European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Large Scale Integrating Project

EXALTED

Expanding LTE for Devices

FP7 Contract Number: 258512

WP5 – Security, authentication, provisioning

Deliverable 5.3
Security Solutions for P2P Relaying

Contractual Date of Delivery:	February 28th 2013
Actual Date of Delivery:	April 04th 2013
Responsible Beneficiary:	Gemalto
Contributing Beneficiaries:	CEA, Gemalto, TKS, UniS, Vodafone
Estimated Person-Months:	26
Security:	Public
Nature	Report
Version:	1.0

Document Information

Document ID: Exalted_WP5_D5.3
Version Date: 3 April 2013
Total Number of Pages: 59

Authors

Name	Organisation	Email
Nick Bone	Vodafone	Nick.Bone@vodafone.com
Herve Ganem	Gemalto	Herve.ganem@gemalto.com
Shahab Mirzadeh	University of Surrey	s.mirzadeh@surrey.ac.uk
Nouha Oualha	Commissariat à l'énergie atomique	nouha.oualha@cea.fr
Aleksandar Obradovic	Telekom Srbija	bojanaja@telekom.rs

Executive Summary

This report is dealing with the security of relayed (or multihop) communications in M2M networks. Relaying occurs when communications from/to one node leverage the communications capabilities of adjacent nodes.

The first part deals with relayed communication occurring in capillary networks for enhanced flexibility, convenience, resilience or energy efficiency.

Relayed communication introduces security threats of its own which are investigated. The importance of security pairing (or security bootstrapping) in the overall security process is explained.

Considering the security bootstrap in an isolated multihop capillary network, group device pairing algorithms are studied and their application in constructing authenticated group key agreement protocols discussed. The algorithms are divided in two categories of protocols: with and without the trusted leader. We show that protocols with trusted leader are more communication and computation efficient.

This study considers both insider and outsider adversaries and presents protocols that provide secure device pairing for uncompromised node even in presence of corrupted group members. Three new group device pairing protocols, namely group numeric comparison, group MANA II, and multichannel group device pairing are proposed and communication efficiency of Nguyen and Roscoe's HCBK protocol and Laur-Pasini SAS-based group key agreement protocol has been enhanced.

The connection of a capillary network to Wide Area Network (WAN), through a gateway acting as a group leader to bootstrap the security on the capillary side is investigated. This WAN is possibly, but not necessarily a 3GPP (2G, 3G, LTE, or LTE-M) network. We investigate possible roles for the gateway in the security architecture and identify three such roles.

- Simple **funnel** for data communication originating from capillary devices. Each capillary device can connect independently to an infrastructure M2M network using its own identity.
- **Data aggregator**, connecting to an infrastructure M2M network with a single identity and relaying data to devices in the capillary network. Only the gateway is seen from the WAN side, hiding the structure of the capillary network.
- **Mediator** connecting to an M2M infrastructure network and **linking** the security of the capillary network with the one of the M2M infrastructure network. We describe how this may be achieved. A first benefit of this idea lies in the possibility to cut the security overhead on the capillary side by collapsing two security layers into a single one, and this may be a bonus for energy constrained devices. In addition it also opens the possibility for an M2M service provider to manage, as a service, the internal security of the capillary networks of its subscribers. This is a new concept which may carry significant business potential.

The WAN can also be used as a bridge between different capillary networks, and we investigate possible bridging scenarios:

- One device belonging to one capillary network, connecting and relaying its communications via a guest capillary network in a mobility situation.
- Devices belonging to one capillary network are able to relay their communications via another capillary network in the immediate neighbourhood.

A security scheme to secure the communications is proposed in both cases.

The second part of the report focuses on relaying occurring in 3GPP Wide Area Networks. This concept is emerging in 3GPP infrastructure communications, with the perspective to become a key feature of tomorrow's networks. The LTE relay node used to enhance coverage in LTE networks is described, and the identified threats associated to this type of technique are listed.

An overview of on-going standardization efforts to specify the use of self-organized network coverage in case of emergency situations is provided.

Table of contents

1 Introduction

The topic of relayed (multihop) wireless communications has been the subject of key research in recent years. It encompasses ad hoc radio networks, sensor networks, wireless mesh networks and mobile multihop relay, and is related to industrial and standardization efforts such as IEEE 802.11s, 802.15.4, 802.16j, etc.

The idea behind multihop communications is to exploit the transmission capabilities of adjacent nodes. Such systems have a number of advantages over traditional communication networks regarding ease of deployment, connectivity, capacity, and coverage extension while minimizing the need for fixed infrastructure.

On the other hand, multihop systems are associated to problems of their own. One such problem is related to the energy consumption of the nodes which will be dependent upon their relaying activity. Energy efficiency through cooperative retransmission has been investigated within the EXALTED project [2]. Routing is another problem which has also been investigated within EXALTED [3].

Security of communications is yet another issue: as nodes are relaying the communications of adjacent nodes, it is important to be able to discriminate friendly nodes from unrelated and potentially malicious ones. This discrimination is done during an initial phase where security pairing (or security bootstrapping) is performed. This phase results in the definition of a secret key shared by all the friendly nodes and enabling them to protect the group communications.

Chapter 2 deals with the security of communication within self-organized capillary networks, with a particular focus on wireless sensors networks.

Securing communications is usually a two-step process: An initial phase (pairing or bootstrapping) involves the distribution of shared secrets between all the parties involved. A second phase makes use of those secrets to actually secure communications.

Section 2.3 explains why pairing is a critical step and section 2.4 contains an extensive compilation of group device pairing mechanisms proposed in the literature which we believe are suitable to achieve pairing within capillary networks. The methods are classified in two categories:

- methods involving a special node acting as a group leader (section 2.4.1)
- methods performing pairing among a group of equal peers (section 2.4.2)

We then compare the methods according to the followings associated requirements:

- The underlying devices structure (for example “need presence of a display”),
- The human/user effort required (for example “press a button”)
- The need for an auxiliary out of band communication channel.
- Security features taking into account the possibility of both insider and outsider adversaries.

Three new group device pairing protocols, namely group numeric comparison, group MANA II, and multichannel group device pairing are proposed and communication efficiency of Nguyen and Roscoe’s HCBK protocol and Laur-Pasini SAS-based group key agreement protocol are enhanced.

Section 2.4 is dedicated to securing communications within an “isolated” capillary network.

Section 2.5 addresses the problem of connecting the capillary network to the outside world and more precisely to a Wide Area Network (possibly but not necessarily a 3GPP (2G/3G/LTE/LTE-M) network) through a gateway. It identifies two roles for the gateway:

- Gateway as funnel
- Gateway as aggregator

- Gateway as mediator:

While the two first ones are well known, the third one is more original, and leads to the idea of bridging LAN and WAN security described in section 3.3.

In addition, we describe in section 2.6 a very economical method based on Physically Unclonable Functions (PUF) which may be used by capillary devices to bootstrap their security with the remote M2M server.

Chapter 3 investigates the wider use of relaying in infrastructure networks. We present the concept of a relay node (RN) and describe on-going efforts in 3GPP investigating the use of relayed communication in infrastructure networks to provide network coverage in case of emergency situations.

Section 3.3 also looks at infrastructure networks, but from a different perspective, considering how a wide area M2M infrastructure may be used to bridge between different capillary networks. We introduce the concept of “global” relaying whereby capillary devices are able to get their communications relayed into (or through) capillary networks belonging to other users and provide two scenarios of “global relaying”. A possible implementation of the security is proposed for each scenario.

Finally, section 4 presents an overview of standardization activity related to multihop and relaying and describes the current targets of standards working groups.

2 Relaying in self-organized capillary networks

2.1 Data propagation in capillary networks

Capillary network communications in EXALTED are based on a hierarchical structure, as shown in Figure 2.1, where cluster head (CH) nodes control and manage non-LTE M2M devices locally. Data collected/generated by the devices are relayed to their corresponding CH node in a single hop. Collected data at CHs are then forwarded to the Gateway node, which has LTE connectivity to one or more eNodeB stations. Any communication that is to take place with specific devices follow the same hierarchy, i.e. packets go through the Gateway and forwarded over the "backbone" of CH nodes to the specific CH whose cluster encloses the requested device.

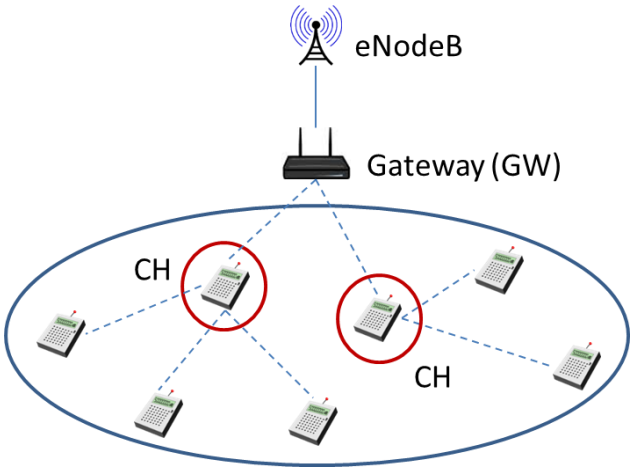

Figure 2.1 : The hierarchical structure of a capillary network

Having a hierarchical capillary network structure as provided in this figure is highly beneficial to support M2M communications. CH nodes perform any necessary data processing procedures in order to reduce the data volume that is forwarded to the Gateway, and eventually to the LTE-M network. In doing so, this data aggregation at CHs not only helps reduce the traffic load on the core network, but also alleviate the potential problems within the capillary network, such as wireless contention, receiver buffer overload, and excessive energy consumption caused by having large number of packet transmissions from a large number of devices. The latter benefit is especially important for the case of remotely located and battery operated devices, which are prone to energy depletion events and hence require efficient energy management procedures. Furthermore, any complex data processing job can be achieved at CHs, rather than at data source M2M devices, if these nodes are chosen as more capable devices.

2.2 Security threats and attacks in Wireless Sensor Network

Wireless Sensor Networks (WSNs) with their resource constrained nodes and unique features such as wireless communication, lack of infrastructure, and unprotected physical environment are vulnerable to different passive and active attacks. Passive adversaries can eavesdrop on sensors' communication channels (even from the distance by using high gain antennas) and process the captured information in their own time. Active adversaries can additionally add corrupted sensors, cluster heads or base station to the WSN (or compromise, replace, or physically damage the current sensors) and modify, insert or delete the transferred information which can be any of sensor readings' data, mobile codes,

bootstrapping messages, and routing and location information. They can perform Man-in-the-Middle (MITM) attacks, hijack a session or even jam the communication channel. They can also send malicious mobile codes to compromise sensors or exploit the location information to locate critical nodes such as cluster heads and base stations for further attacks. They can manipulate the routing information and perform various attacks such as Sinkhole and Wormhole to hijack traffic or disconnect the network. This section will discuss some of these attacks in detail [1][4].

Denial of Service (DoS) attack: DoS attack is one of the most challenging attacks in WSNs which directly targets network and service availability of WSN. It may take several forms. It could be an easily deployable jamming attack in which an adversary jams the communication channels and disturbs the communication among the nodes, or in more serious cases even make it impossible. Power exhaustion or battery depletion attack is another form of DoS attack in which an adversary keeps communicating with sensors and depletes their battery faster than normal. In a specific type of this attack, the adversary does not let sensors go to inactive or sleep mode and for this reason it also known as sleep deprivation attack.

Attacks against networking and routing protocols: The adversary can also attack WSN by disturbing its networking protocols e.g. by modifying the routing tables in a way that makes the network topology disconnected. Adversary can attack the routing protocol by corrupting routing tables, selective forwarding packets, or lunching Sinkhole and Wormhole attacks. In Sinkhole attack, the adversary claims that it has the best path to base station/sink and forces the network traffic through itself. In wormhole attack, the attacker establishes a secret tunnel between two sensors (records messages at one place and moves them to the other place) to makes them think they are neighbours and this results in disruption of network topology and routing mechanisms.

Node compromise attack: Sensor nodes are resource constraint devices which are not generally temper-proof and because of the application and deployment scenario usually tend to be physically unprotected. A sensor node is compromised when adversary somehow gains its control for example by physically capturing the node and accessing its security keys. Without appropriate security countermeasure, the node compromising is also possible by changing the sensors operational codes over the air. Having compromised a node, adversary can exploit it for further attacks or replace it with a malicious node. Compromised nodes are serious problem for WSNs as they have the secret information such as keys and can launch several attacks which are hard to detect and protect [4]. They may demonstrate arbitrary behaviour and may collude with other compromised nodes. They can decrypt the encrypted data and pass it to the adversary. They can report wrong information to the network and subvert its normal behaviour. They can misleadingly report other normal nodes as compromised nodes or infect the routing tables and launch other routing attacks such as selective forwarding and black hole.

2.3 Pairing; an essential operation

To mitigate possible active and passive attacks in capillary networks, where a possibly large number of low-end devices such as sensors communicate with a gateway, we need to use cryptographic primitives such as encryption/decryption algorithms and message authentication mechanisms that all require a priori trust relationship among the communicating peers.

In self-organized capillary networks where devices do not have pre-shared key or common Public Key Infrastructure (PKI), the key establishment can be done through the pairing mechanisms. Pairing (also referred as security bootstrapping) is the process of establishing shared key between two or more devices or authenticating devices' public keys without using pre-shared secrets or Public Key Infrastructure. Based on the application and scenario, the desirable outcome of a pairing mechanism can be pairwise keys, group key, or authenticated public keys.

A number of methods are proposed in this document to achieve pairing and security bootstrapping of capillary networks. More specifically two type of pairing methods are addressed:

- Self-organized pairing methods, where pairing is done within the capillary network and without any infrastructure support (section 2.4)
- Infrastructure assisted bootstrap where the capillary network can rely upon an infrastructure network to perform security bootstrapping (section 2.5, and also section 3.3 in chapter 3)

2.4 Self Organized Pairing Methods

Whitfield Diffie and Martin Hellman were the first to study the issue of establishing shared keys between two parties that do not have a priori trust relationship and only communicate over insecure communications channels. They laid the foundations of public key cryptography in their 1976 seminal paper [5] and proposed a new key agreement protocol known as Diffie-Hellman (DH) protocol. In DH protocol, the two parties initially agree on a finite cyclic group G of order n with a generator g. As a concrete example, they choose a large prime *p* on order of 1024 bits and a generator *g* of order *p*-1 in multiplicative group $\wedge_p^*$. The prime *p* and generator *g* do not need to be confidential and can be some fix and public parameters. Figure 2-2 shows the DH protocol. In order to simplify the notation, the modulus *p* has been omitted in the figure. The protocol's steps are as follows:

- Alice chooses its DH private key *"a"* such that $0 \le a \le p-1$ and calculate its DH public key g^a mod *p.*
- Bob chooses its DH private key "*b"* such that $0 \le b \le p-1$ and calculate its DH public key g^b mod *p.*
- Alice and Bob exchange their DH public keys over the insecure channel and calculate their shared key as g^{ab} mod *p* by bringing the other party's public key to the power of their private key i.e. $K_{AB} = (g^a \bmod p)^b \bmod p = (g^b \bmod p)^a \bmod p = g^{ab} \bmod p$.

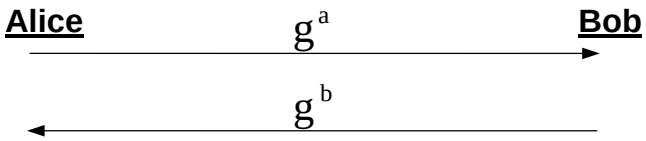

Figure 2-2: Diffie-Hellman key agreement protocol

The DH protocol is secured against eavesdroppers if group G and generator g are chosen properly. To compute the shared secret, a passive adversary has either to solve the Diffie-Hellman Problem (DHP), i.e. computing g^{ab} mod *p* from the observed DH public keys g^a mod *p* and g^b mod *p*, or to work out the Discrete Logarithm Problem (DLP), i.e. computing the DH private key from the respected public key, which are both believed to be hard for certain groups. However, the original Diffie-Hellman key agreement protocol does not provide authentication of the communicating parties and is vulnerable to the man-in-the-middle attack in which the adversary impersonates one or both of the communication parties. Figure 2-3 shows the man-in-the-middle attack scenario in which the adversary (Eve) establishes two DH key $g^{ab'}$ and $g^{a'b}$ respectively with Alice and Bob while they think they share them with each other. In this example, the adversary can even hide her existence with decrypting then re-encrypting the passing messages.

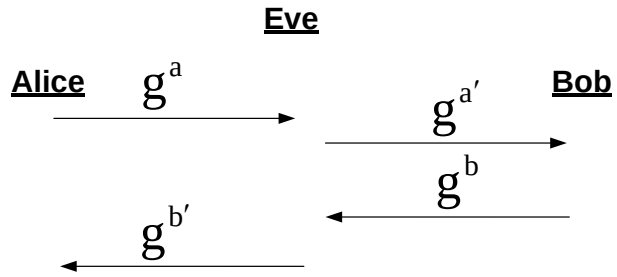

Figure 2-3: MITM attack scenario on DH key agreement protocol

The DH protocol has been extended to group key agreement protocols such as Burmester and Desmedt (BD) protocol. The Burmester and Desmedt (BD) group key agreement protocol [6] is a two-round contributory Diffie-Hellman based group key agreement protocol which is provably secure against passive adversaries in the standard model [7]. The protocol arranges participants in a ring structure in a way that each member $M_i, i \in \{1,...,n\}$ neighbours M_{i-1} and M_{i+1} ($M_0 = M_n$ and $M_{n+1} = M_1$). The protocol works in two rounds:

Round 1:

- Mi selects random $r_i \in_R \{1,\ldots\ldots,p-1\}$ as its private key, computes and broadcasts its public key $Z_i = g^{r_i}$.

Round 2:

- Mi computes and broadcast $X_i = \left(\frac{Z_{i+1}}{Z_{i-1}}\right)^{r_i}$
- Mi computes key K as:

$$K = Z_{i-1}^{nr_i} X_i^{n-1} X_{i+1}^{n-2} \ldots X_{i-2} = g^{r_1 r_2 + r_2 r_3 + \ldots + r_{n-1} r_n}$$

Manulis in [8] gave an elliptic curve equivalent of DB protocol, called μBD, as bellow:

Round 1:

- Mi selects random $r_i \in_R \{1,\ldots\ldots,p-1\}$ as its private key, computes and broadcasts its public key as $Z_i = r_i G$.

Round 2:

- Mi computes and broadcast $X_i = r_i\left(Z_{i+1} - Z_{i-1}\right) = (r_i r_{i+1} - r_i r_{i-1})G$
- Mi computes key K as:

$$K = nr_i Z_{i-1} + (n-1)X_i + \ldots\ldots + X_{i+n-2} = (r_1 r_2 + r_2 r_3 + \ldots + r_{n-1} r_n)G$$

Figure 2-4 shows a schematic diagram of BD group key agreement protocol.

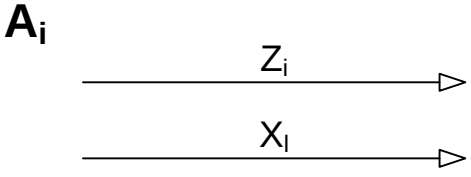

Figure 2-4: Burmester and Desmedt (BD) group key agreement protocol

The BD group key agreement protocol is also vulnerable to the man-in-the-middle attack. In this attack, the adversary impersonates some honest users and tricks the others into believing that they share a key with the trusted users. Figure 2-5 shows a hypothetical scenario of a man-in-the-middle attack. In this scenario, the active adversary E divides the intended group of four participants (A, B, C, D) to two subgroups by impersonating two legitimate members in each subgroup.

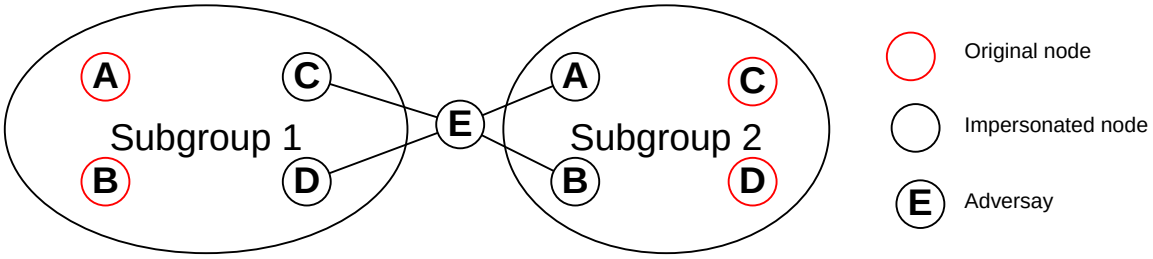

Figure 2-5: Man-in-the-middle attack scenario in group communication

In addition to man-in-the-middle attack, the active adversary can also secretly participate in group key agreement process and establish a group key with legitimate members. In this case all the legitimate users have the same group key with adversary. To mitigate this problem, also known as hidden node issue, it is important that group members somehow agree on the group size and its legitimate participants.

Susceptibility of DH based key agreement protocols to MITM attack, made the authentication of received DH public keys necessary and for this reason a variety of cryptographic authentication solutions was incorporated in Diffie-Hellman based key agreement protocols. For example, the Internet Key Exchange (IKE) protocol [9], Menezes-Qu-Vanstone (MQV) protocol [10], and Station-To-Station (STS) protocol [11] all assume the existence of a common Public Key Infrastructure and use digital signatures for authenticating the DH public keys. In another approach, the Encrypted Key Exchange (EKE) [12] protocol uses a priori shared password for this purpose.

PKI and password based authentication mechanisms are not appropriate solutions in self-organised capillary networks where networks usually form in ad hoc manner without existence of security infrastructure and contain computationally limited devices with inadequate user interfaces. A different approach is using Out-of-Band (OoB) channels in the authentication process. The aim of the out-of-band channel is to exchange some limited amount of confidential or authenticated information between the pairing devices which is then used to authenticate the established key over the main insecure wireless channel. The out-of-band channels provide demonstrative authentication (identification based on physical context), data origin authenticity (giving assurance of the source that the data came from), data integrity (providing assurance that data wasn't tampered with by a MITM attacker), and in some cases data confidentiality. While in insecure wireless channels the Dolev-Yao threat model [13] is a real threat model and adversary is in complete control of the channel and may overhear, block, forge, delay and replay the sent messages, the adversary power is limited on out-of-band channels.

Based on the security properties of out-of-band channels, we can distinguish between three types of private, public and weak out-of-band channels. While private out-of-band channels provide authenticity, integrity and confidentiality; the public and weak out-of-band channels only provide authenticity and integrity. The adversary does not have any power on private OoB channels, but she can block, delay, and eavesdrop information on public OoB channels and even can reply to them on weak OoB channels. Table 2-1 summarises the adversary power on different channels.

Table 2-1: Adversary power on different OoB channels

Adversary Power / Channel	Eavesdrop	Block	Delay	Replay	Forge	Example
Insecure Wireless	✓	✓	✓	✓	✓	Bluetooth, Wi-Fi
Weak OoB	✓	✓	✓	✓	□	Voice Mail
Public OoB	✓	✓	✓	□	□	Manual Data Transfer
Private OoB	□	□	□	□	□	Cable

Typically, proximity authentication in OoB channel is performed by touching the device, or by reading from or keying in some string via the device interface. An OoB channel, or a part of it, can also be implemented offline. For example, the device can be delivered to the user from manufacture with the necessary data such as the device's identity and a hash of its public key stored on an external medium such as piece of paper or Radio-frequency identification (RFID) tag. Such solution might be necessary for low-end accessories without a user interface.

In the literature there are different types of implementation for OoB channels. A typical example of manual date transfer is a user who reads an alphanumeric string from the display of one device and then enters it to the other device using its keypad or compares two short strings displayed on both devices. Other examples of OoB channels are RFID tags, Infrared communication, voice communication, displayed bar codes, and displayed lighting patterns. Malkani and Dhomeja in [14] have provided a survey of recent pairwise device pairing mechanisms; Table 2-2 shows their summary of current pairwise pairing methods.

Arun Kumar et al. in [26] provided a detailed usability study of secure device pairing methods. Based on their analysis of robustness and usability measures, they conclude that number comparison is the best way for OoB channel implementation (if devices' user interfaces allow it). In comparison to phrase and image comparison, their study showed that number comparison is less likely to face user safe errors which require repeating the procedure. They recommend audio pairing such as Human-Assisted Pure Audio Device Pairing (HAPADEP)) for devices with speaker/microphone that do not have display. For interface constrained devices they endorsed button-enabled (Button-Enabled Device Authentication (BEDA)) methods where a device signal its check value via its LED, vibration, or voice (beep) and user press the button on other device.

In this section, we will focus on group device pairing mechanism abstracted from the implementation of OoB channel. Based on the above recommendation and existing implementation such as [27] we also recommend using LED's blinking pattern for implementing out-of-band channels.

In addition to the above out-of-band channel pairing techniques, there are also other categories of pairing methods which exploit physical properties of communication channel between the sender and receiver to establish secure key among them. For example in [28] authors proposed a new in-band wireless pairing mechanism in which the key establishment messages exchange in a way that adversary cannot either block them or alter them without being detected. In Section 2.4.4 we will extend our study to in-band pairing mechanisms and will investigate their suitability for the capillary networks within the EXALTED project.

Table 2-2: Summary of device pairing methods [14]

Pairing Method	Minimum hardware or equipment required in each of the device		Human/User effort required	Out-of-band/Location-limited secondary channel
	Device A	Device B		
Resurrecting Duckling Security Model [15]	A cable and the same physical interface (e.g. USB port) on both of the devices		Set up cable connection between the devices	Cable
Talking to Strangers [16]	Infrared (IrDA) port on both of the devices		Set up infrared (IrDA) connection between the devices	Infrared (IrDA)
Smart-its-Friends [17]	2D accelerometers on both of the devices		Move/shake devices together simultaneously until response signal received	Accelerometer/Motion/Tactile
Are You with Me? [18]	2D accelerometers on both of the devices		Walk around to shake the devices (sensors) for certain time period	Accelerometer/Motion
Shake Well Before Use [19]	2D accelerometers on both of the devices		Move/shake devices together simultaneously until response signal received	Accelerometer/Motion/Tactile
Seeing-is-Believing [20]	Display	Photo Camera	Properly place camera of device B at the displayed bar code on device A with sufficient proximity and take the photograph	Visual
Loud &Clear (Display-Speaker) [21]	Display	Speaker	Compare the MadLib sentence displayed on the screen of device A with the vocalized MadLib sentence from device B	Combination of audio and visual
Loud &Clear (Speaker-Speaker) [21]	Speaker	Speaker	Compare the two vocalized MadLib sentences from both of the devices	Audio
HAPADEP [22]	Speaker	Microphone	Compare two audible sequences/melodies	Audio
Shake Them Up [23]	802.11 network card/interface	802.11 network card/interface	Shake/twirl/move devices around until pairing is done or response signal received	Combination of 802.11 and motion
AMIGO [24]	802.11 network card/interface	802.11 network card/interface	Shake/wave hand near the device until pairing is done or response signal received	Combination of 802.11 and tactile
BEDA (Button-to-Button) [25]	A single button on both of the devices		Press button on both of the devices simultaneously with random time-intervals until response signal received	Tactile
BEDA (Display-to-Button) [25]	Display	A single button	Press and release button on device B whenever display of device A flashes	Tactile
BEDA (Short Vibrations-to-Button) [25]	Vibration capability	A single button	Press and release button on device B whenever device A vibrates	Tactile
BEDA (Long Vibrations-to-Button) [25]	Vibration capability	A single button	Press and hold the button on device B while the device A vibrates	Tactile

2.4.1 Group Device Pairing Protocols with Trusted Leader

We categorized group device pairing mechanisms in two general categories: with and without trusted leader. Group device pairing mechanisms still require that uncompromised nodes be able to authenticate each other's public keys even in presence of corrupted group members. However the presence of a trusted group leader node among the group members, can lead to more efficient communication and computation protocols. In the following, we study group device pairing for a group of n devices with a trusted leader. We denote the

group leader by L with public key PK_n and identity ID_n, normal member by S_i, i=1,…,n-1 (S for slave) with public key PK_i and identity ID_i, and either leader or ordinary group member with A_i, i=1,…,n (A for all).

2.4.1.1 Group Numeric Comparison Device Pairing Protocol

Valkonen et al in [29] studied ad hoc security associations for groups. They extended MANA III and MANA IV pairwise message authentication protocols to group authentication protocols and proposed two protocol of "MANA III for group" and "group numeric comparison". Their group numeric comparison protocol can be extended to a group device pairing protocol as depicted by Figure 2-6.

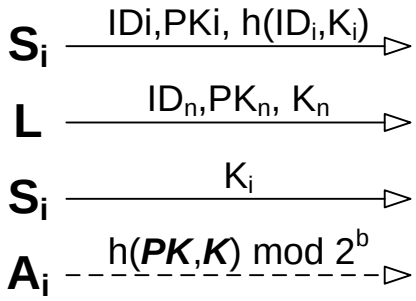

Figure 2-6: Group numeric comparison device pairing protocol

The protocol works as follows:

- Each non-leader member (S_i, i=1,…,n-1) generates a fresh long random number K_i, computes $h_i=h(ID_i,K_i)$ using a hash function h on its identity and random key and broadcasts it along with its identity ID_i and public key PK_i.
- Group leader L waits until it has received all n − 1 above messages, then picks a fresh long random number K_n and broadcasts it along with its identity ID_n and public key PK_n.
- S_i waits until it receives K_n from group leader and h_j from all other members (Sj , j = 1, . . . , n-1, j≠i), then broadcasts K_i.
- Upon receipt of K_j from other devices, every node uses it to verify the received h_j in the first message. If any check fails, the node aborts the process and indicates its failure to the user who will reset others. Otherwise every device waits for a specific timeout (for possible reset from user) and then use a truncated hash function, to derive a short check value (e.g. 16-20 bits) on all the keys and public keys and show the result to user.
- User compares all the check values and accepts or rejects the pairing process by updating all the devices based on the outcome.

The protocol security is inherited from the pairwise MANA IV protocol for which Laur and Nyberg gave a formal proof in [30]. Informally the protocol structure is a way that the adversary in any MITM attack scenario has to fix her inputs to truncated hash function "$h(\mathbf{PK},\mathbf{K}) \bmod 2^b$" before the complete set of others inputs become available. Hence to launch a successful attack the adversary has to either directly attack the hash function "$h(\mathbf{PK},\mathbf{K}) \bmod 2^b$" or try to get some knowledge about the K_i from $h(ID_i, K_i)$. If hash function $h(ID_i, K_i)$ is collision resistant and pre-image resistant and truncated hash function "$h(\mathbf{PK},\mathbf{K}) \bmod 2^b$" preserves its input entropy, which means it produces uniform output when any long-enough part of its input is uniformly distributed, then the protocol is secure.

2.4.1.2 Group MANA II Pairing Protocol

We can also extend the MANA II pairwise protocol to group device pairing mechanism. Figure 2-7 shows our proposal for group MANA II pairing protocol. In this protocol, initially all

the devices exchange their identities and public keys through broadcast communication. Then upon receiving the others public keys and identities (e.g. after a timeout), they use a one-bit OoB channel (e.g. a green light) to acknowledge the receipt and also show that they do not accept any other messages. With getting positive acknowledges from all devices, user commands the group leader (trusted node) to broadcast a short key K (16-20 bits) usable for computing a universal hash function. With having key K, all devices use it in a universal hash function to compute a short digest of 16-20 bits over the hash of all public keys (the public keys are hashed to give a shorter input to universal hash function) and show the result with key K to user. The user compares all the displayed digits and based on result ("all match" or "some differs") accepts or rejects the pairing process and updates all the devices e.g. by pressing a push-button on them.

Protocol's structure is such that all the public keys are fixed before the authentication process is started and so the protocol's security is quite similar to the MANA II pairwise data authentication protocol. Similarly this protocol is not optimal with regards to required data transfer over OoB channel as it provides 2^{-k} protection against MITM attack at the expense of 2k bits data transferred over OoB channel. In spite of this inefficiency, the protocol is communication and computation efficient. It only requires one universal hash and one hash computation per device and one broadcast communication over the wireless channel per participant (2 broadcast for the leader). These properties make the protocol suitable for scenarios such as sensors networks where the lightweight protocols are needed and OoB message can be transferred by using sensors' LEDs.

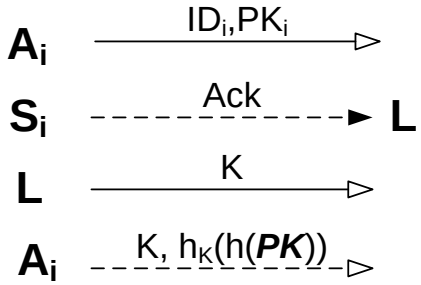

Figure 2-7: Group MANA II pairing protocol

2.4.1.3 Multichannel Group Device Pairing Protocol (MC-GDP)

Wong and Stajano in [31] and [32] studied multichannel security mechanisms including multichannel group key agreement protocols. In [31], they used their unidirectional authentic channel authentication protocol to authenticate Cliques Group Key Agreement (CGKA) Protocol's messages at every round. As Cliques group key agreement protocol is an n-round protocol, the proposed mechanism was not efficient and so in [32] they proposed to just authenticate the final established group key, instead of intermediate keys origin at every round, by using unidirectional OoB channels from a trusted leader towards group members. Their new protocol is extendable to group device pairing protocol as depicted by Figure 2-8.

In this approach, every node (except the leader) broadcasts its identity and public key. Having received all identities and public keys, the group leader broadcasts its identity and public key along with a message authentication code of all public keys, identities and a short and fresh random key *R*. To compute the MAC, the group leader uses a fresh long random key *K* in each protocol run. Having received leader's message, all the group members acknowledge the receipt to leader by using a single bit OoB message e.g. showing a green light to user who will update the leader. After receiving the acknowledgements from all group members, group leader reveals its short key over OoB channel (e.g. show it to user and user enter it on all devices) and long key over wireless channel (broadcast to all). In authentication stage, group members use both short and long keys to compute the same message authentication code on all the public keys and identities and compare it with the previously

received MAC. They show the comparison's result to user who accepts or rejects the process.

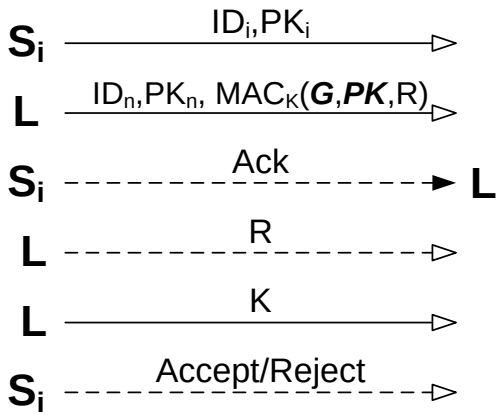

Figure 2-8: Multichannel group device pairing protocol (MC-GDP)

2.4.1.4 Hash Commitment Before Knowledge (HCBK) Protocol

Nguyen and Roscoe in [33] and [34] modified previous Roscoe's work [35] on group message authentication over OoB channels and finalised it as Hash Commitment Before Knowledge (HCBK) protocol. The HCBK protocol is directly applicable to group device pairing as depicted by Figure 2-9.

In this protocol, initially every node (including the leader) broadcasts its public key and identity. Then the group leader broadcasts a secure hash of a freshly generated long key *K* and participants acknowledge its receipt by using OoB channel. With getting all the group member's acknowledgement, the leader reveals its key *K* which is used by everyone (including the leader) to derive a short digest on all the public keys. The nodes compare their digests over OoB channel e.g. they present it to the user who accepts or rejects the pairing process based on the results. The author gave a concrete example of digest functions using Pseudo-Random Numbers Generation (PRNG) which they claimed is more efficient than other universal hash functions or truncated version of a secure hash function.

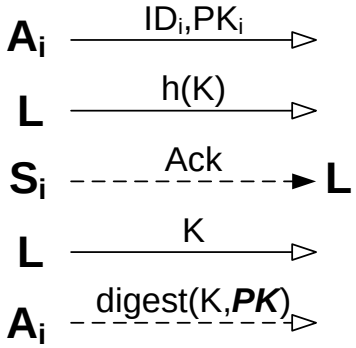

Figure 2-9: HCBK based group device pairing protocol

We can improve communication cost of HCBK protocol by sending leader's identity and public key in combination with h(K) in a single broadcast message instead of two in the original protocol. The enhanced version is shown in Figure 2-10 as HCBK* group device pairing protocol.

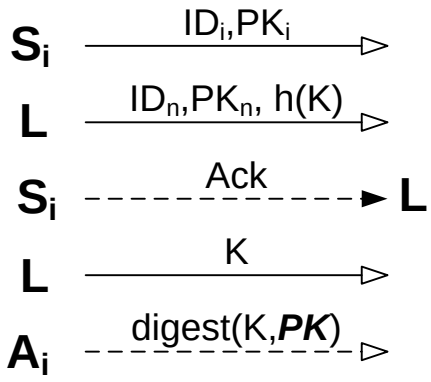

Figure 2-10: HCBK* group device pairing protocol.

2.4.2 Group Device Pairing Protocols without Trusted Leader

In the following group device pairing protocols, there is role symmetry among the participants and every node behaves similarly and sends similar messages. In this category, there is no need for trusted leaders and protocols can tolerate compromised nodes at the expense of more communication and computation cost.

2.4.2.1 Multichannel Group Message Authentication Protocol (GAP)

Perkovic et al in [36] extended Vaudenay's pairwise SAS-based message authentication protocol to multichannel group message authentication protocol and showed that direct extension of this protocol (such as Nguyen and Roscoe's "indirect-binding group protocol" in [33] and [37]) requires ordered communication and fully trusted nodes; otherwise any compromised or corrupted node can impersonate the others. They proposed two protocols: one with assumption of ordered communications and trusted nodes and one without those assumptions and secure against compromised devices. Figure 2-11 shows their later protocol which fulfils our requirements in group device pairing.

In this protocol devices initially broadcast their identities. Then each device computes a secure hash based on its view of all the group participants' identities (i.e. $h_{gi} = h(ID_1,...,ID_{i-1},ID_i,ID_{i+1},...,ID_n)$) and commits to the hash result along with its identity, its public key, long random key K_i, and short random key R_i by sending c_i where $(c_i, d_i) \leftarrow com(h_{gi}, ID_i, PK_i, R_i, K_i)$. With having received all the participant's commitments, each device reveals its long random key K_i and broadcast it to others. After this stage all the devices open their commitments by broadcasting d_i and so reveal their short key R_i. In authentication stage all the nodes compare XOR of all the short keys over the OoB channel.

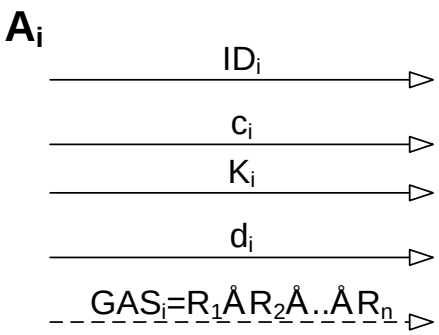

Figure 2-11: GAP protocol secure against compromised devices

The authors discussed their protocol's security heuristically and only provide a sketch of security proof for their first protocol with assumption of fully trusted nodes, collision resistance property of hash function h_{gi}, and non-malleability of the commitment scheme.

2.4.2.2 Symmetrised HCBK Protocol (SHCBK)

Nguyen and Roscoe in [33] and [34] proposed a symmetrised (balanced) version of their HCBK protocol which does not need trusted node. In this protocol, depicted by Figure 2-12, every node initially commits to a long random key K_i by broadcasting its identity, public key along with a hash of the key K_i and its identity. Having received all the commitments, nodes reveal their key K_i with broadcasting it over the wireless channel. In authentication stage, all the nodes derive key ***K*** as XOR of all the individual keys and use it in a digest function to compute a short authentication string from all the public keys and compare it over OoB channel.

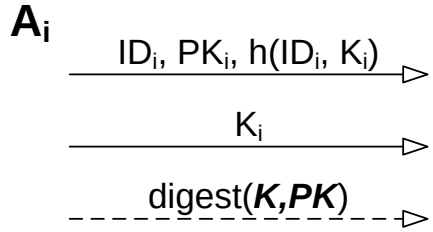

Figure 2-12: Symmetrised HCBK protocol (SHCBK)

The authors discussed their protocol's security informally based on the fact that the key ***K*** is independent for the trusted nodes (which is the result of hiding and biding properties of the hash commitment and also for the final key's XOR operation) and also assuming a uniform output for the digest function.

It should be mentioned that SHCBK could also be seen as a balanced (symmetrised) version of group numeric comparison protocol (discussed at 2.4.1.1). Nguyen and Roscoe also proposed a similar protocol to numeric comparison protocol by de-symmetrising SHCBK and called it de-symmetrised SHCBK protocol. It is worth noting that both SHCBK and group MANA IV protocols are specific forms of SAS-GMA protocol, discussed in the next section, where instead of hash based commitments, non-malleable commitment schemes are used.

2.4.2.3 Laur-Pasini Group Message Authentication Protocol (SAS-GMA)

Laur and Pasini in [38] studied SAS-based group authentication and key agreement protocols. Based on the previous work of MANA IV and also Vaudenay's 4-round SAS-based message authentication protocol, they propose their SAS-based group message authentication (SAS-GMA) protocol and demonstrate its direct application in SAS-based authenticated group agreement protocols.

Figure 2-13 shows the SAS-GMA based group device pairing protocol. In this protocol all the group members initially broadcast their identity (ID_i), public key (PK_i), and commit to a long key (K_i) by sending their commitment c_i where $(c_i, d_i) \leftarrow com(ID_i, K_i)$. Having received all the commitments, group members reveal their long key with opening their commitment by sending decommitment value (d_i). In authentication stage, all nodes use the revealed keys in a multi-keyed universal hash function to derive a short authentication string (SAS) on all the public keys and compare the result (with help of user) over the OoB channel.

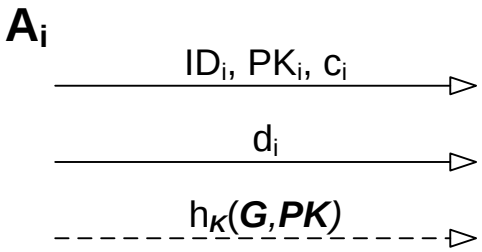

Figure 2-13: SAS-GMA based group device pairing protocol

The security of proposed SAS-based protocol is based on the binding and non-malleability of the commitment schemes and also almost regularity and almost universality of the hash function. Based on these assumptions, the authors provided a detailed security proof of their protocol in the stand-alone model and specified the requirements that should be met to securely compose the protocol with any other protocol.

2.4.2.4 Laur-Pasini Authenticated Group Key Agreement Protocol (SAS-AKA)

Laur and Pasini in [38] used their SAS-based group message authentication protocol (SAS-GMA) to authenticate Burmester and Desmedt (BD) group key agreement protocol's messages. To establish authenticated pairwise keys between the group members in addition to the group key, they proposed that besides BD protocol's messages (Z_i and X_i) the group members also exchange an extra DH public key and authenticate them at the same time with BD's messages. The SAS- based authenticated group key agreement (SAS-AKA) is depicted by Figure 2-14.

In this protocol all nodes initially broadcast two fresh DH public keys Y_i and Z_i. Then they compute Xi based on the BD protocol and broadcast it with Y_i and Z_i and their identities in addition to their commitments to a fresh long key Ki based on SAS-GMA protocol (i.e. $(c_i, d_i) \leftarrow com(ID_i, K_i)$). In the third broadcast all nodes reveal their key Ki by sending de-commitment value d_i. In authentication stage, all the nodes use the revealed keys in a multi-keyed universal hash function to derive a short authentication string (SAS) on all the public keys and compare the result (with user help) over the OoB channel.

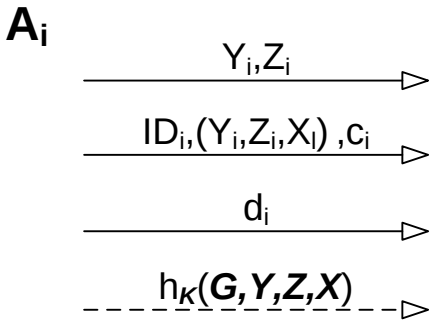

Figure 2-14: SAS-based authenticated group key agreement protocol (SAS-AKA)

In SAS-AKA two public keys Y_i and Z_i are unnecessarily broadcast twice (in the first and the second broadcast messages). To rectify this inefficiency, we proposed a modified version of protocol, called SAS-AKA* depicted by Figure 2-15.

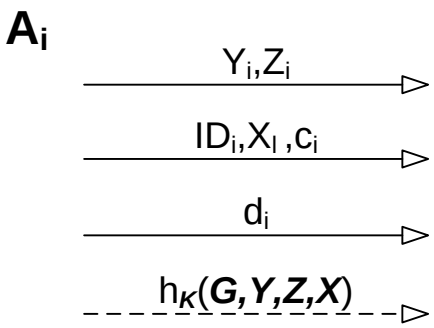

Figure 2-15: Improved SAS-AKA protocol (SAS-AKA*)

2.4.3 Comparison of Group Device Pairing Protocols

Table 2-3 compares the studied group device pairing protocols from the number of broadcast messages, OoB message size, required cryptographic primitives (H: hash function, UH: universal hash function, MAC: message authentication code, D: digest functions, C: cryptographic commitment scheme, XOR: exclusive-or operation), communication and computation cost. In this table, the adversary success probability is limited to 2^{-b} and so the out-of-band message is b-bits for all the protocols except group MANA II protocol which is 2b-bits.

For the computing communication cost we assumed that the commitment value C is a fixed size value |C| and decommitment value D is all the committed values in addition to a randomizer K. We used |ID| for identifiers, |PK| for public keys, |K| for long keys (e.g. 160 bits), and also commitments' randomizers, |R| for short random number (e.g. 32 bits), |H| for hash functions, and |MAC| for MAC functions. To give a better comparison, the table also shows the communication cost without considering public keys and identifiers (which are common for all the protocols) and in line with [37] using W for 160-bits average size of K, H, C, and MAC and W/5 for 32 bits R.

The comparison result indicates that group device pairing mechanisms with trusted leader have clear advantage over symmetrised protocols from communication and computation cost perspectives. In capillary networks, fortunately we can take advantage of gateway as the trusted leader. Among the protocols with trusted leader, group numeric comparison protocol and MC-GDP protocol are not efficient respectively from communication over wireless and OoB channel. Among the HCBK and group MANA II, we recommend using HCBK protocol as it requires less user effort over OoB channel and its inefficiency in sending 2W-bit instead of W-bit is acceptable as it is only for the gateway which is usually a powerful device.

Table 2-3: Group device pairing protocols – communication and computation cost

Protocol	# Broadcast Messages per device	OoB Message Size (bits)	Cryptographic Required Primitives	Computation Cost per Device	Total Communication Cost	Communication Cost without public keys and identifiers	Comments
Group Numeric Comparison	2 (L: 1)	b	CR hash function	N*H	N[|ID|+|PK|+|K|]+ (N-1)|H|	(2N-1)W	- with trusted leader
Group MANA II	1 (L: 2)	2b + 1 (1 bit for Ack. signal)	CR hash function + universal hash function	1*H +1*UH	N[|ID|+|PK|]+|K|	W	- with trusted leader
MC-GDP	1 (L: 2)	(N-1)b+ 2 (2 bits for Ack. signal)	message authentication code function	1*MAC	N[|ID|+|PK|]+|K|+ |MAC|	2W	- with trusted leader
HCBK	1 (L: 3)	b + 1 (1 bit for Ack. signal)	CR hash function + digest function	1*H + 1*D	N[|ID|+|PK|]+|K|+ |H|	2W	- with trusted leader
GAP	4	b	cryptographic commitment + XOR operation	1*H + N*C + (N-1)*XOR	N[|ID|+|PK|+|C|+ 2|K|+|R|]	3NW+ NW/5	
SHCBK	2	b	CR hash function + digest function	N*H + 1*D	N[|ID|+|PK|+|K|+| H|]	2NW	
SAS-GMA	2	b	cryptographic commitment + universal hash	N*C + 1*UH	N[|ID|+|PK| +|C|+2|K|]	3NW	• Provable Secure

2.4.4 Secure in-band group device pairing protocol

Pairing protocol's usability can be improved by applying new in-band wireless pairing methods. Capkun et al. in [40] proposed the Integrity Codes (I-codes) for integrity protection of transferred messages over the wireless channel where the communication peers do not have pre-shared key or authenticate public keys of each other's. Integrity Codes exploit physical properties of the radio channel and consist of three main parts: Unidirectional error-detection codes, on-off keying modulation, and the receiver's awareness of its presence in the sender's transmission range. Unidirectional error-detecting codes are able to detect any number of unidirectional errors (e.g. "0" to "1" but not "1" to "0") in a code word. On-off keying modulation is type of modulation in which bit "0" and "1" are respectively transmitted as the lack or presence of a signal.

Capkun et al. showed several ways to achieve the required signal anti-blocking feature for using unidirectional error-detecting codes in their communication system. They assume that attacker cannot disable the communication channel completely. In their model, attacker can jam the transmission channel and prevent transmission of the information contained in the message but receiver will still receive the original signal. More specifically, in their system the attacker cannot block bit "1" by removing energy of its signal from the channel.

Figure 2-16 shows an example of integrity coding operation. In this figure before sending a message over the insecure wireless channel, it goes through a unidirectional Manchester coding and then gets modulated via an on-off keying modulator. Sender verifies the validity of received code with confirming equal number of "1" and "0" in de-modulated code. The integrity and authenticity of I-coded message digest is guaranteed if and only if receiver

knows that it is in the power range of sender (condition of presence) and ensure that the sender has started transmission (condition of synchronization). Otherwise the adversary can insert its data on the channels and causes the receiver to accept them as valid.

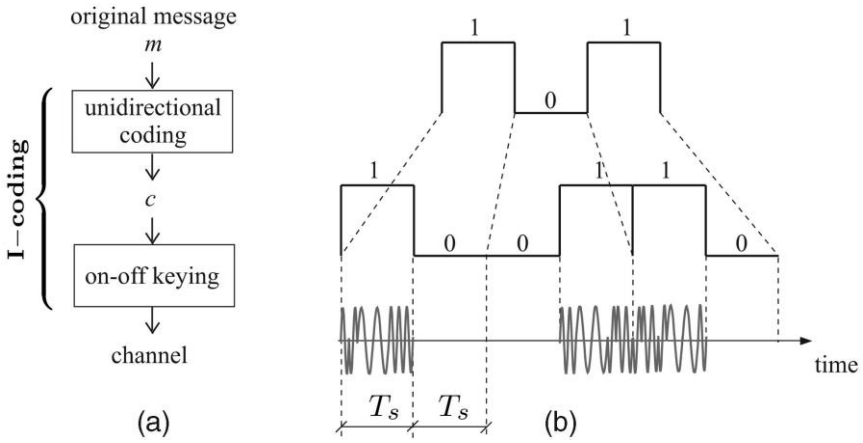

Figure 2-16: (a) I-coding (b) An example of I-coding at the sender using unidirectional Manchester encoding [40]

As in used on-off keying modulation, random signals are sent for “1” bits of derived Manchester code word, the adversary cannot eliminate them and so he/she cannot convert the sent code words to any other valid code word. For example to change original message from 101 to 111 he/she has to change the code word from “100110” to “101010” which requires bit flipping of “1” to “0” which is not possible unless with negligible probability. Adversary can change any symbol “0” to “1” but it makes code word invalid as Manchester code has equal symbols of “1” and “0”.

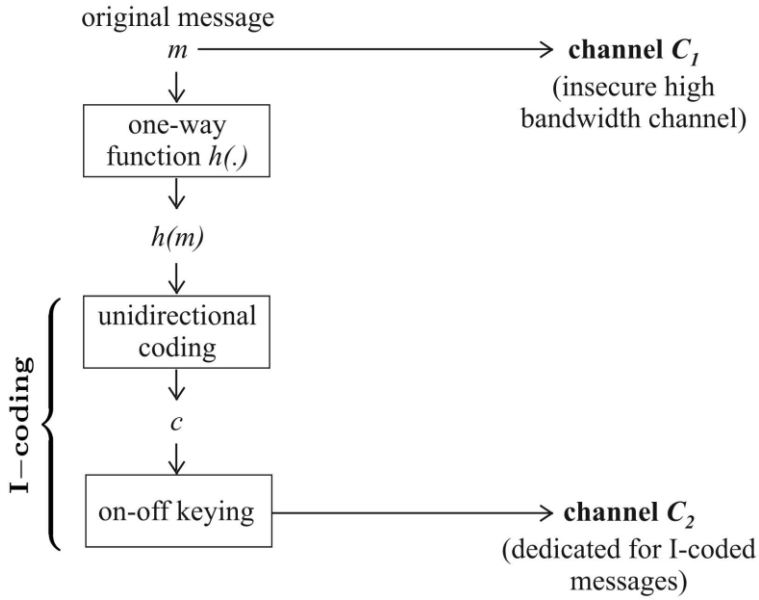

Figure 2-17: Possible usage of I-codes for integrity protection [40]

Figure 2-17 shows possible application of I-codes for message integrity protection. Here message is send over wireless channel and the message digest (hash) is sent I-coded over another dedicated channel. Thanks to the used on-off modulation and unidirectional error correction mechanisms, the dedicated channel protects integrity of message digest and as result protects the integrity of message itself. For a successful attack, adversary has to change message in a way that changed message has the same digest as the original one

which is not possible unless with negligible probability for computationally limited adversaries.

Figure 2-18 shows Capkun et al. I-code application for two-party authenticated key establishment named as I-coded based DH key agreement protocol (DH^{IC}). In this protocol, both Alice and Bob select their secret exponents 'X_A' and 'X_B', and calculated DH public keys as g^{X_A} and g^{X_B} respectively. They proceed by generating k-bit random strings N_A and N_B, and calculate commitment/opening pairs for the concatenations of their ID, DH public key, random strings and two fixed values 0 and 1 (0 and 1 are used to prevent a reflection attack). They send to each other the commitments value c_A and c_B and after that open their commitments with sending d_A and d_B respectively. Both parties check the correctness of the commitment/opening pairs with verifying that 0 and 1 appear at the beginning of the received messages and if the verification is successful, they proceed to the final stage in which they generate the verification strings $N_A \oplus N_B$ ($\oplus$ is the bitwise "XOR" operation) and one party (Alice in Figure 2-18) send its verification string I-coded to other one who verifies its correctness.

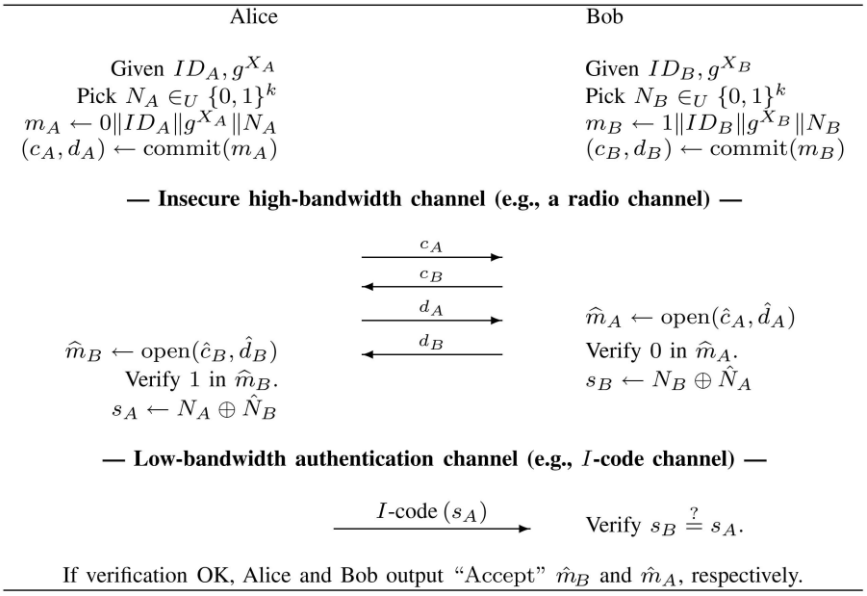

Figure 2-18: DH key agreement protocol based on I-codes (DH^{IC}) [40]

Based on I-codes, Gollakota et al. in [41] proposed Tamper-evident Pairing (TEP) as a new in-band pairing mechanism for IEEE 802.11(Wi-Fi). TEP improves DH^{IC} in this aspect that it does not need dedicated wireless channel for sending I-coded messages. To address MITM attack they introduced Tamper–evident Announcement (TEA) primitive. The main property of a TEA message is that an adversary cannot either hide that the message was transmitted or alter its payload without being detected.

Gollakota et al. have specified three types of actions that an adversary can take to launch a MITM attack:

- Collision: adversary jams the original message, causing collision, and then sends his own message instead
- Capture effect: adversary transmits simultaneously with a significant higher power and as result overshadows the original signal
- Timing control: after a party has sent out his message, the adversary continuously occupies the channel and so does not let the other part to reply it and then sends his reply instead

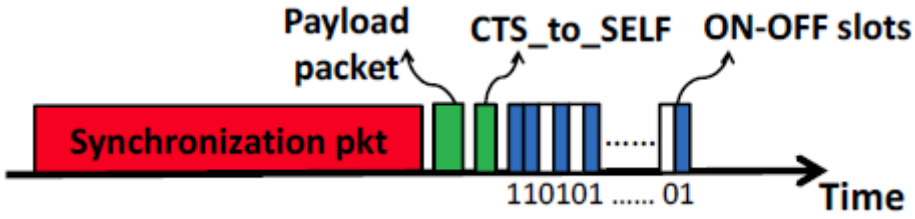

Figure 2-19: The format of a tamper-evident announcement (TEA) [41]

Figure 2-19 shows the format of a TEA message which mitigate above attacks by:

- Collisions on TEA messages are exceptionally longer than normal collisions and so are interpreted as potential attack on a key exchange message.
- The payload of a TEA message is followed by hash of the TEA payload in form of on-off slots. If the adversary overshadows the original message, the hash of the adversary's message with a very high probability does not have 0 in the same place of the original one and so the hash verification stage will invalid the message (hash is different because adversary cannot make "0" on "1" of the original hash).
- TEA uses CTS-to-SELF option of 802.11 protocol to reserve the channel for sending its message and also the receiver's reply. Honest nodes refrain from transmitting during the reserved interval and if adversary does that it will collide with receiver's reply who sends his reply assuming that all nodes are honest.

TEP use TEA to send key exchange messages. It is secure against MITM attack because any attempts to alter or hide a TEA can cause either an invalid TEA message or cause collisions that are interpreted as potential attack on key-exchange protocol.

Capkun et al. in [38] highlighted application of I-code in authenticating broadcast messages of a fixed Access Point (AP). In their suggested method, AP sends its public key I-coded on a dedicated wireless channel and jams the channel otherwise. As receivers know they are in signalling range of AP, they get an authenticated public key of AP and can use it to verify authenticity and integrity of all AP's messages including its broadcast messages. We can use this method to extend our group key agreement protocol for capillary network. In our approach, instead of presenting the digests to "user" for checking, the leader sends its digest I-coded to the group members over the insecure wireless channel. The group members check that the received digest is the same as theirs. This approach exploits the Integrity Codes to protect digest's integrity over wireless channel and it is immune from man-in-the-middle attack, since a MITM can only insert additional bursts into the silences and he/she cannot change bursts into silences and then the hashes won't match.

2.4.5 Incremental addition and revocation in capillary networks

Incremental addition and revocation is an important issue in capillary network. The user should be able to revoke any of current capillary nodes, without tearing down the whole network, in a way that revoked node(s) cannot access the future communications (forward secrecy). This also true for adding new nodes to network which should be done with minimum user effort in a way that new node cannot access previous communications (backward secrecy).

The gateway plays a central role in the capillary network's key management. In the proposed pairing process, gateway has authenticated public keys of all the paired capillary nodes and securely distributes a group key among them. To revoke a paired device, the gateway has to remove the public key of revoked node(s) from its list and distributes a new group key encrypted by the public key of the remaining capillary nodes. To add a new node to capillary network, the gateway has to securely pair with the new node, authenticates its public key and

gives it an encrypted copy of the current group key. In addition, the group key needs to be updated periodically to insure backward secrecy.

For pairwise pairing between the gateway and new device there are two possibilities of using well-tailored pairwise pairing device pairing protocols or applying the current group device pairing mechanisms.

2.5 Securing the connection of a capillary network to a wide area network

While the previous section was dealing with pairing methods to secure communications within an isolated capillary network, this section deals with the connection of a capillary network to the wide area network via the use of a gateway.

This section is dealing with application level security, the minimum requirement being the availability of an IP connection from the capillary network (gateway) to the wide area network. This connection may result from the gateway connection to a 3GPP network or be established via another channel such as an ADSL connection. The discussion below is agnostic of the type of IP connection being used. The scenarios proposed below are applicable to the EXALTED Architecture described in [71] which correspond to the particular case where IP connectivity is obtained via an LTE-M network.

It was shown in section 2.4 that the pairing methods for bootstrapping security in capillary networks may be classified in 2 categories:

- Self-organized pairing methods
- Pairing methods relying upon a group leader

In the last category, it is reasonable to expect that the gateway will play the role of the trusted leader. This particular scenario is investigated here as well as the different ways by which the gateway can act as a group leader.

The EXALTED project has been investigating the topic of data aggregation in capillary networks from the efficiency perspective [71]. We focus here on the different security models which can be used for traffic aggregation, and identify three possible roles played by the gateway:

- It may be only a group leader in order to help achieving pairing on the capillary side and act as a mere funnel enabling every single capillary device to have its own communication on the WAN side.
- It may aggregate data from multiple devices and transmit them as an aggregated stream on the WAN
- It may act as a mediator between the WAN and the LAN side of the net, in order to use on the LAN side security keys defined on the WAN side

In order to detail those 3 solutions we will make the assumption that the security bootstrapping procedure is leading to the definition of a group key.

When using security bootstrapping methods described above based upon asymmetric cryptography and leading to a secure publication of devices' public keys, we will make the assumption that the public key of each device is used to securely transmit a group key shared by all peers belonging to the ad hoc network. This assumption is justified by the reduced computing power required by symmetric cryptography compared to asymmetric cryptography.

2.5.1 Gateway acting as a group leader funnelling capillary devices data

This scenario is sketched on Figure 2-20 . In this case, the gateway is acting as a group leader to bootstrap the security in the capillary network. At the end of this phase the gateway and the other devices will secure their communications using a session group key.

In this scenario, the group key (Kg) is only used to channel data from each source device to the sink (gateway). In order to communicate on the WAN side, each capillary device will have to bootstrap its own WAN security possibly with an M2M service provider. This can be done:

- With a hop-by-hop data protection scheme, where each segment of the data transmission from source to destination is protected with different keys (as described in [42]),
- With end-to-end data encryption obtained with the help of an external authorization server (as described in [43]).

WAN communications security may also be achieved via a peer-to-peer negotiation between one of the capillary devices with a remote peer, using previously established shared secrets.

Whatever the method used, the bootstrapping stage will result in the definition of an application key Ka, used to encrypt data communications from the device.

It is important to note that in this scenario, 2 distinct layers of security are involved:

- Data communication in the capillary network is secured using the Kg key shared by all devices up to the gateway sink.
- Each device numbered [i] in the capillary network defines its own application key Ka[i] to secure its communication on the WAN side of the gateway.

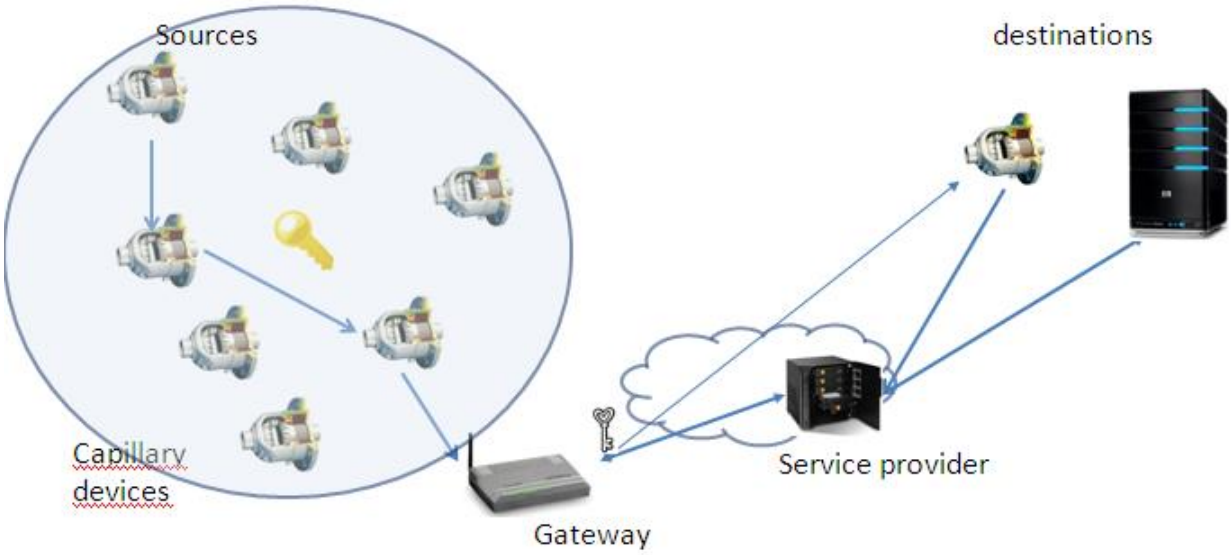

Figure 2-20: security scenarios involving different keys on the LAN and on the WAN side

The advantage of the scenario is the possibility for each device of the capillary network to generate WAN traffic with its own identity and its own security. First key Kg is used to relay data from the capillary node up to the sink (gateway). The second key Ka[i], negotiated at the capillary node [i] level is used to make sure that the data from one node remains opaque to the other relaying nodes. Such scheme is therefore suitable when each node of the capillary network needs to secure its own communications with respect to its peers, with its own identity while still using their data relaying capability.

2.5.2 Gateway acting as a data aggregator

Figure 2-20 is also suitable to describe this scenario. In this case, the gateway is acting as a group leader to help bootstrap the security on the LAN side. This leads as in the previous case to the definition of a key Kg shared between all devices of the capillary network, including the gateway.

The gateway acting as an independent device will bootstrap the security on the WAN side, either by registering with an M2M service provider using a hop-by-hop data protection scheme, or with end-to-end data encryption obtained with the help of an external authorization server (as described in [43]) or else by direct peer-to-peer negotiation with a remote peer, using previously established shared secrets. This will result in the definition of an application Ka, used to encrypt data communications from/to the gateway. The difference with the previous scheme is that from the perspective of the M2M service provider, there is a single identity involved. The fact that the traffic is generated by multiple devices is opaque to the M2M service provider.

Capillary devices will protect their data traffic using Kg up to the gateway, and using Ka from the gateway to the destination. The scenario described is very close to the proxy concept used in computers to achieve protocol translation. The gateway will decode data transmitted from capillary devices and protected with key Kg and re-encode it with the Ka key prior to transmission to the M2M/service provider or directly to a remote peer, possibly using a different transmission protocol. This protocol translation could be achieved at the application layer.

The advantage of this scenario lies in the fact that it reduces the computing burden for the devices which may be constrained both in energy and computing power, by avoiding a dual encryption scheme. Each capillary device can also handle its own communications with whatever source or destination address it may chose. However, all the capillary devices share the same gateway M2M identity.

Such a scheme is suitable when the capillary nodes do not need to protect their communication with respect to their peer. Data exchanged in the capillary network via relaying is readable by all nodes. From the outside world, the gateway appears with a single identity hiding the details about the devices involved in the aggregated traffic generation to the external world.

A disadvantage lies in the fact that data protection is piecewise, and the gateway needs to be trusted in order to achieve suitable security. This leads to a couple of sub-cases:

2.5.2.1 Data aggregation in private gateway

Here we assume that the gateway is “private” to the devices whose traffic is aggregated i.e., it doesn’t carry any other traffic. In this sense it is like a closed-mode femtocell or (home/enterprise) WLAN access point.

The gateway is likely to be owned by the same party that owns the devices, which is often also the application provider. Where there is common ownership, the threats are mostly limited to external parties attempting to corrupt or hijack the gateway or other devices in the capillary network, e.g. by logical attacks exploiting vulnerabilities in the gateway software. There are some physical attack scenarios as well (if, for instance, the gateway is in an accessible outdoor location), but the gateway owner has an incentive to protect against both logical and physical attacks.

2.5.2.2 Data aggregation in public gateway

Here we assume that the gateway carries a mixture of data traffic, not just the traffic that needs to be aggregated. In this sense it is more like an open-mode femtocell or public WLAN hotspot. Probably the gateway will need to be partitioned so that the aggregated private traffic is handled separately from the public traffic (Compare for instance FON[1]).

In this case, the gateway may still be owned by the party that owns the device, but an alternative scenario is for the gateway to be owned (or subsidized) by the M2M service provider, or access network provider. Again this scenario gives the gateway owner an incentive to protect against logical and physical attacks, but not quite the same incentive as the owner of the aggregated devices or application owner. Protection against logical attacks

[1] http://www.fon.com/en/info/security

arising from the open access is one reason to segregate traffic streams in a “public” gateway case.

2.5.3 Gateway acting as a mediator between the LAN and WAN side of the network

A variant of the above scenario consists in linking WAN and LAN security. This involves the gateway to communicate the Ka key to each capillary device to be used as a group key. In this case, the gateway will participate as described in section 2.5.2 in the pairing definition in the capillary network and perform, as well, a WAN side security bootstrap.

However, the pairing defined on the capillary side will be used to safely transmit the application key ka to each of the capillary devices. Each such device will then use this ka key to secure both the LAN part and the WAN part of the transmission.

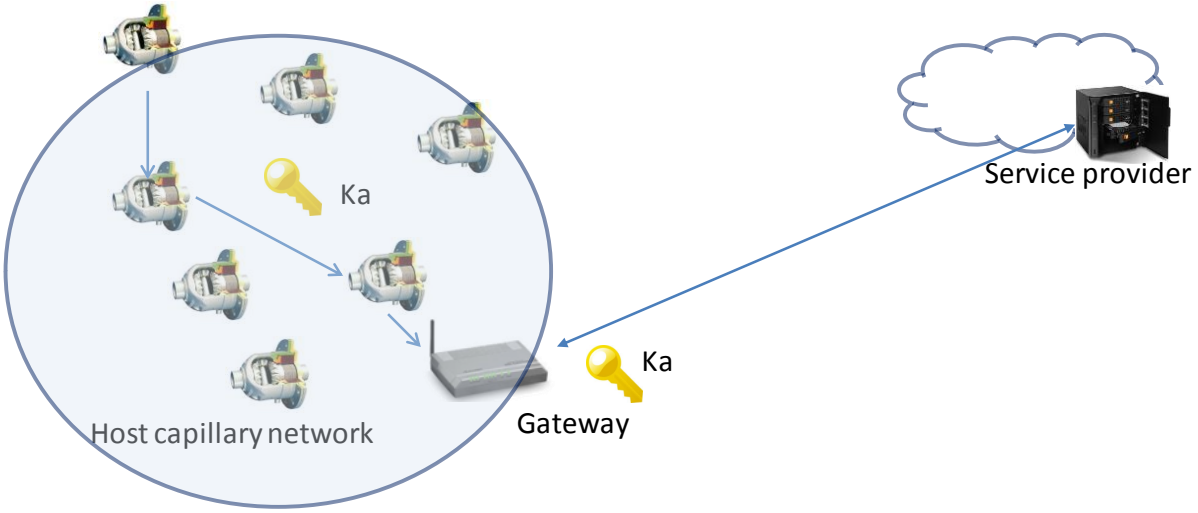

Figure 2-21: security scenario involving the same application key on the LAN and on the WAN side

One advantage of this scheme lies in the fact that a single encryption layer which will secure the whole transmission path. It removes the need for data re-encryption at the gateway. However the size of the Ka key and the cryptographic algorithms used must be compatible with the computing power available in the capillary devices.

Another advantage is illustrated on Figure 2-22. Typically, securing a capillary network involves a first level of security at the network access layer, possibly superimposed if required by a second security layer at the application level. This creates an overhead that may be significant when dealing with energy or computing powered constrained devices. The extension of the application layer security behind the gateway can possibly lead to remove the network access security.

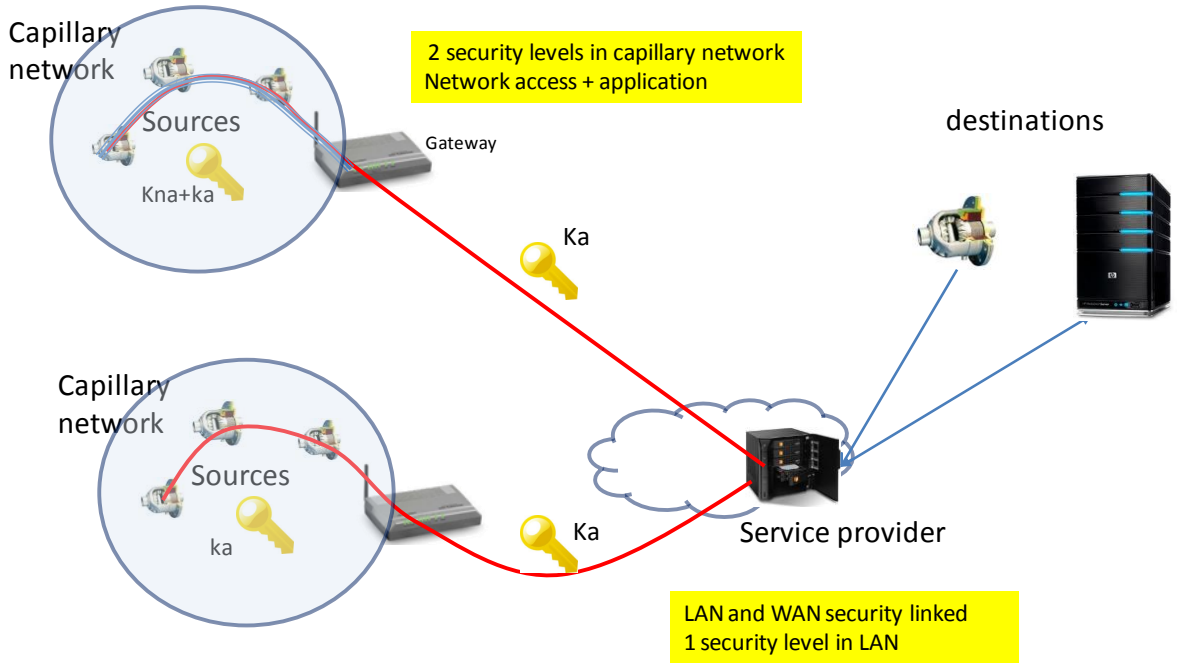

Figure 2-22: Bridging LAN and WAN security

There would be two benefits to this:

- The network access security management in a capillary network is typically under the responsibility of the capillary network owner. The management task can quickly become complex, particularly when a lot of capillary devices are involved. Linking WAN and LAN security opens the possibility for the M2M service provider to remotely manage LAN security as a service, thus relieving the owner of the capillary network from this task.
- The computing overhead associated to cryptographic computation is reduced in the capillary network as there is only one security layer. This can be a bonus for energy constrained devices.

We will see in Section 3.3 that this principle may be used to enable aggregating 2 capillary networks belonging to distinct owners.

Another approach can be possible for bridging LAN and WAN security consists in extending network access (MAC layer) security to the WAN side. This method involves a payload reformatting at the level of the gateway. This approach has not been investigated here.

2.6 Pairing via hardware fingerprint

Section 2.5 identified different roles that may be played by a gateway to secure the connection of a capillary network to a wide area network. In the first role (funnel), the gateway will let each capillary device free to connect to a remote M2M server (after a security bootstrap phase) with its own identity. In the second role, the gateway will be the single point of contact of the capillary network, and will perform its own security bootstrap with the remote server. This proxy role will most likely hide the identity of the devices located behind the gateway.

We describe here the use of PUF (physical Unclonable Functions), exploiting tiny hardware differences (linked to the manufacturing process) to create a kind of hardware biometry, to achieve a security bootstrap between a low cost capillary device and a remote M2M server along with two possible methods to perform an initial security pairing.

The concept of PUF was introduced by Pappu in 2001 [67], [68].We mentioned the interest of PUF in a previous EXALTED deliverable D5.1 [43]. A PUF is implemented with the help of an object. The object, when challenged with a challenge Ci provides an answer:

$$R_i = PUF(Ci)$$

The PUF results from the internal structure of the objects including uncontrollable differences with similar objects resulting from the imperfections in the manufacturing process. Due to this a PUF is very difficult to reproduce.

In 2007, Guajardo and al. [69] introduced the notion of Intrinsic PUF (IPUF), a PUF intrinsic to a device. They showed that the start-up values of SRAM memory cells (present for example in an FPGA) are an IPUF. Due to deep-submicron variations such as doping variations in transistors, some bits of memory will after power up end up initialized in a One state while others will end up with a Zero state. Repeated power up operations will not usually lead to the exact same power up memory pattern, but the use of ”fuzzy extractor processing” applied to those memory patterns can lead to the generation of unique keys, characterizing the hardware which can be steadily reproduced upon multiple power up of the device [66].

Typical implementations of Intrinsic Hardware security involve a software component executed by the bootloader of the device. This component allocates a piece of the RAM memory for its exclusive use and characterizes this memory segment resulting in the derivation of the unique hardware intrinsic key. This key can be used to remotely authenticate a device after an initial security pairing (or bootstrapping phase).

PUF technology has been discussed for a number of years, but this technique has been recently the focus of a renewed attention, due to its potential to provide a solution to a new class of requirements appearing in vertical industries such as for example the need to tie the execution of a software component to a specific piece of hardware. We can now find on the market commercially available PUF solutions.

PUF provides a very economical way to authenticate devices in the Internet-of-things, using their unique fingerprint. This is due to the fact that unlike other security techniques, PUF does not require ”pre-provisioning” of secrets in the device, as the device secrets are created from the device hardware as explained below. The use of PUF to secure communication within wireless sensor networks has been discussed in the literature [69], [70] and different methods have been proposed to bootstrap the security with PUF, with or without a trusted third party.

Another advantage of those techniques often referred to as “Intrinsic Hardware security” is that the credentials are created at power up can only be used when the device is powered. No key is present in the power down state.

We describe in the next section one method to use PUF technology to perform security bootstrap between a M2M device (it may be a small capillary device located behind a gateway) and “a server”. The method is relying upon the presence of an OoB channel used to transmit a passphrase protecting key exchange information. The same method can be used with symmetric or asymmetric cryptography.

2.6.1 Key pair generation of asymmetric keys

This method involves the on-board generation of a pair of asymmetric keys (private and public key) from the hardware intrinsic key. Usually public key cryptography is avoided in small sensor devices because it is expensive in terms of program space and computing power. However the use of Elliptic curves algorithms makes public key cryptography more useable in low end devices.

The device will retain the private key and will send the public key to the server in a protected way. For this, it is possible to use traditional methods. One such method is to establish a DTLS connection between the device and the server, protected with only a server side certificate. However, in order to avoid the possibility of man-in-the-middle attacks, the device should have the possibility to check the validity of the server certificate, and this is not always possible.

Another solution is to transmit the public key to the server, protected by a passphrase. The sequence diagram of this scheme is pictured on Figure 2-23. The passphrase is communicated by the device to the server by an out-of-band (OoB) channel.

Resorting to an OoB channel, as mentioned above, is a popular method to help the authentication of information exchanged in the insecure wireless channel. Most proposed OoB channels rely on some form of human user participation. Documented examples of OoB can be found in [15], [16], [20], and [21] and using short authenticated short strings in [39].

The user involvement may be reduced. For example a telephonic OoB channel may be used. Once the call is setup, the device would send touch tones describing the passphrase through a speaker and through the handset. This method implies the possibility to move the device close to the handset.

Other methods involve communicating the passphrase from the device to the user and having the user rekey to the server, maybe through a web interface to that end:

- The passphrase may be displayed on a display attached to the device (when the device includes a display and the display needs to be accessible by the user)
- The passphrase may be communicated to the user by a sequence of LED flashing. For example a few bursts of lights. This implies the availability of a light on the device (and the LED should be in the user line of sight)
- The passphrase is communicated to the user by audible bips and the device should be within hearing reach of the user

The user may then communicate the passphrase to the server. We have mentioned the option of a web interface and the option of a telephonic interface; we will add to these the possibility to resort to the SMS channel.

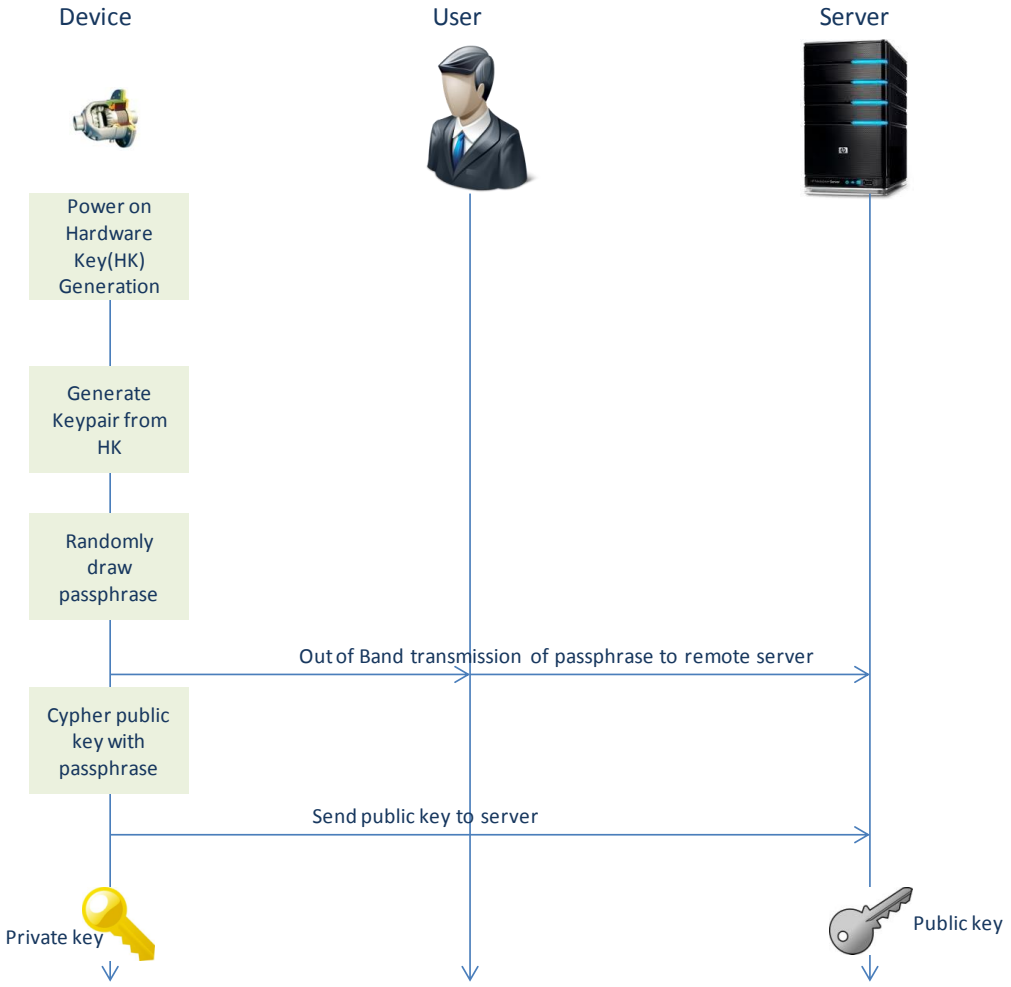

Figure 2-23: hardware security bootstrap with asymmetric cryptography

The server, after receiving the passphrase is able to recover the public key. At this point the server and the device may protect their communications by using the key pair.

At next power up operation, the device will need to regenerate its private key using the intrinsic hardware key in order to authenticate successfully with the server.

A variant of this method is to use the hardware intrinsic key to cipher the private key prior to store it in the non-volatile (flash) memory of the device. At power up, the hardware intrinsic key is used to decipher the private key which is itself used to authenticate the device and secure its communications with the server

2.6.2 Use of symmetric keys

The methods described above to protect the transmission of the public key to the server may also use to protect the transmission of a symmetric key derived from the intrinsic hardware key. At the end of this session, the device and the server share the same symmetric key and may use it for authenticating and securing their communications. The sequence of operations associated to this method is shown on Figure 2-24

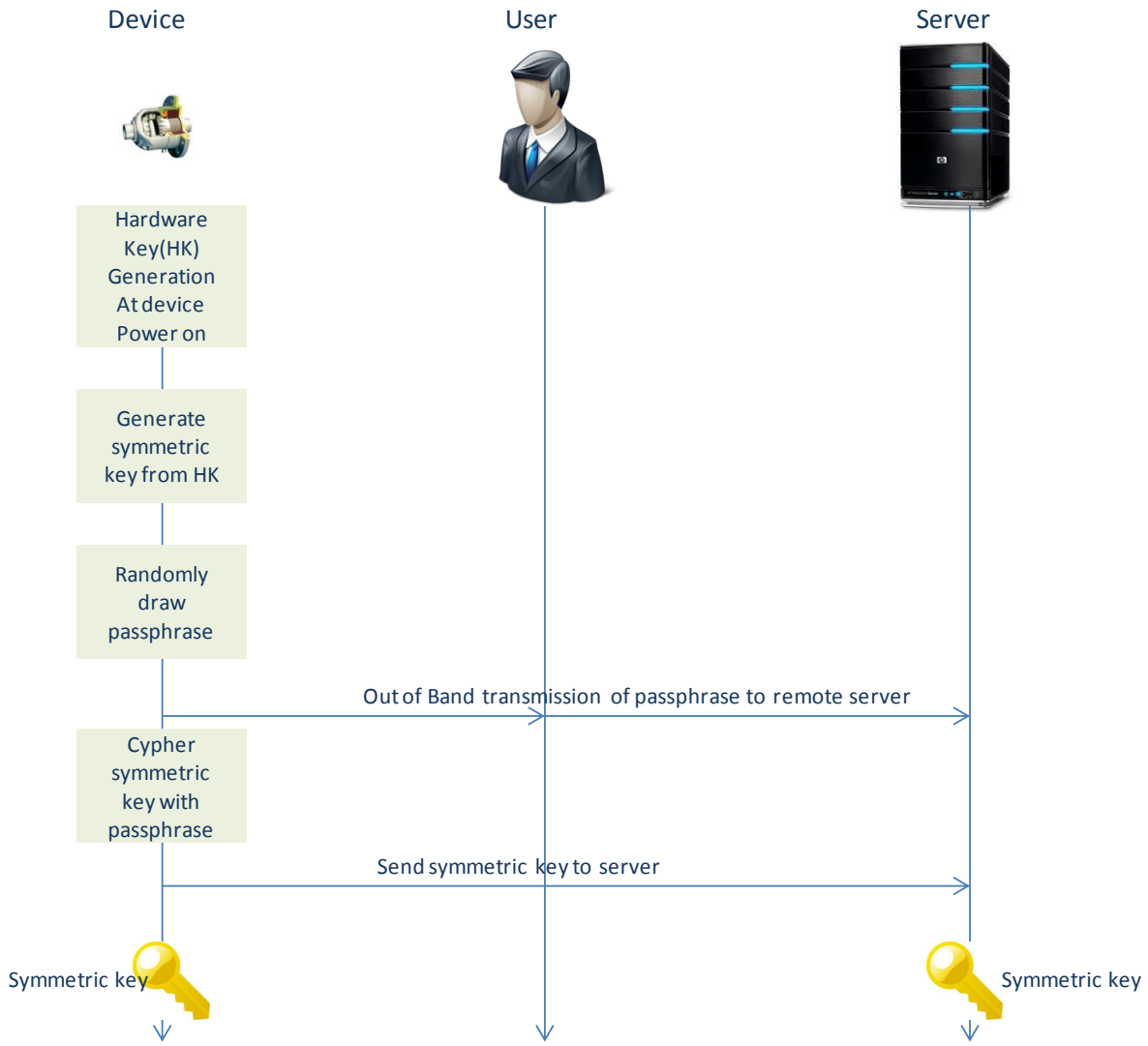

Figure 2-24: Hardware security bootstrap with symmetric cryptography.

3 Relaying in infrastructure networks

After considering the security of isolated capillary networks, and securing the connection of a capillary network to a wide area network, we now investigate the use and the security of relaying in 3GPP and more generally in wide areas IP networks.

After discussing the use of relay nodes in LTE network, we will focus upon a different aspect to understand how multihop capillary networks may be bridged with the help of an infrastructure network.

3.1 Threats related to relaying in infrastructure networks

Relaying involves transmitting the data of a device through adjacent nodes in order to reach a final destination node. This is typically called "multihop".

Generic threats related to multihop networks fall into the following categories

- Impersonation of a relaying node to send transmitted data on its behalf (so compromising authentication of the node).
- Hijack of a relaying node. An attacker takes control of an existing relay node and manipulates its functionality (the effect is similar to impersonation of a relaying node).
- Impersonation of devices that may be using a relay node. An impersonated or hijacked node inserts spoof traffic appearing to originate from other devices.
- Re-routing attacks. An attacker change the destination address of traffic, or sends an additional copy of traffic to an attacker address, or forces the routing path to go via an attacker device or address.
- Man-in-the-middle attack, where the data from source devices may be spied upon or modified without the user being aware of it (so compromising confidentiality and/or integrity of the data transmitted). Relay node impersonation or hijacking attacks or re-routing attacks may all be used to set the stage for a Man In The Middle attack
- DoS attacks, preventing multihop communications to occur (so compromising availability of the network). Again, relay node impersonation or hijacking or re-routing attacks may be used for this attack e.g. data is diverted into a “sink” node, and then never leaves.

The attacks are described against the end user data, but they can also be executed against signalling data, for example signalling data that is used to distribute keys or control the topology of a relaying network.

Such attacks are possible against existing infrastructure networks as data travels through intermediates. However, a particular concern about devices that are explicitly called “Relay Nodes” is that there are likely to be many of them, the nodes are likely to be in much more vulnerable locations (not secured in data-centres for instance), and finally the nodes may not be owned or explicitly controlled by the infrastructure provider. So the relaying environment could be described as a “hostile” environment.

We will provide below a more specific description of the threats associated to relay nodes in LTE (4th generation 3GPP) networks.

3.2 Device-To-Device communication over LTE

Wireless communication over licensed spectrum e.g. GSM, UMTS, LTE etc., are infrastructure based technology. In such communication systems, a device will communicate with one or multiple other devices via the radio network and core network of an operator. In disaster situations such as after an earthquake, a tsunami, a perfect storm, or meteorite

strikes, such infrastructure-based communication could be un-operative. In other special events the infrastructure could be overloaded or so extremely congested that rescue or public safety organizations cannot communicate via any such wireless terrestrial networks.

In the midst of summer 2011, the Federal Communication Commission (FCC) selected LTE as a Public Safety Technology. Several requirement papers were presented to the 3GPP organization (by e.g. Alcatel Lucent, NIST, Nokia Siemens Networks and US Cellular) that express the need for an infrastructure-less or minimal infrastructure support type of communication, using proximity services for LTE radio technology. The use cases assume the total or partial loss of radio coverage due to disaster. It is required that in such situation, public safety team members should still be able to establish direct radio links with nearby members or repeaters that has reliable access link to the infrastructure based cellular access network.

Other use cases were also proposed which could also benefit from direct communication such as for proximity advertising, social networking, multi-player gaming, proximity offloading and so on. All the use cases are based on proximity and direct communication between devices over LTE technology.

3.2.1 Challenges for cellular operators

Though mobile operators can see the new opportunities that such technology development would bring, they also perceive challenges linked to such a direct communication such as control and management of device-to-device communication and interference with existing infrastructure-based LTE communication.

During the numerous debates and discussions in 3GPP working groups, two main aspects of proximity direct communication were highlighted:

- Proximity detection
- Direct communication management

These two aspects can be treated independently. Proximity detection technology could be network based or device based, and was left outside of 3GPP scope.

The challenge of direct communication consists in how the operator could manage and control the activity of the devices using its licensed spectrum. Consequently, the original requirement of public safety whereas communication is performed between devices without the control of the infrastructure, was left for later consideration. Direct communication will be considered first for the user plane data transfer while the signalling or control plane data will still go through the operator's infrastructure.

3.2.2 3GPP relay nodes

3GPP in Release 10 has already completed a work item for "Relay Nodes" (RN) to assist LTE coverage. Effectively a Relay Node acts as a base station (eNodeB) towards the connecting device, but as a UE (user equipment) towards the "donor" base station (DeNB). The work item resulted in the creation of TR 33.816 [44] and a number of CRs (Change Requests) to existing specifications, particularly the LTE Security Architecture TS 33.401 [45], and especially Annex D.

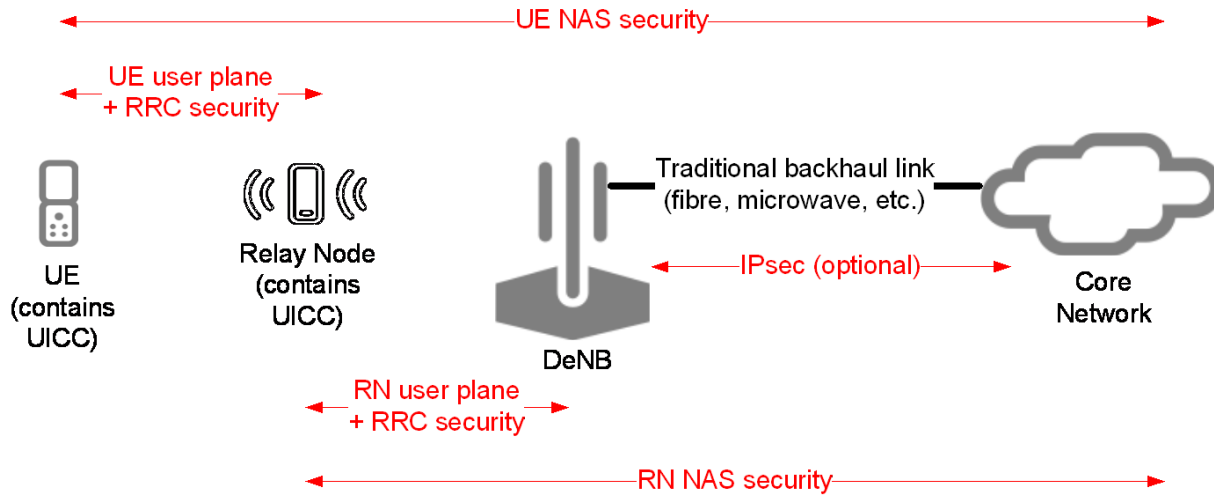

Figure 3-1: Relay Node Architecture in 3GPP Release 10

There have been no further changes requested for Release 11, so the specification can be considered mature. It therefore provides a useful reference model for relaying in EXALTED.

One important observation is that 3GPP Release 10 solution is single or dual hop i.e. it supports only one RN between UE and DeNB. It is not possible to introduce multihop without major architectural changes. So far, 3GPP hasn't seen a requirement for multihop over LTE. The latest EXALTED architecture also does not require multihop over LTE-M (though it does require use of multihop in capillary networks).

TR 33.816 [44] documents a number of detailed threats against Relay Nodes as well as solutions addressing these threats. The most important considerations are new attacks on the Relay Node itself, or attacks which attempt to impersonate the Relay Node (to the UE, or to the DeNB, or both).

As the relay architecture (Figure 3-1) is based on the already existing LTE architecture, the following assumptions are made when analysing the security threats to the relay architecture:

- A removable UICC is inserted into the RN to provide authentication between itself and the network to establish the bearer(s).
- Access Stratum (AS) level encryption is switched on between the RN and DeNB.
- The DeNB will have some secure environment that is assumed that an attacker will not compromise
- Everything from the DeNB upwards (towards the network) is secure and will use macro network security mechanisms (such as NDS/IP).

These assumptions are made purely for the purposes of understanding the security threats and any solution is not restricted to follow these assumptions.

The introduction of a RN into the network introduces some new security threats to E-UTRAN, which are listed in Appendix A. The general protection against attacks on the RN is physical and logical hardening, including use of the UICC hardware to store critical key material. The general protection against impersonation attacks is a form of device authentication of the RN.

However, 3GPP networks are not designed to authenticate devices; they are designed to authenticate subscribers (through the Universal Subscriber Identity Module (USIM) application on UICC). The solution that 3GPP adopted is a form of indirect device

authentication by binding the UICC into the RN, and preventing it being used in a different device (a “rogue” RN). Since Release 10 predated the work on embedded UICC (as discussed in D5.1), a cryptographic binding was chosen, using a cryptographic secure channel between a (physically removable) UICC and (authentic) RN. Physically soldering the eUICC into the device would also address most of the threats, though there would still be a risk of a de-soldering attack.

3.3 Infrastructure assisted bootstrap enabling capillary networks interconnection,

This section identifies a number of scenarios that will create a synergy between the M2M infrastructure network and the self-organized capillary network. The focus here is different from the one in section 2.5 where devices in the capillary network connect via an M2M gateway to an M2M service provider. In this section, the devices belonging to one user will be able to channel their communication via the devices belonging to another user in order to achieve global ad hoc coverage.

Two scenarios will be envisaged along these lines:

- Single device having its data relayed by a guest capillary network, after a phase of security bootstrap in its own network. As an example use case from EXALTED D2.1 [64], M2M devices located within capillary networks inside moving vehicles, may send data to the global public network by multihopping through the capillary networks of adjacent vehicles.
- Bridging two capillary belonging to different users. This means that capillary devices of the first network will have their communications relayed possibly by nodes of the second network using proximity communications. In EXALTED D2.1 [64], one example of related use case is given by the ITS scenario providing gateway vehicle functionality for Car-to-Car (C2C) communications where pre-collision information is sent instantaneously in peer-to-peer mode from the colliding vehicles directly to other vehicles which may be subject to the further shockwave propagation.

3.3.1 Single capillary device connecting to a guest capillary network

This scenario is outlined on Figure 3-2. The capillary device 1 represented on this figure has already performed a pairing process in its home capillary network (capillary network 1). The gateway in this home capillary network acted as a group leader in the pairing process and registered as a gateway with a M2M service provider. The capillary network 2 represented on this picture is another capillary network belonging to another user affiliated to the same M2M service provider.

As a result, device 1, when placed in a mobility situation, will be able to route its data via the self-organized capillary network 2. This can be compared somehow to the problem of roaming between base stations in a 3G or 4G network. In our case however device 1 has probably lost its network connectivity via capillary network 1 before restoring it via its connection to capillary network 2. For the sake of simplifying the discussion, we will assume that the owners of capillary networks 1 and capillary network 2 are affiliated to the same M2M service provider.

The solution described below relies on typical authorization architecture and a delegation scenario resulting in device 1 being able to multihop through capillary network 2. The high level flow of operations may be summarized as follows:

- The gateway of capillary network 1 is authenticated by the M2M service provider and obtains a signed certificate proving the business relationship existing between the owner of capillary network 1 and the M2M service provider.
- Device 1 generates a key pair (public and private key). The gateway signs the public key and delivers a certificate to the device.

- Device 1 when in a mobility situation wants to connect to foreign capillary network 2 through its gateway and authenticate using this certificate and create a security association with the gateway.
- The gateway verifies the chain of trust using the certificate and recognizes that the device enrolment was performed in a network affiliated to the M2M service provider. The fact of being able to check the credentials locally without resorting to a data exchange across the WAN may be critical if connection to the foreign or guest network need to occur quickly (consider a vehicle passing a road-side beacon at speed), so this is quite similar to the problem of performing fast handover in mesh networks as described for example in [65].
- Device 1 is granted access to capillary network 2 and is delivered the group key needed to use relaying in this network.

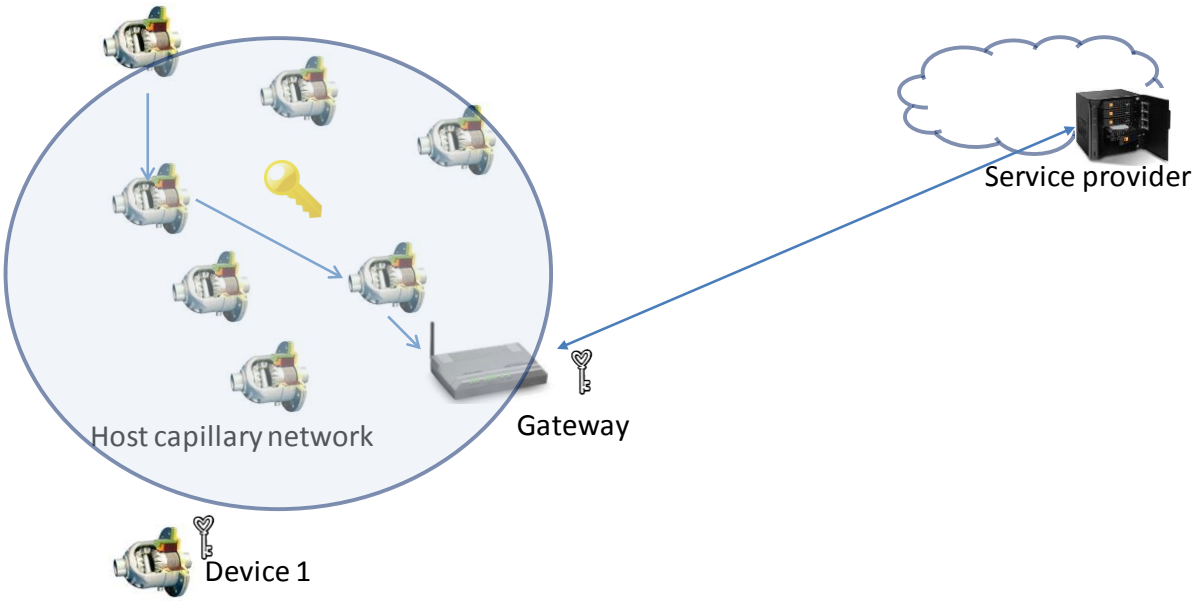

Figure 3-2: Single device connecting to a foreign capillary network

Figure 3-3 describes the flow of operations leading capillary device 1, originally part of its own capillary network (capillary network 1) to be able to communicate when place in a mobility situation through capillary network 2 may be decomposed in 2 stages:

- Operations occurring while device 1 is still within capillary network 1
- Operations occurring when device 1 is in a mobility situation

1. When device 1 is within capillary network 1 :

   a. The gateway of the capillary network 1 performs a bootstrap operation with the M2M service provider using one of the security bootstrap methods described in [43] and this bootstrapping process results in the definition of a shared root key Kr.

   b. The gateway generates a pair of asymmetric keys, creates a certificate signing request and ultimately obtains a certificate from the M2M service provider. Different solutions exist to achieve this. ETSI M2M workgroup has specified different security bootstrapping methods capable of achieving this goal. The use of GBA is one such solution. Performing security bootstrap using TLS or DTLS is another.

c. The device takes part in the pairing process at the level of the capillary network. The gateway plays the role of the group leader in the pairing process. The pairing method used in this case is chosen among the methods enabling the publication of devices public keys. At the end of this process the gateway own a copy of the device public key and creates a signed certificate using the certificate it obtained in step 2. The public key of the device may then be used to distribute a shared group key enabling the capillary devices to relay their communications in the capillary network.

2. When the device is in a mobility situation, outside its home capillary network

a. Device 1 then may want to connect to a guest capillary network. To do so, it performs a pairing operation with any of the devices of the guest network, presenting its signed public key to the device. At this point Three scenarios may be envisaged:

i. The host device may be able to check the signature (and verify the PKI chain) locally and decide on granting access to the guest device. However if one way authentication is feasible in this case (the host device authenticates the guest device), mutual authentication is more difficult to achieve as it involves the host device to own its own signed credentials to be successfully authenticated by the guest device

ii. The host device may resort to the local gateway for granting access to the guest device. The decision is taken locally at the gateway level. Mutual authentication is possible if the gateway presents its own signed certificate obtained from the M2M service provider.

iii. The host device resorts to the gateway to verify the signature locally and then communicate with a remote authorization server, which will ultimately grant or deny access. Here again, the gateway may be authenticate by presenting its own certificate.

b. Once access is granted using any of these 3 methods, the guest device obtains a copy of the shared group key used for relayed communications in the capillary network and is then able to relay its communication via the other devices in the network. Another alternative could be for the guest to obtain a guest key with limited privileges, and simply enabling data relaying.

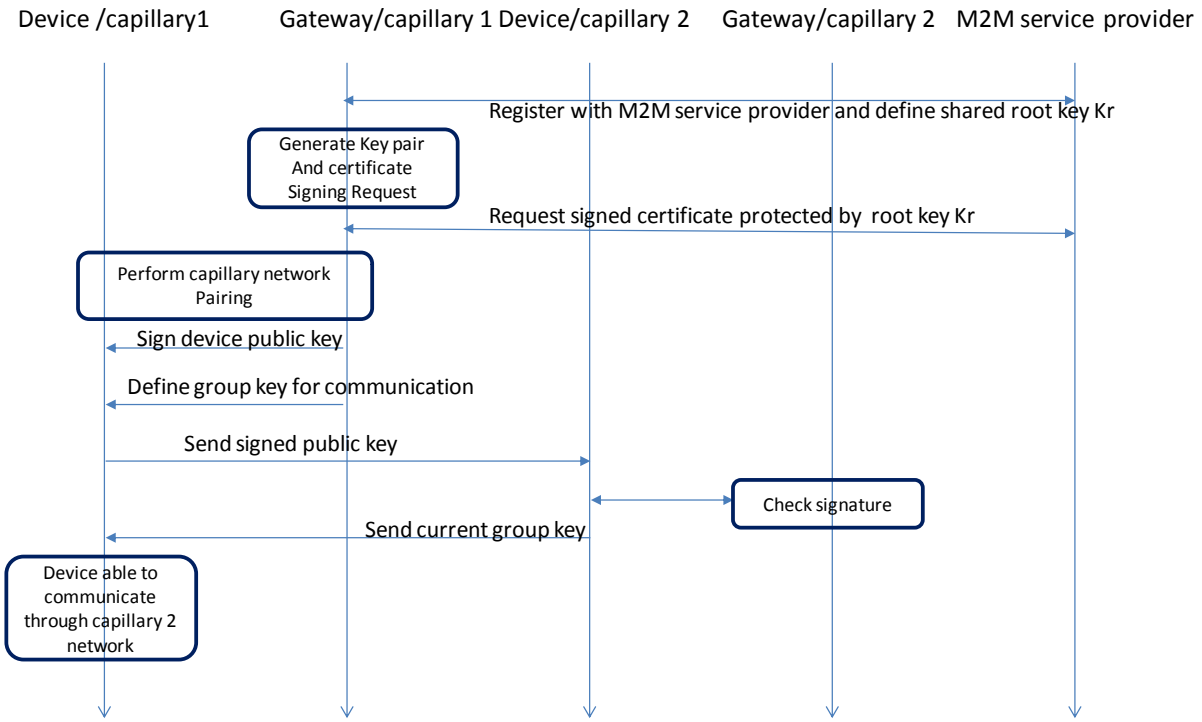

Figure 3-3: Workflow for single device connecting to a foreign capillary network

3.3.1.1 Discussion

The method described above considers data protection at the application level. There is an underlying implicit assumption that the gateway of the host network would be reachable by the guest device. The network access layer is not protected, because in this case, the guest device will not be able to communicate with the gateway.

The workflow defined here does not take into account the definition of security policies. We believe it could be generalized in order to describe more general security architecture, featuring clearly defined policy decision points (PDP) and policy enforcement points (PEP). Such policies could be envisaged at several levels:

At the level of the guest network:

- Policy defining the acceptance of guest request (for example a gateway may be accepting guest requests from 5pm to 7pm)
- Policy definition conditions for granting access (who can be allowed as a guest)

At the level of the connecting guest:

- Policy defining connection attempts (for example, only try to connect to networks affiliated to a specific M2M service provider)

3.3.2 Aggregation of capillary networks

This section describes a second scenario involving capillary networks interconnection. This scenario is depicted on Figure 3-4 and involves direct communication between devices belonging to nearby capillary networks. Thus, as shown on Figure 3-4 , device 1 and Device 2 are able communicate via proximity networks without routing all their communications through gateways 1 and 2 and the wide area network.

Alternatively, the devices of one capillary network could possibly multihop through the devices of the other capillary and send their data through their gateway.

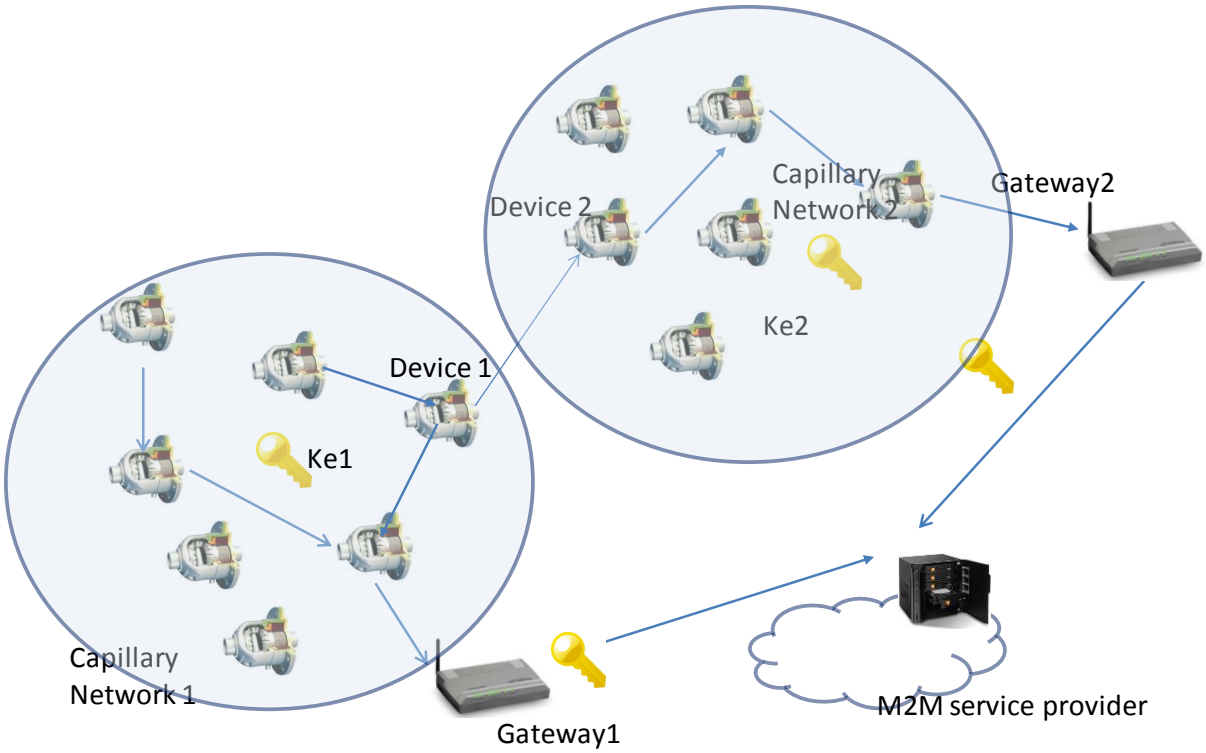

Figure 3-4: bridging capillary networks via M2M infrastructure network

This aggregation scenario can be implemented in two ways:

- Devices in each of the two capillary networks have the possibility to switch to the other capillary network. To do so, they need to obtain the group key used in the "host" capillary network. One solution for this is to use the method described in section 3.3.1 above. Once a device has switched to the other capillary network, it is not able anymore to multihop on its own capillary network. This scenario is similar to the roaming we experience in 3GPP network, where we have only a single connection to a network active.
- Devices may have the possibility to hold 2 group keys. In this case they could communicate at the same time on both networks. Thus device 1 shown on Figure 3-4 can multihop with other devices on capillary network 1 using group key ke1 and with devices on capillary network 2 using group key ke2.

3.3.3 Business perspectives

Two scenarios for bridging capillary networks have been described. The first one involves a guest connection from a capillary device to a guest network while in a roaming situation.

A possible associated business scenario involves an Internet service provider (ISP) offering its subscribers the possibility to connect temporarily their M2M device(s) to the multihop capillary network of other subscribers.

A similar scenario has already be explored in a different context by some ISP[2] in order to provide their subscribers guest WiFi access via the internet boxes of other subscribers when in a mobility situation. The proposal described in Section 3.3.1 is a transcription of this idea to the case of M2M communications, and the use of multihop networks.

[2] Free Mobile, France

The business rationale behind the scenario described in section 3.3.2 is quite similar: One major advantage of multihop networks lies in their self-organization capabilities and their efficiency in spreading the data transmitted.

The possibility to multihop securely through an adjacent capillary network belonging to another user can increase the number of candidate's multihop paths and make multihop even more attractive. Thus, an Internet service provider could optionally offer its subscribers the possibility to multihop their communications over adjacent subscribers capillary networks in order to improve multihop efficiency. To our knowledge, this type of scenario has not yet been deployed commercially, but could be considered by some ISPs in the future.

4 Standardization activity

The purpose of this section is to provide an overview of on-going standardization work related to communications relaying.

4.1 Current work in 3GPP

Since summer 2012, 3GPP requirement group (SA1) has been putting all efforts on a work item for Proximity Services (ProSe). The work item has been pushed by Qualcomm with initially the support of US Public Safety, and now by also US Department of Commerce and Home Office (UK Gov).

The objective was to provide a first study item that includes a list of potential service requirements for ProSe by end-2012 and followed few months after with the requirement specifications by March 2013.

A first set of requirements relative to ProSe security and definitions has been agreed during January SA1 meeting in Prague. Additional meeting for discussing the subject has been agreed and will take place in San Diego in April 2013. The additional meeting will allow members to progress on ProSe as well as Group Communication over LTE work item.

Initially, the Device-to-Device communication work focused on Public Safety needs and environment. However, new use cases were rapidly added to cover commercial/public use cases: Social gaming, advertising and so on. 3GPP operators however prefer to restrict the work to the initial Public Safety communication use cases, with specific radio/handsets.

In second half of 2012, one set of the requirements of Public Safety was spinning off the ProSe work. These requirements are mostly related to group communications: the capability of a ProSe handset to communicate with a group of other ProSe handsets or to all other handsets. Nokia Siemens Network is now the editor of the Group Communication Service over LTE study item document (GCSLTE). The document is expected to be released by May 2013. The requirement document is expected by end of 2013.

In 2013, it is expected that the architecture group in 3GPP (SA2) and the security group (SA3) work on the subject. Public Safety organizations are pushing the subject to be a high priority subject, and provide all resources to make it happen in 2013 (SA2 ad hoc meeting in San Diego collocated with SA1 in April 2013)

New needs/use cases have been recently added covering the relaying function in ProSe communication, involving a ProSe handset behaving as a relaying node for other ProSe devices. Requirements pertaining to this new feature will be included in 2013 requirement document.

4.2 Current work in IEEE 802.16's Relay TG

The standard IEEE Standard 802.16j 2009 [56] on Multihop Relay Specification has been developed within the IEEE 802.16's Relay Task Group. Multihop relay (MR) is an optional deployment that may be used to provide additional coverage or performance advantage in an access network. In MR networks, the base station may be replaced by a multihop relay BS (MR-BS) and one or more relay stations (RS). Each RS is under the supervision of an MR-BS.

In MR system, RS uses the same security architecture and procedures as a subscriber station (SS) to provide privacy, authentication and confidentiality between itself and the MR-BS.

4.2.1 Security zone

In order to satisfy requirements of multihop relay system operation, MR-BS and a group of RSs in MR cell maintain a set of trusted relationships, called Security Zone (SZ). A SZ is a group consisting of one or more RSs and the MR-BS that share key material for the protection of MAC (Media Access) management messages produced and processed by members of the group.

4.2.1.1 Keying material

The primary keying material is SZK (security zone key) and SZKEK (security zone key encryption key), which are provisioned by the MR-BS. SZK and SZKEK are randomly generated at the MR-BS. The SZK is a head of key hierarchy used to satisfy the security requirements such as integrity protection for MAC management messages within a defined security zone. SZKEK and SZK are encrypted by an RS's KEK and SZKEK respectively and transferred to an access RS during authorization phase, and updated periodically via relay multicast rekeying algorithm. In case an RS (i.e., MRS) leaves the security zone, all security zone keys are updated.

4.2.2 Security modes

The standard specifies a security sub layer for authentication and secure key exchange. Two main protocols are used in this security sub layer: an encapsulation protocol for encryption packet data across the broadband wireless network and a Privacy and Key management Protocol (PKM) providing secure distribution of keying data from MR-BS to SS. The PKM protocol use either Extensible Authentication Protocol (EAP) or X.509 digital certificates together with RSA public-key encryption algorithm to carry out key exchanges.

The security sub layer defines two different security modes:

- **The centralized security mode:** this mode is based on key management between an MR-BS and an SS. The security association is established between SS/RS and MR-BS without the involvement from the intermediate RS. The RS does not try to decrypt the user data or authenticate the MAC management message it receives from the SS, but simply relays it. All the SS-related keys are stored and maintained at the SS and MR-BS, and RS does not have any key information associated with the SS. The intermediate RS authenticates management messages it receives from other RSs using relay-specific shared keys.
- **The distributed security mode:** this mode incorporates authentication and key management between an MR-BS and an access RS and between the access-RS and an SS. RS can be configured to operate in distributed security mode based on its capability during the registration process. The authentication key established between SS and MR-BS is distributed to this RS. An RS operating in this mode relays initial PKM messages between the MR-BS and SS/subordinate RS. Upon master session key establishment, access RS securely acquires relevant Authorization Key of the subordinate RS/MS from the MR-BS. Using PKM protocol, the access RS can derives all necessary keys.

4.3 Current work in IETF 6LoWPAN WG

6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) is an active working group in the internet area of the IETF. It aims to enable networks with low-power devices as IEEE 802.15.4-based networks to send and receive IPv6 packets. The 6LoWPAN group has developed a base specification in RFC 4944. The specification defines encapsulation and header compression mechanisms for transmission of IPv6 packets over IEEE 802.15.4 networks.

With respect to security concerns, the RFC 4919 [57] on the 6LoWPAN problem statement points out the need to provide a complete end-to-end security, even with IEEE 802.15.4 AES link-layer security modes that provide authentication and encryption of 802.15.4 frames.

The IETF Internet Draft in [55] discusses the security options for 6LoWPAN, in particular if IPSec or SeND is used. IPSec is expensive to be used in the capillary network, since it requires an extra header (AH or ESP) in every packet and an implementation of security association database at each device. The SeND protocol may be conceivable in such networks if the cryptographically generated addresses (CGAs) are based on elliptic curve cryptography (ECC) rather than RSA.

Key management is also an issue in 6LoWPAN-based networks. In particular, 802.15.4 security mode does not specify key management (presumably group oriented). Because of the intermittent connectivity of devices, a server-based key management solution is not practical for these networks. The use of public key certificates is unendurable by limited computation and energy resources 6LoWPAN devices; though, ECC has proven its feasibility for sensor networks. Probabilistic key pre-distribution schemes whereby keys (network key, group key, pair-wise keys) are pre-distributed at devices prior to their deployment, are the most practical solutions for 6LoWPAN networks.

4.4 Routing in ZigBee standard

The ZigBee Alliance[3] offers two specifications: the core ZigBee specification that defines energy-efficient mesh network and ZigBee RF4CE specification for simple, two-way device-to-device control applications that do not require the full-featured mesh networking capabilities.

In the core ZigBee specification, ZigBee supports star networks where a coordinator device (that is generally a trust centre) provides trust management, network management, and configuration management. The core ZigBee specification supports also the use of ZigBee Router (ZR) devices to extend the communication at the network (NWK) level. The ZigBee NWK layer provides routing and multihop functions needed for creating different network topologies; for example, star, tree, and mesh structures.

The ZR is an IEEE 802.15.4 FFD (Full-Function Device) participating in a ZigBee network, which is not the ZigBee coordinator but may act as an IEEE 802.15.4 coordinator within its personal operating space, that is capable of routing messages between devices and supporting associations.

ZigBee specification builds on Link layer security provided by 802.15.4 frame protection. It relies on three different keys: link key, network key and master key (network-wide) that are pre-installed (for example, during factory installation), transported or established through:

- SKKE (Symmetric Key Establishment)
- CBKE (Certificate-based Key Establishment)
- ASKE (Alpha-secure Key Establishment)

Security amongst a network of ZigBee devices is based on link keys and a network key. Unicast communication between peer entities is secured by means of a 128-bit link key shared by two devices, while broadcast communications are secured by means of a 128-bit network key shared amongst all devices in the network.

The ZR acts as an intermediate device passing data from/to devices. The ZR is not involved with trust management at devices; it simply relays packets between devices and the trust centre without processing them. For instance, for network admission, a ZR is responsible of relaying device join requests and the network authentication exchange between devices and

[3] http://www.zigbee.org/

the trust centre. It is also responsible of securely informing the trust centre of the status (i.e., joined/left) of devices associated with it.

On 2009, ZigBee alliance made an announcement [58] that the future ZigBee specifications will incorporate IETF IP networking stack for IEEE 802.15.4 based platforms like 6LoWPAN (RFC 4944), ROLL (Internet-Draft in progress), UDP, and ICMP standards.

5 Conclusion

This report has been dealing with the security of relayed or multihop networks. In this type of networks, data transmission from/to a node is being relayed by the adjacent nodes.

Chapter 2 has investigated the security of relaying **within a single isolated capillary network**. In this case, capillary nodes relay each other's communications, either in a self-contained environment or towards the gateway, connected to the wide area network. The communication between the nodes is commonly secured using a shared group key defined and distributed through a secure pairing mechanism in a bootstrapping phase.

After identifying the security threats associated to such networks, a comprehensive survey of state of the art pairing protocols was provided. A comparison of the solutions was presented including factors such as communication and computation costs, and also human required effort to perform pairing. Several enhancements were proposed and three new group device pairing protocols were suggested. Furthermore, the application of group device pairing protocols in building manually authenticated group key agreement protocols was discussed and communication efficiency of Laur-Pasini SAS-based group key agreement protocol was enhanced.

The survey shows that the protocols could be classified into two categories:

- Those which require a special node as group leader to orchestrate the definition and distribution of the shared group key,
- Self-organized pairing protocols where all the nodes are equal.

Our comparisons showed that group device pairing mechanisms with trusted leader(s) have clear advantage to symmetrised protocols both from communication and computation cost. This indicates the importance of having a trust anchor in designing security protocols which should be considered in the design of security mechanisms

The important issue of incremental addition or revocation of capillary group members has been discussed but could justify additional work.

Section 2.5 considers the connection of a capillary network to the wide area network via a gateway acting as a group leader to achieve security pairing on the capillary side. We have shown that this situation can lead to different configurations involving distinct roles for the gateway:

- Data funnel, simply merging data from capillary devices having their own identity.
- Data aggregator, to aggregate and anonymise the traffic of nodes behind the gateway.
- Mediator in order to enable the capillary nodes to communicate securely with the external world while benefiting from an M2M security bootstrap operation performed by the gateway with an M2M service provider.

We showed that this last mode opens the possibility to "bridge" LAN and WAN security and use the same keys in the LAN and in the WAN side. We have explored this approach by considering extending application security from the gateway towards the LAN part, with application level security replacing network access security on the LAN side. This not only results in a smaller security overhead, which may be important for energy constrained devices, but opens the possibility for the M2M service providers to offer as a service to remotely manage LAN security.

The approach described here is not the only possible one. Another consists in using only network access security on the LAN side (MAC layer) and extends this security to the WAN side in the gateway. This appears to be an interesting direction to explore.

The second part of the report has examined relayed communications in the context of infrastructure wireless networks. Relaying is already standardised within LTE networks with LTE relay nodes, the purpose of which is to enhance coverage in LTE networks. We have also presented a review of the identified threats associated to this type of relay, as well as a description of the relay node security mechanisms.

For the future, 3GPP is investigating the use of relayed communication in infrastructure networks to provide self-organized network coverage in case of emergency situations. We have described the focus of these investigations.

Section 3.3 of this report has been devoted to identify possible synergies between infrastructure M2M networks and multihop capillary communications. Our goal was to investigate how an authenticated connection to an Infrastructure M2M network can be used to enable secure multihop communications **across** capillary networks. We have introduced the concept of “global” relaying whereby capillary devices are able to get their communications relayed securely through capillary networks belonging to other users, and we have described 2 scenarios of “global relaying”:

- Scenario where a single device, originally part of a capillary network gets access to a guest capillary network while in a mobility situation.
- Scenario involving aggregation of 2 capillary networks belonging to 2 distinct users, enabling devices belonging to one network to relay their communications through the other network.

Those scenarios appear genuinely interesting, and seem to carry potential for Internet-of-things applications. However, we have only explored one possibility of implementation, dealing only with application level security. A future extension of this work could involve exploring also the possibility to use network access security in the LAN and extend it to the WAN side.

Finally, the last part of this report has provided an overview of the various standardization groups dealing with multihop communications and described their focus.

Appendix A. specific threats to the LTE relay architecture

A.1 Security threats

The introduction of a RN into the network introduces some new security threats to E-UTRAN which are described here:

- **Impersonation of a RN to attack a user attached to a RN**

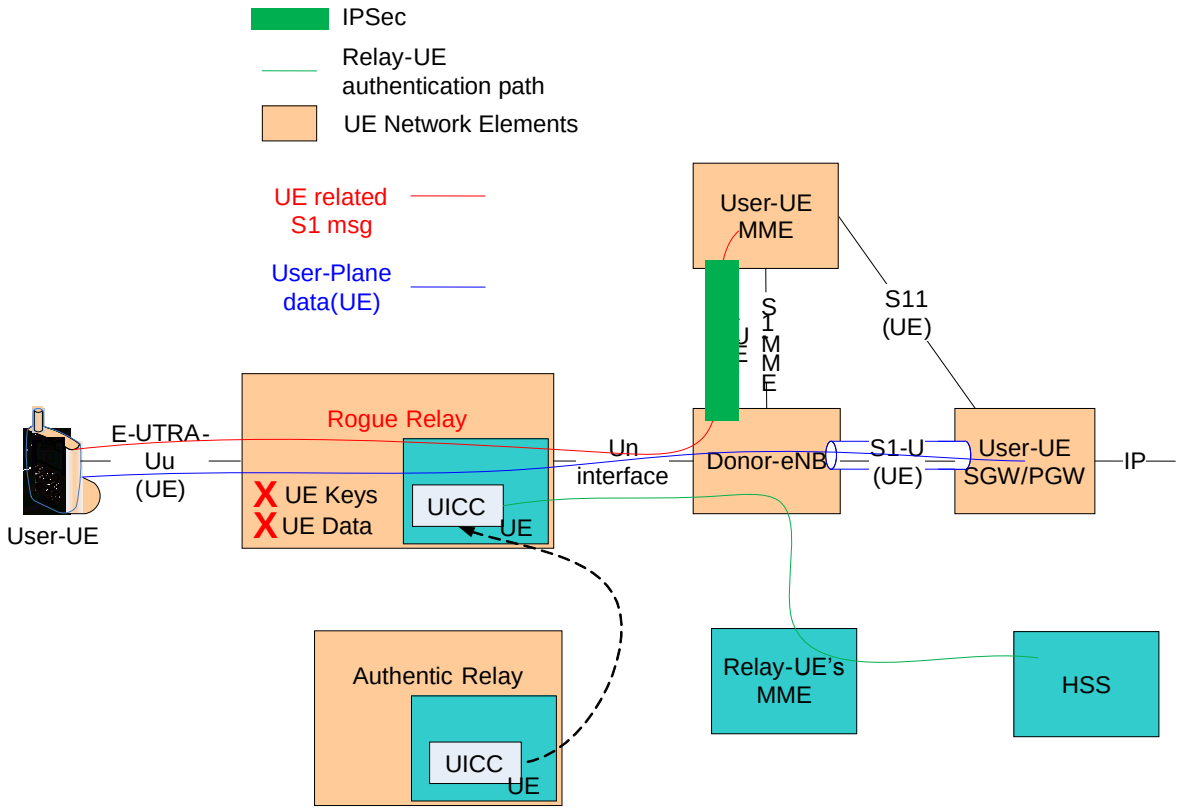

Figure A-1: Impersonation of a RN to attack user attached to RN

To perform the attack, the attacker removes the UICC from a real RN and inserts it into a rogue RN as shown by Figure A-1. As there is no authentication of the RN as a device (only the subscription that is inserted in the RN), the network cannot detect the Rogue RN, and hence keys related to the user-UE will be passed to the Rogue RN. This enables a user to attach to the Rogue RN and hence the user's security will be compromised. This shows that it is essential to perform some type of device authentication of the RN.

- **MITM on the Un interface between RN and DeNB**

This can be considered to be a variant of the above attack, but it is essential to consider as it illustrates that some care must be taken on the method of authenticating the RN device. In this attack, an MITM Node is inserted in between the RN and DeNB. This MITM node is created by taking a real UICC from a real RN and replacing it with a fake UICC for which the attacker has the root key. It also requires inserting the real UICC into the MITM node. This is illustrated in Figure A-2.

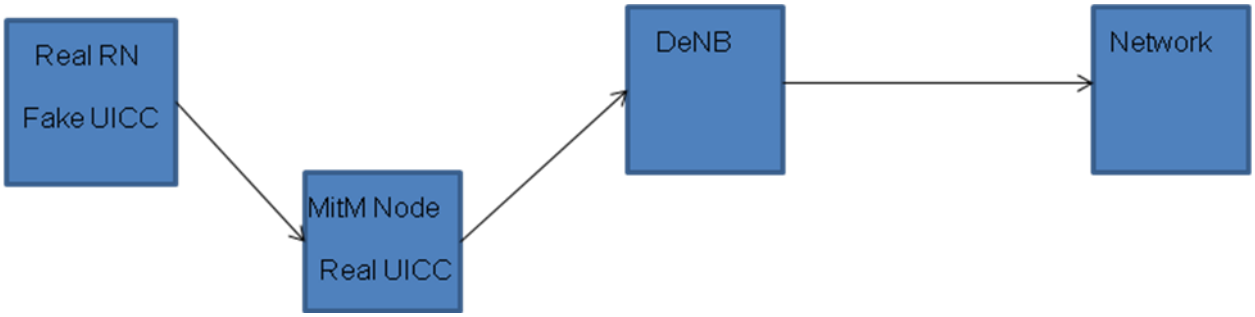

Figure A-2: Man-in-the-middle Node

The real RN will connect to the MITM node and the MITM node can connect to the real DeNB. The MITM node can transparently transmit, receive, view, and modify the traffic between the real RN and the DeNB without either of those nodes being aware of it. Hence the security of any user connected to the real RN is compromised. The MITM can eavesdrop on, modify, and inject user traffic even if the user related keys are protected by IPSec between the MME-UE and the RN. The important security point illustrated by this attack is that not only is it essential to perform device authentication of the RN, it is important to ensure that all security tunnels from the RN terminate in the real network instead of in a MITM node.

- **Attacking the traffic on the Un interface between RN and DeNB**

The interface between the RN and DeNB is based on the standard E-UTRAN air interface. This provides optional confidentiality for all traffic between the EN and DeNB, but all the non-RRC signalling traffic between the RN and DeNB is not integrity protected. The confidentiality protection could be used to encrypt the traffic on this interface, but if this security is not available for RN's node, then some other method of providing confidentiality will be needed.

If there is no integrity protection for the interface between RN and DeNB, an attacker could modify the traffic over this interface.

For user UE traffic, this would be the content as well as the protocol headers of the communication. By changing GTP protocol headers of user traffic over Un, it could be possible to redirect traffic bound for one (victim) UE to another (attacker) UE. This attacker UE would receive the data encrypted with its own UPenc key. In uplink, this may allow IP address spoofing.

While this may be acceptable for user traffic from the UE, this may not be acceptable for signalling traffic (either S1-AP or X2-AP) from RN to network. This means that either the Un interface may to enhanced from a standard E-UTRAN UE-eNB interface or some other method of protecting the S1-AP and X2-AP signalling across the Un interface needs to be used.

- **Impersonation of a RN to attack the network**

A Rogue RN (as described in Threat 1) could insert essentially four types of traffic into the network:

- A NAS signalling towards the MME-RN – the same attacks could be done with a rogue UE so are not important for the RN security analysis
- S1-AP or X2-AP signalling - this can result in DoS against the mobile network, and potentially some forms of traffic hijacking.
- Insert data on behalf of a user
- User plane traffic to get free IP connectivity

These threats could be mitigated by ensuring RN platform authentication of the RN before such traffic is accepted or being aware of such threats and mitigating them in other ways.

Before RN platform authentication has taken place, the network cannot distinguish between a RN and a rogue RN and so there is still a risk for similar attacks.

- **Attacks on the interface between the RN and the UICC**

The data that travels across the RN to UICC interface (and which is used to retrieve key material for protecting the RN to DeNB interface) is not typically protected in UE. This means that while an attacker may not be able to compromise the behaviour of a RN, it may be possible for the attacker to get hold of the keying material that is transferred across this interface. Access to these keys would provide the attacker with access to any data protected by these keys and also allow the attacker to insert data that would be protected using these keys. In particular the attacker could set up a MITM node as described in threat 2.

- **Control of the RN platform**

All traffic, apart from NAS-UE signalling between UE and MME-UE, is available inside the RN platform in the clear. So, when an attacker controls the RN platform eavesdropping and modification of this traffic is possible.

- **DoS type attacks**

When the attacker removes the UICC, RN without UICC can't be authenticated by the network. So the legal RN can't connect to network and provide services. The attacker could also insert the UICC into another RN, then the topology of access network will be changed and cause interference problem to other eNB.

- **RN stays as UE after initial attach**

In this attack, a false RN stays as UE even after RN subscription authentication by not performing detach and also not initiating the S1 interface setup procedure. As a result, the network cannot authenticate the RN as an eNodeB and the RN acts as UE to receive or request services in the network. This will lead to free charging problem even when the network knows the attached user is an RN.

- **Attacks on NAS signalling and AS traffic**

In this attack an attacker intercepts/modifies/injects messages on the UICC RN interface. In Phase I and possibly part of Phase II signalling NAS and AS traffic will be protected with keys that can be derived from information intercepted on the UICC RN interface. It is noted that the attack cannot be stopped, assuming that the RN should be able to attach as UE using legacy eNodeB and MME in Phase I.

Acronyms

3GPP	3rd Generation Partnership Project
3GPP CT	3GPP Core Network and Terminals
3GPP SA	3GPP Service and System Aspects
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
API	Application Programming Interface
AS	Access Stratum
CGKA	Cliques Group Key Agreement
CoAP	Constrained Application Protocol
CPNS	Converged Personal Network Service
CR	Change request
DeNB	Donor Evolved Node B
DH	Diffie-Hellman
DoS	Denial-of-Service
E-UTRAN	Evolved-Universal Terrestrial Radio Access Network
eNB	Evolved Node B
eNodeB	Evolved Node B
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
GAP	Multichannel Group Message Authentication Protocol
GSM	Global System for Mobile Communications
HCBK	Hash Commitment Before Knowledge Protocol
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPSec	Internet Protocol security
IPv6	Internet Protocol version 6
LAN	Local Area Network
LED	Light-emitting diode
LTE	Long Term Evolution
M2M	Machine-to-Machine
MAC	Media Access
MANA	Manual Authentication
MC-GDP	Multichannel Group Device Pairing Protocol
MITM	Man In The Middle
NAS	Non-Access Stratum
OMA	Open Mobile Alliance
OoB	Out-of-Band
PAN	Personal Area Network

PKI	Public Key Infrastructure
PKM	Privacy and Key management Protocol
PNE	Personal Network Element
PN GW	Personal Network Gateway
PRNG	Pseudo-Random Numbers Generation
RAM	Random Access memory
RN	Relay Node
ROLL	Routing Over Low power and Lossy networks
ROM	Read Only Memory
RRC	Radio Resource Control
SAS-AKA	Laur-Pasini Authenticated Group Key Agreement Protocol
SAS-GMA	Laur-Pasini Group Message Authentication Protocol
SHCBK	Symmetrised HCBK Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UE	User Equipment
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
WAN	Wide Area Network
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

References

[1] S. A. Camtepe and B. Yener, “Key Management in Wireless Sensor Networks”, In Book: “Wireless Sensor Network Security”, Javier Lopez and Jianying Zhou (editors), ISBN 978-1-58603-813-7, Cryptology & Information Security Series, IOS Press, 2008.
[2] FP7 EXALTED consortium: “D4.1 - M2M Packet Data Protocols between LTE-M and Capillary Networks project report, June 2012.
[3] FP7 EXALTED consortium: “D4.2 - IP Networking System for M2M communications for EXALTED use cases report, June 2012.
[4] J. Lopez, J. Y. Zhou, “Overview of wireless sensor network security”, In Book: “Wireless Sensor Network Security”, Javier Lopez and Jianying Zhou (editors), ISBN 978-1-58603-813-7, Cryptology & Information Security Series, IOS Press, 2008.
[5] W. Diffie and M. Hellman, “New Directions in Cryptography,” IEEE Trans. on Information Theory, vol. 22, pp. 644-654, 1976.
[6] M. Burmester and Y. Desmedt, “A Secure and Efficient Conference Key Distribution System,” in Advances in Cryptology - EUROCRYPT 1994, LNCS, vol. 950, A. De Santis, Ed., Springer, 1995, pp. 275–286.
[7] M. Burmester and Y. Desmedt, “A Secure and Scalable Group Key Exchange System,” Information Processing Letter, vol. 94, pp. 137–143, 2005.
[8] M. Manulis, "Contributory Group Key Agreement Protocols, Revisited for Mobile Ad hoc Groups," in IEEE International Conference on Mobile Adhoc and Sensor Systems, 2005.
[9] D. Harkins, D. Carrel, Eds., “The Internet Key Exchange (IKE),” RFC 2409, Nov. 1998.
[10] L. Law, et al., “An Efficient Protocol for Authenticated Key Agreement,” Designs, Codes and Cryptography, vol. 28, pp. 119–134, 2003.
[11] W. Diffie, et al., “Authentication and Authenticated Key Exchange,” Designs, Codes and Cryptography, vol. 2, pp. 107–125, 1992.
[12] S. M. Bellovin and M. Merrit, “Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks,” in IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, 1992, pp. 72-84.
[13] D. Dolev and A.C. Yao, “On the Security of Public Key Protocols,” IEEE Trans. on Information Theory, vol. 29, pp.198-208, 1983.
[14] A. Malkani and L. D. Dhomeja, "Secure device association for ad hoc and ubiquitous computing environments," in Emerging Technologies, 2009. ICET 2009. International Conference on, 2009, pp. 437-442.
[15] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues for ad hoc wireless networks," Security protocols, 2000, pp. 172-182.
[16] D. Balfanz, D. K. Smetters, P. Stewart and H. C. Wong, Talking to strangers: Authentication in ad hoc wireless networks, Symposium on Network and Distributed Systems Security (NDSS '02), 2002, pp.
[17] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl and H. W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," Proceedings of the 3rd international conference on Ubiquitous Computing, Springer-Verlag, Atlanta, Georgia, USA, 2001.
[18] J. Lester, B. Hannaford and G. Borriello, Are you with me? – using accelerometers to determine if two devices are carried by the same person, Pervasive Computing, Springer-Verlag (2004) pp. 33-50, 2004.
[19] R. Mayrhofer and H. Gellersen, "Shake well before use: Authentication based on accelerometer data," 5th International Conference on Pervasive Computing (Pervasive 2007), 2007.
[20] J. M. McCune, A. Perrig and M. K. Reiter, Seeing-is-believing: Using camera phones for human-verifiable authentication, Security and Privacy, 2005 IEEE Symposium on (2005), 110 - 124.

[21] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik and E. Uzun, Loud and clear: Human-verifiable authentication based on audio, Distributed Computing Systems, ICDCS 2006. 26th IEEE International Conference, 2006.
[22] C. Soriente, G. Tsudik and E. Uzun, "Hapadep: Human asisted pure audio device pairing," Cryptology ePrint Archive, Report 2007/093, 2007.
[23] C. Castelluccia and P. Mutaf, "Shake them up!: A movement-based pairing protocol for cpu-constrained devices," Proceedings of the 3rd international conference on Mobile systems, applications, and services, ACM, Seattle, Washington, 2005.
[24] A. Varshavsky, A. Scannell, A. LaMarca and E. d. Lara, "Amigo: Proximity-based authentication of mobile devices," Ubicomp 2007: Ubiquitous computing, 2007, pp. 253-270.
[25] C. Soriente, G. Tsudik and E. Uzun, Beda: Button-enabled device association, Internation Workshop on Security and Spontaneous Interaction (IWSSI 2007), 2007.
[26] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun, "A comparative study of secure device pairing methods," Pervasive and Mobile Computing, vol. 5, pp. 734-749, 2009.
[27] L. Ming, Y. Shucheng, L. Wenjing, and R. Kui, "Group Device Pairing based Secure Sensor Association and Key Management for Body Area Networks," in INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1-9.
[28] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi, “Secure in-band wireless pairing”, In USENIX Security Sym., 2011.
[29] J. Valkonen, et al., “Ad hoc Security Association for Groups,” in Security and Privacy in Ad hoc and Sensor Networks - ESAS 2006, LNCS, vol. 4357, L. Buttyan, et al., Eds., Springer, 2006, pp. 150–164.
[30] S. Laur and K. Nyberg, ”Efficient Mutual Data Authentication Using Manually Authenticated Strings,” in Cryptology and Network Security - CANS 2006, LNCS, vol. 4301, D. Pointcheval, et al., Eds., Springer, 2006, pp. 90-107.
[31] F. L. Wong and F. Stajano, “Multi-channel Protocols for Group Key Agreement in Arbitrary Topologies,” in Third IEEE International Workshop on Pervasive Computing and Communication Security - PerSec 06, IEEE Press, 2006.
[32] F. L. Wong and F. Stajano, "Multichannel Security Protocols," IEEE Pervasive Computing, vol.6, pp.31-39, 2007.
[33] L. H. Nguyen and A.W. Roscoe, “Efficient Group Authentication Protocol Based on Human Interaction,” in Joint Workshop on Foundation of Computer Security and Automated Reasoning Protocol Security Analysis - FCS-ARSPA 2006, 2006, pp. 9-31.
[34] L. H. Nguyen and A.W. Roscoe, “Authenticating Ad hoc Networks by Comparison of Short Digests,” Information and Computation, vol. 206, pp. 250-271, 2008.
[35] A. W. Roscoe, “Human-centred Computer Security,” Unpublished draft, 2006, Available: http://www.cs.ox.ac.uk/people/bill.roscoe/publications/113.pdf
[36] T. Perkovic, et al., "Secure Initialization of Multiple Constrained Wireless Devices for an Unaided User," IEEE Transactions on Mobile Computing, 2011.
[37] L. H. Nguyen and A.W. Roscoe, “Authentication Protocols Based on Low−Bandwidth Unspoofable Channels: A Comparative Survey,” Journal of Computer Security, vol. 19, pp. 139−201, 2011.
[38] S. Laur and S. Pasini, “SAS-based Group Authentication and Key Agreement Protocols,” in Public Key Cryptography – PKC 2008, LNCS, vol. 4939, R. Cramer, Ed., Springer. 2008, pp.197–213.
[39] PASINI, S. AND VAUDENAY, S. 2006. SAS-based Authenticated Key Agreement. In Public Key Cryptography - PKC ’06. LNCS Series, vol. 3958. 395 – 409.
[40] S. Capkun, M. Cagalj, R. Rengaswamy, I. Tsigkogiannis, J.-P. Hubaux, and M. Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. IEEE Transactions on Dependable and Secure Computing, 5(4):208–223, October–December 2008.
[41] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In USENIX Security Sym., 2011

[42] ETSI TS 102 690 (2010-09) : Machine- to- Machine communications (M2M); Functional architecture
[43] FP7 EXALTED consortium: “D5.1 - Security and Provisioning Solutions project report, Jan 2012.
[44] 3GPP TR 33.816 “Feasibility Study on LTE relay node security” http://www.3gpp.org/ftp/Specs/html-info/33816.htm
[45] 3GPP TS 33.401 “3GPP System Architecture Evolution (SAE); Security architecture” http://www.3gpp.org/ftp/Specs/html-info/33401.htm
[46] T. Sakurai and H. L. Vu, "MAC access delay of IEEE 802.11 DCF,” IEEE Transactions on Wireless Communications, vol. 6, no. 5, pp. 1702-1710, May 2007.
[47] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, “Performance of reliable transport protocol over IEEE 802.11 wireless LANs: analysis and enhancement,” in Proc. IEEE INFOCOM 2002, New York, Jun. 2002.
[48] K. Schwieger, A. Kumar, and G. Fettweis, “On the impact of the physical layer on energy consumption in sensor networks,” in Proc. European Workshop on Sensor Networks, Istanbul, Jan. 2005.
[49] H. L. Vu and T. Sakurai, “Accurate delay distribution for IEEE 802.11 DCF,” IEEE Communications Letter, vol. 10, no. 4, pp. 317-319, Apr. 2006.
[50] e-SENSE, European FP6 project, Deliverable D3.3.2: Novel Cross Optimization, Available online http://www.ist-e-sense.org, last accessed on 28 Aug. 2010.
[51] IETF RFC 5166, “Metrics for the evaluation of the congestion control mechanisms,” S. Floyd, Mar. 2008.
[52] 3GPP TS 23.203 v. 8.8.0, “Policy and charging control architecture,” Dec. 2009.
[53] H. Holma and A. Toskala, “WCDMA for UMTS,” 4th edition, Willey, 2008.
[54] EXALTED WP7 - Integration & Proof of Concepts. Selection of key algorithms and technologies for proof of concept Testbeds - Integration of selected algorithms into platforms. EXALTED Internal Report D7.2/7.3.
[55] S. Park, K. Kim, W. Haddad, S. Chakrabarti, and J. Laganier, "IPv6 over Low Power WPAN Security Analysis", IETF Internet-Draft draft-daniel-6lowpan-security-analysis-05, March 15, 2011
[56] IEEE Std 802.16j 2009, "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 1: Multihop Relay Specification", IEEE Computer Society and the IEEE Microwave Theory and Techniques Society, IEEE Std 802.16j 2009, 12 June 2009.
[57] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. IETF RFC 4919, August 2007
[58] ZigBee Alliance 2009. ZigBee Alliance Plans Further Integration of Internet Protocol Standards. ZigBee Alliance News Releases Archives https://docs.zigbee.org/zigbee-docs/dcn/09-5003.pdf, 04/27/2009.
[59] OMA-RD-CPNS-V1, “Converged Personal Network Service Requirements Candidate Version 1.0”, May 2011.
[60] OMA-TS-CPNS-Core-V1, “Converged Personal Network Service Core Technical Specification Candidate Version 1.0”, May 2011.
[61] OMA-AD-CPNS-V1, “Converged Personal Network Service Architecture Candidate Version 1.0”, May 2011.
[62] OMA-SEC-CF-V1, “OMA Application Layer Security Common Functions V1.1”, Nov. 2010.
[63] EXALTED D4.2 “First Report on E2E M2M System”,
[64] EXALTED D2.1 “Description of baseline reference systems, scenarios, technical requirements and evaluation methodology”, 31 May 2011.
[65] Yair Amir, Claudiu Danilov, Michael Hilsdale, Raluca Mus aloiu-Elefteri, Nilo Rivera Fast Handoff for Seamless Wireless Mesh Networks MobiSys'06, June 19–22, 2006, Uppsala, Sweden.

[66] Soft Decision Error Correction for Compact Memory-Based PUFs Using a Single Enrollment Vincent van der Leest,Bart Preneel,Erik van der Sluis; Cryptographic Hardware and Embedded Systems – CHES 2012 Lecture Notes in Computer Science Volume 7428, 2012, pp 268-282

[67] Pappu, R. S. (2001). Physical one-way functions. PhD thesis, Massachusetts Institute of Technology, March. http://pubs.media.mit.edu/pubs/papers/01.03.pappuphd.powf.pdf.

[68] Pappu, R. S., Recht, B., Taylor, J., & Gershenfeld, N.(2002). Physical one-way functions. Science 297(6),2026–2030. http://web.media.mit.edu/~brecht/papers/02.PapEA.powf.pdf

[69] Guajardo, J., Kumar, S. S., Schrijen, G.-J., & Tuyls, P. (2007a). FPGA intrinsic PUFs and their use for IP protection. In P. Paillier, & I. Verbauwhede (Eds.), Cryptographic hardware and embedded systems—CHES 2007, 10–13 September LNCS (Vol. 4727, pp. 63–80). New York: Springer.

[70] Sandeep S. Kumar · Thijs Bel · Antoon H. M. Blom ·Geert-Jan Schrijen Anti-counterfeiting, key distribution, and key storagein an ambient world via physical unclonable functions; October 2008 Springer Science + Business Media, LLC 2008

[71] EXALTED WP2 - The EXALTED system concept and its performance. EXALTED public deliverable D2.4.