Building a reliable Internet of Things using Information-Centric Networking

George C. Polyzos & Nikos Fotiou

Journal of Reliable Intelligent Environments

ISSN 2199-4668

J Reliable Intell Environ
DOI 10.1007/s40860-015-0003-5

Springer

J Reliable Intell Environ
DOI 10.1007/s40860-015-0003-5

ORIGINAL ARTICLE

Building a reliable Internet of Things using Information-Centric Networking

George C. Polyzos[1] · **Nikos Fotiou**[1]

Received: 21 December 2014 / Accepted: 31 March 2015

Abstract Recent developments in sensors, devices, identification technologies, and wireless networking have fueled the vision of the Internet of Things (IoT). Small devices with processing, sensing, and connectivity capabilities can be connected to the Internet and produce vast amounts of meaningful information. At the same time identification technologies, such as RFID, enable the association of information with "things". The information produced by things, or associated with things, will be both huge and sensitive. For this reason new architectures for disseminating and processing this information in a reliable and efficient way should be explored. In this paper, we present an architecture for the IoT, based on the Information-Centric Networking (ICN) paradigm. ICN architectures are built around information and information identifiers and they provide mechanisms for advertising, finding, and retrieving information. We leverage a particular ICN architecture, the Publish-Subscribe Internetworking architecture, to design an IoT architecture and we present three security solutions that enable access control, secure delegation of information storage and trust based on information identifiers.

Keywords Publish-Subscribe Internet · Security · Access control · Storage delegation · Trust

✉ Nikos Fotiou
fotiou@aueb.gr

George C. Polyzos
polyzos@aueb.gr

[1] Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, Evelpidon 47A, 113 62 Athens, Greece

1 Introduction

Devices equipped with sensing, actuator, processing, and connectivity capabilities are becoming ubiquitous and cost effective. In 2010, the number of devices connected to the Internet was 12.5 billion, whereas the world's population was 6.8 billion: this was the first time in history where the number of connected devices per person was more than 1 [8]. This rise of the number of connected devices fuels the vision of the Internet of Things (IoT) and more generally the "Internet of Everything." In the IoT, "smart" devices will produce "meaningful" information and will share it with other devices and in some cases, users. The IoT has the potential to create a $7.3 trillion market size [21]. Building a reliable, including secure, IoT is a challenging and at the same time burgeoning problem.

IoT security is challenging for many reasons. Security solutions cannot rely on the traditional end-to-end paradigm since "things" will not be "always connected" and information dissemination will rely on caches, proxies, and gateways. Things can be easily tampered with and thus secrets could be extracted. Most things are not expected to have the same processing and storage capabilities as modern PCs and servers, and updating their software is not expected to be straightforward.

At the same time IoT security is an escalating problem as the intrusive nature of things raises new serious concerns. A thing can be a smartphone, a sensor in our house or at the workplace, a wearable sensor that measures vital body signs etc. Therefore the Internet, through the IoT, moves much more from the virtual to the real world, with more immediate and potentially more significant impact on our lives. Therefore, confidentiality, integrity and availability of the information should be protected and new, more flexible and adaptable to the context, access control mechanisms should

Published online: 01 May 2015

be developed. Moreover, the introduction of third parties that act as indirection points that facilitate information processing and delivery raises the importance of the development of security mechanisms that can vouch for the authenticity and the provenance of each piece of information.

In this paper, we propose to use Information-Centric Networking (ICN) as a technology to integrate the current silos that have been developed in the IoT area and discuss the problem of IoT security, including availability, at the information level. Inspired by recent advances in ICN research [28], we adopt an approach in which we choose to secure the information itself, rather than the communication channels, or the storage and processing nodes.

ICN is a new (inter-)networking paradigm, which brings in the core of all network functions information and information identifiers. Rather than relying on end point, location-dependent identifiers (i.e., IP addresses), ICN provides mechanisms that allow information "advertisement" and "retrieval" using flexible and semantics-rich information identifiers.

In [20], we proposed an information lookup system for the IoT based on the concepts of the Publish-Subscribe Internetworking (PSI) ICN architecture [27]. In this paper, we enhance that proposal and present the design of a reliable IoT architecture that facilitates the development and the deployment of information-centric security mechanisms. To this end, we discuss how various security solutions can be incorporated into the system. In particular, we discuss the applicability of access control delegation mechanisms, proxy re-encryption schemes, and name-based trust. For each security solution, we provide an updated design of the architecture that includes the new, security-related entities. Moreover, we develop new communication protocols and security procedures, demonstrating the capabilities, as well as the impact, of these security solutions onto the IoT architecture.

The remainder of this paper is organized as follows. In Sect. 2, we introduce ICN and the PSI architecture and we survey some ICN-based IoT architectures. In Sect. 3, we detail our PSI-based IoT architecture design. In Sect. 4, we discuss some key security solutions that contribute to the architecture's reliability. Finally, in Sect. 5, we present our conclusions and discuss future work in this area.

2 Background

2.1 ICN and the PSI architecture

Information-Centric Networking (ICN) is an emerging paradigm that has received increasing attention in recent years. ICN is believed to overcome various limitations of the current networking architectures, including inefficient mobility handling, lack of effective multicast, insecurity and distorted business environment. A defining characteristic of ICN architectures is the use of content (information) names as the key identifier, which can also serve as a new abstraction layer between applications and the network.

An ICN architecture is composed of the following entities[1]:

- Owner: A real world or a network entity that creates, owns, and names an information item.
- Publisher: A network device that actually hosts an information item.
- Subscriber: A network device that belongs to a real world entity that is interested in an information item.
- Rendezvous Node (RN): A network entity that acts as an indirection point between subscribers and publishers. The main functionality of a RN is to match subscriptions with publications. Therefore, it accommodates subscriber interests and publisher availability for particular information items. All RNs are organized in a *Rendezvous Network* (RENE).[2]

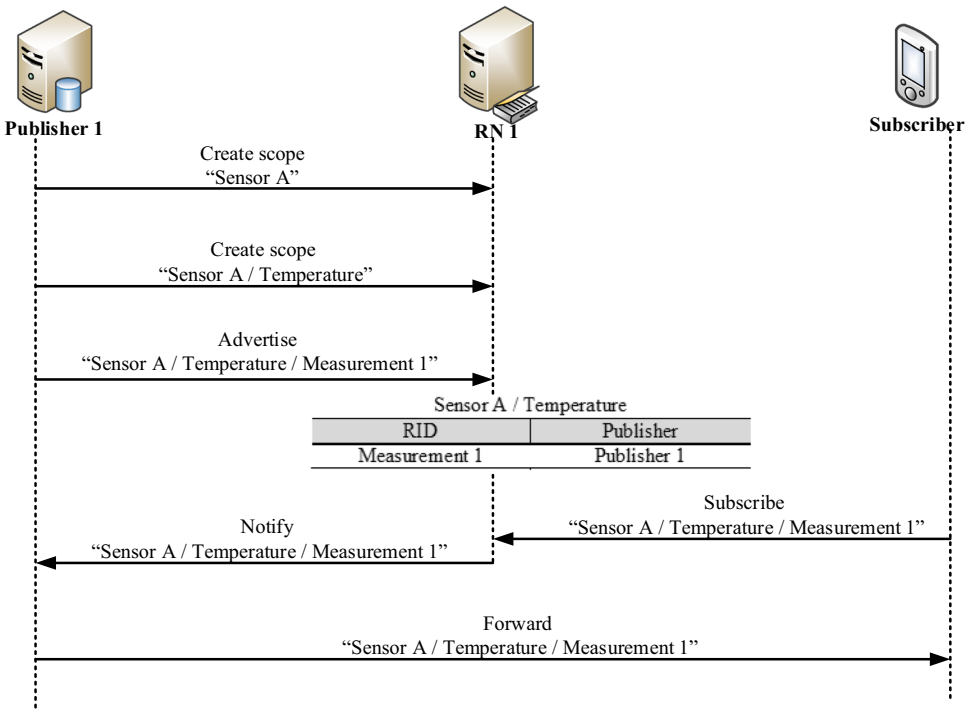
These entities interact with each other in the following manner: An owner creates an information item, assigns a *name* to it and stores a copy of this item in at least one publisher. The publishers `advertise` the information items they host. The advertisement of an item is received and kept by some RNs in the network. A subscriber sends a `subscription` message that is routed through the RENE and eventually reaches a RN that has a matching entry for that item of interest. A successful match will ultimately result in the content being `forwarded` from a publisher to the interested subscriber(s). Intermediate nodes may opportunistically *cache* a forwarded item and act as additional publishers for that item in the future.

In PSI, every information item is uniquely identified by a pair of identifiers, the *Scope* Identifier (SID) and the Rendezvous Identifier (RID). SID denotes the scope in which an item belongs. Scoping allows organizing information items into collections (e.g., a set of temperature measurements). Scopes are hierarchically structured, forming parent-to-

[1] Even though the terminology we use here is based on PSI these entities can be found in all ICN architectures, in many cases they might have different names and sometimes they might be distributed across the network, e.g., in the base CCN and NDN architectures, the rendezvous function is distributed across all network nodes; however, there have been proposed application-specific enhancements that make the function more explicit and introduce specific RNs.

[2] In general, there can be many independent RENEs managed by different providers. Each RENE needs to obtain (potentially incomplete) subscription and publication information and they could have different levels of trust by various publishers/owners and subscribers.

Fig. 1 PSI example



child relationships and this hierarchy is reflected in SIDs (e.g., "Sensor A/Temperature"). The RID is provided by an application-specific function. An RID must be unique in the scope to which it belongs and an SID must also be unique within its parent scope. Every SID (and therefore all RIDs belonging to that SID) is managed by a RN. RENEs in PSI are constructed using overlay networks.

Figure 1 illustrates an example of a typical PSI transaction. In this example a publisher, Publisher 1, creates two new scopes, one with SID "Sensor A" and one with SID "Sensor A/Temperature", hence the second scope is a child of the first. As a next step, Publisher 1 advertises a new item under the second scope with RID "Measurement 1". At this point, the RN creates a lookup table for the scope "Sensor A/Temperature". This lookup table has an entry which indicates that there is an item in this scope, with RID "Measurement 1", and that there is a publisher for that item, i.e., Publisher 1. Later on, a subscriber subscribes for the newly created item. Since the RN has a matching entry, it notifies the publisher. The notification message includes a path to the subscriber which is used by the publisher in order to forward the desired content item.

The form of a RENE is an application-specific design choice which mostly depends on the SID format. For example, (top-level) SIDs can be domain names; in this case the RENE can follow the DNS hierarchy. Similarly, (top-level) SIDs can be arbitrary strings; in that case the RENE can be implemented as a DHT in which each (hashed) SID will be managed by the RN whose identifier is numerically closer to the (hashed) SID (e.g., as in [16]).

2.2 ICN-based IoT architectures

ICN has been regarded as a promising candidate for building IoT architectures. Various research efforts have studied the implications of ICN to IoT (and vice versa) and have proposed requirements that have to be fulfilled. Rayes et al. [26] believe that ICN is expected to be the "most common deployment of the IoT" and they address ICN performance and security requirements of IP-based IoT networks. They argue that architectures should be built using optimal hybrid models that support centralized and distributed systems at the same time. Moreover, they advocate that new security solutions have to be developed to cope with security requirements such as authentication, privacy, resistance to (D) DoS and identity thefts attacks.

In [11], we propose a research agenda for future ICN-based IoT architectures. In particular, they identify research challenges that concern: information naming, efficient and contextual information retrieval, trust models, privacy, access control, and information forwarding. Amadeto et al. [2] devise requirements for an IoT architecture based on the NDN ICN architecture [29]. They argue that such an architecture should support *pull-based* and *push-based* data communication, it should support discovery protocols, it should use naming schemes that achieve optimal routing performance and facilitate data sharing, it should provide authenticated *interests* (the corresponding of subscriptions in the NDN architecture), it should support multiple caching strategies, and it should support data forwarding over heterogeneous networks. Sugang et al. [19] compare NDN and Mobility-

First [25] architectures by considering two IoT scenarios. They consider the discovery and forwarding mechanisms of these two architectures and they measure various performance indicators such as delay, routing state, and control overhead.

Moving a step further, many research efforts have used ICN for implementing an IoT system. Baccelli et al. [6] measure the performance of the CCN ICN architecture [15] when used in an IoT system. In particular, they use a lightweight version of CCN, code-named CCN Lite, and they install it onto 60 devices equipped with the RIOT operating system [5]. They measure the performance of various routing protocols and the impact of caching. Their experiments indicate that CCN offers advantages over an approach based on 6LoWPAN/IPv6/RPL in terms of energy consumption and RAM and ROM footprint.

Biswas et al. [7] utilize the CCN ICN architecture in order to implement a contextualized information-centric home network. CCN is used in their architecture in order to provide automatic node and service discovery and policy-based service publication and subscription. Francois et al. [12] explore optimizations for the routing strategies of the CCN ICN architecture. In particular, they propose the support of both *pull* and *push* strategies. Since CCN already supports pull-based routing (i.e., a subscriber first subscribes for an item and then the item is forwarded), they develop a mechanism for pulling data. In particular, they assumed that the size of the data to be pulled is small, therefore, it "fits" within a subscription message. With this in mind, they encode the pulled data in a subscription message which is broadcasted to all intended recipients.

Grieco et al. [14] utilize the NDN ICN architecture to implement an Overlay Service Capability Layer (OSCL) that can be used in ETSI M2M systems. This layer encodes available services as information items and it can be used for distributed service discovery and invocation. This is an improvement to the centralized Service Capability Layer (SCL) proposed by ETSI. Piro et al. [22] develop a platform that can be used for enabling services in a "Smart-City" environment, using the NDN ICN architecture. A service in their platform is executed in three phases. First comes the *Discovery* phase during which a subscriber finds potential publishers that can satisfy a request, the second phase is the *Security Initialization* phase during which a subscriber retrieves some security-related information, and the final phase is the *Service Usage* phase which is the actual invocation of the service. All operations are encoded as information items and they can be invoked using subscription messages.

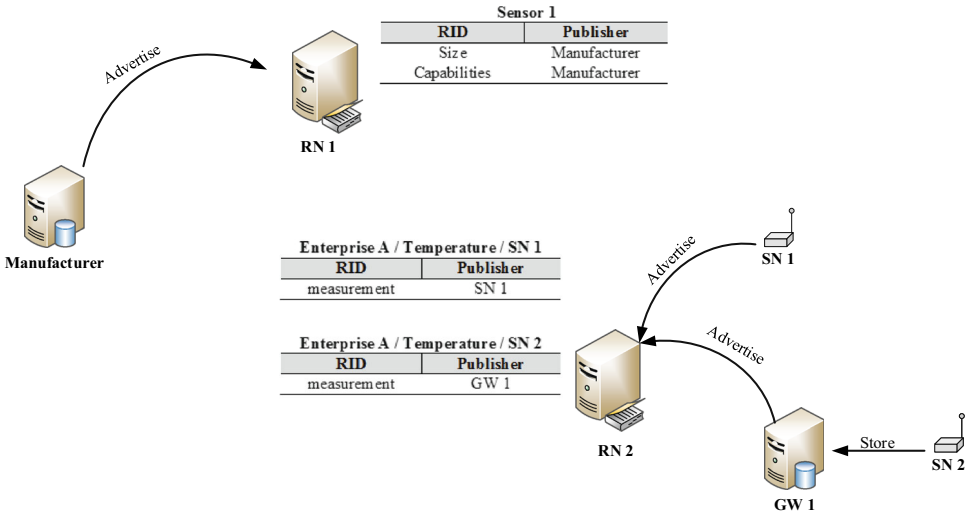
Amadeo et al. [3] utilize the NDN ICN architecture in order to build an IoT architecture that supports multi-source data retrieval. In particular they extend the NDN architecture and add support for "prefix-based" interests. For example, a subscriber interested in learning temperature measurements from sensors deployed in his home would issue an interest of the form "home/temperature". On the other hand, sensors should advertise their measurements using "home/temperature" as a prefix (e.g., a sensor deployed in the kitchen would advertise "home/temperature/kitchen"). This procedure will result in the user issuing a single interest packet and receiving multiple information items.

3 A PSI-based architecture for the IoT

In our PSI-based architecture for the IoT (hereafter will be referred to as PSI4IoT) all things have identifiers. The granularity and the purpose of these identifiers are application specific; there can be identifiers that are thing specific (e.g., vehicle identification numbers), or there can be identifiers that are specific to a group of similar things (e.g., a barcode that identifies a product). A thing may have multiple identifiers and identifiers can be context specific. For example, a sensor may have an identifier that represents its brand and another identifier that represents its current network attachment address. Information can be created by, or associated with a thing. For example, a sensor that measures temperature can create an information item that represents the current measurement, its manufacturer may associate with it an information item that represents its components, and a retail store may associate with it an information item that represents its price. A thing may act as the publisher of the information it generates, or it may appoint other network devices to hold this role. In the temperature measurement example, a sensor may either store the measurements itself, or delegate them to a gateway or proxy.

Figure 2 illustrates the above concepts. In this example, a manufacturer has created a series of sensors under the brand name "Sensor 1". The manufacturer has created an information item named "Size" that represents sensor's size and another named "Capabilities" that represents sensor's sensing capabilities. Both these items have been advertised under the scope "Sensor 1". An enterprise ("Enterprise A") uses two of these sensors to measure the temperature in a data center. Enterprise A has assigned to each sensor an identifier that represents its network location (it can be for example an IP address). The sensor with location identifier "SN 1" has a built-in SD card in which measurements are stored. Therefore, "SN 1" is able to fulfill the role of publisher. On the other hand, the sensor "SN 2" does not have storage capabilities, therefore it stores its measurements on a gateway ("GW 1"); GW 1 now becomes the publisher of SN 2 measurements. All measurements are stored as information items identified by an RID of the form "measurementXX". The measurements of "SN 1" are advertised under the scope "Enterprise A/Temperature/SN

Fig. 2 PSI4IoT example



1", whereas the measurements of "SN 2" are advertised under the scope "Enterprise A/Temperature/SN 2". Both these scopes are managed by the rendezvous node "RN 2".

Subscribers wishing to receive an information item, should subscribe to the item's SID, RID pair in the appropriate RN. Back to our use case, consider a temperature monitoring application that is interested in learning "measurement22" of "SN 1". The application knows that this information is stored under the scope "Enterprise A/Temperature/SN 1" managed by RN 2, therefore, it constructs the appropriate subscription message and sends it to RN 2 which in return notifies SN 1. PSI and PSI4IoT consider two information delivery modes: the *channel* mode and the *document* mode. When a subscriber subscribes for an information item that is delivered using the channel mode, every time this item is updated, the subscriber will "automatically" receive the new version of the item. This mode therefore creates a "persistent" state and it is suitable for cases such as real-time feeds. On the other hand, with the document mode a subscription will result in a single item being transferred from a publisher to a subscriber; with the completion of this transfer any state that has been created for this subscription is discarded. Subscribers may also subscribe for items that do not currently exist (or they are "unavailable"). When these items are created (or become available) they are forwarded to the subscribers. This "delay tolerant" mode of operation is of particular importance for the IoT.

Two features of the PSI architecture that contribute to the reliability of PSI4IoT are its *multicast* and *caching* capabilities. Subscriptions for the same item can be "merged" and the corresponding response can be delivered using multicast. Moreover, items can be cached by intermediate nodes which then act as publishers.

3.1 Information naming

An important design choice that PSI4IoT applications have to make is the exact form of the information items names. Various options can be considered, including the following:

A name can be bound to the item data. The name of an information item can be directly bound to its data (e.g., part of the name is the hash of the item data).Of course, this is only applicable to immutable data. A typical example would be a large document after it has been finalized (or a video, or video chunk). Such a binding has some interesting security properties. For example, a network node can easily verify that a forwarded item is what a user asked for, therefore it can prevent unwanted traffic (e.g., spamming). Moreover, these names can be easily disseminated (e.g., using a QR-code). On the other hand, such names are not memorable therefore a "search engine" or directory like mechanism may be required for finding them.

A name could be independent of (not bound to) the item data. This is particularly relevant for small pieces of information that are changing rapidly and are defined through their location of origin or more generally the acquisition context. In this case we name the source (real or virtual), rather than the information itself. For example, the temperature at some point in time and space could be named and then provided directly by a sensor or approximated (interpolated) through various sensor readings.

A name can be human readable. Human readable names are memorable and can be easily communicated by human beings. Such names may require a registry-like authority, which will resolve issues such as trademarks, copyrights, etc. A disadvantage of human readable names is that they require additional mechanisms for binding them to security primitives (e.g., digital certificates that map a name to a public key).

A name can be mutable or immutable. Mutable names are short lived and facilitate the deployment of privacy preserving mechanisms: it is difficult to continuously block (censor) a particular item if it changes name periodically and it is difficult to "watch" the subscribers of such item. Of course, the privacy preserving properties of such a scheme are highly dependent on how the names are generated, i.e., if it is easy for an attacker to predict the next name of the item, it may be easy to deploy privacy related attacks. Immutable names are long lived and facilitate information replication and caching: if an item does not (often) change name the probability of a cache hit increases.

An item may have multiple names. Supporting multiple names per item is essential for achieving *contextual information retrieval*. For example consider an information item that represents a temperature measurement. The following names can be valid "latest temperature measurement", "temperature in Athens", "temperature measurement of sensor 1", "temperature in my area". Such naming schemes require complex and intelligent mechanisms for mapping a name to an item.

3.2 Intelligent rendezvous

So far, RNs have been described as mere lookup "boxes" that match supply with demand. Nevertheless, RNs can be more intelligent and capable of orchestrating publishers in order to produce "meta-information". Consider for example the case of a network of temperature sensors deployed in a building. A monitoring application (running on some network node) might be interested in learning the "average temperature of the first floor." This information in PSI4IoT is considered yet another named information item, therefore the application can simply subscribe to it. In order to generate this item there should be a "special purpose" node in the network (e.g., the building gateway) that aggregates temperatures and calculates "averages." In order for this node to aggregate (to average) temperatures it has only to subscribe for "temperature measurements." (in this case on the first floor) Upon receiving the subscription to "average temperature of the first floor" the RN notifies the sensors deployed in the first floor to forward their measurements to the special purpose node (the gateway). This node calculates the average and advertises it to the RN as a new information item. Immediately, the RN notifies it (the gateway) to forward the newly created item to the application (node) that requested it.

Another related example presents another feature of the architecture. If instead of the "average temperature of the first floor" the "temperature of the first floor" (somewhat more loosely specified) is requested, then the RN could select any sensor on the first floor to send the temperature (e.g., if it had data about temperature publishers of the first floor, i.e., such publications, or it could anycast a request to discover temperature publishers of the first floor). In principle, the application would not need to know about or deal with low-level details of sensor distribution or addresses or placement on the first floor, nor would it need to do the math, which the PSI4IoT software would undertake, presenting a more abstract view of the things network to the application. On the other hand, the application might need to accept a loosely defined function of the temperature as the answer (allowing for example for failed or unresponsive sensors etc.).

4 Security mechanisms

In this section, we discuss security solutions for three significant security requirements: access control enforcement, secure delegation of information storage, and information-based trust. Although, these security requirements exist in any networking architecture, as we discuss in the following subsections, they are of particular importance for the IoT.

4.1 Access control delegation

Access control is an integral component of any IoT architecture, including PSI4IoT. It should be possible for owners to define access control policies that govern access to information items and/or scopes. As far as access control is concerned, the IoT introduces many new challenges. These challenges arise from the fact that an information item may be stored (i) in devices that can be easily tampered with or even be stolen (i.e., the things), or (ii) in locations that are not controlled by the information owner, e.g., caches, gateways, CDNs. These also apply to PSI4IoT: publishers may be devices with low processing capabilities that are not tamper resistant or devices that do not belong to the administrative realm of the owner. In these conditions access control is a challenging task.

Access control policies may be enforced either by a publisher or a RN. However, as already argued, publishers cannot be trusted neither to store an owner's user management system (which in most cases is essential for authenticating and authorizing subscribers) nor to process subscribers' credentials. RNs, on the other hand, are more powerful and better protected devices. Therefore, they are better candidates for enforcing access control policies. Nevertheless, this is not trivial since RNs are general purpose devices that usually do not belong to the administrative domain of an information owner. This raises two challenges. Firstly, a RN should be trusted to store a user management system and/or to process subscribers' credentials. Secondly, a RN should be able to "interpret" access control policies defined by different owners with different requirements.

To overcome these problems, we use the access control delegation scheme proposed in [9]. This scheme introduces

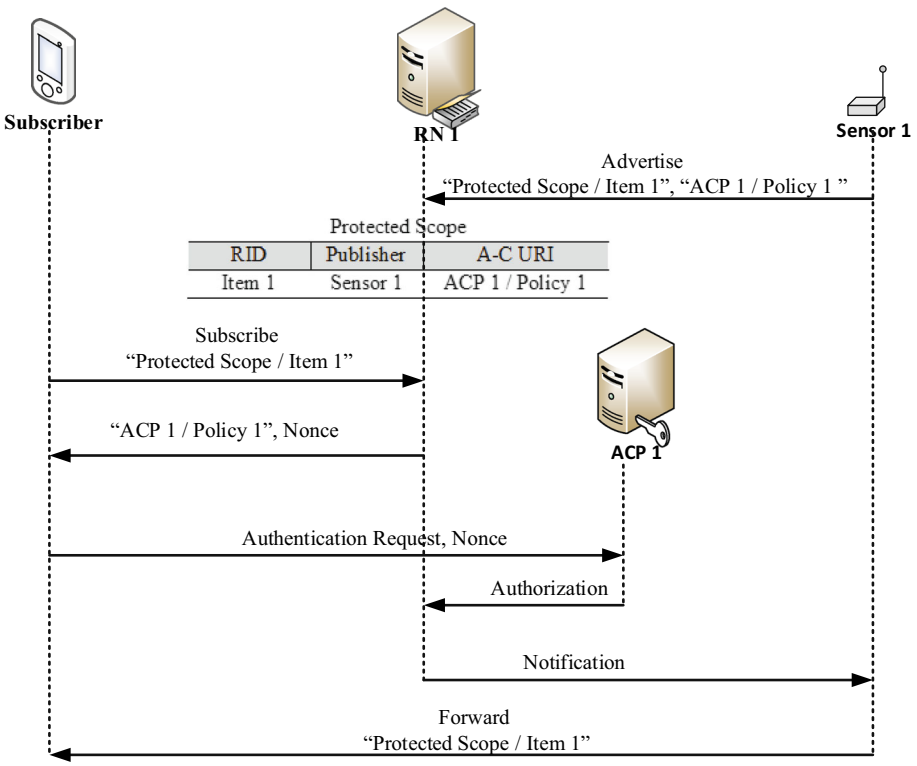


Fig. 3 Access control delegation example

a new entity, the *Access Control Provider* (ACP). ACPs are trusted network entities that may be owned by an owner, or may be provided as a services by a 3rd party, ACPs "host" user management systems, as well as access control policies. Each access control policy hosted in an ACP is identified by a URI. Figure 3 illustrates how this scheme is used in the PSI4IoT context. An owner creates and stores an access control policy in an ACP and receives back the corresponding URI. This URI is included in the advertisements of the items this policy protects and it is stored in the lookup tables of RNs. In the example of Fig. 3, the publisher "Sensor 1" has advertised an information item protected by an access control policy with URI "ACP 1/Policy 1". When a subscriber tries to access a protected item the following procedure takes places. Firstly, the RN generates a random number (token) and transmits it securely to the subscriber along with the URI of the access control policy. Secondly, the subscriber locates the ACP, authenticates himself, transmits the token, and requests authorization for the particular policy. If the subscriber is authorized the ACP creates and digitally signs an *approval* that includes the token and the URI of the policy and sends it to the RN. Finally, the RN notifies the publisher about the successful subscription.

In this system the following trust relationships exist: subscribers trust ACPs to store their identification details, publishers and owners trust ACPs to authorize subscribers, owners trust publishers to respect ACPs' decisions, owners and ACPs trust subscribers not to share their approvals with other subscribers.

The amount of trust required by this systems is much less compared to a system where access control policies are stored in RN. Moreover, the amount of trust required by this system is similar to capabilities tokens-based systems. The system is protected against malicious subscribers that: (i) can be authorized by URI_A and they want to subscribe to an item secured by URI_B or (ii) they used to be authorized by URI_A (but are not anymore) and they want to subscribe to an item secured by URI_A. The first type of adversarial subscribers is mitigated by including the URI of the policies in the (digitally signed) approvals and the second type of adversarial subscribers by including the token.

A feature of the proposed system is that subscribers' privacy is enhanced. What a RN learns about a subscriber is that she has business relationships with an ACP, as well as, that she is interested in an item. This information is much less, compared to the information that a thing would have learned if user credentials or access control policies were stored in RNs. Moreover, provided that access control policies are generic, subscribers' interests can be hidden from ACPs. Indeed, an ACP does not have to know in which item a subscriber is interested in order to evaluate her identification data against an access control policy.

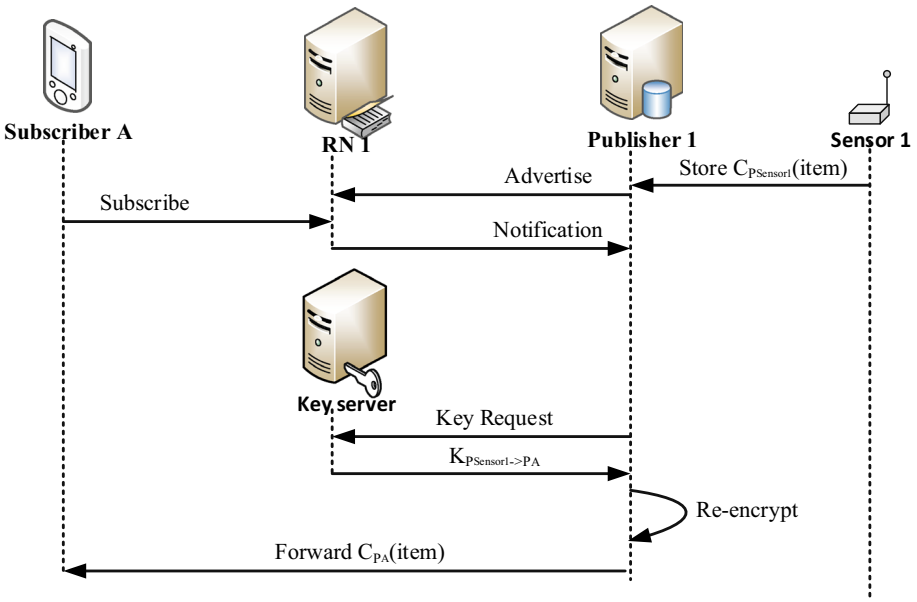


Fig. 4 Proxy re-encryption example

4.2 Secure publisher proxies

As it has already been discussed a thing that has generated an information item may appoint another network device to act as the publisher of this item. This is a highly desirable feature of the architecture, since it allows things to preserve computational power and storage capacity. Usually these network devices are a commodity used by many things, therefore information should be stored in a secure way. A straightforward approach for achieving this is by encrypting the stored information. In cases where all subscribers are reliable and they are known before the encryption process, the thing may share with them a symmetric encryption key and use this key for encrypting the information item in question. However, this is a rare case. An alternative approach would be the thing to encrypt information items using its own public key P_{thing}, store the ciphertext $C_{P_{\text{thing}}}$ in a publisher and every time a subscriber (that owns a public key P_{Sub}) requests an item, the publisher would "generate" $C_{P_{\text{Sub}}}$. A trivial approach to achieve this functionality is to reveal to the publisher the private key of the thing, i.e., K_{thing}. Then the publisher will be able to decrypt any ciphertext and re-encrypt it using the public key of the subscriber. Of course, this entails severe security threats. A better approach to implement this functionality is using a *proxy re-encryption* (PRE) scheme (e.g., [4]).

PRE schemes are cryptosystems which allow third parties, called proxies, to alter a ciphertext, encrypted with the public key of a user A (the delegator), in a way that another user B (the delegatee) can decrypt it with her own appropriate key (i.e. in most cases her secret private key). Generally a PRE scheme is composed of five algorithms, namely `KeyGen`, `Encrypt`, `RKGen`, `Reencrypt`, and `Decrypt`.

- `KeyGen`: is performed by a trusted party and generates user public-private keys.
- `Encrypt`: takes as input a public key P_{key} and a message M and returns the encryption of M using P_{key}, i.e., $C_{P_{\text{key}}}$.
- `RKGen`: takes as input a private key K_{key1}, a public key P_{key1}, and a public key P_{key2} and returns a re-encryption key $RK_{P_{\text{key1}}->P_{\text{key2}}}$.
- `Reencrypt`: takes as input a re-encryption key $RK_{P_{\text{key1}}->P_{\text{key2}}}$ and a ciphertext, $C_{P_{\text{key1}}}$, and returns a new ciphertext that can be decrypted with the private counterpart of P_{key2}.
- `Decrypt`: takes as input a ciphertext $C_{P_{\text{key}}}$ and the decryption key K_{key} and outputs a message M.

Figure 4 illustrates how secure delegation of information storage can be implemented using PRE. In this example it is assumed that there is a trusted entity that generates public-private key pairs, as well as, re-encryption keys. This entity is referred to as the *Key server* in Fig. 4. Initially a sensor generates an information item. Assume that this sensor cannot act as a publisher itself (e.g., because it has very low processing capabilities). For this reason it appoints another entity to act as a publisher, i.e., "Publisher 1". In order to protect the confidentiality of this item, the sensor encrypts it with its public key P_{Sensor1}. Then it stores it at Publisher 1 and goes offline. Publisher 1 follows PSI4IoT procedures and advertises this item. Then, "Subscriber A" subscribes for this item. Publisher 1 requests from the Key server the appropriate re-encryption key, transforms the ciphertext and sends it to the subscriber.

An interesting observation is that this solution can be combined with the access control delegation scheme described in Sect. 4.1. In that case, the ACP holds the role of the

Key server. Upon a successful subscriber authentication and authorization, the ACP generates the appropriate re-encryption key and sends it to the RN. Finally, the RN includes the re-encryption key in the notification message.

As a proof of concept we implemented a secure publisher proxy using the Identity-Based Encryption (IBE) PRE scheme proposed in [13]. The PRE scheme has been implemented using the Charm-Crypto library [1] in Python 2.7.[3] The secure publisher proxy has been implemented as follows. A thing generates information items and encrypts them using a symmetric encryption key (different for each item). Each symmetric key is then encrypted using IBE and the identity of the thing. The encrypted content items and the encrypted symmetric keys are stored in a publisher. To access the encrypted content, a subscriber needs to decrypt the symmetric encryption key. This can be achieved by having the publisher re-encrypting the symmetric key with the help of the Key server. In order to achieve a security level equivalent to RSA with key size 1024 bits, the size of the public system parameters of the IBE scheme should be 1024 bits, resulting in encrypted symmetric keys of size 2288 bits and in re-encryption keys of size 832 bits. Supposedly, RSA public key cryptography was used and the thing had encrypted every symmetric encryption key with the public keys of all subscribers. In that case, if x items had to be shared with y subscribers, the thing would have to generate $x * y$ different ciphertexts. In contrast, the described PRE Scheme would required only y re-encryption keys and x IBE ciphertexts.

4.3 Name-based trust

Trust in PSI4IoT (and in ICN in general) should be built around information and information identifiers, i.e., names, rather than on (secure) communication channels and storage (and processing) nodes. Therefore, subscribers should be able to verify the *integrity* and the *authenticity* of the information they receive, no matter the publisher and/or the communication channel. The integrity property of an information item guarantees that this item has not been tampered with during transmission. The authenticity property assures that an item is what a subscriber really asked for, i.e., it binds the item name with the item data. Integrity and authenticity are not the same: an item may have not been tampered with (therefore its integrity can be verified) but it may not be authentic. Information-based trust is of particular importance for the reliability of our architecture, as it facilitates information caching and replication; since trust is not based on end-hosts information can be cached even at unreliable nodes. In this subsection, we discuss how information-based trust can be achieved using *Hierarchical Identity-Based Encryption* (HIBE).

An Identity-Based Encryption (IBE) scheme is a public key scheme where an arbitrary string can be used as the public key. HIBE is a generalization of IBE reflecting organizational hierarchy. An IBE scheme is specified by four algorithms, namely `Setup`, `Extract`, `Encrypt` and `Decrypt`.

- `Setup`: takes as input a security parameter k and returns a `master-secret key` (MSK) and some `system parameters` (SP). The MSK is kept secret in a trusted key server whereas SP are made publicly available.
- `Extract`: takes as input SP, MSK, and an arbitrary string ID, and returns a `secret key` K_{ID}. ID can be used as a *public key* and K_{ID} is the corresponding private decryption key
- `Encrypt`: takes as input an arbitrary string ID, a message M, and SP, and returns a ciphertext C_{ID}.
- `Decrypt`: takes as input a ciphertext C_{ID}, the corresponding private decryption key K_{ID} and returns M.

The `Setup` and `Extract` algorithms can only be executed by the key server, the `Encrypt` algorithm can be executed by any entity that knows SP, and the `Decrypt` algorithm is executed only by the entity that possess the corresponding K.

In addition to the above algorithms, a HIBE scheme specifies the `Delegate` algorithm:

- `Delegate`: takes as input SP, K_{ID_1}, and a string of the form $\mathrm{ID}_1.\mathrm{ID}_2$ and outputs $K_{\mathrm{ID}_1.\mathrm{ID}_2}$.

The `Delegate` algorithm is of particular importance, as it enables the creation of a private key (K) without the involvement of the PKG. Another interesting property of HIBE is that a ciphertext $C_{\mathrm{ID}_1.\mathrm{ID}_2\ldots\mathrm{ID}_n}$ can be decrypted using any of the following secret keys K_{ID_1}, $K_{\mathrm{ID}_1.\mathrm{ID}_2}$, $\ldots$, $K_{\mathrm{ID}_1.\mathrm{ID}_2\ldots\mathrm{ID}_{n-1}}$, $K_{\mathrm{ID}_1.\mathrm{ID}_2\ldots\mathrm{ID}_{n-1}.\mathrm{ID}_n}$.

HIBE can be used in PSI4IoT for providing information-based trust as follows. For simplicity reasons we assume that there is a single RN, a single key server which has generated the MSK and the SP are well known. Moreover, we consider that information and scope names are human readable and that the creation of a root scope is a "controlled" operation, i.e., the RN assures that the entity that creates such a scope is eligible to perform this operation. For each root scope, the key server generates the corresponding K_{SID}, which is communicated to the entity that created that scope. We will refer to these entities as *scope owners*. Scope owners can authorize publishers to create sub-scopes and/or advertise items in these sub-scopes. The "authorization to create sub-scopes" process is achieved by generating $K_{\mathrm{SID/subscopeSID}}$ using the delegate algorithm and by dis-

[3] The source code of our implementation is included in latest release of the library.

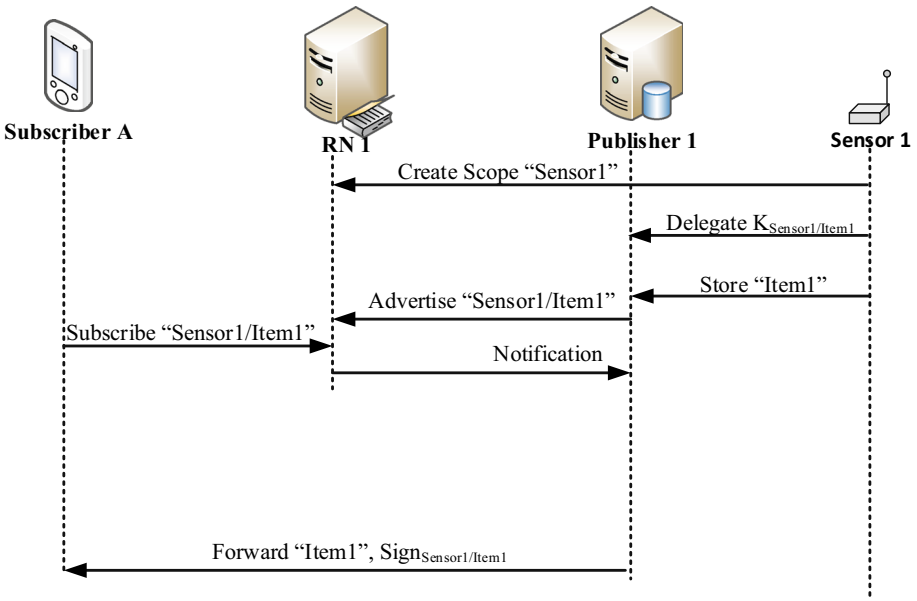


Fig. 5 Granting authorization to advertise an item

tributing it to "authorized" publishers. The "authorization to advertise a particular item under a (sub-)scope" is achieved by generating $K_{\text{SID/(subscopeSID)/RID}}$ using the delegate algorithm and by distributing it to "authorized" publishers. This process is illustrated in Fig. 5.

All authorized publishers calculate a digital signature for all sub-scopes they create and for all items they advertise. The digital signature of a sub-scope identified by SID is K_{SID} whereas the digital signature of an item identified by RID is $K_{\text{SID/RID/H(item)}}$, where $H()$ is a secure hash function. Any entity that knows SP can verify this signature using the following procedure. Firstly, it selects a random number r. Then, it encrypts r using as key either SID or SID/RID/H(item). Finally, it verifies if the calculated ciphertext can be decrypted using the digital signature of the item. The digital signatures are calculated using the delegate algorithm, therefore only authorized publishers can calculate them. Digital signatures are used during the creation of a scope and the advertisement of an item. With digital signatures a RN is able to verify that a publisher is authorized to perform the action in question. Moreover, given that authorized publishers are reliable, the digital signature of an item can be used for providing integrity and authenticity since it (a) contains the hash of the item and (b) it binds this hash to the item name. An unauthorized publisher is not able to calculate or modify a signature of an item, since it does not possess the necessary secrete information.

As a proof of concept, we have implemented[4] the Lewko-Waters [18] HIBE scheme using the Charm-Crypto library [1] in Python 2.7. The Lewko-Waters scheme is fully secure and it is based on bilinear maps applied over the elements of a group G of order p, where p is a prime number.[5] In [10] we use this implementation to build name-based trust mechanism and we show how name resolution infrastructure can be used for delivering the necessary system parameters.

5 Conclusion

It is a common belief that the IoT is going to be one of the next "big things." In this paper we argued that a new reliable IoT architecture is required, based on the expectation that the information that will be produced by the IoT will be vast and sensitive. This architecture should allow information dissemination in an effective and secure way. To this end, we presented the design of an IoT architecture based on the PSI ICN architecture. Being an ICN architecture, PSI provides efficient mechanisms for advertising, finding and retrieving information. Moreover, we described three security mechanisms that allow access control enforcement, secure information proxies, and trust establishment. These three security features are of particular importance for the IoT, as things usually should rely on the delegation of various functions to third parties due to their limited processing and storage capabilities. These third parties are usually employing or implementing shared infrastructures, which introduces new security challenges.

ICN architectures are not designed to be specifically IoT architectures; they are designed to be generic Internet archi-

[4] Source code available at: https://github.com/nikosft/HIBE_LW11.

[5] We have considered modification of the scheme for prime order settings [17].

tectures that take into consideration, among other things, the particular requirements that IoT introduces. This feature has a significant advantage: by applying the ICN paradigm to the IoT, we can bridge the gap between various "silos" of things, as well as, between the IoT and the rest of the Internet (to realize the Internet of Everything).

It can be argued that anticipating ICN to be the new "thin waist" of the Internet, above which all applications will be built, is improbable. Even though we are optimistic, we currently explore incremental deployments of ICN. In particular, in the I-CAN project [24] we investigate the potential of applying ICN in end-user devices and access networks. Early findings of this project demonstrate that in this environment and even with incremental only deployment, ICN brings significant advantages (in performance, access network planning and deployment, and application flexibility and security). In the POINT project [23] we focus on ICN deployment within a single ISP domain, consider the interactions with other domains using IP and specifically investigate the use of CoAP protocol for IoT, including over ICN.

Acknowledgments This research has been co-financed by the European Union (European Social Fund–ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)–Research Funding Program: Aristeia II/I-CAN.

References

1. Akinyele JA, Garman C, Miers I, Pagano MW, Rushanan M, Green M, Rubin AD (2013) Charm: a framework for rapidly prototyping cryptosystems. J Cryptogr Eng 3(2):111–128
2. Amadeo M, Campolo C, Iera A, Molinaro A (2014) Named Data Networking for IoT: An architectural perspective. In: 2014 European Conference on Networks and Communications (EuCNC), pp 1–5. doi:10.1109/EuCNC.2014.6882665
3. Amadeo M, Campolo C, Molinaro A (2014) Multi-source data retrieval in IoT via Named Data Networking. In: Proceedings of the 1st International Conference on Information-Centric Networking, INC '14, pp. 67–76. ACM, New York, NY, USA. doi:10.1145/2660129.2660148
4. Ateniese G, Fu K, Green M, Hohenberger S (2006) Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans Inf Syst Secur 91(1):1–30. doi:10.1145/1127345.1127346
5. Baccelli E, Hahm O, Gunes M, Wahlisch M, Schmidt T (2013) RIOT OS: Towards an os for the Internet of Things. In: 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp 79–80. doi:10.1109/INFCOMW.2013.6970748
6. Baccelli E, Mehlis C, Hahm O, Schmidt TC, Wählisch M (2014) Information centric networking in the IoT: Experiments with NDN in the wild. CoRR abs/1406.6608. arxiv:1406.6608
7. Biswas T, Chakraborti A, Ravindran R, Zhang X, Wang G (2013) Contextualized Information-centric home network. SIGCOMM Comput Commun Rev 43(4):461–462. doi:10.1145/2534169.2491691
8. Dave Evans D (2011) The Internet of Things how the next evolution of the Internet is changing everything. Cisco white paper
9. Fotiou N, Marias GF, Polyzos GC (2012) Access control enforcement delegation for Information-Centric Networking architectures. SIGCOMM Comput Commun Rev 42(4):497–502. doi:10.1145/2377677.2377773
10. Fotiou N, Polyzos GC (2015) Enabling NAME-based security and trust. IFIP Trust Management
11. Fotiou N, Polyzos Polyzos GC (2014) Realizing the Internet of Things using information-centric networking. In: 2014 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), pp 193–194. doi:10.1109/QSHINE.2014.6928688
12. Francois J, Cholez T, Engel T (2013) CCN traffic optimization for iot. In: 2013 Fourth International Conference on the Network of the Future (NOF), pp 1–5. doi:10.1109/NOF.2013.6724509
13. Green M, Ateniese G (2007) Identity-based proxy re-encryption. In: Katz J, Yung M (eds) Applied Cryptography and Network Security, vol 4521., Lecture Notes in Computer Science, Springer, Berlin Heidelberg, pp 288–306
14. Grieco L, Ben Alaya M, Monteil T, Drira K (2014) Architecting information centric ETSI-M2M systems. In: 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp 211–214. doi:10.1109/PerComW.2014.6815203
15. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pp 1–12. ACM, New York, NY, USA. doi:10.1145/1658939.1658941
16. Katsaros KV, Fotiou N, Vasilakos X, Ververidis CN, Tsilopoulos C, Xylomenos G, Polyzos GC (2012) On inter-domain name resolution for Information-Centric Networks. In: Bestak R, Kencl L, Li L, Widmer J, Yin H (eds) NETWORKING 2012, Lecture Notes in Computer Science, vol 7289. Springer, Berlin Heidelberg, pp 13–26. doi:10.1007/978-3-642-30045-5_2
17. Lewko A (2012) Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval D, Johansson T (eds) Advances in Cryptology-EUROCRYPT 2012, vol 7237., Lecture Notes in Computer Science, Springer, Berlin Heidelberg, pp 318–335
18. Lewko A, Waters B (2011) Unbounded HIBE and Attribute-Based Encryption. Advances in Cryptology—EUROCRYPT 2011, vol 6632., Lecture Notes in Computer Science, Springer, Berlin Heidelberg, pp 547–567
19. Li S, Zhang Y, Raychaudhuri D, Ravindran R (2014) A comparative study of MobilityFirst and NDN based ICN-IoT architectures. In: 2014 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), pp 158–163. doi:10.1109/QSHINE.2014.6928680
20. Marias GF, Fotiou N, Polyzos GC (2012) Efficient information lookup for the Internet of Things. In: 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp 1–6. doi:10.1109/WoWMoM.2012.6263786
21. Microsoft: Create the Internet of your Things. The Microsoft Cloud OS Vision (2014)
22. Piro G, Cianci I, Grieco L, Boggia G, Camarda P (2014) Information centric services in smart cities. J Syst Softw 88(0):169–188. doi:10.1016/j.jss.2013.10.029. http://www.sciencedirect.com/science/article/pii/S01641212130
23. POINT: Project home page (2015). http://www.point-h2020.eu/. (Last accessed 27 Mar. 2015)
24. Polyzos GC, Siris VA, Xylomenos G, Marias GF, Toumpis S (2014) I-CAN: Information-centric future mobile and wireless access networks. In: 2014 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), pp 139–141. doi:10.1109/QSHINE.2014.6928676

25. Raychaudhuri D, Nagaraja K, Venkataramani A (2012) MobilityFirst: a robust and trustworthy mobility-centric architecture for the future Internet. SIGMOBILE Mob Comput Commun Rev 16(3):2–13. doi:10.1145/2412096.2412098
26. Rayes A, Morrow M, Lake D (2012) Internet of Things implications on icn. In: 2012 International Conference on Collaboration Technologies and Systems (CTS), pp 27–33. doi:10.1109/CTS.2012.6261023
27. Xylomenos G, Vasilakos X, Tsilopoulos C, Siris VA, Polyzos GC (2012) Caching and mobility support in a publish-subscribe internet architecture. IEEE Commun Mag 50(7):52–58. doi:10.1109/MCOM.2012.6231279
28. Xylomenos G, Ververidis CN, Siris VA, Fotiou N, Tsilopoulos C, Vasilakos X, Katsaros KV, Polyzos GC (2014) A survey of Information-Centric Networking research. IEEE Commun Surv Tutor 16(2):1024–1049. doi:10.1109/SURV.2013.070813.00063
29. Zhang L et al (2010) Named data networking (NDN) project ndn-0001