Security Access Protocols in IoT Capillary Networks

Romeo Giuliano, Franco Mazzenga, Alessandro Neri, Anna Maria Vegni

***Abstract*—Smart City services are enabled by a massive use of Internet of Things (IoT) technologies. The huge amount of sensors, and terminals with a great variety of typologies and applications, requires a secure way to manage them. Capillary networks can be seen as a short range extension of conventional access network in order to efficiently capture the IoT traffic, and are enablers for Smart City services. They can include both IP and non-IP devices, and security can become an issue, especially when simple unidirectional communication devices are considered.**

The main goal of this paper is to analyze security aspects in IoT capillary networks including unidirectional and bidirectional IP or non-IP devices. We propose an algorithm for secure access for uni- and bi-directional devices. The security procedure is based on a secure key renewal (without any exchange in air), considering a local clock time and a time interval of key validity. Following previous work in [2], in this paper we assess the duration of the validity of the time window, and present extended simulation results in terms of (average) transmission time in a realistic scenario *i.e.*, including the presence of disturber(s), then providing indications for the setting of the duration of the key validity time window. Finally, we present the benchmark analysis in order to assess the effectiveness of our approach with respect to other existing standards, as well as the security analysis in terms of typical attacks.

I. Introduction

Smart Environment (SE) services exploit the massive data collected by sensors, connected devices and mobile terminals, and also social applications, in order to provide advanced services to users [1]. In this scenario the environment will be disseminated by millions of simple and sometime tiny devices (*i.e.*, smart things), with autonomous sensing, processing, and communication capabilities. Devices as varied as soil moisture sensors, street lights, diesel generators, video surveillance systems, and many others will be connected with each others anytime, and anywhere, in a seamless way.

As a vision towards 2020, we expect to reach the threshold of 50 billion connected devices [4]. This number is so large that devices cannot be individually managed, and then techniques for self-addressing and self-classification will be mandatory. In a more general setting, SE services will be enabled by a massive use of the Internet of Things (IoT) technologies [5], [6], where the data traffic produced by IoT devices will be very different by traditional human-machine oriented Internet traffic. Many sensors are very simple, and due to their hardware/software limitations they are unable to directly support IP-based communications.

Typical radio access infrastructures are claimed for supporting IoT using existing Machine-to-Machine (M2M) services, and are commonly seen as the integration of one or more short/medium range radio technologies, such as WiFi, Bluetooth, Zigbee, 6LoWPAN etc. and cellular radio technologies such as High Speed Packet Access (HSPA), Long Term Evolution (LTE), and so on. However, this vision could be limiting since IoT/M2M access networks should be able to accommodate any kind of device. In addition, it is well known the Internet will run out of IPv4 addresses in next few years. As a consequence, new sensors requiring unique IP addresses will need IPv6 addressing. Moreover, sensors will need to be self-sustaining, in order to keep alive connections in IoT networks.

A lot of effort has been made in the area of standards, especially in the areas of access IoT network architectures, communications and security. All the aforementioned challenges rely on efficient data communications, being them within a fixed infrastructure or with mobile devices and users.

Capillary networks are seen as the fundamental enabling infrastructure(s) required for IoT, and more in general, for Internet of Everything (IoE), and then for the realistic development of the SE concept. The typical capillary network is realized through gateways densely deployed in the serving area, and it can be seen as a short-range extension of conventional wireless/wired access network(s). It allows to collect traffic from any sensor device. As a result, the capillary networks [3] represent a fundamental part within the IoT framework, allowing local wireless sensor networks to connect to and efficiently extend the own local connectivity through the use of gateways.

The concept of capillary networks implicitly assumes the coexistence of heterogeneous devices (*i.e.*, unidirectional, bidirectional, IP, and non-IP). Unidirectional devices cannot be coordinated/commanded in any way by the gateway/mediator, neither they can coordinate among them. Thus, they interfere among them, as well as, unintentionally, with bidirectional devices. This largely justifies the performance analysis of bidirectional devices in the presence of interference due to unidirectional terminals that need to coexist in the same capillary network, thus complicating the operating scenario.
On the other side, IP and non-IP devices distinguish among the hardware and software capabilities of capillary terminals. It is obvious that unidirectional terminals are non-IP devices, and have very reduced functionalities (maybe only sensing and simple transmission capabilities over non-standard signal formats).

Romeo Giuliano is with the Department of Innovation & Information Engineering, Guglielmo Marconi University, Via Plinio 44, 00193, Rome, Italy, Email: *r.giuliano@unimarconi.it*.

Franco Mazzenga is with the Department of Enterprise Engineering "Mario Lucertini", University of Rome Tor Vergata, Via del Politecnico 1, 00133, Rome, Italy, Email: *mazzenga@ing.uniroma2.it*

Alessandro Neri and Anna Maria Vegni (*corresponding author*) are with the Department of Engineering, COMLAB Laboratory, Roma TRE University, Via Vito Volterra 62, 00146, Rome, Italy, Email: *neri@uniroma3.it*, *annamaria.vegni@uniroma3.it*

The access security of heterogeneous devices in IoT capillary networks is analyzed in this paper. This topic has been addressed in part in [2]. In particular, in this paper we extend previous work [2], where we consider an architecture for the IoT non-IP terminal access, based on the usage of a *mediator* in the network. The mediator entity is not simply a gateway, but it represents *all* the non-IP terminals inside the network, in order to allow the applications running on the service application platform to address and communicate with them, as if they were IP devices. The mediator is at one hop with non-IP devices and communicates with them by using a secure connection.

Leveraging on the previous work [2], in this paper we aim to extend the discussion on security issues in IoT platforms, and we present further simulation results in realistic scenarios specific for IoT paradigm, even including intentional disturbers aiming of disrupting the correct operations of the proposed security algorithm.

The proposed secure access algorithm is based on the extension of the concepts in [10], [11], where a common time reference is used to generate/renew the session keys *i.e.*, no secure server for keys generation/exchange is required. The procedure is then extended to bi-directional communications between a non-IP device and its mediator. Our proposed approach belongs to the class of pre-distribution key management schemes [26], with specific features that distinguish from existing related works. Indeed, the main advantages of our technique is the use of locally generated secure keys, based on a time approach, that can be renewed if the time window elapses.

The considered time-based security technique correctly operates assuming a specific duration for the time window used to assess the validity of the secure key. The setting of this parameter can be related to the transmission time required by sensors to correctly deliver the message to the gateway/mediator. Since terminals operate in an interfered environment characterized by many terminals that access the channel in accordance with different protocols, many delays can occur *i.e.*, delays due to access the channel, and delays due to retransmissions caused by packet collision at the receiver. Such delays can strongly affect the transmission time. In particular, events where the transmission time can be larger than the key validity time window can seriously impair the considered security procedure.

In fact, the packet arrival outside the time window is interpreted as a typical reply attack, and then discarded even though the key is correct and the packet owns the timestamp indicating the emission time (inserted at security/application level) is in the correct time window. In addition, the presence of an intelligent disturber that can lead to a reduce availability of the idle channel, can cause a significant increase of the channel access time. Then, the time window shall be set accounting for the typical overall transmission time achievable in a realistic scenario including several and different types of terminals and even intentional disturbers (if any).

In order to analyze this important aspect, we adopt a simulation-based approach and some results are presented and discussed in this paper. It is observed that the overall transmission time increases above one second in the presence of the disturber even for low/medium network load.

The paper is organized as follows. In Section II we provide a background on security and privacy issues in IoT scenarios, and present the main technologies used to face with these issues. Section III illustrates the considered scenarios, as well as the list of the mediator's functionalities. In Section IV we discuss the proposed algorithm for providing secure connections for both uni-directional and bi-directional (non-IP)–to–mediator communications. In order to assess our proposed technique with respect to other existing standards, in Section V we present the benchmark analysis and compare our approach to ZigBee and 6LoWPAN. In Section VI, simulation results carried out to evaluate the performance of the secured network are reported. Then, in Section VII we provide the security analysis of our technique. Finally, conclusions are drawn at the end of the paper.

II. Security, and privacy issues in IoT

Providing security in IoT scenarios is a challenge, mainly due to a huge amount of heterogeneous devices globally accessible via insecure connections.

Security issues are extended from confidentiality, authenticity and integrity of end-to-end communications to network aspects, such as authenticity and integrity of devices, and networks. By fact, hackers may use IoT devices as attack platforms to perform distributed denial-of-service (DDoS) attacks. For instance, in 2014 Proofpoint researchers discovered an IoT cyber-attack, where home appliances like TVs, and a refrigerator, sent malicious email spam. *Thingbots* were created in order to compromise things. Finally, privacy issues emerge as more complicated to be fixed, since devices in IoT networks may be associated with persons.

Security issues in the framework of IoT have been largely investigated. A recent review on the security aspects of IoT is deeply investigated in [12]. Traditional security countermeasures cannot be applied to IoT framework, due to specific features of the different standards and communication stacks employed. Several works have addressed the topic of security issues in IoT [26]–[29]. In [28] Sicari *et al.* survey the main research challenges and the existing solutions in the field of IoT security, as well as in [29] Granjal *et al.* present a survey on existing protocols and mechanisms to secure communications in the IoT. Finally, in [26] Simplcio *et al.* present a review of different key management solutions applied to wireless sensor networks, compared in terms of security, efficiency, and flexibility. Finally, in [27] Chang *et al.* present a survey on main key management protocols applied to Body Sensor Networks (BSN), a dedicated area within the IoT framework.

In IoT scenarios, a number of technologies have been developed in order to achieve information privacy and security goals [13], such as the Transport Layer Security (TLS), which could also improve confidentiality and integrity of the IoT, and the Onion Routing, which encrypts and mixes Internet traffic from different sources, and encrypts data into multiple layers, by using public keys on the transmission path.

IoT platforms will become a reality due to two main pillars *i.e.*, (*i*) 6LoWPAN [14], and (*ii*) Constrained Application Protocol (CoAP) [15]. 6LoWPAN enables embedded nodes to use a restricted subset of IPv6 addresses, while CoAP –a software protocol targeted for small low power sensors– allows these devices to offer services to other machines, enabling resource-efficient implementations. More in details, the idea of 6LoWPAN is a combination of IPv6 and IEEE 802.15.4. The most important difference is the size of the IPv6 packet, so that the IETF 6LoWPAN working group proposed an adaptation layer that optimizes IPv6 packets through fragmentation, and assemblies to be supported by the IEEE 802.15.4 link layer. This new layer is located at the Edge Routers (called also Border Router) that controls flows incoming and outgoing from the LoWPAN that represents the collection of 6LoWPAN nodes sharing the same address prefix IPv6.

A 6LoWPAN network consists of one or more LoWPAN networks connected to the Internet through the Edge Router that controls flows incoming and outgoing from the LoWPAN. LoWPAN devices are characterized by their short radio range, low data rate, low power and low cost. In a LoWPAN, there are two types of devices *i.e.*, (*i*) the FFD (Full Function Devices), and (*ii*) RFD (Reduced Function Devices) connected to the Edge Router, responsible for communications with the Internet. Moreover, the LoWPAN supports two types of topologies *i.e.*, (*i*) star topology, where nodes communicate with one coordinator responsible for managing communications within the network, and (*ii*) mesh topology, where nodes can communicate with each other directly. Within LoWPAN, devices do not use the IPv6 address or UDP full header to communicate, but it remains at the Edge Router to communicate with the outside. Finally, routing issues in 6LoWPAN are addressed by IETF-ROLL (Routing over Low-power and Lossy Network) working group, in order to seek a proper routing solution to this kind of networks. IETF-ROLL proposed RPL (Routing Protocol for Low-power and Lossy-networks) [16], which has opened a new area of research and development.

Security issues in 6LoWPAN are analyzed by Rghioui *et al.* in [17]. 6LoWPAN networks can suffer from several attacks aimed to cause a direct damage to the network, or just to spy the network confidential information. These attacks can be classified into two types *i.e.*, (*i*) *internal* attacks provided by malicious nodes, and (*ii*) *external* attacks by unauthorized devices. Moreover, these attacks are *i.e.*, (*i*) *passive*, when the attacker has as main purpose to spy the network, and catch secret information, and (*ii*) *active* when interfering directly on the network performance and then causing its malfunction as Denial-of-Service (DoS) attacks. In [18] Kasinathan *et al.* present a DoS detection architecture for 6LoWPAN, where they integrate an Intrusion Detection System (IDS). Finally, threats are several, and each layer in the 6LoWPAN stack can undergo specific attacks, occurring at different layers [17]. Surveys on main protocol stacks for IoT are presented by Palattella *et al.* in [19], and by Tan and Koo in [21].

Traditional robust static security can result not sufficient, especially for wireless communications (lack of fixed infrastructure), meaning constant surveillance, and required privacy. Moreover, cooperative wireless protocols are more vulnerable, and dynamic network conditions do not allow distinguishing normalcy and anomaly.

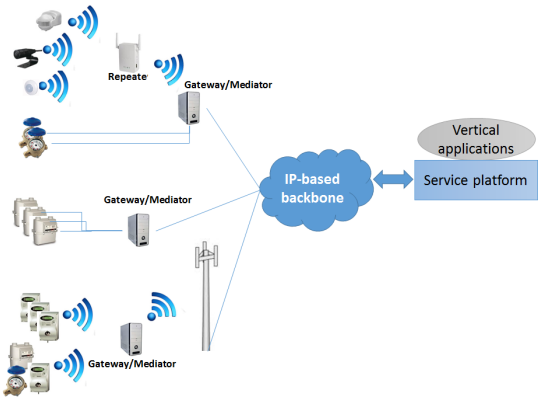

Fig. 1. Schematic of the considered access network architecture for IoT devices.

With the huge deployment of wireless technologies, and the rapid evolution of mobile devices and applications, and then fully distributed control loose security management, mobile devices are subject to security tradeoff. Today, we need a new approach to providing security, because even adaptive security is insufficient. This new approach is termed as *cognitive security* [22]. As well known, cognitive involves conscious intellectual activity as knowing and perceiving, and is based on the capable of being reduced to empirical factual knowledge. It follows that cognitive security is to add cognition by exploiting technologies such as machine learning, knowledge representation, network control and management, etc., while solving security problems. Cognitive security authenticates a user through the properties, patterns or knowledge peculiar to the user that have been continuously learned and updated. A more detailed description of cognitive security concept, and its applicability to IoT capillary networks is given in the following sections.

III. IoT SCENARIOS

We consider the access network architecture for IoT devices as depicted in Fig. 1. We have assumed that IP/non-IP devices can connect via short range radio links, as well as via wired links to the network by the gateways, as indicated in Fig. 1. Gateways are densely deployed over the area and realize a *capillary* access network, which can be seen as a short range extension of conventional access network in order to efficiently capture the IoT traffic. Gateways are equipped with additional functions to act as mediators for anyone of the non-IP devices connected to them. Moreover, gateways/mediators can connect to the rest of the network by using several alternative wireless (*i.e.*, cellular, WiFi, etc.) and wired (*i.e.*, xDSL) technologies. The link selection is based on the available technologies and on energy-optimization strategies [23]. Notice that the mediator platform described in this article is very similar to the gateways defined for IoT platforms, such as Moebious [9].

Data collected by gateways/mediators are then sent through the IP-backbone to the service platform running vertical ap-

plications. Devices supporting bidirectional communications can receive commands issued by those remote applications. In principle, the service platform can communicate with both IP and non-IP devices in the area by using IPv6 addressing. Non-IP devices communicate with gateways in accordance with standardized protocols that, in general, could not adopt IP stack for communication with the gateway. In this case, we assume non-IP devices are represented over the network by a mediator.

The considered protocol stack architecture of the IP/non-IP IoT to service platform link including the mediator is depicted in Fig. 3. The mediator interfaces the non-IP devices by implementing (instances) of their specific communication protocol stacks, including PHY and DLC/MAC sub-layers, at least. The Mediator Application (MA) is at the top of the stack, and includes the security sub-layer, as detailed in the following Section IV. The MA interfaces with Internet by means of the TCP/IPv6 stack, and adopts L2/L1 technologies for transferring/receiving data packets over the IP backbone to/from the service platform. The main function of MA is to represent each non-IP device as an IP addressable entity over the network.

Since the number of non-IP devices in the area could be very high, it could be convenient the MA to act like a sort of Network Address Translation (NAT). In this case only the gateway/mediator is identified by a static/dynamic IP address assigned by the network, and non-IP devices are connected to the local network managed by the gateway/mediator. As an alternative, the gateway/mediator can internally assign a (static) IP address to a non-IP device. This IP address is used by the gateway to "represent" the device over Internet. It can be extracted by a list of IP addresses that have been assigned by the service platform (or any other entity) to the gateway. On the other side, for non-IP unidirectional devices data received by the gateway/mediator are directly re-transmitted to the service application platform by using the IP address of the gateway as source address, and specifying the origin of packet payload, the destination IP address, and the application(s) to which data are directed to. When necessary, the service platform can command the gateway/mediator to change the destination of unidirectional packet data.

As previously outlined, the MA manages the access of IP/non-IP IoT devices to the gateway. It is out of the scope of this paper to start an in-depth discussion on the (best) multiple access protocols to be selected for unidirectional and bidirectional terminals. In general, we can assume that access protocols can be different for different kinds of devices. In particular, as indicated in the performance assessment, we assume (*i*) ALOHA, and (*ii*) non-persistent Carrier Sense Multiple Access (CSMA) for unidirectional and bidirectional devices, respectively. Furthermore, for simplicity it is assumed that IoT devices cannot directly communicate among them, so that no-cooperative strategies are considered.

Data originated by IoT devices are first received by the gateway/mediator and then sent to the service platform over secure connections based on typical protocols, such as IPsec, HTTPs, WS-Security, etc. For bidirectional devices, the application running on the service platform can address the IoT devices served by the gateway/mediator layer to issue commands and then wait for (possible) answers.

IV. Protocol for Security Network Access

Our approach belongs to the class of pre-distribution key management schemes, with specific features that distinguish from existing related works [26]. The algorithm supposes to pre-distribute a long set of keys in each node. The transmitter (*e.g.* the sensor node) selects one secure key over this set based on the current time, and sends the message encrypted by this key. Thanks to the attached timestamp to the received message, the receiver (*e.g.* the mediator) uses the corresponding key to decrypt the message. The mediator and the generic node should just agree on the sequence of hopping/changing between one secure key and the next one after an agreed time interval. The sequence can be obtained by any pseudo-random number generator. The security parameters that need to be agreed between the node and the mediator, in charge the communication for that node, are the following:

1) The same random number generator;
2) The seed from which the random number generator creates the sequence [1];
3) The time interval over which the key is changed;
4) The overall set of pre-distributed and stored keys in the sensor node.

During the bootstrap, the node and the mediator need to agree on these security parameters. This procedure can be different for uni-directional and bi-directional nodes due to different communication capabilities. For the uni-directional nodes, the procedure needs an initial phase to register the node in the network. The (human) installer assistant reads the code on the novel node label and communicates to the security server by using the mediator or any other technology (*e.g.* the available cellular system) all the security parameters assigned to that node.

When the mediator receives an encrypted message by the novel node (*i.e.* not stored yet in its local database), it has to search out the security parameters. Then, it interrogates the security server, which provides it with the requested parameters. At the end of the bootstrap procedure, the mediator has all the security parameters to correctly decrypt the message received by the novel node. In fact, it is able to generate the same key sequence as for the novel node, since both have the same number generator, the same seed and the same security key pool, and it has received also the timestamp attached to the message. This procedure is briefly described in Fig. 2.

Bi-directional devices can use the same bootstrap procedure used by uni-directional devices, or eventually they can exchange data with the mediator due to duplex channel. In the second case, the novel bi-directional node starts the bootstrap procedure by sending a REQUEST_PARAMETERS message and its identity. When the mediator receives this message, it recovers the enclosed information by the security server or generates it locally and sends to the security server for further accesses. Then, the mediator provides the parameters

[1] Of course, the same random number generator will generate the same sequence if the seed is the same

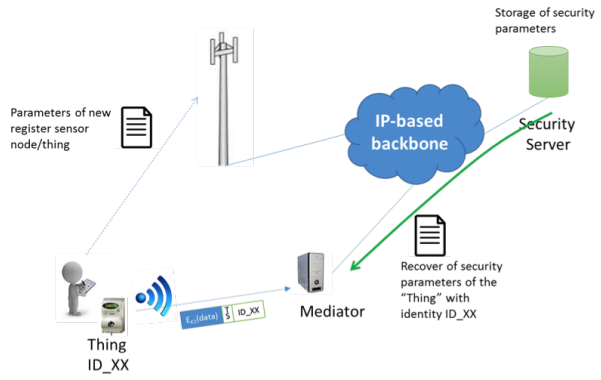

Fig. 2. Bootstrap procedure for uni-directional devices.

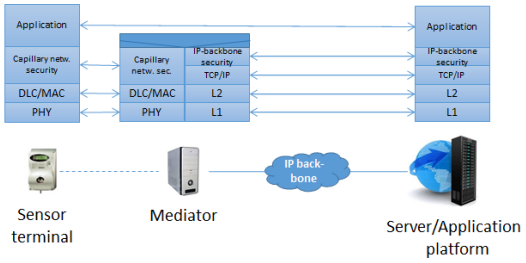

Fig. 3. General end-to-end protocol stack architecture of the link connecting the generic IP/non-IP IoT device to the service platform; the gateway/mediator is included.

through the RESPONSE_PARAMETERS message. After receiving that, the novel node is able to produce the hopping sequence and encrypts the messages based on its time and the pre-stored secure keys.

Thanks to the duplex channel, the bi-directional node and its mediator can agree on changing periodically some security parameters (*e.g.* the seed and the periodicity of the key pool). This information can be exchanged in plain without affecting the system security. In fact, the attacker is not able to know the specific secure key pool pre-loaded into each device. Moreover, through the sniffing of the seed, the attacker is just able to generate the same pseudo-random sequence without knowing anything on the pre-loaded secure keys on devices, neither the length of the hopping sequence, nor the amount of the key pool. Also, by hacking one device, the key pool is different and the hopping sequence generation is specifically set for the single device.

In our selected scenario, in order to guarantee secure connections of non-IP terminals with the gateway/mediator, several assumptions need to be taken. First, we assume that security layer is available below the application level. The considered protocol stack of a non-IP device should be as depicted in Fig. 3. We also consider that the identities of the IP and non-IP devices have been trusted with the server application during the installation of the device in the area, although this can be of scarce relevance for bi-directional devices, since an authentication procedure based on common and well established protocols can be set up. Notice that existing protocols for non-IP devices, like WirelessHART and ISA 100.11a [20], can be easily integrated with the capillary network, so that the security is managed on the application layer by the proposed protocol.

On the contrary, the terminal trustiness is very important for uni-directional devices that are assumed to be unable to receive any message, and then to perform any authentication procedure. The problem of authentication of unidirectional terminals could be solved in many ways. One of them could assume that during first setup the installer uses its private key to communicate the gateway the identity of the devices. The gateway/mediator uses the installer public key to decipher the message and, given the identity of the device it securely accesses the server of the device manufacturer to download any (reserved) information on the device that will be used for subsequent (secure) communications as, for instance, the device public key.

Finally, we assume that the security access protocol is provided on the mediator-to-sensors link, as shown in Fig. 3. This avoids a (centralized) service platform to manage security for a huge number of devices *i.e.*, we implicitly have considered a hierarchical security architecture where the secure channel from gateway/mediator to the service platform is achieved by standard security protocols (*e.g.*, IPsec or HTTPs).

In the following Subsection IV-A, we first recall the aspects concerning the key generation and renewal based on time. Then, in Subsections IV-B, and IV-C, we propose an approach to secure the IoT terminals' access, and distinguish the case for uni- and bi-directional terminals, respectively.

A. *Time-based Secure Key Generation and Renewal*

The time-based secure key generation approach has the aim to efficiently manage, and renew, the keys to provide the secure connection, while guaranteeing the integrity of data transmitted over an insecure channel. As a main characteristic, the local key synchronization and generation occur by means of the generation of symmetric encryption keys at both sides of the communication channel (*i.e.*, at the transmitter and receiver sides). The transmitter (receiver) will encrypt (decrypt) data by means of an encryption (decryption) key extracted from a shared sequence of keys. Moreover, in order to enhance the security level of the data transmission, the selected key will be changed during transmission.

The key change can be planned on time or event basis, and obviously, must be synchronized between the two communication parties. The principle of time-based secure key generation is schematically depicted in Fig. 4. In this approach, the key generation process is an operation performed independently by each communication party. Differently by any other key management algorithms, no additional message is required to be exchanged in order to agree about a key, and the only requirement is that the key generation function should create the same keys for both two communications parties. The validity of the secure keys is restricted to a time interval, and then reply attacks based on valid messages sent using keys generated in past time intervals are discarded.

Following such features, we derive that, as a main advantage of the time-based secure key generation approach, there is no

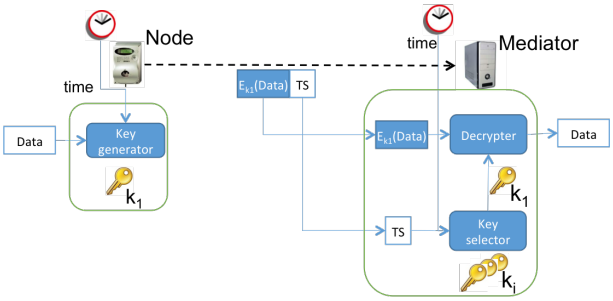

Fig. 4. Principle of time-based secure keys generation.

need of a server that manages secure keys. Moreover, the keys are generated locally on both sides of the communication link (*i.e.*, at the transmitter and at the receiver), and then are not shared along the connectivity link. Notice that in Fig. 4 the clocks are synchronized through the Network Time Protocol (NTP). This is a viable assumption since in our algorithm high accuracy synchronization is not necessary since decryption is performed by using more than one key (*i.e.*, all those indicated in the key validity time window).

Another protocol that uses a time-based key renewal is μ-TESLA [8], for secure authentication in wireless sensor networks. Differently from our approach, it applies symmetric cryptosystem to broadcasting authentication, through the disclosure of keys that are broadcasted once per epoch. Thus, μ-TESLA cannot be applied to our scenario due to the presence of unidirectional terminals.

B. Security Access Algorithms for Uni-directional Data Transmissions

Due to their simplicity, unidirectional devices cannot perform any secure procedure for secure keys exchange with the mediator. Indeed, the transmitter just sends a message without any feedback *i.e.* it fails to receive any signal, and is equipped with an internal clock, assumed as not accurate. A typical example of applications of unidirection devices is the smart metering [32]. A smart meter is a new kind of gas/electricity/water meter that can transmit meter readings to the energy supplier. In this way, a more accurate energy bill is guarantee.

A generic non-IP unidirectional terminal executes the following steps in order to send a data to the gateway/mediator in a secure way:

1. The terminal generates locally the encryption key, based on the time measured by a local clock;
2. It creates the message and encrypts it with the generated key; the message includes the payload and (possibly) any other data to be used to enhance security.
3. It computes the hash values using the message text and the generated key and attaches it to the message;
4. It sends the message to the gateway/mediator.

The message includes fields that can be grouped in (*i*) plain part, and (*ii*) encrypted part, as reported in Fig. 5:

1. *Plain part*: it comprises the timestamp (obtained by the local clock), the plain part identity (allowing the gateway/mediator to locally identify it for security procedure such as the key generation), the hash (for assessing message integrity; if the hash is calculated using the message texts (see Fig. 5) and the generated encryption key, then the hash can also be used to verify the identity of the transmitter), a security level parameter, which is present when several security degrees are allowed at the application level for different types of messages (*e.g.* simple state data and setting sensor data can be secured differently);
2. *Encrypted part*: it comprises the encrypted part identity (optional field), which could be used to enhance authentication, the frame counter that is increased by one at each sent frame, and the payload (*i.e.* used to convey information to the related application running on the remote server).

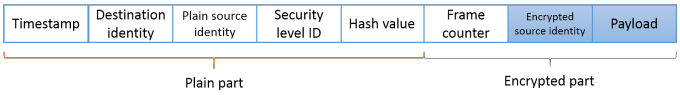

Fig. 5. Format of the message sent by a non-IP terminal to the mediator.

When the mediator receives the ciphered message, it can decipher it by generating the correct decryption key starting from the attached timestamp. The key generation is tied to the clocks, such that the gateway checks the timestamp of the received packet and calculates the key according to that particular timestamp. In fact, based on the information provided by the timestamp, the mediator can calculate/select the key to decrypt the given message; if the temporal difference between the current time and the timestamp exceeds a predefined threshold, the message is discarded.

Consecutive values of the timestamps could also be used by the gateway/mediator to estimate the behavior of the clocks of the unidirectional devices in terms of phase and drift. This could allow the gateway/mediator to follow the evolution of the device's clock and then to easily adapt the temporal window in which the timestamp is considered to be valid. We note that verifying that the received timestamp time series is monotonically increasing allows to avoid replay attacks.

The gateway/mediator can organize message reception with all connected terminals in a receiving table. Each table entry is indexed by the *tri*-ple field *i.e.*, `<plain identity, timestamp, security level parameter>`. The Security Level Parameter can indicate the security algorithm to be used for decryption (*e.g.* AES for confidentiality or SHA for integrity) obviously pre-defined in the installation phase. The other fields of each entry contain the key to decrypt the received message. The entry is deleted from the table when the validity related to the timestamp expires. The organization allows having different parallel communications with a simple IoT terminal related for example to a periodic sensor detection, a setting parameter or a critical detected datum. In TABLE I, we report an example of the receiving table.

TABLE I. EXAMPLE OF THE RECEIVING TABLE.

Plain ID	Timestamp	SLP	Key
ID 51	dd mm yy, 17:35:45	1	$a_0 a_1 \dots a_{n-1}$
ID 27	dd mm yy, 18:10:11	1	$b_0 b_1 \dots b_{n-1}$
ID 74	dd mm yy, 17:44:57	2	$c_0 c_1 \dots c_{n-1}$
...	...	...	...

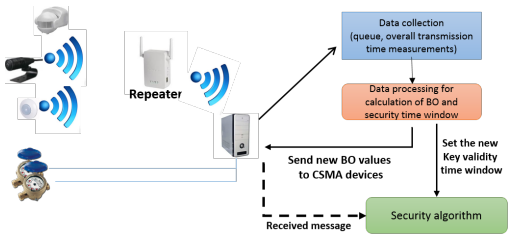

Fig. 6. Principle scheme of cognitive security work in the capillary network.

C. Security Access Algorithms for Bi-directional Data Transmissions

For bidirectional terminals (*i.e.*, each device can send and receive packets), the mediator can periodically broadcast its clock timing in a dedicated message [11], and its identity in the plain part of the message. Terminals can align their local clocks to the gateway/mediator terminal, and then generate the security keys in accordance with the algorithm described in the previous Subsection IV-B.

Since devices are close to the gateway/mediator, propagation delays can be neglected. Furthermore, as for the unidirectional case, the security keys have a validity time interval sufficiently longer to transmit one or more packets, and to absorb possible retransmissions or any other unwanted delay. In this case, even the key renewal can be performed with a time-based generation algorithm.

Note that each terminal can be served (*i.e.*, being in the coverage area) by more than one mediator gateway. Thus, the mediator gateway identity is fundamental for bi-directional transmissions, in order to distinguish several mediator gateways, which could also have clocks running at (slightly) different time. Thus, the terminal should insert the mediator gateway identity in the sent message, otherwise the message cannot be correctly decrypted due to possible gateway desynchronization causing the encryption with a wrong key.

Notice that the analysis of possible solutions based on public-key cryptography is out of the scope of this paper. As an example, for gateway-to-IoT devices transmissions, the gateway/mediator could broadcast the public-key in the area. IoT devices use this key to encrypt their identity and data, and also to communicate with the gateway/mediator. In this case the only problem to be solved to guarantee secure reception is to preserve the integrity of the transmitted packet. This can be solved by adding a hash to the packet transmitted by the IoT device.

D. Cognitive Security

The concept of *cognitive security* [22] arises from the need of applying a more sophisticated security level w.r.t well-known adaptive security approaches. The basic idea behind cognitive security is that the user authentication occurs through the properties, patterns or knowledge peculiar to the user that have been continuously learned and updated.

In this paper, we exploit this concept and apply it to the capillary networks, as shown in Fig. 6. The cognitive engine collects all the received data from the terminals in the capillary network at the mediator. Possible parameters to be collected are the transmission-reception time difference of frames for each terminal, the transmission frequencies, the packet lengths, the queue lengths, and so on. In the case of unidirectional terminals, the timestamp difference related to received frames provides information about the emission rate of the source, which should be compared with their target emission rate. For bidirectional terminals, their timestamp difference measured at the mediator should be compared with the set value.

Based on these parameters and on the comparison with historical data, a cognitive security based algorithm should be able to adapt security thresholds to counteract possible intruders/disturbers or terminals that are not correctly working. As an instance, the cognitive security engine can modify the backoff time (*i.e.*, BO) of the same terminals in order to increase their possibility to access shared channel and transmitted frames. When a traffic anomaly at a certain terminal is detected, the mediator analyzes the identity *i.e.*, ID parameter, of this terminal, which is considered as a potential disturber. If the disturber is declared non-trustworthy (*i.e.* secure), the mediator modifies the transmission parameters of terminals of the capillary network, in order to increase the bidirectional sent frames, and also notifies the ID disturber about the management entity of the capillary network. On the contrary, if the anomalous terminal is trustworthy, the mediator notifies the terminal ID to management entity that the terminal has been compromised.

Thus, possible countermeasures are:

- The mediator modifies access parameters to a set of terminals, based on the information of the application level. Possible access parameters that can be modified are: (*i*) the generation rate of the frames; (*ii*) the reduction of the back off time to repeat a new access to the channel; (*iii*) the reduction the measured time to detect the presence of the transmission of another terminal (*e.g.* acting on the Short InterFrame Space);
- The terminal can perform packet aggregation in order to improve its performance;
- The mediator modifies the time validity of the security keys to avoid replays attacks or clock desynchronization.

Thanks to a collection of data in the capillary network, the mediator is able to toughen the security in the network by properly modifying some parameters related to (*i*) the channel access, (*ii*) the security techniques or (*iii*) the transmission characteristics of the traffic. In one term, by applying the cognitive security paradigm.

V. BENCHMARK ANALYSIS

To evaluate the incidence of the header overhead, we compare the performance of the proposed security algorithm with

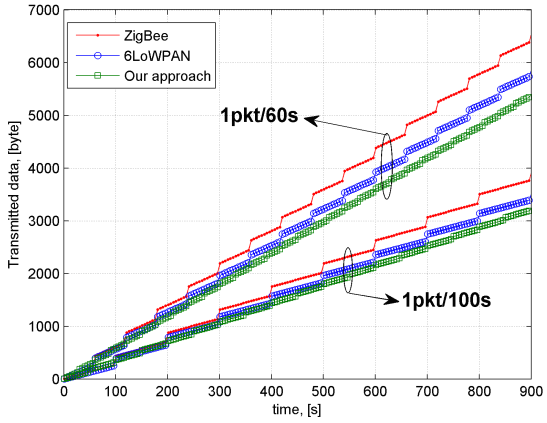

Fig. 7. Transmitted bytes by capillary nodes for ZigBee, 6LoWPAN and our technique when one packet is transmitted every 60 s and 100 s.

the two standards typically used for capillary networks *i.e.*, ZigBee [24] and 6LoWPAN [25]. In particular we compare the amount of transmitted bytes, assuming the same security level. The payload of a single packet is 100 bytes, and the overheads introduced by ZigBee, 6LoWPAN, and our technique's security are 35, 34, and 39 byte/packet, respectively. Nevertheless, ZigBee and 6LoWPAN have to transmit further bytes to perform the key exchange required to guarantee the same security level.

We consider 168 byte for ZigBee and 128 byte for 6LoWPAN, as detailed in [11]. However, for supporting and improving the synchronization of the capillary nodes implementing our security algorithm, we assume the gateway has to periodically transmit a 41 bytes packet containing the time information. In Fig. 7 the transmitted bytes are reported considering a packet rate of (*i*) 1/60 pkt/s, and (*ii*) 1/100 pkt/s, in the case the security is applied according to ZigBee, 6LoWPAN or our approach.

Note that to guarantee the same security level all security algorithms must exchange the secret key with the same frequency and for each packet. Finally, we assume that the time info packet is sent twice with respect to the data packet, in order to obtain a strict time synchronization among nodes. From Fig. 7, we notice that the data transmitted by nodes implementing the proposed approach are reduced respect to the data transmitted by ZigBee and 6LoWPAN nodes. The extra fields in the header do not cause any degradation in capillary network. The difference of transmitted data (*i.e.*, overhead gap) between our approach and the other two standards significantly increases with the packet transmission rates.

VI. Performance Evaluation

In this Section, we evaluate the overall transmission time in a mixed traffic scenario including uni-directional terminals (adopting ALOHA protocol for accessing the wireless channel) and bi-directional terminals (adopting CSMA protocol), which coexist over the same area. An intentional disturber with variable traffic characteristics has been included in the scenario. A Monte Carlo simulation-based approach has been considered and details are provided in the following.

A. Assumption and simulation description

We assume that uni-directional and bi-directional terminals transmit on the same band, and are connected to the same gateway/mediator.

Performance of the considered security access protocols are closely related to the (*i*) collision probability among uni-directional and bi-directional terminals, and (*ii*) the latency required for a CSMA terminal to correctly deliver the packet. Both parameters depend on the overall number N of uni- and bi-directional devices attached to the gateway/mediator. A non-persistent CSMA scheme is considered, thus allowing to save batteries, but other access schemes can be adopted as well.

For the case of ALOHA terminals, one terminal just broadcast encrypted data messages in accordance with the packet format depicted in Fig. 5.

In the CSMA case, we assume the terminal first listens to the channel, and if the channel is busy (*i.e.*, due to the transmission of another uni-directional or other active bi-directional terminals), it is not enabled to transmit. Then, a BO interval is randomly generated in accordance with an uniform distribution, and packet transmission is re-scheduled after the backoff time. When the BO elapses, the terminal listens to the channel and, if free, transmits, otherwise it backoffs again and so on, until the packet will be transmitted. After the correct packet reception, another packet is scheduled for transmission at a new randomly generated time instant.

We assume that CSMA transmissions can only be interfered at the receiver side by ALOHA packets *i.e.*, we assume that no hidden or exposed CSMA nodes are present in the area. In this case, only ALOHA traffic acts like a background noise, thus disturbing the normal operations of the considered non-persistent CSMA access protocol. However, the ACK message sent by the receiver is a short message, and we assume it is not interfered by ALOHA or other CSMA transmissions. If no ACK has been received, we consider that the transmitted packet has been interfered at the receiver and then lost. In this case, the CSMA protocol re-schedules the packet transmission by generating a new BO value.

In both two cases, we do not assume any capture effect nor at the receiver terminal, neither at the gateway/mediator receiver *i.e.*, two simultaneous messages (even partially superimposed) always generate a collision causing the loss of both messages. Moreover, we consider the inter-arrival times between packet transmission events in ALOHA and CSMA are exponentially distributed, and the backoff time interval is generated in accordance with an uniform distribution with average μ_B [s]. Finally, we assume the traffic originated by gateway/mediator is $(N_{CSMA} \times \lambda_b \times p)$, where N_{CSMA} is the number of bidirectional devices served by the gateway/mediator, λ_b [pkt/s] is the packet generation frequency, and $0 \leq p \leq 1$ is a sort of activity factor of the gateway/mediator, which reduces the overall gateway CSMA transmission activity. Notice that we assume that a percentage (*e.g.*, $1-p$) of the messages generated by the CSMA devices are directed to the server and do not need reply.

TABLE II. PARAMETERS USED FOR SIMULATIONS, FOR UNI- AND BI-DIRECTIONAL DEVICES.

Description	Value
Msg Length - Unidir.	784 bit
Msg Length - Bidir.	736 bit
Channel Coding Rate (R_c)	1/2
TX Bit Rate - Unidir.	2.4 kbit/s
TX Bit Rate - Bidir.	4.8 kbit/s
Pkt gen. freq. (λ_u) - Unidir.	0.0017 pkt/s
Pkt gen. freq. (λ_b) - Bidir.	0.0033 pkt/s
Avg. BO	5 s
Gateway activity (P)	25%

TABLE III. PARAMETERS OF THE MESSAGE FORMAT USED IN THE SIMULATIONS, FOR UNI- AND BI-DIRECTIONAL DEVICES.

Description	Unidirectional [bit]	Bidirectional [bit]
Timestamp	32	32
Destination Identity	64	64
Plain identity	32	16
Security level + frame counter	16	16
Hash value	64	32
Encrypted identity	64	64
Payload	512	512

To summarize, the overall simulation parameters are listed in TABLE II, where the message lengths are computed in accordance with the values in TABLE III.

To characterize the intelligent disturber we have modeled it as an extra terminal emitting packets similarly to a uni-directional device. The packet length T_D [bit] and the corresponding emission frequency λ_D [pkt/s] have been related to the uni-directional terminal parameters (*i.e.*, the index u) as follows:

$$\lambda_D \cdot T_D = J\lambda_u \cdot mT_u = K \cdot \lambda_u T_u, \quad (1)$$

where $J = \lambda_D/\lambda_u$ is the multiplicative factor of the disturber emission frequency respect to uni-directional emission frequency, $m = T_D/T_u$ is ratio between the disturber packet length and the uni-directional packet length, while $K = J \cdot m$. During simulations, we set J and K, and then we determined m.

B. Simulation Results

Simulation results are presented in terms of mean transmission time and collision probability for those terminals served by one gateway/mediator.

In Fig. 8 the collision probability for ALOHA terminals P_{coll}^{ALOHA} is reported as a function of the total number of active terminals N in the area. We report the case of 25% of ALOHA terminals, and 75% of CSMA terminals. Fig. 8 depict the case without the disturber (*i.e.*, no Jammer), which is considered to be the reference case. In the same picture we plot performance when the jammer interferes ALOHA terminals. As expected, performance degrades when jammer increases its emission frequency λ_D from 50 to 200, and its

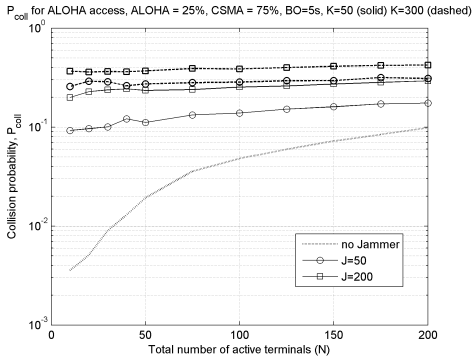

Fig. 8. Collision probability of ALOHA terminals vs. total number of terminals.

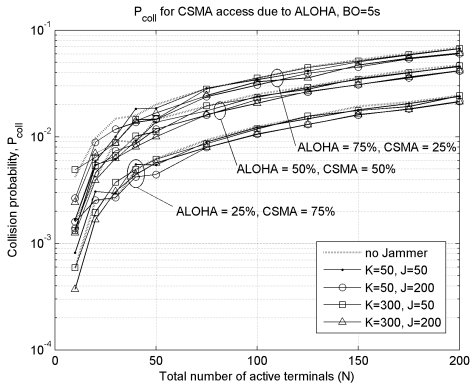

Fig. 9. Collision probability of CSMA terminals vs. total number of terminals.

packet duration T_D for which we varied K from 50 to 300. Fig. 9 depicts the collision probability of CSMA terminals P_{coll}^{CSMA} as a function of the total number of terminals N in the area served by the gateway/mediator. We have considered three groups of cases reporting the percentage of CSMA vs. ALOHA terminals, respect to the total number of terminals served by the gateway/mediator *i.e.*, (*i*) ALOHA $= 25\%$ and CSMA $= 75\%$, (*ii*) ALOHA $= 50\%$ and CSMA $= 50\%$, and (*iii*) ALOHA $= 75\%$ and CSMA $= 25\%$.

The P_{coll}^{CSMA} increases with the number of active terminals in the area, so leading to a reduction of the overall throughput. Since no hidden or exposed CSMA terminals exist in the area, CSMA can collide at the receiver with ALOHA transmissions only. On the other hand, ALOHA is affected only by other ALOHA transmissions, due to the carrier sense procedure of CSMA terminals.

Both the CSMA, and the ALOHA, collision probabilities increase with the number of ALOHA terminals, and this leads to a significant increase in the transmission time required to correctly deliver the message. In Fig. 9 we report both the case with and without the presence of a jammer, assuming several traffic configurations for the jammer, in terms of transmission frequency (*i.e.*, $J = 50$, and $J = 200$), and packet duration (*i.e.*, $K = 50$, and $K = 300$). It can be observed that thanks to the adoption of CSMA non-persistent

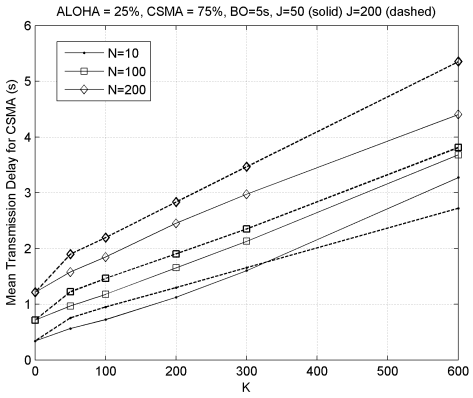

Fig. 10. Mean transmission time vs. K.

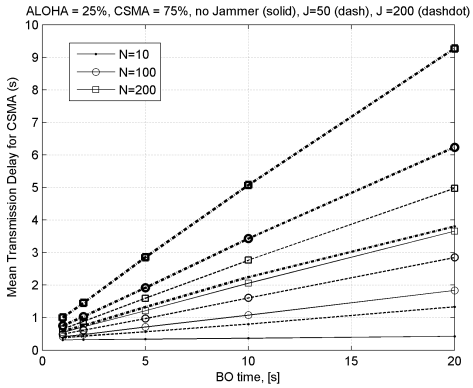

Fig. 11. Mean transmission time vs. BO.

protocol, in all cases the P_{coll}^{CSMA} is not dependent by the jammer characteristics. However this desirable behavior is paid in terms of the achievable transmission time as shown in the following results.

In Fig. 10 the mean transmission delay for CSMA terminals is reported as a function of the K parameter for 25% of ALOHA terminals respect to a total of $N = [10, 100, 200]$, and a BO set to 5 s. On one hand, the mean delay increases with the number of total numbers in the area, since more collisions are probable to occur. On the other hand, fixing the packet generation *i.e.* J, the mean delay increases with the packet duration *i.e.* K. Of course, increasing the BO parameter causes an increase in the mean delay.

Finally, in Fig. 11 the mean transmission delay for CSMA terminals is reported as a function of BO for 25% of ALOHA terminals respect to a total of $N = [10, 100, 200]$. The case of no jammer is indicated with solid line. The other cases refer to $J = 50$ (dashed lines), and $J = 200$ (dotted lines).

From information about monitoring of traffic load parameters (*e.g.*, the status of the transmission queues of the authorized transmitter, the BO time, the average frame length, etc.), which can be collected by capillary nodes, cognitive security algorithms can be applied. In this paper, it is possible to infer the proper setting for the BO time parameter based on the assumed (*a priori*) type of disturber, and on the number of terminals served by the gateway/mediator. For example, if a mean transmission delay of 2 s is required, and $N = 100$, we should select BO equal to 5.3 s when the disturber is $J = 200$, and BO equal to 13.2 s for a disturber with $J = 50$. In general, the reduction of BO duration allows CSMA terminals to better cope with the presence of disturbers. This could be achieved in practice by allowing the gateway/mediator to assess for the presence of the disturber and then to broadcast the optimal BO value to the CSMA terminals in the area to counteract the disturber action. The same information can be used by the gateway/mediator to properly set the duration of the key validity time windows cognitive security algorithm.

Curves shown in Fig. 8 and 9 can be used in the design stage to assess the maximum number of terminals that can be connected to a gateway/mediator for a given level of collision probability, and/or a maximum tolerable packet delay for CSMA. Since secure access procedures are implemented inside the gateway, this number provides an indication on the physical processing resources to be added in the gateway to support security functionalities. Usually, processing associated to security algorithms (*i.e.*, including encryption/decryption, authentication, etc.) could be very intensive, and this is an important aspect to be accounted for the design of the gateway/mediator terminals. As an instance, if a maximum CSMA/ALOHA collision probability level of about $4 \cdot 10^{-2}$ is required, the overall number of terminals connected to the gateway/mediator shall be restricted to $N = 100$. In this case, from Fig. 8 it can be observed that to preserve the CSMA/ALOHA collision probability requirement, the maximum number of ALOHA terminals is 75. Furthermore, from Fig. 9 the average delay is 0.4 s, while the standard deviation is slightly higher than 1 s.

Similar considerations apply if CSMA requirement is specified in terms of the overall transmission time of a CSMA packet. In general, design requirements should always focus on the desired CSMA performance since ALOHA traffic acts like a background noise, and CSMA has to coexist with it.

Finally, a consideration about synchronization issue in such scenario needs to be taken. In principle, for CSMA terminals it is not necessary to insert the timestamp in each packet. This allows to reduce the overhead and the packet transmission time, and also permits to greatly simplify the transmitter sub-system since it would be not necessary to add clock information to the packets at some protocol layer. In fact, for CSMA, it is not difficult to setup a synchronization procedure between the gateway and the terminal; the simplest way is to assume that the gateway sends its time information to terminals in the area so that they can align their clocks. In this case, depending on the number of active terminals in the area, the time required for successful packet reception could be not negligible (see Fig. 9) and, as shown from simulation, this depends on the entity of ALOHA background traffic.

Leveraging on such results, it follows that the information on the average (and even on the standard deviation) time, required for a correct reception, should be accounted for, in order to determine the extension of the secure key validity time interval, for the selected performance target.

TABLE IV. SECURITY ANALYSIS AGAINST THE MOST POPULAR ATTACKS.

Attack		Typical countermeasures	Proposed algorithm's countermeasures	Attack effectiveness
Release of message content		Payload encryption	Payload encryption	Ineffective
Traffic analysis		Padding-based countermeasures Distribution-based countermeasures	Medium access randomness and unknown identity	Partially effective. Not critical for the capillary applications
Masquerade		Authentication mechanisms	Renewal key generation pattern specifically assigned to each user/node	Ineffective
Replay attack		Timestamp, OTP, SSL protocol	Timestamp and frame counter field in the frame header	Ineffective
Intentional DoS				
LEVEL	above MAC	IDS	(out of the scope)	–
LEVEL	MAC	WIPS	Transmission parameter changes based on cognitive security	Ineffective or limited
LEVEL	Physical	Spread spectrum techniques	None	Limited
Modification of messages		Hashing	Hash field in the frame header	Ineffective
Man-in-the-middle		HMAC, SSL protocol, mutual authentication	Mutual authentication and a hash field in the frame header	Partially effective or ineffective

VII. SECURITY ANALYSIS

In TABLE IV, we report the security analysis for the proposed algorithm, with respect to specific adopted countermeasures. We considered the following most popular attacks, and related countermeasures available in the literature [30], [31] *i.e.*,

1) **Release of message content**: it is a passive attack where the opponent attempts to break the system based on observed messages. As a typical countermeasure, this attack can be prevented by encryption of payload, as occurs in our proposed technique.
2) **Traffic analysis**: it is another passive attack where an eavesdropper analyzes the traffic pattern, so to predict the nature of communication. Typical countermeasures are padding- and distribution-based approaches. This attack is partially effective against our technique. In fact, due to the random access to the medium, the eavesdropper is unable to observe the traffic patterns generated by sensors, and then to associate patterns to specific sensors. Anyway, this attack is not critical for the capillary networks applications.
3) **Masquerade**: it is a type of attack in which the attacker impersonates a user. Whereas in replay attack, the attacker just sends the same data packet to some user assuming to have the same effect. Masquerade attack is traditionally solved through authentication mechanisms. This attack is ineffective against our technique. In fact, the attacker cannot decode the key, generated according to the renewal algorithm, which is specifically assigned to each user/sensor in the network. Moreover, any authentication mechanism can be implemented at application layer to further improve the security.
4) **Replay**: it is a specific type of masquerade attack. Sometimes, replay attack may not relate to impersonation (*e.g.*, the attacker captures a password, or a cookie, in order to obtain unauthorized access with false identity). Traditional countermeasures exploit timestamps, One Time Password (OTP), and Secure Sockets Layer (SSL). The replay attack is ineffective against our technique, due to the present of a frame counter and the timestamp in the header of the proposed protocol.
5) **Intentional Denial of Service (DoS)**: this attack aims to affect user's resources (*e.g.*, a resource is unavailable to its intended users). It acts in different levels, mainly at MAC, and physical level. Above MAC level, typical countermeasures are implemented in Intrusion Detection Systems (IDSs); in our approach, the countermeasures are left to the application running on the server that should implement proper security algorithm to counteract against DoS. At MAC level, typical countermeasures are the Wireless Intrusion Prevention System (WIPS), including DoS detection, and location tracking. However, DoS at MAC level is ineffective or limited with respect to the proposed algorithm. Indeed, the cognitive security against malicious nodes tries to properly set transmission parameters to avoid DoS (see Subsection IV-D). On the other hand, at physical level, spread spectrum techniques can be adopted. As a result, DoS attack is limited with respect to the proposed technique, since when the opponent behaves as a jammer, it can be detected and removed by authorities.
6) **Modification of messages**: this attack results in accessing messages and modifying the content. Typical countermeasures exploit hashing techniques. This attack is ineffective with respect to our technique, due to the presence of an integrity field (*i.e.*, hash value) in the header.
7) **Man-in-the-Middle (MITD)**: this attack allows the attacker coming in between two hosts, so that all the communication between them passes through the attacker. Typical countermeasures exploit the HMAC (Hashed Message Authentication Code), SSL protocol, and mutual authentication. MITD attack results partially effective or ineffective with respect to our technique. Indeed, if the code of the protocol is cracked, no countermeasure exists (difficult to be performed). Nevertheless, the mutual authentication and hashing adopted in our protocol strongly reduce the possibility of this attack.

VIII. CONCLUSIONS

In this paper the problem of security access to a gateway/mediator for non-IP uni- and bi-directional IoT terminals has been addressed, by exploiting the cognitive security concept.

A time-based solution to generate the keys for secure connections has been proposed, and adapted to the case of uni-directional terminals that cannot receive any message from the gateway. The selected time-based technique does not require the presence of a (centralized) server for secure keys management for a huge number of terminals. Performance of the considered secure protocols are closely related to the (*i*) collision probability among uni- and bi-directional terminals, and (*ii*) the latency required for bidirectional terminals to correctly deliver packets. Thus, the problem of coexistence between ALOHA and non-persistent CSMA terminals transmitting in the same area, and in the same band, has been analyzed by simulations. The obtained results be used to assess the maximum number of ALOHA-, and CSMA-based terminals that can be served in the area for the specified performance target. Finally, the security analysis has shown the effectiveness of our countermeasures to main security attacks.

REFERENCES

[1] K. Su, J. Li, and H. Fu, "Smart city and the applications," in Proc. of International Conference on Electronics, Communications and Control (ICECC), pp. 1028-1031, 9-11 Sept. 2011.

[2] R. Giuliano, F. Mazzenga, A. Neri, and A.M. Vegni, "Security Access Protocols in IoT Networks with Heterogenous Non-IP Terminals," in Proc. of IEEE Intl. Conf. on Distributed Computing in Sensor Systems (DCOSS), 2014, vol., no., pp.257-262, 26-28 May 2014.

[3] O. Novo, N. Beijar, M. Ocak, J. Kjallman, M. Komu, and T. Kauppinen, "Capillary Networks – Bridging the Cellular and IoT Worlds," in Proc. of IEEE 2nd World Forum on Internet of Things, 2015.

[4] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," White Paper, April 2011, available online http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[5] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey", Computer Networks, Vol.54, 2010, p. 2787-2805.

[6] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements and future direction", Future Generation Computer Systems, Vol.29, 2013, p. 1645-1660.

[7] R. Rajsuman, "System-on-a-Chip: Design and Test," Boston, MA, Artech House, 2000.

[8] A. Perrig, R. Canetti, J. D. Tygar, D. Song, "The TESLA Broadcast Authentication Protocol," in RSA CryptoBytes, vol. 5, no. 2, pp. 2-13, 2002.

[9] J. Yun, I. Y. Ahn, N. M. Sung and J. Kim, "A device software platform for consumer electronics based on the internet of things," in IEEE Transactions on Consumer Electronics, vol. 61, no. 4, pp. 564-571, November 2015.

[10] R. Giuliano, F. Mazzenga, A. Neri, A.M. Vegni, and D. Valletta, "Security implementation in heterogeneous networks with long delay channel," in Proc. of 2012 IEEE 1st AESS European Conference on Satellite Telecommunications, ESTEL 2012, Rome, Italy, p.1-5.

[11] R. Giuliano, A. Neri, and D. Valletta, "End-to-end secure connection in heterogeneous networks for critical scenarios", WIFS 2012 - Proc. of the 2012 IEEE Intl. Workshop on Information Forensics and Security, Tenerife, Spain, p. 264-269.

[12] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A Review," in Proc. of Intl. Conf. on Computer Science and Electronics Engineering (ICCSEE), vol. 3, no., pp. 648-651, Mar. 2012.

[13] R. H. Weber, "Internet of Things – New security and privacy challenges," Computer Law & Security Review, Vol. 26, No. 1, Jan. 2010, pp. 23-30.

[14] G. Mulligan, "The 6LoWPAN architecture," in Proc. 4th ACM workshop on Embedded networked sensors (EmNets '07), pp. 78-82, 2007.

[15] B. Frank, Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol," (CoAP), IETF draft, Jul. 2011.

[16] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, R. Kelsey, R. Kelsey, R. Struik, J.P. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," Request For Comments (RFC): 6550, Mar. 2012.

[17] A. Rghioui, M. Bouhorma, and A. Benslimane, "Analytical study of security aspects in 6LoWPAN networks," in Proc. of 5th Intl. Conf. on Information and Communication Technology for the Muslim World, 2013.

[18] P. Kasinathan, C. Pastrone, M.A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in Proc. of IEEE 9th Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013, vol., no., pp.600-607, 7-9 Oct. 2013.

[19] M.R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L.A. Grieco, G. Boggia, M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," IEEE Communications Surveys & Tutorials, vol.15, no.3, pp.1389-1406, Third Quarter 2013.

[20] M. Nixon, "A Comparison of WirelessHART‡ and ISA100.11a," White Paper, Sept 23, 2012.

[21] J. Tan, and S.G.M. Koo, "A Survey of Technologies in Internet of Things," in Proc. of IEEE Intl. Conf. on Distributed Computing in Sensor Systems (DCOSS), 2014, vol., no., pp.269-274, 26-28 May 2014.

[22] K. Witold, "Towards cognitive security systems", in Proc. of Cognitive Informatics Cognitive Computing (ICCI*CC), 2012 IEEE 11th International Conference on, pp. 539-539, Aug. 2012.

[23] R. Giuliano, F. Mazzenga, and M. Petracca, "Consumed Power Analysis for Mobile Radio System Dimensioning", IEEE International Conference on Communications (ICC 2013), Budapest, Hungary, Jun. 2013.

[24] ZigBee Alliance, "ZigBee Specification", Document 053474r17, January 17, 2008.

[25] Z. Shelby, C. Bormann, "6LoWPAN: The Embedded Internet", ed. John Wiley & Sons Ltd, 2009, ISBN 9780470747995.

[26] Marcos A. Simplcio Jr., Paulo S.L.M. Barreto, Cintia B. Margi, and Tereza C.M.B. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks," Computer Networks, Volume 54, Issue 15, 28 October 2010, Pages 2591-2612, ISSN 1389-1286.

[27] S. Chang, S. Ji, J. Shen, D. Liu, and H. Tan, "A Survey on Key Management for Body Sensor Network," 2015 First International Conference on Computational Intelligence Theory, Systems and Applications (CCITSA), Yilan, 2015, pp. 217-221.

[28] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," Computer Networks, Volume 76, 15 January 2015, Pages 146-164, ISSN 1389-1286.

[29] J. Granjal, E. Monteiro, and J. S Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," IEEE Communications Surveys & Tutorials (Volume:17, Issue: 3), August 2015, pp. 1294-1312.

[30] W. Stallings, "Cryptography and Network Security: Principle and Practice", 5th ed., Prentice Hall, 2011.

[31] A. Jesudoss and N.P. Subramaniam, "A survey on authentication attacks and countermeasures in a distributed environment", Indian Journal of Computer Science and Engineering (IJCSE), Vol. 5 No.2 Apr-May 2014.

[32] S. Jain, Vinoth Kumar N., A. Paventhan, V. Kumar Chinnaiyan, V. Arnachalam and M. Pradish, "Survey on smart grid technologies- smart metering, IoT and EMS," Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students Conference on, Bhopal, 2014, pp. 1-6.