2), enabling the creation of dynamic groups of smart objects. When encrypted data are outsourced (5.3) and received by consumer smart objects, they try to decrypt this information by using their corresponding group key (5.4).

A smart object can be managed, either directly by another smart object, or more commonly, by the infrastructure layer (as shown in Fig. 3). The management stage implies an authentication and authorization process (6.2, 6.3), so only legitimate and authorized users (i.e., the smart object's owner (6.1)) are able to perform the main management tasks of the smart object. An exhaustive set of these tasks is already provided by the management FG from IoT-A, so ARMY functionality is intended to complement them with security aspects. Finally, the smart object can be *decommissioned* (or *recommissioned*), through appropriate revocation procedures.

## ARMY Instantiation and Deployment

The instantiation and deployment of ARMY has been primarily driven by two European projects, SocIoTal and SMARTIE, whose overall goal is the application of secure and privacy-preserving mechanisms to different IoT use cases and scenarios. ARMY has been instantiated in the scope of both projects, through the definition of several deployment components that instantiate and implement the functionality provided by the different FCs of the proposed architecture. Table 2 summarizes which FCs are instantiated for each deployment component. It should be pointed out that authentication and KEM FCs are instantiated by the whole set of deployment components (including smart objects), since they provide basic functionality for establishing authenticated communications among them. Furthermore, given the heterogeneity of IoT devices, the instantiation of certain FCs is denoted as "optional" (O). For example, while group manager and T&R FCs can be instantiated directly in certain IoT devices, they may also be partially instantiated by other infrastructure components (i.e., gateway T&R service) in the case of more resource-constrained devices.

Figure 4 shows the interactions between deployment components and smart objects, to accomplish ARMY's functionality for each stage of the life cycle. In this representation, an *IoT domain* encompasses a local IoT-enabled environment (e.g., a smart building) as part of a more global ecosystem (e.g., a smart city). The *device layer* consists of the set of heterogeneous devices (or things) composing an IoT environment. Following IoT-A notation, a device may be com-

The instantiation and deployment of ARMY has been primarily driven by two European projects: SocIoTal and SMARTIE, whose overall goal is the application of secure and privacy-preserving mechanisms to different IoT use cases and scenarios.

This instantiation has been driven by the design and development of different security and privacy mechanisms. In this sense, our ongoing work is focused on the extension of such mechanisms to devices with strong resource constraints.

| Deployment component | ARMY functional component | | | | | | |
|---|---|---|---|---|---|---|---|
| | Authentication | Authorization | KEM | T&R | IdM | Group manager | Context manager |
| Anonymous credential issuer | ✓ | – | ✓ | – | ✓ | – | – |
| Attribute key issuer | ✓ | – | ✓ | – | – | – | – |
| Authentication service | ✓ | – | ✓ | – | – | – | – |
| Trust service | ✓ | – | ✓ | ✓ | – | – | – |
| Authorization service | ✓ | ✓ | ✓ | – | – | – | – |
| Local/global resolution service | ✓ | ✓ | ✓ | – | – | – | – |
| Management service | ✓ | ✓ | ✓ | – | – | – | – |
| Context broker | ✓ | – | ✓ | – | – | – | ✓ |
| Gateway | ✓ | ✓ | ✓ | – | – | O | – |
| Smart object (producer) | ✓ | ✓ | ✓ | O | ✓ | ✓ | ✓ |
| Smart object (consumer) | ✓ | ✓ | ✓ | O | ✓ | ✓ | ✓ |

**Table 2.** ARMY components instantiation.

posed of sensors, actuators, or tags, as well as other devices. The *IoT services layer*, while being part of the IoT domain, is considered part of the *infrastructure* level and consists of deployment components that support devices in managing security and privacy aspects within the domain.

For the sake of clarity, let us consider two domains where a smart object *producer* in *IoT Domain A*, and a *consumer* within *IoT Domain B*, are intended to interact with each other. Following the lifecycle stages, the producer initiates a *bootstrapping* process by which it contacts the *gateway* to join securely in IoT Domain A. Optionally, this process may require additional interaction (dotted line) with the *authentication service* to authenticate the identity provided by the smart object. As a result of a successful bootstrapping, the producer obtains a domain identity, which is *registered* in the *local resolution service*, and then in the *global resolution service* to make this smart object globally available.

As already mentioned, the root identity and the domain identity make up the complete identity of the smart object. Then, during provisioning, the producer tries to get an anonymous credential and a group key associated with its complete identity. This process requires the interaction with two infrastructure components; the *attribute key issuer*, which has been implemented to generate and deliver CP-ABE keys, and the *anonymous credential issuer*, instantiated as an *Idemix issuer* generating credentials associated with such attributes. Also, the object can be managed by its owner through the *management server* by using LWM2M.

From the consumer side, the discovery process is performed through the local resolution service and the global resolution service. Then, to operate with the discovered device, it tries to get an authorization token by contacting the *authorization service* that is responsible for evaluating XACML policies, and generating DCapBAC tokens in case of a successful authorization. Furthermore, the authentication process has been implemented by considering traditional approaches (through the use of certificates as the complete identity), as well as privacy-preserving techniques (employing a partial identity) [8]. For the latter, the consumer contacts the authorization service using a partial identity (a cryptographic proof of certain identity attributes) through the use of Idemix. Then, the token is bound to a pseudonym that is proved to the producer, which acts as an Idemix verifier and evaluates the token. In addition, it queries the *trust service* to get the trust value associated to the consumer. This service is built on the trust model based on fuzzy logic presented in [7].

Finally, group communication is implemented by using CP-ABE, which has been deployed on non-heavily constrained devices, to communicate context information to groups of devices. Specifically, this has been developed in the scope of the SocIoTal project by integrating these devices through OMA NGSI-9/10 with the *context broker* (implementing a publish/subscribe model) of the European initiative FI-WARE. The resulting mechanism is intended to provide a flexible encryption approach for highly uncoupled environments with a huge number of devices.

The integration of these components and technologies represents an instantiation of the main ARMY functionality. However, it should be noted that the proposed architecture can be instantiated by other initiatives or technologies tailored to specific IoT scenarios or use cases, where security and privacy must be preserved.

## Conclusions

In a hyper-connected world, we claim that security and privacy are a must, which requires stringent efforts from different disciplines and IoT stakeholders to achieve a unified view about their requirements, including incentives to make the society aware of the associated risks. In this

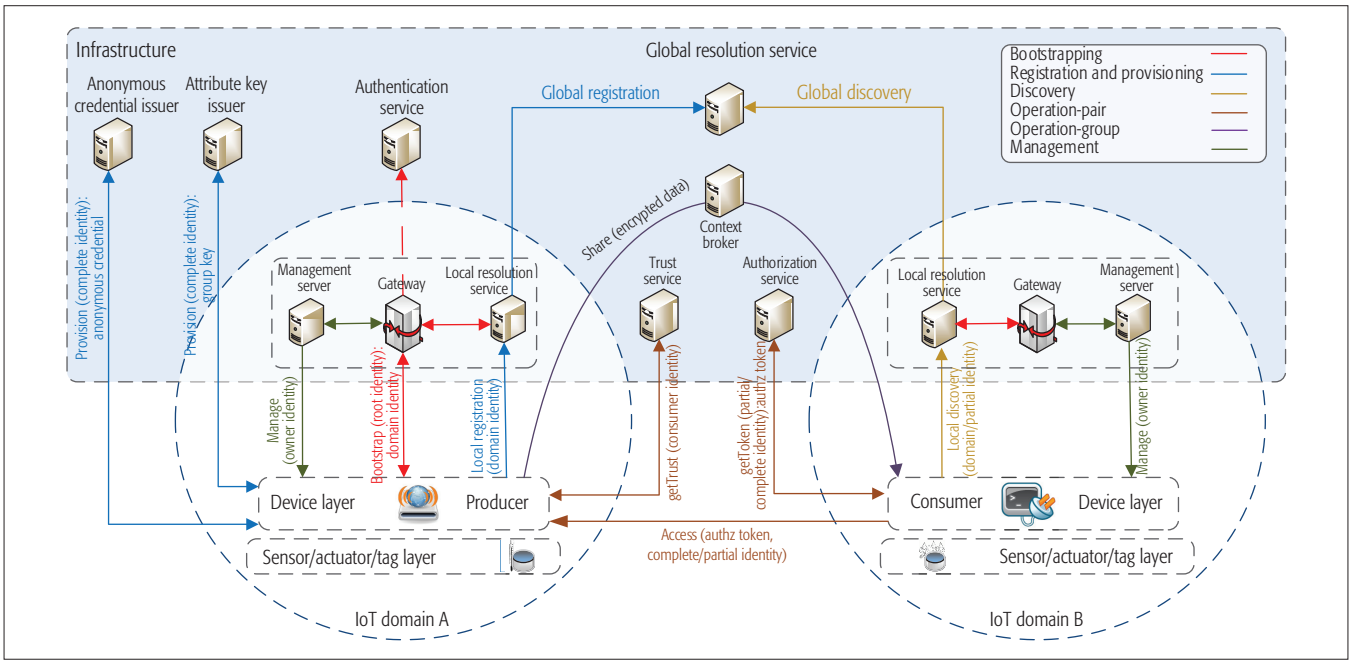


**Figure 4.** Proposed ARMY deployment.

work, we proposed a high-level architecture in order to note some of the major security and privacy needs to be managed during the lifecycle of smart objects. This architecture has already been instantiated and deployed under the umbrella of two European initiatives, and it is intended to be considered by other specific IoT architectures and deployments where security and privacy are required. This instantiation has been driven by the design and development of different security and privacy mechanisms. In this sense, our ongoing work is focused on the extension of such mechanisms to devices with strong resource constraints. This trend has recently sparked great interest from the IETF with the establishment of different working groups. Additionally, the application of appropriate revocation procedures is currently a key challenge to cover the whole spectrum of security and privacy needs throughout all the lifecycle stages of smart objects.

## Acknowledgment

This work has been sponsored by the European Commission through the FP7-SMARTIE-609062 and the FP7-SOCIOTAL-609112 EU projects, and the Spanish National Project CICYT EDISON (TIN2014-52099-R) granted by the Ministry of Economy and Competitiveness of Spain (including ERDF support).

## References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Elsevier Computer Networks*, vol. 54, no. 15, 2010, pp. 2787–805.
[2] A. Bassi *et al.*, "Enabling Things to Talk," *Designing IoT Solutions with the IoT Architectural Reference Model*, 2013, pp. 163–211.
[3] G. Kortuem *et al.*, "Smart Objects as Building Blocks for the Internet of Things," *IEEE Internet Computing*, vol. 14, no. 1, 2010, pp. 44–51.
[4] D. Garcia-Carrillo and R. Marin-Lopez, "EAP-based Authentication Service for CoAP," IETF Internet Draft, draft-marin-ace-wg-coap-eap-03 (work in progress), 2016.
[5] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 3, 2015, pp. 1294–312.
[6] J. L. Hernandez-Ramos *et al.*, "Toward a Lightweight Authentication and Authorization Framework for Smart Objects," *IEEE JSAC*, vol. 33, no. 4, Apr. 2015, http://dx.doi.org/10.1109/jsac.2015.2393436, pp. 690–702.
[7] J. B. Bernabe, J. L. H. Ramos, and A. F. S. Gomez, "TACIoT: Multidimensional Trust-Aware Access Control System for the Internet of Things," *Soft Comp.*, vol. 20, no. 5, 2015, http://dx.doi.org/10.1007/s00500-015-1705-6, pp. 1763–79.
[8] J. L. Hernández-Ramos *et al.*, "Preserving Smart Objects Privacy through Anonymous and Accountable Access Control for a M2M-Enabled Internet of Things," *Sensors*, vol. 15, no. 7, 2015, pp. 15, 611–639.
[9] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Advances in Cryptology — CRYPTO 2001*, Springer, 2001, pp. 213–29.
[10] J. Camenisch and E. Van Herreweghen, "Design and Implementation of the Idemix Anonymous Credential System," *Proc. 9th ACM Conf. Computer and Commun. Security. ACM*, 2002, pp. 21–30.
[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *IEEE Symp. Security and Privacy*, 2007, pp. 321–34.
[12] T. Heer *et al.*, "Security Challenges in the IP-based Internet of Things," *Wireless Personal Commun.*, vol. 61, no. 3, 2011, pp. 527–42.
[13] C. Perera *et al.*, "Context Aware Computing for the Internet of Things: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 1, 2014, pp. 414–54.

## Biographies

José L. Hernandez-Ramos (jluis.hernandez@um.es) received the B.Sc. and M.Sc. degrees in computer science from the University of Murcia, Spain. Since 2013 he has been a research fellow and Ph.D candidate in the Department of Information and Communications Engineering at the University of Murcia, where he has participated in different European research projects. His research interests are mainly related to the application of security and privacy mechanisms for the Internet of Things.

Jorge Bernal Bernabe (jorgebernal@um.es) received the Ph.D. and M.Sc. in computer science from the University of Murcia, Spain. He is a research fellow in the Department of Information and Communications Engineering at the University of Murcia. He has been a visiting researcher at HP-Labs (UK), and he has participated in several European research projects. His scientific activity is mainly devoted to security, privacy, trust, and identity management applied to distributed systems and the Internet of Things.

Antonio Skarmeta (skarmeta@um.es) received the M.S. degree in computer science from the University of Granada, and B.S. (Hons.) and the Ph.D. degrees in computer science from the University of Murcia, Spain. Since 2009 he has been a full professor at the same department and University. His main interests are the integration of security services, identity, IoT, and smart cities. He has published more than 200 international papers, and he has been a member of several program committees.