A Secure ICN-IoT Architecture

Sabrina Sicari*, Alessandra Rizzardi*, Luigi Alfredo Grieco§, Alberto Coen-Porisini*
*Dipartimento di Scienze Teoriche e Applicate Università degli Studi dell'Insubria, v. Mazzini 5 - 21100 Varese, Italy, Email: {sabrina.sicari; alessandra.rizzardi; alberto.coenporisini}@uninsubria.it
§Department of Electrical and Information Engineering Politecnico di Bari, v. Orabona 4 - 70125 Bari, Italy, Email: a.grieco@poliba.it

***Abstract*—Security, interoperability, scalability, and mobility support are key challenges for the Internet of Things (IoT). Information Centric Networking (ICN) is an emerging paradigm for the Future Internet design that brings all the potential to face these challenges thanks to name-driven networking primitives. As a matter of fact, ICN natively supports multicast, mobility, content oriented security, and can be effectively used to design deverticalizing middlewares for the IoT. In this context, the so called ICN-IoT middleware has been proposed within an initiative of the ICNRG (Information Centric Networking Research Group) of the Internet Research Task Force (IRTF) to encompass the key requirements of IoT systems. An important issue to be addressed is how to guarantee the security and the privacy of the information transmitted over the network, which may include sensitive data. Herein, the current security functionalities of ICN-IoT are analyzed and enriched in order to strengthen its resilience to violation attempts. The proposed approach is validated, at a design level, by means of a running example.**

***Index Terms*—Internet of Things, Information Centric Networks, Security**

I. Introduction

Internet of Things (IoT), an emerging paradigm which promises to connect a wide number of heterogeneous devices to the Internet, is now facing the challenge to build an infrastructure able to cope with this dynamic environment [1]. In fact, in a typical IoT context, not only different technologies and protocol standards are involved, but also applications pertaining to different domains, which should cooperateto provide interested users innovative and integrated services. What emerges is the need to design and develop a unified defragmented IoT platform which should ease the accessibility and the cooperation among IoT objects and applications. Since one of the main factors limiting the growth and the diffusion of IoT systems is the lack of a standardized platform, a crucial point is the introduction of a set of tools and interfaces able to manage the interoperability across different vendors of hardware and software solutions and across diverse vertical application domains [2]. Several middleware solutions have been presented with the aim to enforce the integration of the devices connected in the IoT, considering the communication mediums involved [3] [4] [5]. While many smart devices can natively support IPv6 communications, existing deployments might not support IP protocol within the local area scope, thus requiring ad hoc gateways and middlewares. Furthermore, the emerging technologies have amplified the problem of mobility, since the number of the devices used within IoT scenarios is dynamic. As a consequence, the effectiveness of information retrieval as well as the security of the transmitted information are other two requirements to be addressed. In this direction, Information-Centric Networking (ICN) has received increasing attention by research community [6] [7]. Its main features are: (i) the identification of network entities (mobile devices, contents, services) by their name instead of their IP address; (ii) a routing system based both on names and addresses; (iii) native support to mobility and multicast; (iv) content security. Such functionalities allow to ease the scalability, the moblity support, the multicasting, and the caching of the contents. As regards security, few existing solutions specifically addresses this requirement in the ICN context, such as [8], which focuses on authentication aspects. However, also the integrity and the confidentiality of the handled information has to be preserved, and the privacy for user sensitive data. Several attempts have already been made in IoT scenarios, for example regarding the enforcement of policy constraints [9] [10], or the design and the development of a flexible middleware able to deal with security and data quality aspects [2]. Nevertheless, security and privacy in ICN have still to be investigated and an ad hoc solution has to be proposed for such a specific infrastructure. Hence, in this paper, a secure architectural model for ICN-IoT is defined to face security issues in ICN-based IoT deployments. This contribution can significantly enrich the security functionalities of the so called ICN-IoT middleware [11], proposed within an initiative of the ICNRG of the IRTF to encompass the key requirements of IoT systems, in order to strengthen its resilience to violation attempts. The proposed approach is validated, at a design level, by means of a running example using the potentiality of UML sequence diagrams. Note that ICN-IoT scenarios put in light new challenges towards the definition of security solutions able to: (i) provide a trust model suitable for the involved entities and relationships; (ii) preserve the privacy for sensitive information transmitted within the network; (iii) guarantee an effective access control system, based on well-defined and cross-domain policies.

II. Reference Architecture

The proposal to build a unified IoT platform using an ICN-centric approach started with the middleware proposed in [11]. Here, IoT services run separately from ICN functionalities, which are in charge to manage the IoT data discover and delivery. The core functions to be supported by the underlying ICN infrastructure are: (i) device and network service discovery, through an efficient content publish/subscribe management; (ii) naming service, which implies the ability to assign unique names to the device resources, guaranteeing the persistence also in presence of mobility and security issues; (iii) context processing, storage and caching, in order to reduce content access latencies. The architecture includes five components:

1) *Embedded Systems*, which enable both sensing and actuating functions and the transmission of the data to the *Aggregator*
2) *Aggregator*, which plays two roles: (i) it acts as a local network gateway in order to bridge the communication among the resource-contrained nodes belonging to the network and the other aggregators; (ii) it integrates sensing and/or actuating services in the local network
3) *Local Service Gateway (LSG)*, which connects the local IoT system to the global one, handling the local name assignment and enforcing the data access policies for the local IoT devices
4) *ICN-IoT Server*, which manages both the lookup services and the subscriptions; it does not represent a bottleneck for the content provision, since it is only involved in the control of the name and certificates exchanged among publishers and subscribers
5) *Services/Consumers*, which are application interfaces able to interact with the *ICN-IoT Server*.

The goal is to move the data discovery, processing and delivery closer to the distributed network nodes; the aggregators and the *LSG*s have to own self-configuration capabilities and provide not only local services, but also scaling to large IoT services, thanks to integration functionalities in real-time. Fig. 1 shows a schema of the proposed ICN-IoT middleware, along with its interactions and functions.

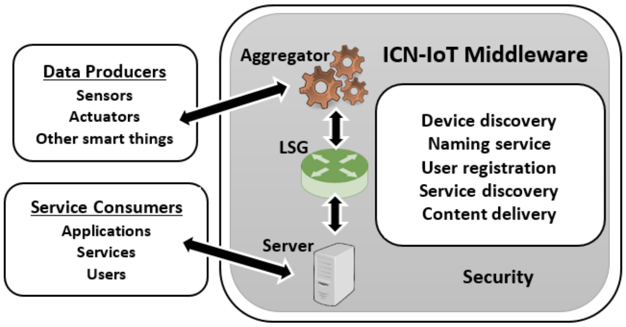

Fig. 1. ICN-IoT middleware

III. Secure functionalities

Besides designing a proper middleware (as done in [11]), which specifically addresses ICN-IoT paradigm, in order to guarantee its real diffusion, it is fundamental to deal with security and privacy requirements, since the system may handle sensitive information (user habits, device location) [1]. The following interactions among the ICN-IoT entities have to be integrated with security solutions: (i) device discovery; (ii) service discovery; (iii) naming service (iv) user registration; (v) content delivery. Such transactions are analyzed in depth in the following subsections within a running example. In the proposed application scenario, users want to be informed about a set of GPS location information regarding the vehicles in a particular area. A device reports the GPS values and the users receive the relative notifications (the presence of a particular vehicle on which the GPS device is installed), in order to monitor traffic conditions and decide to take the best route to their destination. Such information may also be useful for healthcare systems.

A. Secure device discovery

The final goal of device discovery is the establishment of relationships among nodes. In the actual IP-based IoT systems, such a transaction is complicated by the fact that a translation service is required in order to maintain the mapping from IP-addresses to the physical node attributes; moreover, this often involves manual configurations, which are not efficient in the dynamic IoT environment. ICN-IoT approach overcomes such issues, since device discovery does not require any manual configuration or address translation because names and related contexts are directly used to discover new devices. During the discovery, the new devices pass to the *Aggregator* their device-level information (manufacture identifier and model number) as well as application-level information (service and/or data type) in order to have a name assigned by the naming service (Sec. III-C). It is important to point out that it is possible to discover two kinds of devices: (i) with pre-load secure keys; (ii) without pre-load secure keys. In both cases, the performed tasks do not depend on the kind of device (temperature sensor, GPS device, video camera). In the first case, where the embedded system is programmable before deployment, the owner can preload identity information (such as secure ID, a pair of public/private key and a certificate), or has some manufacture ID and a pair of public/private key (which is certified by the manufacturer). Therefore, the device is associated with information including device identity, public/private keys (PK_{device}, SK_{device}) and a certificate, either from the owner or the manufacturer, who certifies the device identity and public/private keys. When such a device is discovered, the *Aggregator* first verifies the device identity (the device can generate a signature with the private key SK_{device} and present the signature and the certificate to the *Aggregator* so that the *Aggregator* can verify it). If the verification succeeded, then the *Aggregator* sends back the action key AK_{device} (encrypted with the signature key SK_{device}), which will be used for the transmission of data by the device, in order to guarantee confidentiality and integrity of the transmitted information. The *Aggregator* locally stores the action key AK_{device}, the signature key SK_{device}, and the secure ID. Summarizing, each device owns two keys: a signature key SK_{device} and an action key AK_{device}. The former is used for access control operations, while the latter for data encryption. In fact, in order to guarantee confidentiality and integrity, the *Aggregator* makes two operations, as illustrated in Fig. 2. In the first one, it verifies the signature key SK_{device} (function *identityVerification*). While, in the second one, the *Aggregator* sends back the action key AK_{device} (the symmetric key or the private key) encrypted with the device signature key SK_{device} (function *actionAssign*). These actions represent an enforcement mechanism, which improves the robustness of the proposed approach, guaranteeing the policy satisfaction. In fact, without the right execution of the first action the second transmission is not performed at all, thus blocking any other actions. In presence of GPS information, it is important to guarantee integrity and confidentiality, since several issues could arise if data is compromised (violation of user privacy, bad decisions taken by the traffic managers or by healthcare operators). Moreover, the proposed solution is suitable for the integration with the available algorithms for key distribution and revocation, which would improve the robustness of the system [12] [13]. According to device resources, it is possible to use either public/private encryption techniques (RSA, DH) or symmetric ones. Hence, the information are encrypted and a set of keys is used. In the second case, where devices are only associated with the secure manufacture ID, without public/private keys and the certificate being pre-loaded, it is critical to assure that devices are authenticated. A solution could be to use another trust model. For example, the system could take advantage of the web-of-trust model or the contextually semantic information, so that the devices manufactured by the same vendor can authenticate each other. Moreover, in order to comply with the capability of resource-restricted devices, light-weight cryptographic primitive may be used. A possible solution should be the following, as reported in Fig. 3. The *Aggregator*, after receiving the secure ID from the device (function *newDevice*), sends back to the device the signature key SK_{device} along with the certificate (function *signatureAssign*); then the device sends to the *Aggregator* its secure ID encrypted with the received signature key SK_{device} (function *signatureConfirm*). As a response, the *Aggregator* sends to the device the action key AK_{device} encrypted with the signature key SK_{device} (function *actionAssign*), as in the previous case. Clearly, such a solution is less robust than the first one, since the secure ID is transmitted in clear the first time, but such a device discovery phase is executed into a three-way form. Finally, the network service discovery would be hosted on *LSG*s or *Aggregator*s. The devices periodically broadcast their services/data, which will be responded by other devices that need these services/data.

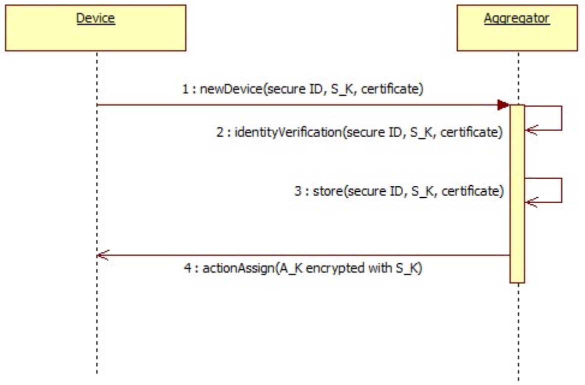
Device
Aggregator
1 : newDevice(secure ID, S_K, certificate)
2 : identityVerification(secure ID, S_K, certificate)
3 : store(secure ID, S_K, certificate)
4 : actionAssign(A_K encrypted with S_K)

Fig. 2. Secure device discovery - with pre-load secure keys

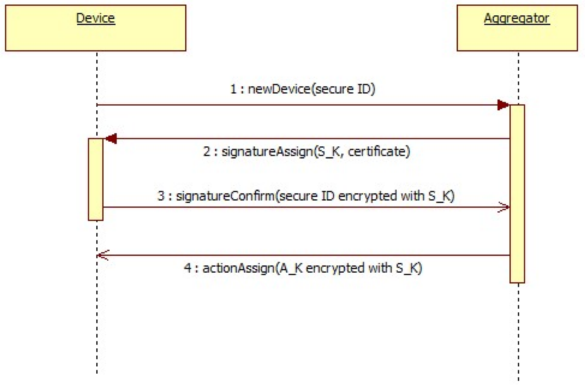
Device
Aggregator
1 : newDevice(secure ID)
2 : signatureAssign(S_K, certificate)
3 : signatureConfirm(secure ID encrypted with S_K)
4 : actionAssign(A_K encrypted with S_K)

Fig. 3. Secure device discovery - without pre-load secure keys

B. Secure service discovery

The scope of service discovery transaction is to discover and advertise IoT services to the rest of the IoT system. In our example, GPS information should be propagated to interested users/vehicles/healthcare systems in real time. Many research activities on service discovery has been conducted, but privacy has often been ignored, even if it is essential for legitimating the users to discover the desired services. It is also necessary that services were hidden from illegitimate users: a malicious entity might know that a specific user is in a particular route at a certain time, and not at home, thus causing serious damages. In our reference architecture, a host, which wants to exploit the information of the GPS service makes a request to *ICN-IoT Server*. As shown in Fig. 4, first of all it makes an access using its signature key SK_{device} and its secure ID (function *hostAccess*); then, in order to guarantee confidentiality, integrity and non repudiation, the request is encrypted with the host action key AK_{device} and signed with the signature key SK_{device} (function *hostServiceRequest*). *ICN-IoT Server* forwards the request to the *Aggregator* (function *hostServiceRequest*) that is able to decrypt the request using the stored keys (function *serviceDecryption*) and try to satisfy it (function *serviceRequestSatisfaction*). Note that a host could request a known service, as just described, or only explore the available services. In the latter case, the *Aggregator* sends back a list of the services, which could satisfy the host request. The possibility for a host to access a particular service is determined by its credentials, which establish the access permissions policies. The proposed solution also fits in a distributed approach, as among IoT embedded systems, but an orchestration entity is needed (*ICN-IoT Server*).

C. Secure naming service

Naming service is in charge to assign and authenticate the device names. In IoT perspective, the naming is based on IP addresses, which are insecure, not persistent, and not scalable. Instead, ICN introduces the name concept (names are separated from the data locators). The names assigned to each

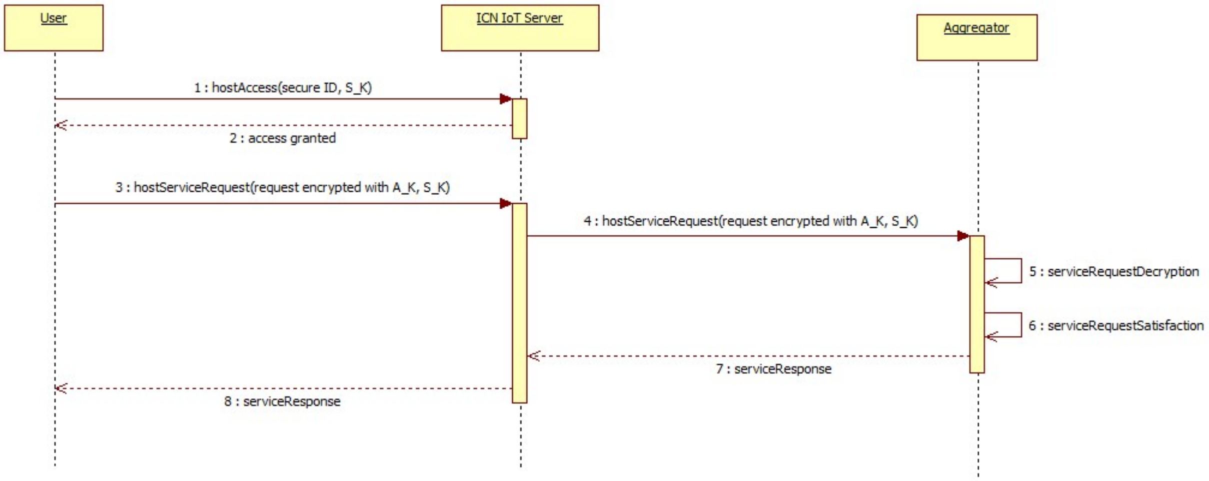
User
ICN IoT Server
Aggregator
1 : hostAccess(secure ID, S_K)
2 : access granted
3 : hostServiceRequest(request encrypted with A_K, S_K)
4 : hostServiceRequest(request encrypted with A_K, S_K)
5 : serviceRequestDecryption
6 : serviceRequestSatisfaction
7 : serviceResponse
8 : serviceResponse

Fig. 4. Secure service discovery

device are unique and persistent. Some existing solutions [14] [15] [16] are available, but only for ICN field, without taking into account the features of IoT paradigm (mobility, devices heterogeneity). Wong et al. [14] aim to provide a flexible naming system to enable content retrieval from multiple, and also untrusted, sources with a security mechanism embedded in the name. The security functionality are separated from the routing and storage primitives, allowing the content to be verified regardless of its location. [15] introduces a group-oriented naming concept that integrates various available group schemes and simplifies rendezvous processes. An open-source middleware prototype of this name-oriented multicast access has been implemented. In [16] the naming scheme offers persistent IDs even though the content or location change. In our approach, as represented in Fig. 5, the *Aggregator* sends the secure device ID, just received by the new GPS device (which performed the device discovery operation - Sec. III-A), the signature key SK_{device} and the action key AK_{device} to the *LSG* (function *newDevice*). Then, the *LSG* assigns an ICN name and generates a certificate, thus certifying the binding of ICN name and signature key SK_{device}. Then, the *LSG* communicates such a name to the *Aggregator*, which sends it, encrypted with the related action key AK_{device}, to the new GPS device (function *nameAssign*). Finally, the *Aggregator* and the *LSG* store the ID, the signature key SK_{device}, the action key AK_{device}, the ICN name and the certificate for the future communications with the device itself (function *nameStore*). The same naming mechanism can be used to name higher-level IoT devices such as *Aggregator*s and *LSG*s.

D. User registration

A user, who wants to access/subscribe to a service (in this example, the traffic monitoring service or an healthcare system), has to make the registration operation by sending his/her username, along with other sensitive/useful information (birth date, address, nation, language), as shown in Fig. 6 (function *registration*). Such information depend on the specific application domain, but, for our example, may be limited to those useful for customizing the offered service (language, nation, address). The corresponding *ICN-IoT Server* assigns an identifier, a user signature key SK_{user} and a password to the user and sends them to him/her (function *registrationData*) who, after the reception, has to access to the system modifying his/her password (function *changePassword*). At this point, the *ICN-IoT Server* sends to the user his/her action key AK_{user}, encrypted by means of his/her signature key SK_{user} (function *actionAssign*). The user action key AK_{user} is used for future information exchanging, in order to guarantee confidentiality and integrity to the content. Also in this case, the action key AK_{user} should be a symmetric key or a private key, according to the adopted encryption technique. With respect to existing secure application-layer solutions, a further benefit of the presented approach is the introduction of a second level of security, represented by the use of a temporary password (immediately replaced) and a couple of keys (signature SK_{user} and action AK_{user}), which is well suited for the heterogeneous and distributed IoT environment.

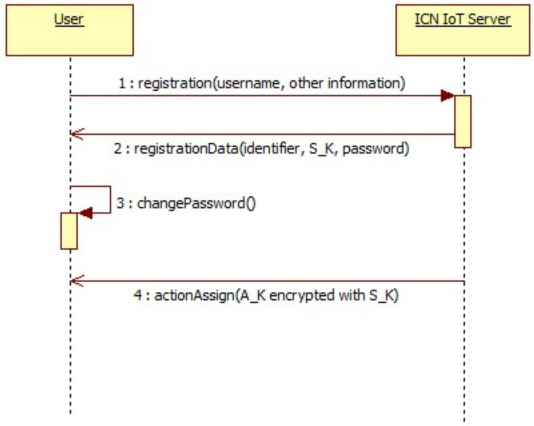
User
ICN IoT Server
1 : registration(username, other information)
2 : registrationData(identifier, S_K, password)
3 : changePassword()
4 : actionAssign(A_K encrypted with S_K)

Fig. 6. User registration: secure subscribe

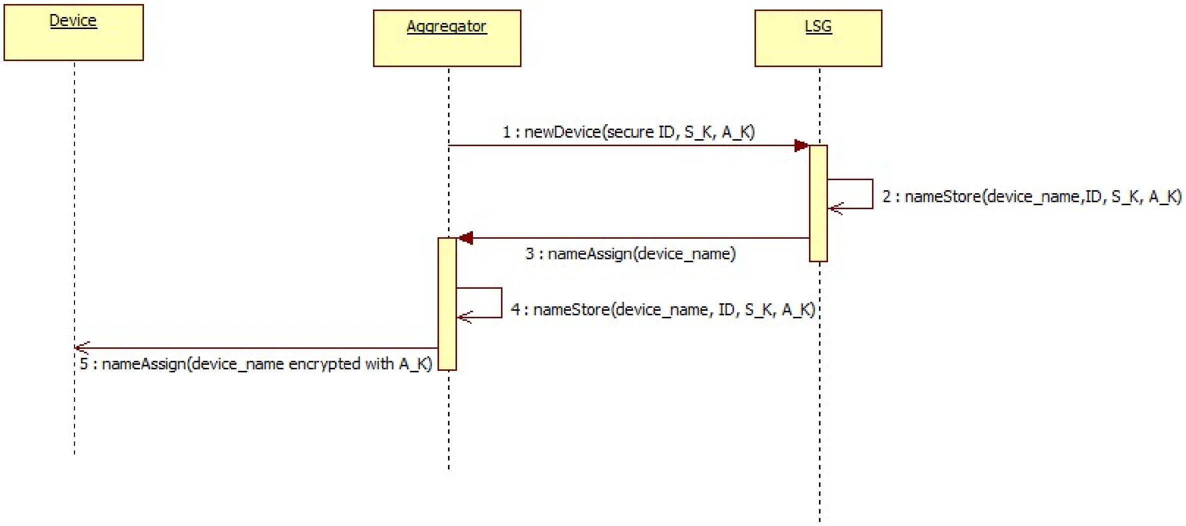
Device
Aggregator
LSG
1 : newDevice(secure ID, S_K, A_K)
2 : nameStore(device_name,ID, S_K, A_K)
3 : nameAssign(device_name)
4 : nameStore(device_name, ID, S_K, A_K)
5 : nameAssign(device_name encrypted with A_K)

Fig. 5. Secure naming service

E. Secure content delivery

Another imporant issue regards the security of the content delivery within the ICN-IoT system. In literature, there are few solutions able to guarantee content security in ICN. [17] aims to ensure a high availability of the cached data only to legitimate users. The authors design a security framework for ICN able to deliver trusted content securely and efficiently to legitimate users/subscribers. Instead, Zhang et al. [18] propose a name-based trust and security protection mechanism. Their scheme is built with identity-based cryptography, where the identity of a user or device can act as a public key string. However, a solution able to take into account both ICN and IoT features is still missing. Therefore, content delivery has been revised here to guarantee the security requirements. In our method, GPS device, which has already perfomed the device discovery and naming service, sends to the *Aggregator* its ICN name, its ID encrypted with its signature key SK_{device} and the GPS data acquired in real time, encrypted with its own action key AK_{device}, in order to guarantee confidentiality and integrity, as shown in Fig. 7 (function *contentDelivery*). The action key AK_{device} has been distributed during the device discovery (Sec. III-A). The *Aggregator* is able to decrypt the GPS data using the corresponding action key AK_{device}, stored with the device ID, the signature key SK_{device} and the device ICN name obtained during the name service (Sec. III-C), in particular the aggregator uses the device name to go back to the related action key AK_{device} (function *contentDecryption*). Data are encrypted only if it is required by the application domain (some contexts may not have any security requirements, so the function *contentDecryption* is not applied). As regards the content delivery towards a user who subscribes to GPS service, the *ICN-IoT Server* transmits to the user the GPS data in real time, encrypted with the user action key AK_{user}, in order to guarantee security and privacy (function *contentDelivery* in Fig. 8), if it is a requirement of the application domain, as this case. The user decrypts the received GPS data using his/her action key AK_{user} (function *contentDecryption*). Following the presented approach, the services are treated as multiple-unicast ones, since the aggregator has to use different keys for different devices (two different GPS devices will not be provided with the same credentials). In order to address a multicast approach, a group signature key system may be adopted.

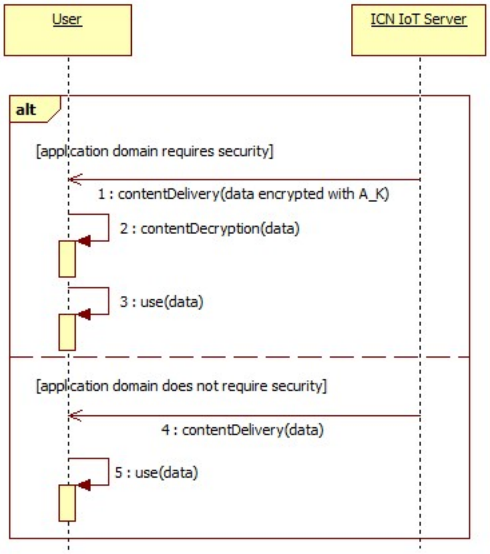
User
ICN IoT Server
alt
[application domain requires security]
1 : contentDelivery(data encrypted with A_K)
2 : contentDecryption(data)
3 : use(data)
[application domain does not require security]
4 : contentDelivery(data)
5 : use(data)

Fig. 8. Secure content delivery - user

IV. Conclusion

The real spreading of ICN-IoT services requires proper security and privacy levels to be guaranteed. The scientific community has to keep in mind such issues, since the early stage of network design, as done in this paper. The paper

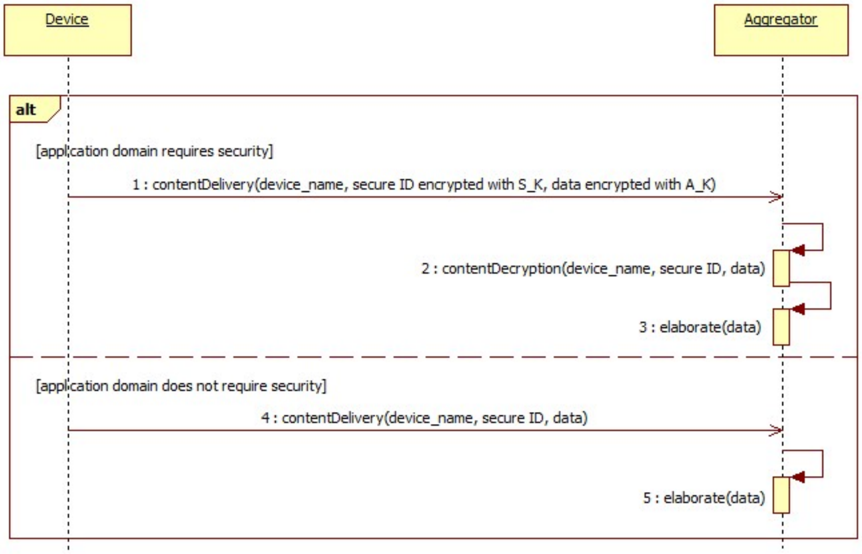
Device
Aggregator
alt
[application domain requires security]
1 : contentDelivery(device_name, secure ID encrypted with S_K, data encrypted with A_K)
2 : contentDecryption(device_name, secure ID, data)
3 : elaborate(data)
[application domain does not require security]
4 : contentDelivery(device_name, secure ID, data)
5 : elaborate(data)

Fig. 7. Secure content delivery - device

has presented the integration of security functionalities into an ICN-IoT middleware architecture, proposed as IRTF draft [11] within the ICNRG. Device and service discovery, naming service, content delivery have been integrated with security solutions, in order to prevent violation attempts. UML have been exploited to better clarify the operations involved. It represents a valuable starting point, which should allow the development of an ICN-IoT platform. For the future, we are planning to test the effectiveness of the proposed approach in a real scenario, evaluating the level of security and privacy of transmitted information. Moreover, an integration with the existing *NOS (NetwOrked Smart object)* IoT architecture [2] [19] will be considered.

Acknowledgment

This work was partially supported by the Department of Theoretical and Applied Sciences of University of Insubria (Varese, Italy) and the “Bonvoyage” and ”Green Community Efficiency Systems” research projects funded by the EU H2020 program (grant agreement No 635867) and the Italian MISE (code B01/0768/03/X24), respectively.

References

[1] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead,” *Com. Net.*, vol. 76, pp. 146–164, 2015.

[2] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, “A secure and quality-aware prototypical architecture for the Internet of Things,” *Inf. Sys.*, vol. 58, pp. 43–55, 2016.

[3] D. Conzon, T. Bolognesi, P. Brizzi, A. Lotito, R. Tomasi, and M. Spirito, “The VIRTUS middleware: An XMPP based architecture for secure IoT communications,” in *21st Inter. Conf.on Computer Communications and Networks*, Munich, Germany, Jul 2012, pp. 1–6.

[4] C. H. Liu, B. Yang, and T. Liu, “Efficient naming, addressing and profile services in Internet-of-Things sensory environments,” *Ad Hoc Net.*, vol. 18, no. 0, pp. 85–101, 2013.

[5] A. Gòmez-Goiri, P. Orduna, J. Diego, and D. L. de Ipina, “Otsopack: Lightweight semantic framework for interoperable ambient intelligence applications,” *Computers in Human Behavior*, vol. 30, pp. 460–467, Jan 2014.

[6] M. Amadeo, O. Briante, C. Campolo, A. Molinaro, and G. Ruggeri, “Information-centric networking for {M2M} communications: Design and deployment,” *Comp. Comm.*, vol. 89-90, pp. 105–116, 2016.

[7] L. A. Grieco, M. B. Alaya, T. Monteil, and K. Drira, “Architecting information centric etsi-m2m systems,” in *IEEE nternational Conference on Pervasive Computing and Communications*, 2014, pp. 211–214.

[8] M. Aiash and J. Loo, “An integrated authentication and authorization approach for the network of information architecture,” *Journ. of Net. and Comp. Appl.*, vol. 50, pp. 73–79, 2015.

[9] R. Neisse, G. Steri, and G. Baldini, “Enforcement of security policy rules for the internet of things,” in *IEEE 10th Inter. Conf. on Wireless and Mobile Computing, Networking and Communications*, Oct 2014, pp. 165–172.

[10] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, “Security policy enforcement for networked smart objects,” *Com. Net.*, vol. 108, pp. 133–147, 2016.

[11] Y. Zhang, D. Raychadhuri, L. Grieco, R. Ravindra, and G. Wang, “ICN based architecture for IoT,” in *http://trac.tools.ietf.org/group/irtf/trac/attachment/wiki/icnrg/draft-zhang-icn-iot-architecture-01.txt*, Jul 2015.

[12] G. Jolly, M. Kuscu, P. Kokate, and M. Younis, “A low-energy key management protocol for wireless sensor networks,” in *8th IEEE Inter. Sym. on Comp. and Comm.*, Jun 2003, pp. 335–340.

[13] G. Dini and L. Lopriore, “Key propagation in wireless sensor networks,” *Computers & Electrical Engineering*, vol. 41, pp. 426–433, 2015.

[14] W. Wong and P. Nikander, “Secure naming in information-centric networks,” in *Re-Architecting the Internet Workshop*, 2010, p. 16.

[15] T. C. Schmidt, M. Wahlisch, D. Charousset, and S. Meiling, “On name-based group communication: Challenges, concepts, and transparent deployment,” *Comp. Comm.*, vol. 36, no. 15-16, pp. 1657–1664, 2013.

[16] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, “Secure naming for a network of information,” in *IEEE Conf. on Comp. Comm.*, Mar 2010, pp. 1–6.

[17] S. Misra, R. Tourani, and N. E. Majd, “Secure content delivery in information-centric networks: Design, implementation, and analyses,” in *3rd ACM SIGCOMM Workshop on ICN*, 2013, pp. 73–78.

[18] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi, and G. Wang, “Towards name-based trust and security for content-centric network,” in *9th IEEE Inter. Conf. on Network Protocols*, Oct 2011, pp. 1–6.

[19] A. Rizzardi, S. Sicari, D. Miorandi, and A. Coen-Porisini, “AUPS: An open source AUthenticated Publish/Subscribe system for the Internet of Things,” *Inf. Sys.*, vol. 62, pp. 29–41, 2016.