OnboardICNg: a Secure Protocol for On-boarding IoT Devices in ICN

Alberto Compagno
Sapienza University of Rome
compagno@di.uniroma1.it

Mauro Conti
University of Padova
conti@math.unipd.it

Ralph Droms
Cisco Systems
droms@cisco.com

ABSTRACT

Information-Centric Networking (ICN) is an emerging networking paradigm that focuses on content distribution and aims at replacing the current IP stack. Implementations of ICN have demonstrated its advantages over IP, in terms of network performance and resource requirements. Because of these advantages, ICN is also considered to be a good network paradigm candidate for the Internet-of-Things (IoT), especially in scenarios involving resource constrained devices.

In this paper we propose OnboardICNg, the first secure protocol for on-boarding (authenticating and authorizing) IoT devices in ICN mesh networks. OnboardICNg can securely onboard resource constrained devices into an existing IoT network, outperforming the authentication protocol selected for the ZigBee-IP specification: EAP-PANA, i.e., the Protocol for carrying Authentication for Network Access (PANA) combined with the Extensible Authentication Protocol (EAP). In particular we show that, compared with EAP-PANA, OnboardICNg reduces the communication and energy consumption, by 87% and 66%, respectively.

CCS Concepts

•Networks → Security protocols; Mobile and wireless security; *Sensor networks;* **•Security and privacy → Security protocols;**

Keywords

ICN Security; Internet-of-things; Authentication

1. INTRODUCTION

In many large scale IoT scenarios, such as *Smart City*, *Electricity Metering* and *Environmental Monitoring*, resource constrained devices (i.e., devices with limited computational power, memory and energy) play a central role. Due to their low cost, they enable cost-effective scalable and reliable systems, in which thousands of devices form a wireless mesh network.

Interconnection of such devices is currently achieved through two different approaches: proprietary stacks and protocols such as ZigBee [34]; or open standards and protocol stacks such as IPv6 [17] combined with 6LoWPAN [29]. Unfortunately, both approaches have many drawbacks. The former does not provide a seamless connection to the Internet, thus requiring a translation step between the proprietary IoT network and the rest of the Internet. The latter allows for a seamless connection, but deploying the entire IP stack is still a challenge in term of computational resource requirements [5].

Recently, new Internet architectures have been proposed with the aim of improving the network performances and security of current Internet. Two such proposals, Named-Data networking (NDN) [25] and Content-Centric networking (CCNx) [14], are currently receiving a lot of attention from the research community in both IoT and general purpose scenarios [32, 3, 4]. Both NDN and CCNx implement the same Information-Centric Networking (ICN) paradigm, in which the current IP's host-centric model is replaced in favor of a content-centric model. Due to their simple design, NDN and CCNx resource requirements are smaller than the IP counterpart [5]. This makes them interesting network paradigm candidates for the Internet-of-Things.

Our contribution is the proposal of the first secure onboarding protocol for ICN that: (1) authenticates a new joining device and authorizes it to be part of the network, (2) provides the authentication of the network to the joining device in order to prevent its deployment in an untrusted network, (3) bootstraps the necessary key material to later secure the communication in the IoT Network (at link and network layer), and (4) is resilient to both *insider* and *outsider* adversaries.

We show that OnboardICNg can easily outperform, in both communication and energy overhead, the Extensible Authentication Protocol (EAP) combined with the Protocol for carrying Authentication for Network Access (PANA) [19]. EAP-PANA is the authentication protocol selected for the ZigBee-IP specification [35], thus we believe it is the most representative and adopted authentication protocol for IoT-over-IP mesh networks. Our results show that OnboardICNg and ICN reduce the communication and energy consumption, by 87% and 69,5%, with respect to IP and EAP-PANA. Such results confirm that OnboardICNg and ICN are a valid choice for the IoT world.

ICN'16, September 26 - 28, 2016, Kyoto, Japan

ISBN 978-1-4503-4467-8/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2984356.2984374

Motivation. Previous papers have already investigated confidentiality, integrity and access control of data in ICN communications between applications and sensors [8, 28]. However, none of them focus on the authentication and authorization phase of new devices joining an ICN *mesh* network, which we will call *on-boarding* phase. Although authenticating devices seems to collide with the ICN content-based security, we believe there are at least two good reasons for doing so, especially in the on-boarding phase:

1. Authenticating devices allows the distribution of proper keys for signing and encrypting ICN data to the appropriate devices. This will prevent unauthorized devices from accessing and altering ICN data.

2. Authenticating devices allows to create a *mesh* network of trusted devices and distribute the necessary cryptographic material to trigger integrity and confidentiality between neighboring devices, namely link-layer security. Especially in resource-constrained scenario, this is a key requirement to prevent untrusted devices from launching Denial of Service attacks on network and device resources (e.g., interest flooding [13] and link exhaustion [31]), as well as attacks on privacy (e.g., timing attacks [2, 12]) [20].

Organization. Section 2 gives an overview of the EAP-PANA protocol and briefly reports the basis of ICN networking. In Section 3, we present our system model, we analyze the IoT-over-ICN communication needs and outline our adversarial model. In Section 4, we present our on-boarding protocol. In Section 5, we analyze the security aspects of our proposal with respect to the adversary outlined in Section 3.3. In Section 6, we provide an evaluation of our proposal, and in Section 7 we present the related work. Finally, we conclude in Section 8.

2. OVERVIEW

In this section we give a brief overview of the current on-boarding approaches adopted in IoT-over-IP mesh networks (Section 2.1), as well as the ICN fundamental principles and basis (Section 2.2).

2.1 IoT-over-IP on-boarding approaches

Existing on-boarding approaches for IoT-over-IP mesh networks adopt a two-party protocol that provides an end-to-end mutual authentication between the new joining device and an authorization server. The most notable and common protocols are the well known EAP-PSK and EAP-TLS [30]. The former exploit symmetric encryption and a pre-shared secret between the two parties to mutually authenticate them. The latter uses the Transport Layer Security protocol (TLS) to establish a secure connection between the joining device and the authentication server. Compared to EAP-PSK, EAP-TLS has a greater overhead in terms of communication, computation, energy and memory.

Both EAP-PSK and EAP-TLS protocols assume that the two involved parties can communicate, i.e., either they are adjacent nodes or there exists a transport protocol that carries EAP messages among the two. For this reason, in IoT-over-IP mesh networks, EAP-PSK and EAP-TLS rely on an additional transport protocol, i.e., PANA. PANA defines three different entities: the new joining device called PANA client (PaC); the authentication server called PANA Authentication Agent (PAA); an intermediate node, called PANA Relay Element (PRE), which has the routing and forwarding information to relay packets between PaC and PAA.

PANA was designed as a *one-size-fits-all* solution to accommodate as many types of network as possible. Such solutions are usually inefficient for any specific application because of the inclusion of special features for specific deployment scenarios. In particular, the combination of EAP protocols with PANA has the following consequences: (1) EAP messages are encapsulated into PANA messages that in turn are enclosed into UDP packets, thus having a final packet grater than the IEEE 802.15.4 frame size specification; (2) every EAP-PANA message travels all the way from the PaC and the PAA involving intermediate nodes to use their limited resources (e.g., energy, computation and bandwidth) to forward EAP-PANA messages.

2.2 ICN

Rather than emphasizing communication between hosts as in IP networking, communication in ICN is achieved via content distribution. The ICN communication model can be characterized as using a pull model: content is delivered to consumers only upon (prior) explicit requests for that content, i.e., each content delivery is triggered by a request, carried in an interest message, for that content, which is then returned in a content message. ICN interest and content messages contain the name of the content of interest, along with several other fields. In this paper we are only interested in the following: `name` and `key locator` in the interest packet; `name`, `payload` and `signature` in the content packet.

Security in ICN follows a data-centric model. Each content is signed by the producer, allowing consumers to verify integrity and data-origin authentication, no matter what entity (either the producer or an intervening router) provided for the content (or its copy). Consumers can restrict the answer from a particular producer by including a key locator in the interest. This locator will contain the identifier of the key which will be used to verify the signature of the content. OnboardICNg uses these security mechanisms as an integral part of its secure on-boarding service.

3. SYSTEM MODEL AND DESIGN GOALS

In this section we present the system model, the design requirements and the adversary model, for which we build our secure on-boarding protocol and create a trusted IoT network.

3.1 System Model

We consider a wireless network composed of a number of constrained devices (sensors/actuators with low computational and storage resources, limited energy) and one (or few) Application Gateway devices (AGW). For simplicity, in the remainder of this paper we refer to any device, be it a sensor, an actuator, or both, as a constrained device (or device).

We take into account the case in which the span of the wireless network is greater than the range of the radios on each device. Due to this consideration, the topology of the network reflects the typical mesh network topology. We call this network setup *wireless mesh network*. A typical use case

for this type of network is given in Figure 1, where a neighborhood implements an *Advanced Metering Infrastructure* (AMI) network.

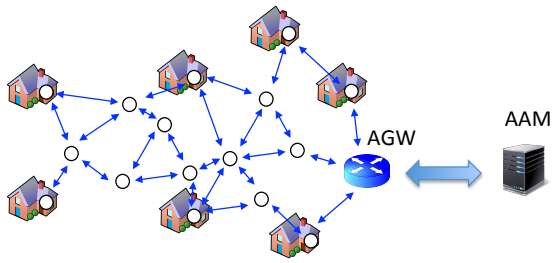

Figure 1: AMI Scenario

The devices are deployed in static locations, and can be deployed in the network over the time (i.e., not all of them are present at the initial time). An *AGW* provides an external connection to the IoT network, and moderates every access to and from the Internet according to a predefined access control list[1]. We further consider the existence of an Authorization and Authentication Manager (*AAM*) which stores the necessary information to securely bootstrap new devices into the network. It is worth mentioning that the *AAM* can be collocated with the *AGW*, or can be reached through it. In the latter case, an additional secure connection is required between the *AGW* and the *AAM*.

We consider a trusted IoT network to be a set of trusted devices that have been authenticated and authorized to be part of the network by the Authorization and Authentication Manager (*AAM*). In a trusted IoT network, trusted devices have already established the necessary key material to support link-layer security, as defined in the IEEE 802.15.4 standard [1], and the content-based ICN security. Moreover, we assume that trusted devices already have the necessary routing and forwarding information to communicate with the *AGW*.

Each device and the *AGW* can be both producer and consumer of content. All the content published in the IoT network shares a common namespace (e.g., */my_AMI*)[2] and it can be further organized into different sub-namespaces that collect common features. For example, all energy readings can be published under the namespace */my_AMI/readings*. We further consider the *AGW* as a secure storage that records and manages all the keys used to secure (authenticate and encrypt/decrypt) the namespace */my_AMI* and its subnamespace. The *AGW* will distribute such keys to the IoT devices that can access (publish or request content) the corresponding namespace. The access to a namespace is defined in the access control list the *AGW* maintains. However, in this work we only distribute the minimum key material to later allows secure key distribution. A complete key management is out of scope of this work. Moreover, we do not focus on the authentication and authorization of IoT operations (e.g., issuing queries or commands to devices) which have already been addressed in [28].

[1]In the ICN parlance, the access control list will specify the namespaces the IoT devices are allowed to access for either requesting or publishing content

[2]We use human-readable names for the sake of clarity. However, names within an IoT network are expected to be short and binary encoded due to the limited available space in the packets.

Before running OnboardICNg, a new joining device, hereafter d_j, must exchange some preliminary information with the *AAM*. This information will be used later to guarantee a secure on-boarding of the new device to the IoT network. We call this phase the *provisioning phase* and we consider the following information as the minimum requirement to run our on-boarding protocol.

- **A pre-shared key with the *AAM*:** The device d_j and the *AAM* must share some initial secret. This step can be achieved with different existing technologies, such as RFID and NFC. In this paper we indicate with psk_{d_j-AAM} a symmetric key that d_j and *AAM* share in this phase.
- **Knowledge of the network namespaces:** The device d_j must know the name prefix used to retrieve desired content (e.g., content published by the *AGW*) and the namespace under which d_j publishes its own content.

3.2 Design Requirements

We aim at designing an on-boarding protocol that satisfies the following requirements:

Mutual Authentication: The trusted network and the joining device d_j are able to mutually authenticate. In particular, *AAM* will verify that the joining request is coming from a device d_j that has previously performed the provisioning phase. On the other side, d_j will be able to verify that the authorization is: (1) received from an adjacent neighbor d_{nbr} that is already part of the trusted network it wants to join, (2) generated from the *AAM* of such network.

Fresh Authorization: The protocol guarantees that the authorization to join the network is fresh and unique, generated specifically for the current protocol session.

Minimal network traffic: The protocol minimizes the interaction with the *AAM* in order to preserve the overall network's and devices' resources.

Bootstrap the initial key material: The protocol must distribute the necessary cryptographic material to later allow a secure key management and communications. In particular, OnboardICNg will distribute a symmetric pairwise key k_{d_j-AGW}[3] between the *AGW* and d_j. Such key will be used to provide confidentiality during the distribution of the keys securing the namespace d_j can access. Moreover, the protocol will distribute a secure pairwise key, $k_{d_j-d_{nbr}}$, between d_j and each of its adjacent neighbors to support link-layer security (as defined in the IEEE 802.15.4 standard) unicast packet forwarding. The key $k_{d_j-d_{nbr}}$ can then be used to distribute other cryptographic material to secure 1-hop broadcast communication.

3.3 Adversary model

In our system, we consider both an insider and an outsider adversary. The insider controls a set of devices that are already part of an IoT trusted network (from now on *controlled devices*) and optionally a set of devices that are

[3]For a simple exposure, in this paper we assume a single key for integrity, authentication and confidentiality protection. However, different keys can be used without requiring any change to the protocol, thus adhering to the common best practices.

not part of the trusted network (from now on *malicious devices*). The outsider only controls a set of malicious devices. These malicious devices are deployed, without performing the provisioning phase, within the wireless range of the IoT network. Malicious and controlled devices have the same or even greater capabilities than the trusted devices (e.g., no energy, memory and computational resources constraint).

We consider an adversary that performs the following attacks against the on-boarding protocol: (i) *fraudulently join the trusted network*; (ii) *impersonate a trusted network*; (iii) *obtain the distributed symmetric keys*. In the first attack, the adversary tries to mislead the AAM and the AGW in order to force the authorization of a malicious node to be part of the trusted network. In the second attack, the adversary aims to convince a honest device that, after completing the on-boarding phase, it became part of a trusted network while it is actually connected to an adversary controlled network. In the third attack, the adversary tries to obtain the symmetric keys k_{d_j-AGW} and $k_{d_j-d_{nbr}}$.

4. SECURE DEVICE ON-BOARDING FOR IOT-OVER-ICN NETWORKS

We design OnboardICNg starting from the well known Authenticated Key Exchanged Protocol (AKEP2) [6], the authentication scheme that inspired EAP-PSK. While AKEP2 implements an end-to-end authentication between the two entities involved in the protocol, OnboardICNg involves three entities: the new joining device d_j, a device d_{nbr} adjacent to d_j that is already part of the trusted network, and the AAM. In the following we present our proposal. We start presenting the implementation of AKEP2 in the ICN communication style (i.e., a sequence of interest and content packets). On presenting AKEP2, we consider the two entities d_j and AAM as the two parties involved in the authentication protocol. Then, we present our proposal by extending AKEP2, including the third party, d_{nbr}. Table 1 reports the notation we use in the following to present both AKEP2 and OnboardICNg.

rn_X	Random number generated by the entity X
k_{X-Y}	Symmetric key shared between entities X and Y
id_X	Identity of entity X
MAC_k	MAC of the message calculated with the key k
$KDF()$	Key Derivation Function
$E_k(payload)$	Encryption of payload with the key k
$D_k(payload)$	Decryption of payload with the key k

Table 1: Notation

4.1 AKEP2 over ICN

AKEP2 is a two-party scheme assuming a pre-shared key between the two authenticating parties. Figure 2 depicts the three messages forming the AKEP2 scheme.

AKEP2 authenticates the two parties, in our case d_j and AAM, through the generation of two Message Authentication Codes (MACs), calculated with the shared secret psk_{d_j-AAM}.[4]. The validity of the MAC proves the ownership of the pre-shared key and thus the authenticity of the

[4]A well known best practice consists of deriving further keys from the pre-shared key avoiding its use to provide integrity and confidentiality.

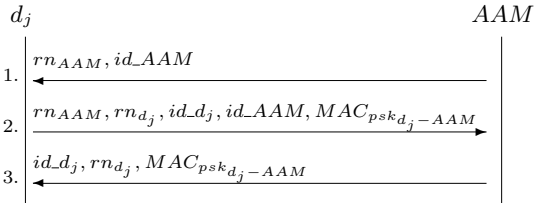

Figure 2: AKEP2

entity that generated it. Figure 3 depicts our implementation of AKEP2 over ICN.

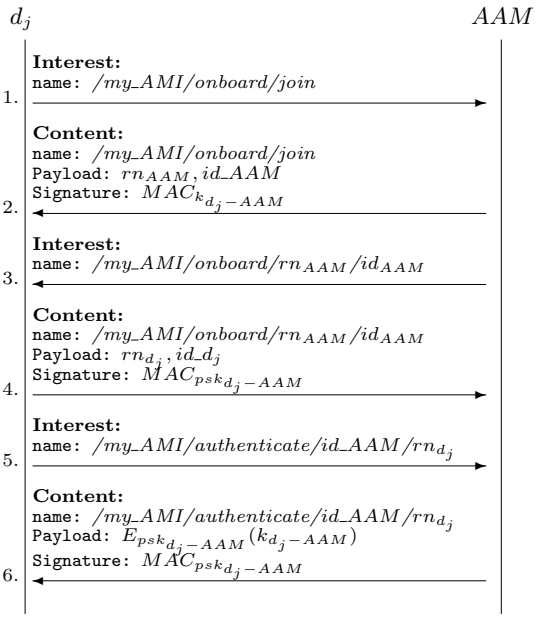

Figure 3: AKEP2 over ICN

The AKEP2 implementation in ICN translates each step of the original AKEP2 scheme into a pair of interest and content packets. The information exchanged at each step is encoded in the `name` or in the `payload`, while the MACs required by the AKEP2 scheme are encoded in the `signature` field present in every content packet. The only notable difference with the original AKEP2 scheme resides in Step 2 of Figure 3. In particular, the original AKEP2 scheme does not require any MAC in the first message; instead the AKEP2 implementation in ICN introduces a MAC, $MAC_{k_{d_j-AAM}}$, for the corresponding translated message (i.e., the content packet at Step 2 of Figure 3). This is due to the fact that ICN requires every content packet to be authenticated. Moreover, unlike the other MACs (steps 4 an 11 of Figure 3) that are calculated with the key psk_{d_j-AAM}, $MAC_{k_{d_j-AAM}}$ is calculated with a temporary key, k_{d_j-AAM}, generated by AAM. The key k_{d_j-AAM} will then be disclosed to d_j in the last step of the protocol, thus allowing d_j to verify the authenticity of the content received at Step 2. Even though k_{d_j-AAM} does not contribute to accomplish the mutual authentication, i.e., it is not used in the AKEP2 scheme, OnboardICNg will make use of such key.

The motivation for using k_{d_j-AAM} to calculate $MAC_{k_{d_j-AAM}}$ is that, at Step 2, AAM is not yet aware of the identity of d_j. Therefore, AAM cannot identify (and use) the pre-shared key corresponding to d_j.

4.2 OnboardICNg

OnboardICNg modifies the implementation of AKEP2 on the ICN protocol introducing an intermediate entity, d_{nbr}, between d_j and AAM. The role of d_{nbr} is to: (1) perform the AKEP2 scheme with the d_j, (2) prove to d_j that d_{nbr} is currently part of the trusted network, (3) act as a filter preventing adjacent unauthorized devices to forward packets to the trusted network. As for the original AKEP2 scheme, the pre-shared key is still known by AAM and d_j. Figure 4 depicts our three-party authentication protocol. For the sake of clarity, we describe OnboardICNg as performed only by one of the devices, d_{nbr}, adjacent to d_j. However, it is worth mentioning that the interest at Step 1 is broadcasted to all the adjacent devices. All the available adjacent devices (i.e., devices that have been configured to accept requests to join the network and that are not in a sleeping state to preserve energy) can continue to execute the protocol independently.

From Step 1 to Step 4, OnboardICNg reflects the first four steps of AKEP2 on ICN (Figure 3). During these four steps, d_j presents to d_{nbr} the proof of its authenticity, that is the MAC of the content packet in Step 4 (Figure 4). Such proof is also the first difference between OnboardICNg and AKEP2 on ICN. Indeed, while AKEP2 on ICN uses psk_{d_j-AAM} to calculate such MAC, OnboardICNg uses k', a session key derived from the pre-shared key as follows:

$$k' = KDF(psk_{d_j-AAM}; rn_{d_{nbr}}, rn_{d_j}, id\_d_j, id\_d_{nbr})$$

The generation of this key is fundamental to prevent the disclosure of psk_{d_j-AAM} to d_{nbr}, thus keeping psk_{d_j-AAM} secret between d_j and AAM.

At Step 5, d_{nbr} must retrieve the secret k' to conclude the AKEP2 scheme. Thus, d_{nbr} issues an interest requesting k' to the AGW and encodes in the interest the necessary information to derive k'. The name contains $rn_{d_{nbr}}$, rn_{d_j} and id_d_j, while the `key locator` field is used to both identify the key used to authenticate the corresponding content and to reveal the identity of d_{nbr}[5]. On receiving of such interest, AGW passes to AAM the necessary information to grant or to prevent the authorization of d_j to join the network. Then, if AAM grants the authorization, the protocol continues with AAM calculating k' (Step 6 in Figure 4) and sending back k' to d_{nbr}. On Step 8, d_{nbr} validates $MAC_{k'}$ received at Step 4 and then, at Step 9, it issues an interest to d_j. Such interest does request any specific content, but it acts as a notification for d_j to complete the AKEP2 scheme. In the final step, Step 11, d_j validates the $MAC_{k'}$ proving that d_{nbr} obtained k' from the AAM. Moreover, because AAM disclosed the key k' to d_{nbr}, d_j can also conclude that: (1) d_{nbr} can communicate with AAM, thus is already part of the trusted network, and (2) AAM authorized d_j to join the network through d_{nbr}. Moreover, the freshness of the authentication and authorization is provided by the two random numbers rn_{d_j} and $rn_{d_{nbr}}$.

[5] $id\_k_{nbr-AGW}$ identifies a symmetric pairwise key shared between d_{nbr} and AGW. We assume that AGW keeps track of the devices' identity associated with each symmetric pairwise key AGW owns. AGW can record such association on receiving k' from the AAM.

Besides the protocol changes explained above, OnboardICNg distributes a symmetric pairwise key between d_j and AGW, i.e., k_{d_j-AGW}. This key is generated by AAM and distributed to AGW and d_j (steps 7 and 8). To protect the confidentiality of k_{d_j-AGW} during its distribution to d_j, AAM encrypts the key with the pre-shared key psk_{d_j-AAM}. Moreover, OnboardICNg exploits the key $k_{d_j-d_{nbr}}$, whose distribution is described in Section 4.1, to secure the link-layer communication between d_j and d_{nbr}.

5. SECURITY DISCUSSION

In this section we provide a security discussion regarding the adversary goals detailed in Section 3. We first present an analysis considering an adversary that tries to mislead the on-boarding phase in order to connect a malicious device in the trusted network (Section 5.1). Then, we discuss the case in which the adversary aims at connecting a honest joining device to a malicious network (Section 5.2). We conclude with the scenario in which the adversary tries to obtain k_{d_j-AGW} and $k_{d_j-d_{nbr}}$.

In the following discussion we assume that: (1) an outsider can only inject, modify and eavesdrop packets between d_j and d_{nbr}; (2) an insider extends the outsider either controlling d_{nbr} during an execution of OnboardICNg or interacting with AGW from one of its controlled device; (3) the function KDC is secure, i.e., it is not possible to derive k' without the knowledge of psk_{d_j-AAM}, (4) encrypted messages can be decrypted only with the proper key.

5.1 Fraudulently join a trusted network

An adversary that wants to connect a malicious joining device md_j to a trusted network has to successfully complete the on-boarding protocol and retrieve the necessary key material. During the attack, md_j performs the steps depicted in Figure 4 for d_j.

Outsider. To fraudulently join the network, md_j has to be able to mislead d_{nbr} and pass the authentication step. We recall that OnboardICNg uses the AKEP2 scheme to perform the authentication and AKEP2 has been proven secure against an outsider [6] that does not know the shared secret. Therefore, the only way to mislead d_{nbr} and pass the authentication step is, for md_j, to obtain a valid k', i.e., the secret that OnboardICNg uses in the AKEP2 scheme. However, because d_{mj} does not know psk_{d_j-AAM}, deriving k' is not a possible action for it. The other way to obtain k' is to eavesdrop it from a previous session, while being sent from AGW to d_{nbr}. In this case, both the confidentiality provided be the IEEE 802.15.4 and the fact that k' is encrypted with $k_{d_{nbr}-AGW}$ prevent md_j from obtaining it. For these reasons, we can assume that OnboardICNg is secure too.

Insider. An insider can perform the attack following two different approaches. In the first approach, the insider tries to connect a malicious device md_j to a controlled device of the network. In order to be part of the network, md_j needs to be able to pass the authorization step at the AAM (Step 6 in Figure 4) and obtain a symmetric key, k_{md_j-AGW}, to later communicate with the AGW (Step 11 in Figure 4). However, the authorization phase is performed at the AAM and it requires md_j to express a valid identity (i.e., an identity associated to a pre-shared key stored in the AAM). Because md_j has not performed the provisioning phase, the

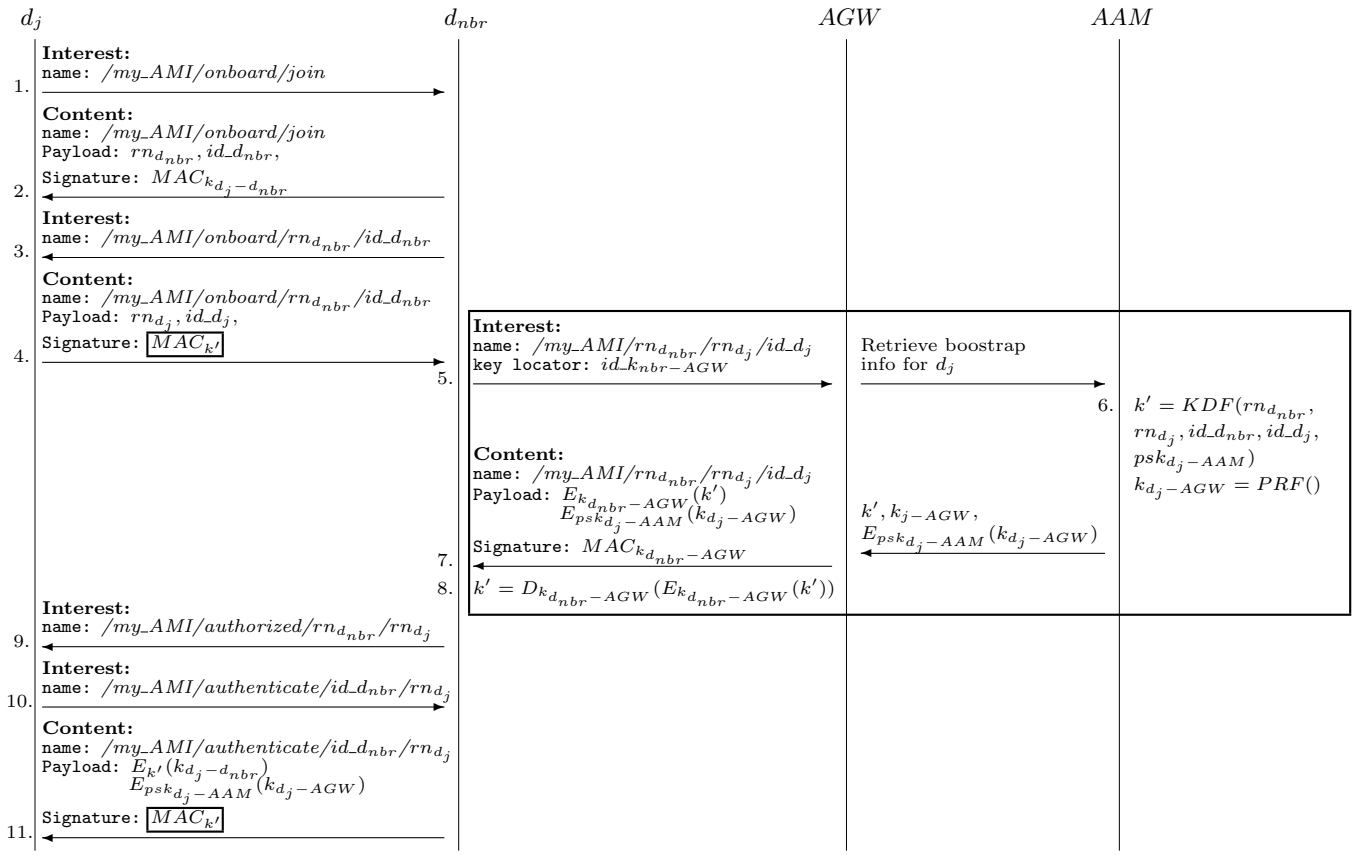

Figure 4: OnboardICNg. The additional steps and changes we made on AKEP2 are enclosed in a box

AAM will reject any authorization request containing the identity of md_j (Step 6 of Figure 4).

In the second approach, the insider performs a *clone attack*: the adversary replicates a controlled device that is already part of the trusted network and deploys the replica in a different part of the network. Because the replica has the necessary information to perform the on-boarding phase, our protocol itself cannot defeat such attack. However, adding the knowledge of the entire network topology to the AAM can reveal such attack. At Step 7, the AAM will in fact notice that the replica is trying to connect to a device that, according to the topology owned by the AAM, is not adjacent to the replica.

5.2 Impersonate a trusted network

In this attack, the adversary aims at connecting a honest joining device d_j to a malicious network. The malicious network is composed of a malicious application gateway $mAGW$ and a malicious authentication and authorization manager $mAAM$. In the outsider scenario, the malicious network also is composed of a number of malicious device md_{nbr} that are not part of the trusted network that d_j wants to join. In the insider scenario, the malicious network is composed of a number of controlled devices cd_{nbr} that are part of the trusted network that d_j wants to join. Both md_{nbr} and cd_{nbr} perform the steps depicted in Figure 4 for d_{nbr}.

Outsider. In this attack the outsider must be able to force d_j to authenticate the malicious device md_{nbr} as a trusted device. To pass such authentication step, the outsider must be able to either retrieve a valid k' or brake the AKEP2 scheme. We have already discussed in Section 5.1 that an outsider cannot do any of those actions.

Insider. The adversary exploits the situation in which an honest device d_j performs the on-boarding protocol with a controlled device cd_{nbr}. The adversary can successfully connect d_j to the malicious network if and only if d_j exchanges a symmetric key, k_{d_j-mAGW}, with the malicious $mAGW$. In order to achieve its goal, cd_{nbr} must be able to generate a spoofed packet containing k_{d_j-mAGW} at Step 11. While cd_{nbr} can perform OnboardICNg up to Step 10, it cannot generate a spoof packet at Step 11. In fact, generating such a spoofed packet requires the knowledge of psk_{d_j-AAM} to encrypt k_{d_j-mAGW}. However, psk_{d_j-AAM} is exchanged between d_j and AAM during the provisioning phase. Therefore, as long as the adversary is not able to compromise AAM, it cannot retrieve psk_{d_j-AAM} and generate a valid packet at Step 11.

5.3 Obtain the distributed symmetric keys

In this attack, the adversary tries to obtain the symmetric keys k_{d_j-AGW} and $k_{d_j-d_{nbr}}$, distributed during the an execution of on-boarding protocol. Since during the distribution, confidentiality of such keys is protected by the encryption, in the following we discuss an outsider and an insider that try to obtain the corresponding decryption keys, namely psk_{d_j-AAM} and k'. In the insider scenario, we assume the adversary does not control the devices owning the keys k_{d_j-AGW} and $k_{d_j-d_{nbr}}$, that are d_j and d_{nbr}. The in-

sider tries instead to obtain those keys from other controlled devices.

Outsider. The key psk_{d_j-AAM} is never sent through the network, and kept secret between d_j and AGW. Therefore, because an outsider does not control any of them, it has no way to obtain psk_{d_j-AAM}. Considering k', such key is encrypted with the key $k_{d_{nbr}-AGW}$ and then distributed to d_{nbr}. Thus, to access k', the outsider must first retrieve $k_{d_{nbr}-AGW}$ that was provided to d_{nbr} during its on-boarding. However, during its distribution, $k_{d_{nbr}-AGW}$ was encrypted with $psk_{d_{nbr}-AAM}$ which, as we have already discussed, is never revealed to the outsider.

Insider. As we discussed in the outsider scenario, psk_{d_j-AAM} is kept secret between d_j and AAM, which we assume cannot be controlled by the insider.

To obtain the key k', the insider can request such key from a controlled device cd_{nbr}. Since cd_{nbr} is able to forward interest to the AGW, the insider can try to request k' by forging and issuing the same interest at Step 5 (Figure 4). However, the derivation of k' follows the context binding principle [11], namely the identity of every entity that has access to the derived key is included in the input of the key derivation function. This forces the adversary to include the value $id\_k_{d_{nbr}-AGW}$ in the `key locator` which will result in obtaining k' encrypted with the key $k_{d_{nbr}-AGW}$. Because we have already discussed that neither an insider nor at outsider are able to obtain $k_{d_{nbr}-AGW}$, we can conclude that the insider cannot obtain k' too.

6. ANALYTICAL EVALUATION

In this section we compare the execution of OnboardICNg (i.e., the protocol depicted in Figure 4) with the execution of PANA protocol transporting the EAP-PSK [7] authentication messages (EAP-PSK/PANA). We choose the EAP-PSK/PANA implementation done in [27]. As in [27], we assume that the PANA protocol involves PANA relay nodes (PRE).

In our evaluation, we compare constrained devices having a similar role in the two protocols. The role of the constrained devices, and their similarity, is the following: PaC and d_j are both new joining devices in the corresponding protocol; PRE and d_{nbr} are trusted devices adjacent to PaC and d_j respectively. For such devices, we consider the following costs:

- **Communication cost**: the number of bytes sent and received during the authentication and authorization phase.
- **Computation cost**: the number or cryptographic operations performed during the authentication and authorization phase.
- **Energy cost**: the energy consumption as the sum of the energy required by the communication cost and the computation cost.
- **Memory cost**: the amount of memory required during the execution of the authentication and authorization phase.

In our analysis, we consider devices equipped with a MSP430 MCU combined with the CC2420 radio chip. The latter provides an hardware implementation of `AES-128`. We exclude PAA and AGW in our evaluation because they are deployed in a general purpose machine (e.g., router or server). Since AGW and PAA perform similar operations, we believe a general purpose machine can easily afford AGW operations. Finally, we assume that a single pair of devices is involved during the execution of both OnboardICNg and EAP-PSK/PANA (i.e., d_j and d_{nbr} in OnboardICNg; PaC and PRE in EAP-PSK/PANA).

Communication cost. We assume the underlying link-layer is the IEEE 802.15.4, with frames of 127 bytes. Moreover, we assume that frames exchanged between the new joining device and its adjacent neighbor (i.e., d_j and d_{nbr} in OnboardICNg; PaC and PRE in EAP-PSK/PANA) do not carry the 802.15.4 message authentication code. Thus, the 802.15.4 header and footer only requires 36 bytes. Instead, the 802.15.4 frames exchanged within the trusted network (i.e., d_{nbr} and AGW in OnboardICNg; PRE and PAA in EAP-PSK/PANA) carry the full 802.15.4 header and footer that is 52 bytes length.

We estimate the communication cost of OnboardICNg considering the 1+0 Encoding proposal for CCN [24]. We choose $rn_{d_{nbr}}$ and rn_{d_j} as 8 bytes random numbers while the size of the device's id (i.e., d_j and d_{nbr}) is 2 bytes. Moreover, we assume that the size of k' and k_{d_j-AGW} and k_{d_j-dnbr} is 16 bytes. Finally, we encode each name component different from $rn_{d_{nbr}}$, rn_{d_j} and id_d_j in 1 byte. Under these assumption all the interests fit in a single 802.15.4 frame, while only one content packet requires fragmentation. Table 2 reports the estimation of the communication cost, detailing the amount of sent and received bytes for each step of OnboardICNg. The content that requires fragmentation is indicated with (*). Such content is fragmented in two 802.15.4 frames[6].

Protocol message	d_j/d_{nbr}	d_{nbr}/AGW
Step 1 - Interest $d_j \rightarrow d_{nbr}$	52	0
Step 2 - Content $d_{nbr} \rightarrow d_j$	93	0
Step 3 - Interest $d_{nbr} \rightarrow d_j$	62	0
Step 4 - Content $d_j \rightarrow d_{nbr}$	95	0
Step 5 - Interest $d_{nbr} \rightarrow AGW$	0	88
Step 7 - Content $AGW \rightarrow d_{nbr}$	0	230*
Step 9 - Interest $d_{nbr} \rightarrow d_j$	68	0
Step 10 - Interest $d_j \rightarrow d_{nbr}$	62	0
Step 11 - Content $d_{nbr} \rightarrow d_j$	117	0

Table 2: Communication Cost (in bytes) in OnboardICNg

To estimate EAP-PSK/PANA communication cost, we refer to the size of the EAP-PSK/PANA messages reported in [27]. On such evaluation, we added the IPv6 and UDP headers compressed following the IPHC defined in 6LoWPAN. We assume that IPv6 header requires 2 bytes for the packets exchanged between PaC and PRE, while it requires 7 bytes for the packets exchanged between PRE and PAA. In both cases the UDP header requires 2 bytes. In case of fragmentation, each fragment carries the 802.15.4 header and footer, IPv6 and UDP header. In Table 3 we compare the total communication cost for each entity (i.e., the sum-

[6]We assume an end-to-end fragmentation, therefore each frame must contain the full ICN name.

mation of all the bytes sent and received during the authentication phase) of EAP-PSK/PANA and OnboardICNg.

EAP-PSK/PANA		OnboardICNg	
PaC/PRE	PRE/PAA	d_j/d_{nbr}	d_{nbr}/AGW
1380	2481	549	318

Table 3: Total cost for communications (in bytes) in EAP-PSK/PANA and OnboardICNg

In particular, it is interesting to note that the cost of the communication between d_{nbr} and AGW is 87% less expensive when compared with the communication between PRE and PAA. This achievement is mainly due to:

- The design of OnboardICNg that performs the main part of protocol between two neighboring devices, thus reducing the number of packets forwarded up to AGW. Instead, EAP-PSK/PANA performs the authentication protocol between PaC and PAA, involving all the devices in the path between PaC and the PAA to forward every EAP-PSK/PANA message.
- The exploitation of the ICN transport facilities which allow to use names to multiplex and demultiplex data among applications. Instead, EAP-PSK/PANA transports EAP packets over the PANA protocol, which in turn runs over the UDP protocol. This allows OnboardICNg to reduce the overhead in the packet header with respect to EAP-PSK/PANA.

Computation cost. Computation cost is driven by the security operations involved in the protocols. For both OnboardICNg and EAP-PSK/PANA we choose the following cryptographic functions: `AES-CMAC` for the message authentication code function and the `prf`, `AES-128` for encryption and the `kdf` function is defined as the `prf+` used in PANA [21]. The derivation function used by EAP-PSK is comparable to `prf`, therefore in the following we refer to it as `prf`. We assume that the size of the pre-shared secret PSK in EAP-PSK is 16 bytes (as reported in the EAP-PSK specification [7]) while psk_{d_j-AAM} in OnboardICNg is 128 bytes. In this analysis, we consider a further encryption key that both d_j and AAM derive from psk_{d_j-AAM} to encrypt/decrypt the key k_{d_j-AGW}. Moreover, we consider that both d_j and d_{nbr} derive an authentication key and an encryption key from k'.

In Table 4, we report the time needed to perform the security operations in OnboardICNg and EAP-PSK/PANA. Considering PaC and d_j, both entities have to calculate two MACs and verify two MACs which requires approximately the same time. In PaC, both EAP-PSK and PANA requires one MAC generation and one MAC verification each. In d_j, MAC generation and verification is driven by the number of content packets received and sent. Considering the keys generation phase, it requires comparable time in PaC and in d_j. This is because the number of keys derived in EAP-PANA and OnboardICNg is similar. PaC applies once `prf+` to derive `PANA_AUTH_KEY`, it applies `prf` to derive `AK`, `KDK` and `MSK`[7]. d_j uses twice `prf+` to derive k' and the encryption key from psk_{d_j-AAM}. Moreover, d_j uses twice `prf` to derive the authentication and encryption keys from k'. Finally, encryption and decryption costs are instead negligible. This is due to the fact that in both protocols the data encrypted are of one block, thus they can take full advantage of the `AES-128` hardware implementation.

Considering PRE and d_{nbr}, it is clear as d_{nbr} computation cost is greater than the PRE computation cost. This is because PRE devices do not perform any cryptographic computation during the authentication protocol. Instead d_{nbr} has to apply `prf` to generate $k_{d_j-d_{nbr}}$ and the authentication and encryption keys from k'. Moreover, d_{nbr} must verify/generate four MACs and decrypt k'.

	EAP-PSK/PANA		OnboardICNg	
Crypto op.	PaC	PRE	d_j	d_{nbr}
MAC gen./ver.	49,90	0,00	37,68	53,87
Keys gen./der.	22,75	0,00	23,05	0,90
Encrypt	0,00	0,00	0,00	0,30
Decrypt	0,30	0,00	0,60	0,30

Table 4: Cryptographic operation (in milliseconds) in EAP-PSK/PANA and OnboardICNg

Energy cost. We calculate the energy cost of EAP-PSK/PANA and OnboardICNg as the energy consumption of each cryptographic operation shown in Table 4 and data transmission reported in Table 3. We estimate the energy consumption using the measurement for transmitting and receiving one byte reported in [16] and the energy consumption for cryptographic operations reported in [22]. Table 5 shows the total energy cost in microjoules of the two compared protocols.

EAP-PSK/PANA		OnboardICNg	
PaC	PRE	d_j	d_{nbr}
10905	20695	5993	7082

Table 5: Total energy cost (in microjoules) in EAP-PSK/PANA and OnboardICNg

Table 5 shows that the entities in OnboardICNg consume less energy than the entities in EAP-PSK/PANA. Moreover, it is interesting to note how d_{nbr} consumes about 66% less than its counterpart in EAP-PSK/PANA. This is due to the communication cost that is the dominating component in the energy cost. Thus, even if d_{nbr} has a higher computational cost than PRE, it is still able to preserve a considerable amount of energy.

Memory cost. The memory cost can be estimated as the amount of information each device must store in RAM during the execution of EAP-PSK/PANA and OnboardICNg. In EAP-PSK/PANA, PaC and PAA must store: PANA cryptographic material and EAP-PSK cryptographic material. As specified in [19], PANA requires: `PANA_AUTH_KEY` (16 bytes), `I_PAR` and `I_PAN` (40 bytes each[8]), the two nonces `PaC_nonce` and `PAA_nonce` (8 bytes each). The work in [7] defines the requirement for EAP-PSK, that is: `PSK` (16 bytes), `AK` (16 bytes), `KDK` (16 bytes) and `MSK` (64 bytes) and the two random number `RAND_S` and `RAND_P` (16 bytes each). We consciously exclude `TEK` and `EMSK` keys because they are not needed in the EAP-PSK/PANA protocol. PRE

[7]Deriving MSK needs the execution of prf four times.

[8]`I_PAR` and `I_PAN` size is taken from [27].

does not need to store any additional information while running the protocol because it just acts as a relay. In OnboardICNg, the information that must be stored in d_j are: psk_{d_j-AAM}(128 bytes), $rn_{d_{nbr}}$ and rn_{d_j} (8 bytes each), k' (16 bytes), the authentication and encryption key derived from k' (16 bytes each), the content generated at Step 2 of Figure 4 (93 bytes), k_{j-nbr} (16 bytes), k_{j-AGW} (16 bytes), and the encryption key derived from psk_{d_j-AAM} (16 bytes). The device d_{nbr} must store: $rn_{d_{nbr}}$ and rn_{d_j}, k', the authentication and encryption key derived from k', k_{j-nbr}. In Table 6 we report the total amount of memory requires by each entity. The amount of RAM required by OnboardICNg is negligible and it can fits in almost all the current IoT platforms.

EAP-PSK/PANA		**OnboardICNg**	
PaC	***PRE***	d_j	d_{nbr}
224	0	332	159

Table 6: Total memory cost (in bytes) in EAP-PSK/PANA and OnboardICNg

7. RELATED WORK

OnboardICNg is not the first authentication protocol involving the use of three entities. A similar approach can be found in the IEEE 802.1x framework in which three entities, i.e., *supplicant*, *authenticator* and *authenticator server*, participate in the authentication phase. Similar to PANA, the IEEE 802.1x defines the encapsulation of existing authentication protocols (EAP protocols) exploiting an underlying network protocol to carry the authentication packets to the destination. However, while PANA is built upon UDP, the IEEE 802.1x defines the encapsulation over the IEEE 802 media. Even though the IEEE 802.1x is widely adopted in many 802 media, nowadays there is no specification for its adoption in IEEE 802.15.4 networks.

Research in wireless sensor network comprises several papers that investigate constrained devices authentication and key distribution approaches in [10, 18, 23, 26, 33]. Mainly two approaches can be distinguished among the many proposal: centrally based solutions in which a trusted server authenticates and distributes the crypto material to the joining device [26, 18, 23, 15]; distributed approaches in which trusted devices can authenticate new joining device without the involvement of a central authority [33]. An example of central based solution is give by by SPINS [26] in which two building blocks, SNEP (Secure Network Encryption Protocol) and uTesla, guarantees: data confidentiality, two-party data authentication, evidence of freshness and authenticated broadcast communication.

An example of distributed approach is given by LEAP+ [33]. LEAP+ proposes a set of efficient security mechanism for large-scale wireless sensors network. LEAP+ assumes sensors to be tamper resistant a short interval and the on-boarding phase of each node is accomplished during such tamper-resistant short interval.

One prior work [9] proposed a secure building automation system based on an ICN wireless sensors network. In the proposal [9], a sensor directly communicates through Internet with applications running on external devices (i.e., devices and sensors do not share a local network). Pairs of private/public keys (or shared symmetric keys) are used to provide an end-to-end secure communication between a sensor and an application. However, the work does not focus on the on-boarding phase of joining devices and does not distribute cryptographic material for link-layer security.

8. CONCLUSION

In this paper we presented OnboardICNg, the first protocol that securely authenticates and authorizes new devices to an IoT-over-ICN network. OnboardICNg is resilient to both insider and outsider adversary. In particular, OnboardICNg prevents malicious devices from joining a trusted network, as well as mislead honest devices to be part of a malicious network. Moreover, OnboardICNg securely bootstraps the necessary cryptographic material to later secure the communication in the trusted network.

We showed that OnboardICNg outperforms the current authentication and authorization protocol designed for IP-based IoT networks. Due to its design, OnboardICNg minimizes the number of packets forwarded in the network and exploits the ICN transport facilities to limit the overhead in the packet header. Compared to EAP-PSK/PANA, OnboardICNg reduces the communication cost up to 87% and the energy cost up to 66%, while it maintains a comparable amount of memory occupation.

We believe this work is important because it addresses the preliminary steps that are paramount for the security of an IoT-over-ICN network, thus contributing to improve one of the best candidates for the future Internet architecture.

9. ACKNOWLEDGMENT

Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). This work is also partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061), the EU-India REACH Project (agreement ICI+/2014/342-896), the Italian MIUR-PRIN TENACE Project (agreement 20103P34XC), and by the projects "Tackling Mobile Malware with Innovative Machine Learning Techniques", "Physical-Layer Security for Wireless Communication", and "Content Centric Networking: Security and Privacy Issues" funded by the University of Padua.

10. REFERENCES

[1] IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2011*, pages 1–314, Sept 2011.

[2] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik. Cache privacy in named-data networking. In *ICDCS*, pages 41–51. IEEE, 2013.

[3] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro. Named data networking for IoT: An architectural perspective. In *EuCNC*, pages 1–5. IEEE, 2014.

[4] M. Amadeo, C. Campolo, and A. Molinaro. Internet of Things via Named Data Networking: The support of push traffic. In *NOF*, pages 1–5. IEEE, 2014.

[5] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch. Information Centric Networking in the IoT: Experiments with NDN in the Wild. In *ICN*, pages 77–86. ACM, 2014.

[6] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *CRYPTO*, pages 232–249. Springer, 1994.

[7] F. Bersani and H. Tschofenig. The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method. RFC 4764.

[8] J. Burke, P. Gasti, N. Nathan, and G. Tsudik. Securing instrumented environments over content-centric networking: the case of lighting control and NDN. In *INFOCOM Workshops*, pages 394–398. IEEE, 2013.

[9] J. Burke, P. Gasti, N. Nathan, and G. Tsudik. Secure Sensing over Named Data Networking. In *NCA*, pages 175–180. IEEE, 2014.

[10] D. W. Carman, P. S. Kruus, and B. J. Matt. Constraints and approaches for distributed sensor network security (final). *DARPA Project report,(Cryptographic Technologies Group, Trusted Information System, NAI Labs)*, 1(1), 2000.

[11] L. Chen. Recommendation for key derivation using pseudorandom functions. *NIST special publication*, 800:108, 2008.

[12] A. Compagno, M. Conti, P. Gasti, L. V. Mancini, and G. Tsudik. Violating consumer anonymity: Geo-locating nodes in named data networking. In *ACNS*, pages 243–262. Springer, 2015.

[13] A. Compagno, M. Conti, P. Gasti, and G. Tsudik. Poseidon: Mitigating interest flooding DDoS attacks in named data networking. In *LCN*, pages 630–638. IEEE, 2013.

[14] Content centric networking (CCNx) project. http://www.ccnx.org.

[15] M. Conti, R. Di Pietro, and L. V. Mancini. Secure cooperative channel establishment in wireless sensor networks. In *PerCom Workshops*, pages 5–9. IEEE, 2006.

[16] G. De Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira. On the energy cost of communication and cryptography in wireless sensor networks. In *WiMob*, pages 580–585. IEEE, 2008.

[17] S. E. Deering. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460.

[18] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *CCS*, pages 41–47. ACM, 2002.

[19] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for carrying authentication for network access (PANA). RFC 5191.

[20] C. Karlof, N. Sastry, and D. Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *SenSys*, pages 162–175. ACM, 2004.

[21] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296.

[22] J. Lee, K. Kapitanova, and S. H. Son. The price of security in wireless sensor networks. *Computer Networks*, 54(17):2967–2978, 2010.

[23] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *TISSEC*, 8(1):41–77, 2005.

[24] CCN and NDN TLV encodings in 802.15.4 packets. https://www.ietf.org/mail-archive/web/icnrg/current/pdfs9ieLPWcJI.pdf.

[25] Named Data Networking project (NDN). http://named-data.org.

[26] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.

[27] P. M. Sanchez, R. M. Lopez, and A. F. G. Skarmeta. Panatiki: A network access control implementation based on PANA for IoT devices. *Sensors*, 13(11):14888–14917, 2013.

[28] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang. Securing building management systems using named data networking. *Network*, 28(3):50–56, 2014.

[29] Z. Shelby and C. Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011.

[30] D. Simon, B. Aboba, and R. Hurst. The EAP-TLS Authentication Protocol. RFC 5216.

[31] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[32] Y. Zhang, D. Raychadhuri, R. Ravindran, and G. Wang. Icn based architecture for iot. *IRTF contribution, October*, 2013.

[33] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *TOSN*, 2(4):500–528, 2006.

[34] ZigBee Alliance. Zigbee specification, 2006.

[35] ZigBee Alliance. ZigBee IP specification, 2010.