A Cyber-Physical Approach to Secret Key Generation in Smart Environments

Paolo Barsocchi *, Stefano Chessa * †, Ivan Martinovic ‡, Gabriele Oligeri *

* ISTI-CNR, Pisa Research Area, Pisa, Italy

† Computer Science Department, University of Pisa, Largo B. Pontecorvo 3, 56127 Pisa, 8 Italy

‡ Distributed Computer Systems Lab, Computer Science Department, University of Kaiserslautern, Germany

Abstract

Encrypted communication in wireless sensor networks oftentimes requires additional randomness and frequent re-keying in order to avoid known-plain text attacks. Conventional approaches for shared secret generation suffer however from various disadvantages, such as necessity of a trusted third party, protocol scalability, and especially, the computational resources needed for performance-demanding public-key protocols.

To appropriately respond to the increasing disproportions between a computationally powerful adversary and lightweight wireless devices, a *cyber-physical* approach has recently attracted much attention. The general idea is to leverage the properties of the physical world and include them in a design of lightweight security protocols. Especially valuable physical property is the erratic and unpredictable nature of multi-path signal propagation which has already shown itself as an efficient source of randomness.

This work presents a new cyber-physical approach in order to make secure wireless sensor communications and proposes a secret key extraction algorithm that leverages signal strength fluctuations resulting from dynamic physical environments, e.g. environments experiencing human movements. In particular, this work presents a systematic experimental evaluation by using a real-world sensor network, and analyzes the impact of different moving patterns on legitimate devices and an eavesdropper. Finally, this work quantifies the main factors that influence the key establishment algorithm and propose a protocol which allows secret sharing in an effective and efficient way.

I. Introduction

Wireless sensors networks (WSNs) [1] are one of the most important enabling technology for the smart environments[1]. Sensors in a WSN are micro-systems embedding a number of transducers and a wireless radio interface. Often they are small sized and battery powered. Sensors collect information from the environment in order to undertake actions on the environment itself. Some of the sensors are statically deployed in the environment, while some others are mobile (typically those deployed on the users body). The sensors are also interconnected with the

[1] In 2007, the MIT Technology Review ranked short-range transmission technologies, such as wireless sensor networks, among 10 emerging technologies that will change the world. The analysis company ON World estimates the global market for, e.g., wireless sensor networks to about 9000 million Euros in 2010.

application framework of the Ambient Assisted Living (AAL) system by means of suitable gateways [2].

One of the main issues in security design for wireless sensor networks (WSNs) is the secret key generation and its distribution, since sensors are commonly computationally-weak and battery-limited devices. In this field, a majority of existing works focus on adapting conventional security solutions into the world of WSNs. Such solutions include manual installation of secrets by the user, their imprint during the production process, or/and the use of public key cryptography. Yet, common to all of them is the abstraction of the physical properties of communication. On the other hand, a realistic high-powered adversary takes advantage of performance-based disproportions and, with the help of the broadcast nature of the wireless channel, it is able to launch various resource-depletion attacks such as, e.g., flooding a sensor with public-key requests.

A secure and efficient key-distribution mechanism is needed to allow simple key establishment in WSNs [3]. Recently, a cyber-physical approach to this problem has shown to be an interesting alternative to the conventional security design. Instead of ignoring the physical properties of communication, the peculiarities of the wireless channel are used as valuable security primitives. Such physical properties, specifically, *reciprocity* of radio wave propagation and its *erratic and unpredictable* behavior in temporal and spatial domains can successfully be turned against an adversary. The reciprocity of the wireless channel causes correlated measurements of the channel statistics at both transmitters; at the same time, the physical environment and the multipath fading impact the channel response in an unpredictable fashion. Hence, independently of adversarial computational capabilities and only assuming different physical positions, two transmitters can generate and share secrets derived from their inherent physical behavior. Importantly, the messages exchanged between two legitimate transmitters do not carry any content and only serve to provide channel statistics. Hence, high computational costs or complex statefull protocols can be avoided.

Recently, there is a number of security schemes that are based on such an approach (more detailed discussion is given in the Related works Section). However, most of them require arbitrary movements of the sensors in order to generate secrets keys. On the other hand, many sensors in typical AAL deployments are stationary (think of sensors for environmental parameters such as temperature, etc.) and, therefore, existing schemes cannot be applied. This brings to the central question of this work: how can static WSNs generate and share secrets derived from the physical environments under realistic application scenarios? The general idea is to take advantage of physical environment in which different events, such as human movements, influence the signal propagation in an unpredictable but yet correlated fashion. This allows legitimating transmitters to opportunistically take advantage of such disturbances as a source of randomness and to derive fresh shared secrets.

Specifically, this work presents a novel key generation scheme in which secrets are not brought to devices in a conventional way, but are harvested from the physical world by the sensors themselves. This work follows an experimental and system-oriented approach by introducing a new scheme in a real-world environment. Collected data has been used to evaluate how the human movements impact on the signal propagation and how resulting signal deviations can be used to generate secret keys. WSNs have been adopted as application scenario, because they are usually deployed within indoor environments, such as public and private residential areas. **Contribution.**

This work can be summarized as follows :

- Design of a robust key generation protocol for static WSNs, which takes advantage of environmental events to extract shared secrets.
- Implementation of the protocol by using off-the-shelf sensor devices and its evaluation in a real environment.
- Systematic analysis of the collected data with respect to primary factors which impact the rate of the successful key exchange.
- A new adversarial strategy is evaluated, the Receive Signal Strength (RSS) based attack, that attempts to guess a secret by guessing the RSS from legitimate positions.
- A real scenario constituted by a crowded environment is presented: results related to key agreement performance and adversarial key guessing capabilities are compared.

II. Related works

The problem of generating shared secrets by using correlated random sources and communication over broadcast channels has been intensively studied from information-theoretic perspective [4], [5], [6]. Concretely, results from [7], [8] analyze how information-theoretically secure keys can be derived in settings where three participants observe correlated random variables. They show that an eavesdropper remains completely ignorant of derived secrets even though its observations are also correlated with the random source. Recently, the wireless communication attracted many researchers to instantiate these information-theoretical results and adapt them to more practical settings. The appealing property lies in the nature of the signal propagation which, while being erratic and unpredictable, provides correlated observations of the channel response between two transmitters [9].

Several papers take advantage of this property and use narrow-band communication to generate secret keys from the wireless channel. In [10], the authors assume random movement of transmitters as a source of entropy. By frequently sampling the RSS, both parties can create a sequence of channel states that are strongly correlated. The fading behavior on a single sampling frequency is strongly dependent on the physical position, and the movement introduces uncertainty for an adversary that is captured in these sequences. The authors apply a level-crossing algorithm that uses two thresholds for signal strength values to generate bit strings. However, their secret key generation scheme requires powerful transmitters such as software-defined radios or laptops, since a high sampling rate is necessary to provide usable secret-key generation rate. Similarly, in [11], the authors propose a protocol that focuses mainly on the robustness of the key generation process, i.e., tolerance against deviations in the wireless channel and a high success rate of key-agreements. They employ a single threshold for detection of strong deep fades, an event that is reliably detectable, but also rare (in the order of Hz). Their quantitative evaluation is based on theoretical channel model and simulations that does not shed light on performance of such key-generation scheme in an everyday, real-world environment. In addition, they also assume movements of transmitters as a source of randomness. Several other contributions use highly specialized hardware, such as steerable antennas, ultra-wideband (UWB) radio or multi-antenna systems with performance-capable processors [12], [13], [14]. The most recent overview with comparison of existing schemes is given in [15]. Although these existing results justify

the possibility of generating secret keys from the wireless communication, they cannot be applied in static WSNs. Realistic "off-the-shelf" sensor devices can neither rely on random (and fast) movements, nor on sophisticated hardware such as steerable multi-array antennas. Authors in [16] present a similar scenario where they also consider static WSNs. In fact, their key-generation scheme is based on using frequency-hopping as a source of randomness. However, the few number of wireless channels is not suitable to generate secret random keys, the secrecy of the extracted secret is limited and static. In [**?**], authors proposed preliminary results regarding the key establishment in WSNs leveraging RSS fluctuations in indoor environment.

This work presents a new detailed analysis regarding the security issues in smart environments featuring WSNs, and new results related to the algorithm performance. Moreover, this work presents the results of extensive real-world measurements and shows that the generated secrets can be used for key-refreshment in WSN environment.

III. Scenario and Security Issues

WSNs undergo to intrinsic issues related to the data sampling procedures performed in the smart spaces. There are mainly three important points to address:

- Commodity sensors are usually cheap devices constituted by **simple hardware**.
  Asymmetric crypto, and more generally, computational expensive procedures should not be used in order to save the battery and, thus, to preserve the sensor lifetime.
- The sampled data has a **low numerical resolution** (granularity).
  This is mainly due to the low-cost architecture featured in the sensor. Each data sample lacks the randomness needed by the crypto algorithms to make the cypher-text cryptographically secure. Hence, sampled data cannot be used as source of entropy for the generation of secrets.
- Data samples are strongly **time-correlated**.
  The sample at time $t+1$ is strongly correlated with that one at time t. As an example, Fig. 1 shows 12h of sound (Fig. 1(a)), light (Fig. 1(b)) and temperature (Fig. 1(c)) samples measured in a generic office. As it can be seen, such raw data is strongly time-correlated due to the nature of the observed physical phenomena.

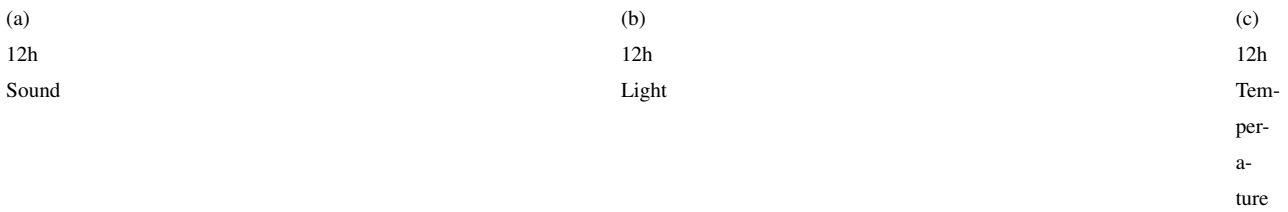


Fig. 1. Half a day of sensor measurements from an indoor WSN. The sensed data shows a low numerical resolution and a highly time-correlated behaviour

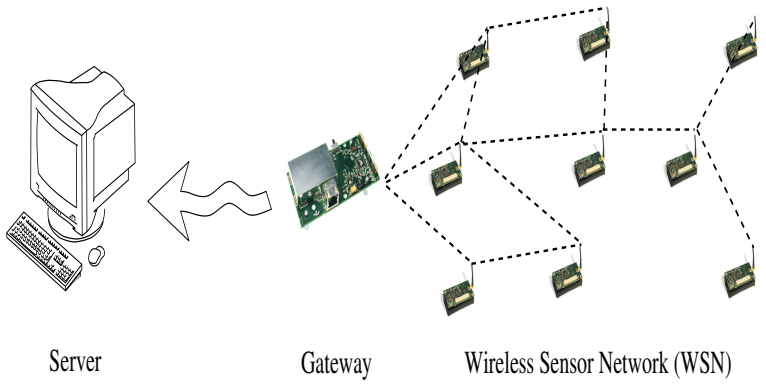


Fig. 2. The envisioned scenario: wireless sensors report measured environmental data to a base station, e.g., a gateway, using a multihop WSN.

A. Network Scenario

Figure 2 shows the envisioned scenario: a multi-hop WSN and a gateway, allowing the data forwarding to a server for storing and processing. Each sensor senses the environment, encrypts the sensed data, and finally transmits it to a neighbor. In this work, secrets are extracted from RSS estimations, and therefore, the only way for $\mathcal{E}$ to access the secret data is to guess the secret keys by overhearing the RSS values. The most important sensor characteristics can be summarized as follows:

- RSS estimation capability: The core of the system is the RSS procedure. Sensors must be able to evaluate the RSS for each received packet according to a data collection strategy. This capability is usually provided by every wireless interface, since the RSS measurement and estimation is required by medium access protocols.
- Security: Each sensor is able to perform symmetric key encryption and hash computations. Note that, no pre-shared secrets are requested for guarantee the security of the WSN since the adversary is interested in passively learning about the sensed data, i.e. as already defined $\mathcal{E}$ is a perfectly passive adversary. Hence, for the evaluation proposed in this paper all the sensors and their transmitted packets can be considered as authentic, yet vulnerable to off-line analysis.
- Data collection: Each sensor communicates according to a given network-wide data collection strategy; each sensor is able to communicate with one or more neighbors in the WSN and, by means of multihop protocols, to reach the gateway.

B. The Bad Guy: behavior and goals

The adversary, hereafter Eve ($\mathcal{E}$), may have an easy life in a WSN environment. Firstly, the adversary can try both to brute-force the transmitted cypher-text or collect cypher-text data from the radio channel in order to foresee some values. Secondly, sensors are easy to tamper due to their cheap architecture; in this work the adversary is able to identify and localize the sensor, dismantle it, and eventually, disclose all its secrets stored in the memory. Nevertheless, this work does not deal with *active* adversaries, i.e, $\mathcal{E}$ will not attempt to tamper with the protocol

or sensors. Hence, while she is interested to collect as many information as possible from the sensor network, she wants to avoid any risk of being detected. In addition, countermeasures against an active adversary have been widely studied in literature [17] [18] and are out of the scope of this work.

Figure 3 shows the three different status in which a sensor can stay when dealing with an adversary. The sensor starts *healthy*, i.e. it has at least one shared secret with its own peer that $\mathcal{E}$ does not know. Subsequently, $\mathcal{E}$ identifies the sensor and dismantles it accessing to all its secrets. This is the *compromise phase*, nothing can be done here because the sensor is under the control of the adversary. Then, $\mathcal{E}$ releases the sensor, but again, she still knows all its secrets. Therefore, from now on she can acts as a pure eavesdropper and access all the cypher-texts transmitted by the sensor. Only a *new secret injection* can make the sensor recover the secret status.

This issue cannot be solved using traditional solutions like Diffie-Hellman key generation [**?**], in fact if the adversarial behaviour is such that she can disclose all the secrets inside the sensor, she can subsequently foresee all the pseudo-random values generated by all the sensors in the WSN. It is worth noticing, how a true-random number generator can easily fix the previous issue, i.e. a real source of randomness make possible the healthy status regain, but so far commodity sensors are not provided with such feature.

The new established secret must be unknown to $\mathcal{E}$, who is still eavesdropping the channel, and shared with the peer of the sensor. $\mathcal{E}$ will never be able to compromise all the sensor in the WSN at the same time. In particular, for the sake of simplicity, it is assumed that the adversary compromises one sensor at time, i.e. before compromising a new sensor the adversary must release the old one. Concluding, the behavior of Eve can be summarized as follow:

- **RSS overhearing capabilities**: $\mathcal{E}$ is able to evaluate the RSS of all the packets transmitted in the network, provided that the sending sensor is within her transmission range.
- **Sensor behavior knowledge**: $\mathcal{E}$ knows all the procedures and algorithms running on the sensors.
- **Environment passive player**: she is not able to inhibit environmental event occurrences; she does not know the disturbing events speed, their path, the time they appear, and finally, their duration. Her knowledge about event occurrences is exactly the same of the sensors.
- **RSS-attack strategy**: The only way for $\mathcal{E}$ to access the encrypted data on sensors is to compute the shared secret, that is, collecting as much as possible RSS values with almost the same values of one of the two peers. This new kind of approach is defined as *RSS-attack*, that is, $\mathcal{E}$ wants to get access to the secret key establishment by means of a RSS sniffing process. The *RSS-attack* is more and more efficient as $\mathcal{E}$ gets closer to one of the peers. This is possible if and only if $\mathcal{E}$ belongs to a *privacy region* of one of the two peers, and in this work a bound for that region is proposed. Moreover, when she is out of the privacy region, we she cannot perform a successful *RSS-attack* due to the fact that she is not able to control the events occurrences in the environment around the sensors.

IV. Our solution in a nutshell

In this section, a new key generation scheme is presented: it extracts randomness from events occurrence (i.e. persons motion) to share new secrets between pair of sensors. The received power can be considered as a shared

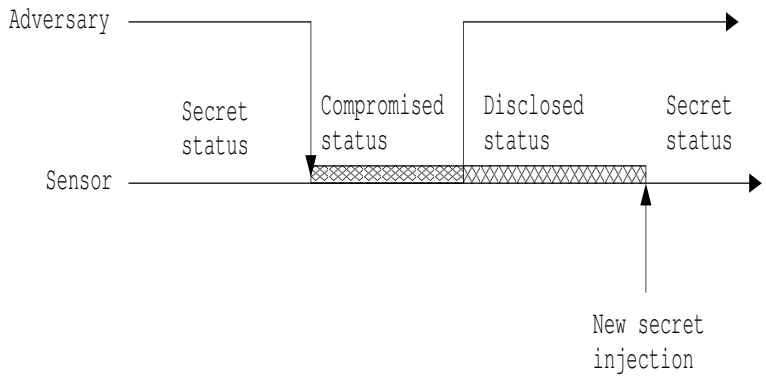


Fig. 3. WSN activity timeline during an attack: by generating new secrets from environmental data the secret state of a sensor can be recovered.

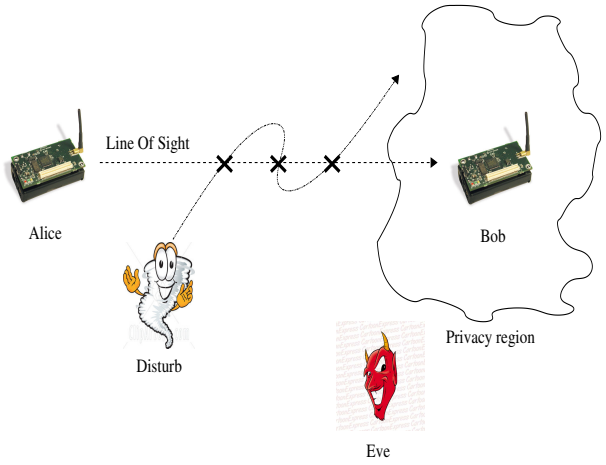


Fig. 4. Generating new secrets from from frequent Lo-S disturbances.

information between the transmitter and the receiver under the hypothesis of signal propagation on a symmetric channel [19] [10]. Channel reciprocity is leveraged in order to to make an agreement on a shared secret key, i.e. the received power is correlated at the both transmitters.

The RSS evaluated for each received packet turns out to be a good estimation of the signal to noise ratio as witnessed in [20][21][22]. RSS variation has been widely used to generate shared secret keys [16][23], but multipath fading in static scenarios proves to be an insufficient source of entropy, and therefore malicious entities can easily find the generated keys. In particular, this work leverages the RSS variation due to disturbing events that cross the Line of Sight, hereafter Lo-S, for generating shared secrets between pairs of sensors. The disturbing event, hereafter $\mathcal{D}$, has movement characteristics, i.e. speed and direction, that dynamically changes and cannot be foreseen by malicious third entity. Figure 4 shows how the various entities behave in the proposed scenario. Alice and Bob, hereafter $\mathcal{A}$ and $\mathcal{B}$ respectively, are two sensors deployed in a WSN, and they want to share a secret key to communicate safely among them. $\mathcal{E}$, the adversary, wants to achieve the knowledge of the shared secret between $\mathcal{A}$ and $\mathcal{B}$. Finally, $\mathcal{D}$

Fig. 5. Measurement scenario: Alice on the left, Bob and the gateway on the right, the server in the top right side, and finally the four different positions considered for Eve.

breaks the Lo-S between $\mathcal{A}$ and $\mathcal{B}$, generating a RSS variation Δ_P for a time interval Δ_T on the received packets in both the sensors: we leverage the previous effect to extract a shared and secret key among them. Obviously, $\mathcal{E}$ can detect $\mathcal{D}$ but cannot experience the same variation of $\mathcal{A}$ and $\mathcal{B}$. Whereas, $\mathcal{E}$ can move and take place close to $\mathcal{A}$ or $\mathcal{B}$ with the aim of experimenting the same RSS variations produced by $\mathcal{D}$. Therefore, it is necessary to define a *privacy region* around $\mathcal{A}$ and $\mathcal{B}$ to protect them from RSS sniffing. In the following sections an in-depth analysis of this issues will be provided.

V. RSS SAMPLING: MEASUREMENTS AND CONSIDERATIONS

The measurement campaign was performed in a 7x5 m_2 room. Figure 5 shows a picture of the measurement layout. Alice, on the left side, and Bob, on the right side, are placed over tables of one meter height. The solid blue line represents the Lo-S between $\mathcal{A}$ and $\mathcal{B}$, the distance between them is about 3 meters. Bob is connected to the gateway, which in turn is directly connected a server. Three different positions for $\mathcal{E}$have been considered, all of them at 3 meters from Bob, therefore the distance between $\mathcal{A}$ and $\mathcal{B}$ is always equal to the distance between $\mathcal{B}$ and $\mathcal{E}$. Position 1 is the closest one, just by $\mathcal{A}$ side, position 2 is at 1 meter of distance from $\mathcal{A}$, and finally, position 3 (the further) is at 2 meters. Alice, Eve and Bob are MICAz sensor motes [24], which are equipped with a 2.4Ghz radio, IEEE 802.15.4 compliant. MICAz motes feature the Atmel ATmega128L low-power microcontroller and ChipCon CC2420 radio. The software developed for testing the algorithm is written in nesC [25] and executed on TinyOS [26]. The software behaves as follows: $\mathcal{B}$ sends a packet to $\mathcal{A}$, who receives it, evaluates the RSS, injects that value in the payload of a new packet, and sends that packet back to $\mathcal{B}$. $\mathcal{E}$ behaves in the same way, sending back to $\mathcal{B}$ her estimation of the RSS; therefore $\mathcal{B}$ knows the RSS estimation computed at both $\mathcal{A}$ and $\mathcal{E}$, consequently, he can forward these information to the server for further analysis. The experiments have been conducted sampling the channel every 100 ms; this is a critical parameter since increasing the sampling frequency it is possible to collect more information related to the disturbing event, nevertheless energy saving issues should be taken into account. Actually, in a real scenario, RSS estimation does not occur during a dedicated transmission, but leveraging packet exchanging due to all the communications between the peers.

A. Wireless Channel Considerations

The wireless channel is a symmetric time-dependent and spatial-varying filter, i.e. it has the same response for the signals sent from $\mathcal{A}$ to $\mathcal{B}$ as well as from $\mathcal{B}$ to $\mathcal{A}$: multipath properties of the radio channel i.e. gains, phase shift, and delays are identical on both directions of the link. However, noise sources like radio equipments, moving persons, and environmental changes in general do not have a symmetric impact on $\mathcal{A}$ and $\mathcal{B}$ radio receivers, and therefore, $\mathcal{A}$ and $\mathcal{B}$ are unable to obtain identical measurements of the radio channel. Moreover, the IEEE 802.15.4

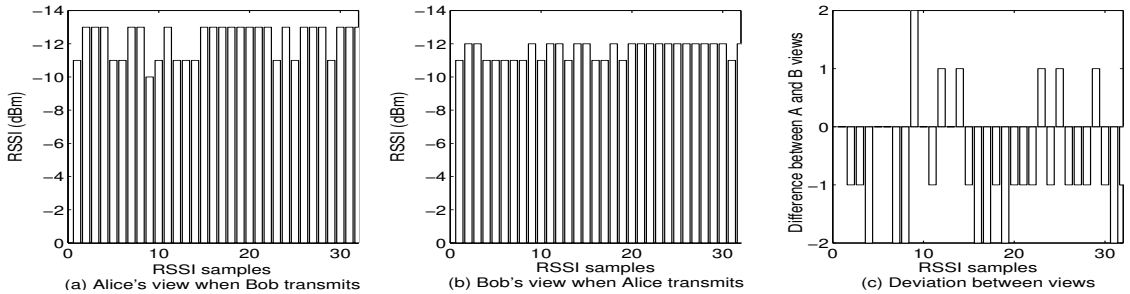


Fig. 6. Reciprocity of the wireless channel in a static office environment.

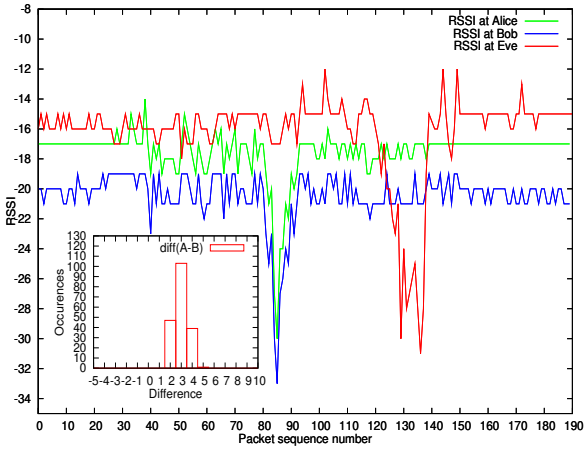
transceivers are half duplex, i.e., each sensor cannot transmit and receive simultaneously; thus $\mathcal{A}$ and $\mathcal{B}$ do not have the same snapshot of the radio channel because they measure the signal strength one after the other. Nevertheless, the time between $\mathcal{A}$ and $\mathcal{B}$ channel sampling is smaller than the channel coherence time, and therefore $\mathcal{A}$ and $\mathcal{B}$ exhibit a similar received signal power [27].

The Static environment: Figure 6 shows the $\mathcal{A}$ and $\mathcal{B}$ channel snapshot performed in the environment of Fig. 5. It refers to a deployment of 2 MICAZ sensors at 3 meters of distance from each other. Figure 6(a) shows about 30 RSS samples evaluated on packets transmitted by $\mathcal{B}$, while Fig. 6(b) shows the RSS samples evaluated in the same period by $\mathcal{B}$. Finally, Fig. 6(c) shows the difference between $\mathcal{A}$ and $\mathcal{B}$ views. This simple experiment witnesses that a static environment like that one in Fig. 5 cannot be used to extract shared secret; in fact, key agreement can be easily performed in such conditions, but no sufficient randomness can be extracted from that time period, and consequently, the shared secret can be easily guessed by the adversary. Therefore, different solutions have been proposed to undertake the previous issue [16][10], however none of them exploits disturbing events that may occur in the environment: this work shows how it is possible to leverage the randomness introduced by RSS fluctuations caused by unexpected disturbing events that appear on the scene.

The Dynamic environment: Nine different scenarios configurations have been considered, i.e. the communication between $\mathcal{A}$ and $\mathcal{B}$ is repeated in 9 different configurations: 3 different positions for $\mathcal{E}$ and 3 different disturbing events. The envisioned disturbing event $\mathcal{D}$ is represented by a person who walks through the room in 3 different ways:

- *Simple cut*: $\mathcal{D}$ breaks the Lo-S between $\mathcal{A}$ and $\mathcal{B}$ one time only.
- *Crosswalk*: $\mathcal{D}$ goes in the direction of $\mathcal{A}$, it turns 90 degrees clockwise, it walks straight in the $\mathcal{A}$-$\mathcal{B}$ Lo-S direction, and after reaching $\mathcal{B}$, it turns 90 degrees anticlockwise, and finally, it walks again till it stops.
- *Zig-zag*: $\mathcal{D}$ walks in a zig-zag fashion around the Lo-S. The event enters the scene from the $\mathcal{B}$ side, and the overall Lo-S breaks sum up to 4.

Figure 7(a) shows the simple cut scenario. $\mathcal{D}$ walks straight on the red line and breaks first the $\mathcal{A}$-$\mathcal{B}$ Lo-S, and subsequently the $\mathcal{E}$-$\mathcal{B}$ Lo-S. Figure 7(b) shows the estimated RSS for the 3 different sensors. The solid green line represents the RSS estimated at $\mathcal{A}$, the solid blue line at $\mathcal{B}$, and finally, the solid red line at $\mathcal{E}$. It is worth noticing



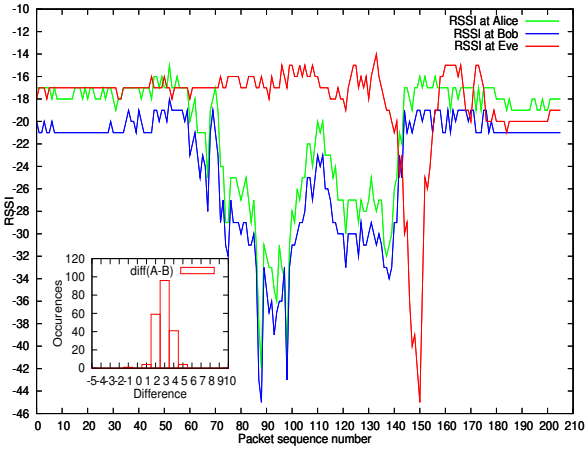
(a) (b)

Fig. 7. Simple cut: The disturbing event walks straight on the red line, and breaks both the $\mathcal{A}$-$\mathcal{B}$ and $\mathcal{E}$-$\mathcal{B}$ Lo-Ss one only time.

the presence of a stationary offset between $\mathcal{A}$ and $\mathcal{B}$ RSS values; this offset has been estimated in the sub-picture showing the occurrences of the differences between $\mathcal{A}$ and $\mathcal{B}$ values: the peak at 3 witnesses the average difference between the $\mathcal{A}$ and $\mathcal{B}$ values. The $\mathcal{D}$ effect is clear both at $\mathcal{A}$ and $\mathcal{B}$ with a strong correlation both in time and amplitude: a fluctuation of about 10dBm is present from the 80th to the 90th received packet. $\mathcal{E}$ (solid red line) observes a completely different channel: the disturbing event $\mathcal{D}$ produces just a sudden variation when it breaks the $\mathcal{E}$-$\mathcal{B}$ Lo-S, from the 120th to the 140th packet.

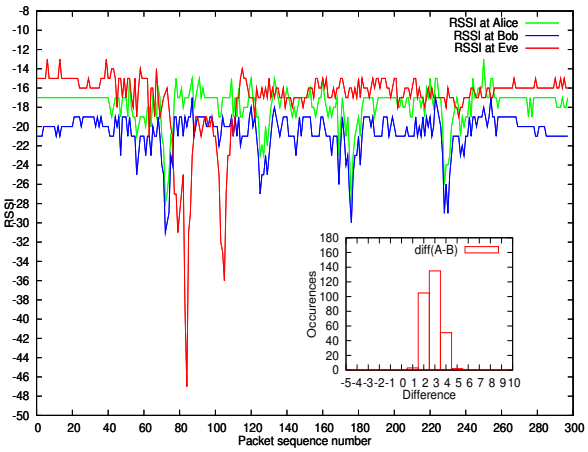
Figure 8(a) shows the crosswalk scenario. $\mathcal{D}$ walks straight on the red line and breaks the $\mathcal{A}$-$\mathcal{B}$ Lo-S for a long time. Figure 8(b) shows the estimated RSS for the 3 different sensors. The solid green, blue and red lines represent the RSS estimated at $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{E}$, respectively. The same stationary offset (of about 3 dBm) appears as in the previous case. The $\mathcal{D}$ effect is clear both at $\mathcal{A}$ and $\mathcal{B}$ with a strong correlation both in time and amplitude: a fluctuation of about 22dBm is present from the 60th to the 140th received packet. $\mathcal{E}$ perspective (solid red line) is completely different: she experiences just a Lo-S break when $\mathcal{D}$ finishes his path.

Figure 9(a) shows the zig-zag scenario. $\mathcal{D}$ walks in a zig-zag fashion around the Lo-S and breaks the $\mathcal{A}$-$\mathcal{B}$ Lo-S 4 times. Figure 9(b) shows the estimated RSS for the 3 different sensors: the green line for $\mathcal{A}$, the blue line for $\mathcal{B}$. and finally, the red line for $\mathcal{E}$. As in the previous cases, there is a stationary offset between $\mathcal{A}$ and $\mathcal{B}$ RSS values which can be approximated with 3dBm. The $\mathcal{D}$ effect is clear both at $\mathcal{A}$ and $\mathcal{B}$ with a strong correlation both in time and amplitude: 4 fluctuations of about 10dBm are present at packets 70, 130, 180, and 230. $\mathcal{E}$ perspective (solid red line) is completely different: she experiences two Lo-S breaks just at the beginning of the path.



(a) (b)

Fig. 8. Crosswalk: The disturbing event walks straight on the red line, and breaks the $\mathcal{A}$-$\mathcal{B}$ Lo-S for a long period.



(a) (b)

Fig. 9. Zig-zag: The disturbing event walks in a zig-zag fashion around the $\mathcal{A}$-$\mathcal{B}$ Lo-S.

VI. Protocol Design

Figure 10 shows the proposed key generation and agreement protocol: the *sampling phase* where RSS values are collected, the *key generation phase* where RSS values are translated into symbols, the *entropy detection phase* assures generated symbols have sufficient entropy and secret key is generated, and finally, the *key verification phase* where sensors pair commit on the generated secret key.

A. Sampling Phase

Sampling phase is depicted in the upper part of Fig. 10. $\mathcal{A}$ and $\mathcal{B}$ sample the channel by exchanging each other k packets and collecting a set of k RSS values $m_j = \{m_1, \cdots, m_k\}$ at $\mathcal{A}$, and $m'_j = \{m'_1, \cdots, m'_k\}$ at $\mathcal{B}$, respectively. $\mathcal{A}$ starts the sampling process sending a packet to $\mathcal{B}$ who answers by sending back to her the channel estimation

Fig. 10. Key generation protocol in four phases: *(i)* sampling phase, *(ii)* key generation phase, *(iii)* entropy detection phase, and *(iv)* verification phase.

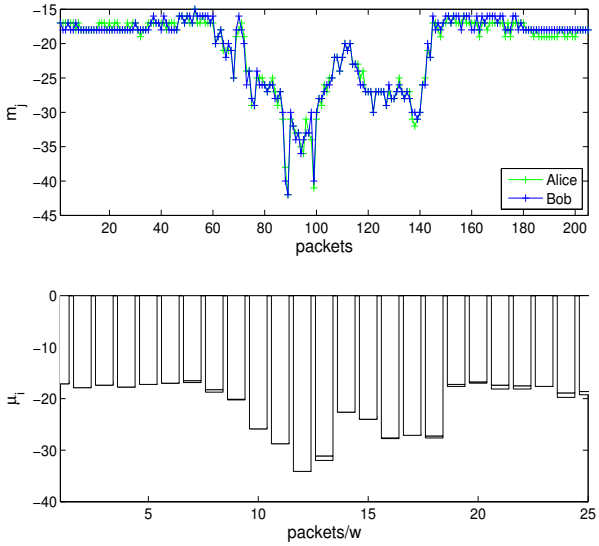


Fig. 11. RSS time series and mean RSS bars over a window of eight samples.

as soon as possible. In order to remove the stationary offset between m_j and m'_j (as shown in Section V-A) a normalization function *offset*$(\circ)$ is defined, obtaining m^o_j and m'^o_j. In practice, the *offset*$(\circ)$ function computes the difference between the current RSS value and an old RSS value sampled when the channel was in a steady state (no disturbing event on the scene). From the perspective of $\mathcal{A}$ and $\mathcal{B}$, the behavior of the channel can be modeled as follows:

$$\begin{aligned} X_{\mathcal{A}} &= \mu_{\mathcal{A}} + N_{\mathcal{A}} \\ X_{\mathcal{B}} &= \mu_{\mathcal{B}} + N_{\mathcal{B}} \end{aligned} \tag{1}$$

where $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{B}}$ are mean values evaluated on w elements of the k sampled RSSs; while $N_{\mathcal{A}}$ and $N_{\mathcal{B}}$ are the random variables representing the errors in the channel reciprocity (noise), experienced at $\mathcal{A}$ and $\mathcal{B}$, respectively. Figure 11 shows a snapshot of a typical channel perspective of a sensor pair. The upper part shows the RSS values sampled at $\mathcal{A}$ and $\mathcal{B}$, that is $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$, respectively; while in the lower part, the bars show the mean computed on the previous values considering $w = 8$, i.e. $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{B}}$, respectively. It is worth noticing that $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{B}}$ are almost the same, and consequently, can be considered as a shared secret between $\mathcal{A}$ and $\mathcal{B}$. Thus, the sampling phase concludes by computing μ_i and μ'_i on the last w RSS samples by $\mathcal{A}$ and $\mathcal{B}$, respectively.

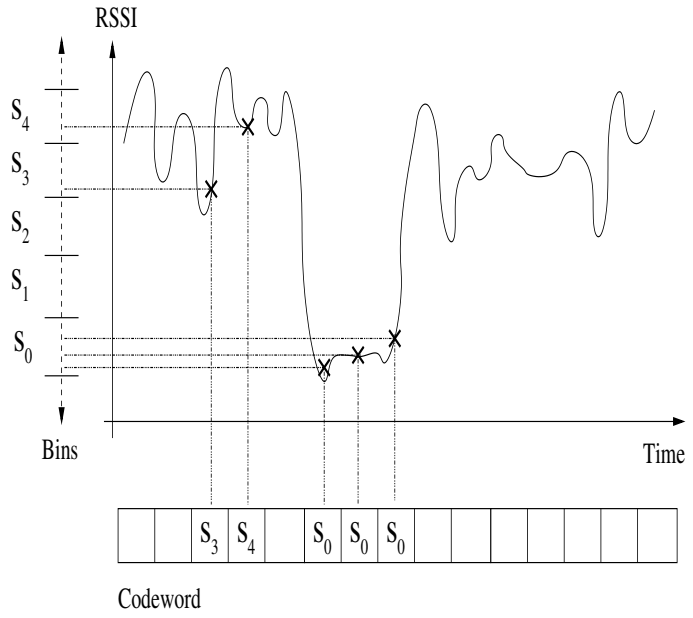


Fig. 12. Symbols generation from RSS values; RSS is sampled and coded to a sequence of symbols.

B. Key Generation Phase

RSS values sampled at $\mathcal{A}$ and $\mathcal{B}$ are not the same and suffer from asymmetry effects due to the environmental noise; in this section, it is presented a way to remove such asymmetries and how $\mathcal{A}$ and $\mathcal{B}$ can eventually agree on a set of values. Let us assume $\mathcal{A}$ and $\mathcal{B}$ collected from the sampling phase a set of RSS samples $X_{\mathcal{A}}$ and $X_{\mathcal{A}}$, respectively, and moreover they computed $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{B}}$, which are the same (see Section VI-A), that is $\mu_{\mathcal{A}} = \mu_{\mathcal{B}} = \mu$. It is worth noticing that the actual shared secret between $\mathcal{A}$ and $\mathcal{B}$ is the mean value μ, and in the following it is shown how the key extraction depends on this value. $\mathcal{A}$ can easily compute her noise component in this way: $N_{\mathcal{A}} = X_{\mathcal{A}} - \mu_{\mathcal{A}}$, then she transmits it to $\mathcal{B}$. Now it is easy for $\mathcal{B}$ to recover $X_{\mathcal{A}}$ values: $X_{\mathcal{A}} = \mu + N_{\mathcal{A}}$. It is worth noticing that $N_{\mathcal{A}}$ transmission on a public channel does not give any information to $\mathcal{E}$ provided that $\mu_{\mathcal{E}}$ (mean value computed by $\mathcal{E}$) is different from μ, that is, $\mathcal{E}$ is experiencing a completely different behavior of the channel due to her position respect to $\mathcal{A}$ and $\mathcal{B}$.

A quantization function ends the symbols generation phase (Fig. 10), where both $\mathcal{A}$ and $\mathcal{B}$ use their RSS values m_j and the recovered ones m_j^r, respectively, to generate the codeword S_j. Each sensor uses a multi-level quantization scheme of width q_w, in order to generate a codeword of k elements $S = \{S_1, S_2, \ldots, S_k\}$ to correct the RSS differences between $\mathcal{A}$ and $\mathcal{B}$. An example of how the codeword generation helps to correct channel asymmetries is shown in Figure 12: let us consider a simple configuration with 4 only quantization levels (bins), S_1 , S_2 , S_3 , and S_4; each RSS value $X_{\mathcal{A}}$ is converted into the correspondent symbol, and in particular, it is worth noticing how little fluctuations due to the noise factor $N_{\mathcal{A}}$ are completely taken up.

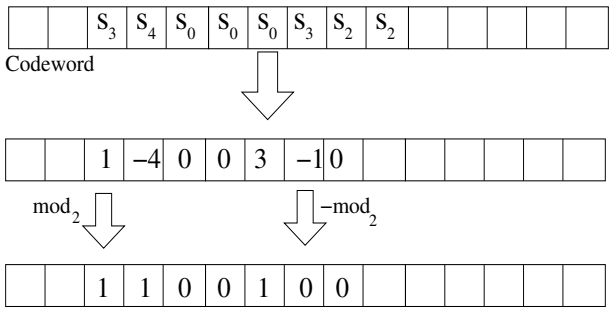


Fig. 13. Symbol translator: the codeword is translated into a string of bits.

C. Entropy Detection Phase

The channel conditions are not always suitable for the key generation and agreement, e.g. stationary periods (like which ones happening at the beginning and at the ending of Fig. 7, 8, 9) cannot be used to extract secrets because $\mathcal{E}$ can easily guess that channel conditions despite of her position and thus guess the key. In order to give a mathematical formulation of "*usable variation of RSS*", we consider the entropy of the k extracted symbols $S = \{S_1, S_2, \ldots, S_k\}$ by the key generation phase:

$$H(S) = -\sum_{i=1}^{k} P(S_i) \log_2 P(S_i), \tag{2}$$

where $P(S_i)$ is the probability mass function of the generated symbol S_i. $H(S)$ is a measure of the channel randomness, and can be used to estimate if the channel conditions are suitable for generating shared secret keys. Let E_{th} be the threshold to overcome in order to accept a key as safe, that is, the generated sequence of symbols S has sufficient randomness to be considered as secure if $E(S) > E_{th}$.

The random symbols S is converted to a binary string by means of a derivative-based algorithm, yielding:

$$K_C = \begin{cases} mod_2(S_i - S_{i-1}) & \text{if } S_i - S_{i-1} \geq 0 \\ -mod_2(S_i - S_{i-1}) & \text{if } S_i - S_{i-1} < 0 \end{cases} \tag{3}$$

Figure 13 shows how the symbol translator works: the codeword S is accepted as safe after the entropy detection phase, a differentiate is performed over the symbol sequence, and finally, Eq. (3) is applied.

D. Verification Phase

The bottom part of Fig 10 shows the key verification phase: $\mathcal{A}$ and $\mathcal{B}$ check if they committed on the same secret. After the entropy detection phase, $\mathcal{A}$ sends the hash value $h(K_C)$ to $\mathcal{B}$. $\mathcal{B}$ computes his hash value $h(K'_C)$, and compares it with that one sent by $\mathcal{A}$. The found shared secret can now be used to support security services, otherwise $\mathcal{A}$ and $\mathcal{B}$ start again with the sampling phase.

VII. The security algorithm

The system is basically constituted by 2 entities: the key generation protocol and a Message Authentication Code (MAC). Sensors pairs are pre-loaded with a shared secret k_0, in this way, $\mathcal{A}$ and $\mathcal{B}$ can authenticate each other

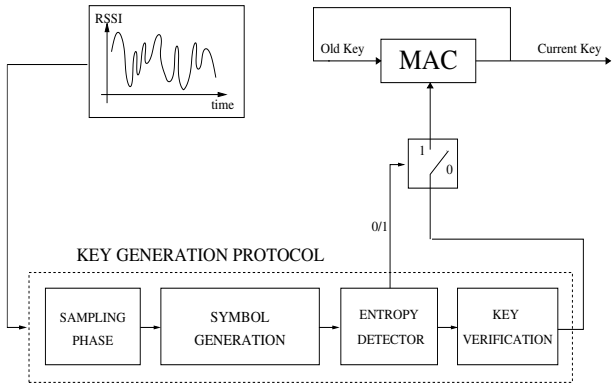


Fig. 14. Key generation algorithm run on each sensor.

during the bootstrap phase. Note that secret extraction from RSS cannot avoid key pre-distribution, in fact $\mathcal{E}$ can try to steal the identity of $\mathcal{B}$ and starting to communicate with $\mathcal{A}$, looking for a successful pairing process with $\mathcal{B}$. For this reason, the system core is basically constituted by a switch: when there are no events in the environment, a classical key generation scheme is adopted, i.e. the old key k_{j-1} is hashed to obtain the actual key k_j. Therefore, a new key is periodically generated in a hash chain fashion, i.e. $k_j = \mathbf{H}(k_{j-1})$, where $\mathbf{H}(\circ)$ is a cryptographic hash function like RC5 [28]. Otherwise, the verification process inhibits a different key generation process.

The new key is now generated as $k_j = \mathbf{MAC}(k_{j-1}, K_C)$, where K_C is the key generated from the channel entropy, and $\mathbf{MAC}(\circ)$ is a CBC-MAC [29]. The subsequent key is generated according to the traditional scheme, re-switching to the hash chain generation algorithm. It is worth noticing the extremely opportunistic nature of the proposed approach, i.e. only channel conditions that allow entropy harvesting are took into account, and managed as a source of entropy to generate new keys. Figure 14 shows the key generation algorithm run on each sensor. The output of the *entropy detector* is always 0 except when a sufficient entropy is experienced in the channel, than the output is set to 1. When the channel entropy goes back to the steady state, the output is reset to 0. When no source of entropy comes from the channel, the $\mathbf{MAC}(\circ)$ is short-circuited, and the key generation process relies on a hash chain. When sufficient source of entropy is detected on the channel, a new secret key K_C is computed by the *key generator*, and the current key k_j is a function of the old key k_{j-1} and the secret key K_C.

VIII. Evaluation of the Key Extraction and Agreement Algorithms

This section presents the performance of the proposed system in terms of key extraction and agreement frequency between $\mathcal{A}$ and $\mathcal{B}$, hereafter F_a, and between $\mathcal{A}$ and $\mathcal{E}$, hereafter F_g. The perfect solution must be oriented to maximize the former and minimize the latter, and eventually proposing a set of optimal parameters for making our system working with best performance.

The frequency transmission has been fixed to 1 packet every 100ms, the secret key length to 32 bits, and finally, the entropy threshold $E_{th} = 0$. It is worth noticing how the previous parameters have been chosen: (i) transmission

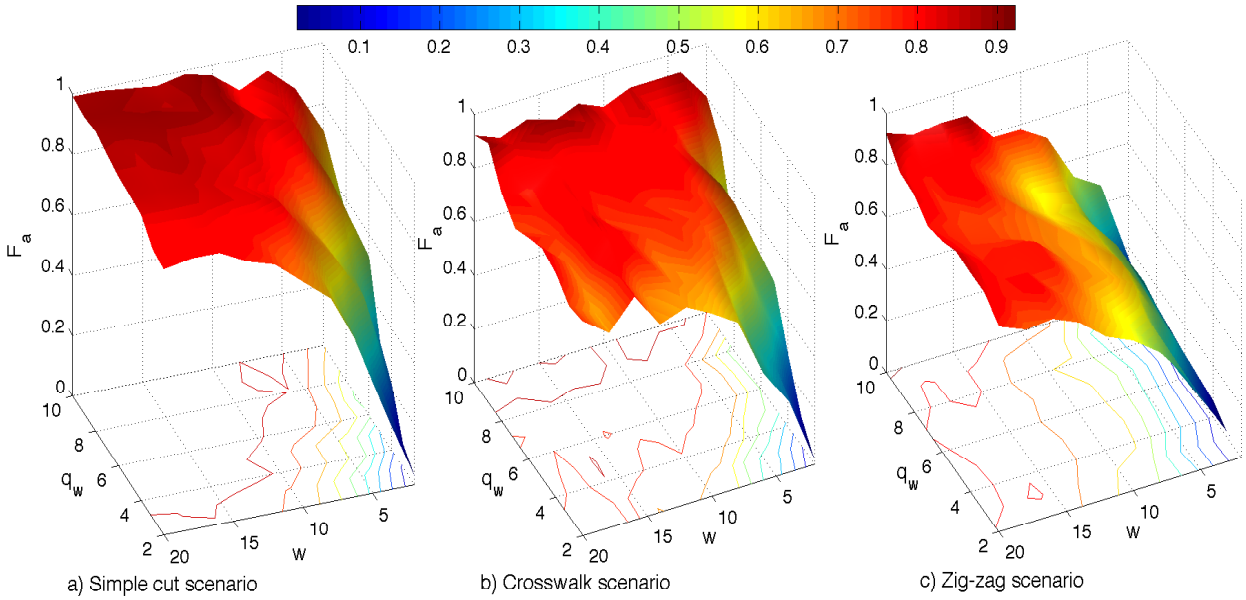


Fig. 15. Secret key agreement frequency as function of q_w and w for the different Lo-S scenario.

timing has been selected as frequent as possible in order to collect as many RSS values as possible in the shortest time period: this does not affect the protocol performance, in fact our analysis depends only on the RSS values and not in the transmission timings; (ii) key length is a critical parameter: longer keys means more security but also more packets to transmit in order to collect a sufficient high number of RSS values. In this work we consider 32 bits key [**?**] but loprotnger keys are easily achievable collecting more RSS samples.

A. Secret Key Agreement Performance

Two parameters affect the proposed system: w, that is the number of samples used to compute the μ value (see Fig. 11), and the bin width, hereafter q_w, used to perform the quantization of the RSS values (see Fig. 12). Figure **??** shows the shared secret generation frequency experienced during the measurement campaign performed in the environments of Fig. 7, 8, 9, respectively. The shapes are basically constituted by two regions: a flat area where the $F_a > 0.8$, and a steep area where shared secret generation is difficult and eventually impossible ($F_a < 0.1$). The influence of the two parameters, w and q_w, is almost the same: reducing the window length to be used for computing the RSS mean values (w assumes smaller values) is actually the same of decreasing the q_w parameter (larger number of bins). It is worth noticing how all of 3 scenarios behave almost in the same way independently from the disturbing event dynamics.

B. $\mathcal{E}$ in Action: Shared Secret Guessing Frequency

The effectiveness of $\mathcal{E}$depends on w and q_w parameters used by $\mathcal{A}$ and $\mathcal{B}$ to perform the shared secret agreement but also in her position.

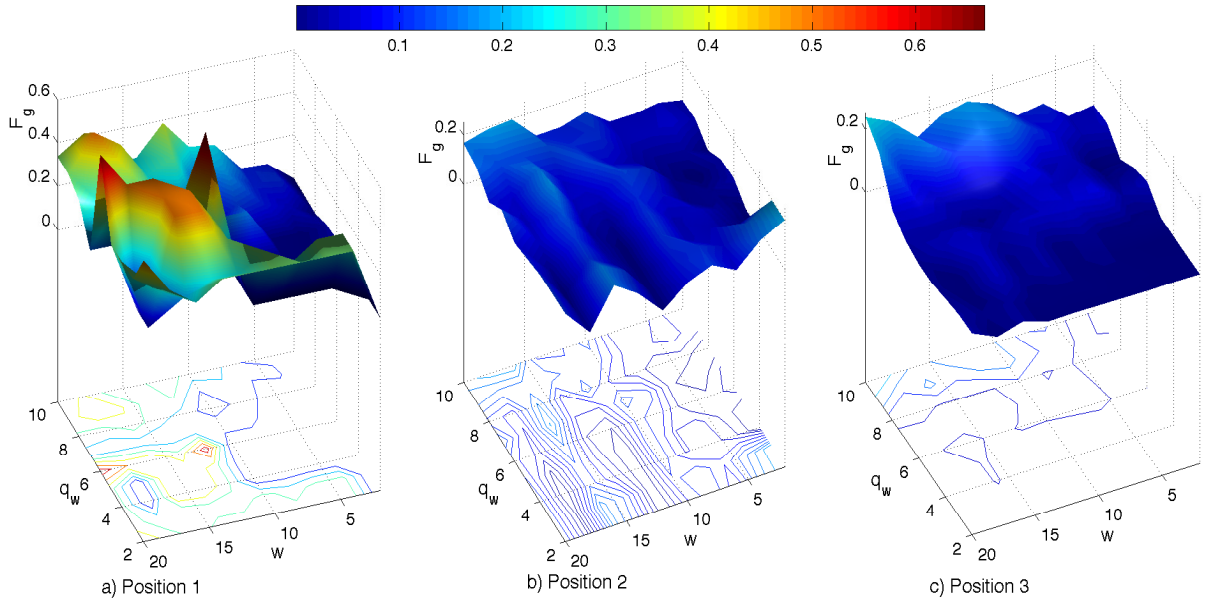


Fig. 16. Shared secret guessing frequency as function of q_w and w for different positions of $\mathcal{E}$.

Firstly, w and q_w parameters are used by $\mathcal{A}$ and $\mathcal{B}$ to correct RSS asymmetries, i.e. to be more tolerant to environmental noise, but the previous settings may result on $\mathcal{E}$ to have the same perspective of the channel, and consequently guessing the shared secret key.

Secondly, the position of $\mathcal{E}$is definitely the only parameter she is able to control in the environment, therefore this work takes into account a set of measurements results in which the position of $\mathcal{E}$changes, and in particular, is getting closer and closer to $\mathcal{A}$, (See Fig. 5). Figure 16 shows the shared secret guessing frequency F_g considering the 3 different positions for $\mathcal{E}$. Note that no difference has been made among the Lo-S breaking scenarios: $\mathcal{E}$ is blind and she is not able to control the environment, therefore for each position of $\mathcal{E}$ put together all the collected data in all the 3 different Lo-S breaking scenario.

Position 1 is the best one from $\mathcal{E}$ perspective: putting $\mathcal{E}$ closer and closer to the position of $\mathcal{A}$ increases the frequency F_g to guess the shared secret key. The previous can be explained observing that a shorter distance between $\mathcal{A}$ and $\mathcal{E}$ gives to the last one higher chances to experience almost the same channel fluctuations of $\mathcal{A}$. We suggest a safe privacy region of 1 meter all around the position of $\mathcal{A}$ that bounds F_g to 0.2.

Finally, it is worth noticing how it is possible to mitigate the $\mathcal{E}$ effectiveness when her position is the closest to $\mathcal{A}$ (position 1, Fig. 16(a)). In that case, $\mathcal{E}$ can even guess the key with a frequency of 0.8, but choosing properly w and q_w parameters can reduce her effectiveness significantly. In particular, using $w = 10$ and $q_w = 8$ bounds F_g to 0.1 and allows to achieve an overall shared secret with $F_a > 0.8$ (Fig. **??**).

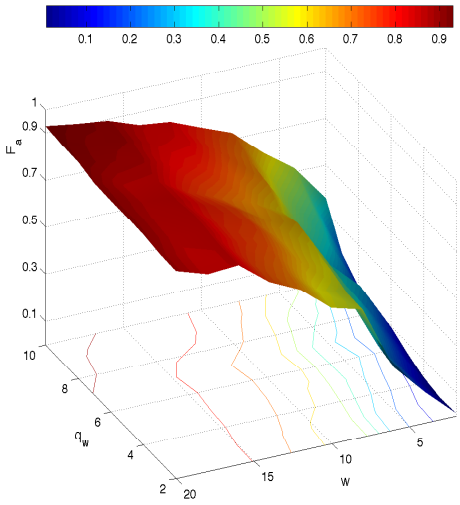


Fig. 17. Secret key agreement in a crowded office environment.

IX. The Showcase: $\mathcal{E}$ against $\mathcal{A}$ and $\mathcal{B}$ in a Crowded Environment

Let us now consider a real crowded environment where the disturbing event $\mathcal{D}$ is not alone and is not moving in a controlled way: Figure 5 shows a real-world scenario with 5 persons moving randomly. Figure 17 shows the secret key agreement frequency F_a in a crowded office environment as function of w and q_w parameters. F_a performance are a little bit worst respect to that ones experienced in the controlled environments of Fig. **??**. For example, in this case using a $w = 10$ may guarantee only $F_a \approx 0.7$, whereas in the previous cases $F_a \approx 0.8$.

Key robustness basically relies only on symbol randomness and consequently on channel fluctuations. Therefore, key randomness depends only on q_w parameter and not w, that is, only the number of bins influences the randomness. Figure 19 shows key agreement frequency F_a and the entropy estimation as function of the q_w parameter fixing the $w = 10$. The solid blue line represents F_a which is monotonic increasing when q_w increases, that is, choosing larger bins introduces a higher noise tolerance, and therefore, increases the probability to perform key agreement. Whereas, key entropy (solid green line) is monotonic decreasing, i.e. larger bins produces key symbols that are almost the same.

Figure 18 shows F_g considering the 3 different positions of $\mathcal{E}$. As in the previous case, $\mathcal{E}$ is not able to guess the key when she is in positions 2 and 3, but her effectiveness in position 1 is significantly reduced respect to the controlled environment (Fig. 16). Actually, the increased noise in the room, constituted by 5 persons walking around plays against the adversary, reducing her capability to experience the same channel variations of $\mathcal{A}$.

X. Discussion

Extracting shared secrets from RSS fluctuations is a challenging issue that can be definitely solved by leveraging environmental dynamics.

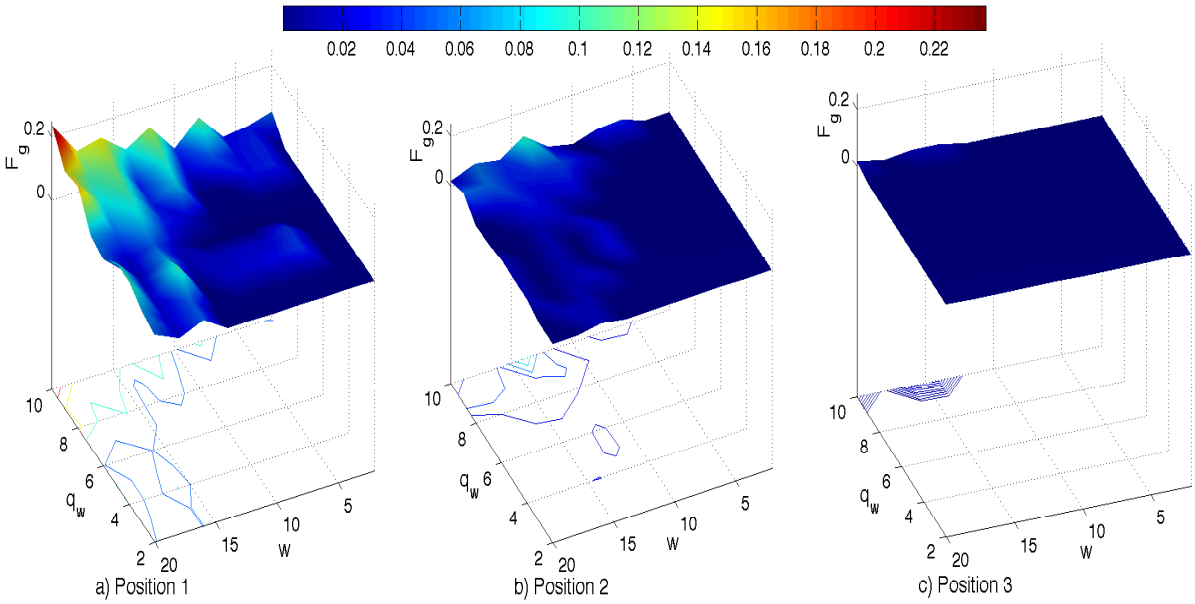


Fig. 18. Secret key guessing in a crowded office environment.

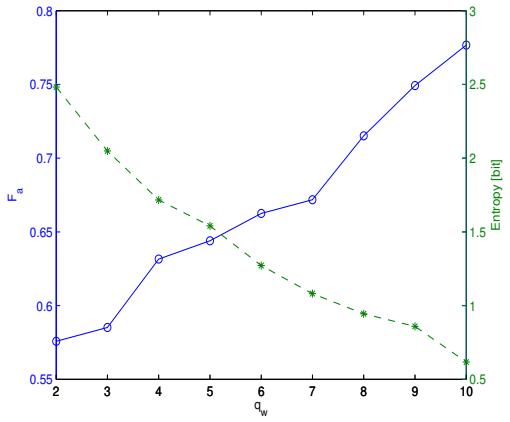


Fig. 19. Secret key agreement and entropy as function of the q_w parameter.

This work does not deal with completely static environments, since, as it has been showed, while being simple they are not random enough, i.e., although the key agreement can be performed easily, the generated key lacks randomness due to the insufficient changes experienced on the channel.

Dynamic environments are more suitable to extract randomness from RSS values, in particular, two important aspects have been identified: (i) event occurrences behave differently in the randomness generation, and (ii) the position is a key point in the adversary effectiveness.

The more the environment is noisy (in terms of Lo-S breaking), the less the key agreement is frequent, that is, the

flat red area (high rate key agreement) reduces itself, passing from the simple cut, to the crosswalk, and finally, to the zig-zag scenario: Fig. **??** (a), (b), and (c), respectively. The crowded environment is actually the worst in terms of frequency agreement.

Nevertheless, noisy environments protect $\mathcal{A}$ and $\mathcal{B}$ secret agreement procedure, in fact, comparing Fig. 16 with Fig. 18, it can be seen as the adversary effectiveness is significantly reduced: the few peaks in Fig. 16(a) completely disappear from Fig. 18(a).

Clearly, adversary effectiveness is influenced by her position, both Fig. 16 and Fig.18 show that a closer adversary position can slightly increase her performance. The measurement results suggest to defined a privacy region of about 1 meter all around the position of $\mathcal{A}$. In fact, using $w = 10$, $q_w = 8$, and placing $\mathcal{E}$ out of the privacy region, the secret key guessing frequency is almost 0 and the key agreement frequency is higher that 0.7. It is worth noticing that the proposed system eventually guarantees no chances for the adversary to guess the key in a real environment. Actually, key agreement frequency performance is not a critical parameter for the system, in fact key refreshment is guaranteed not only by the randomness harvesting but also by the MAC re-keying procedure.

XI. Conclusions and Future Works

This article presented a new cyber-physical approach in securing wireless sensor networks and proposed a novel secret key generation protocol for performance-limited wireless sensors. The proposed protocol leverages both environmental changes and channel reciprocity to allow sensor pairs to commit on mutually generated secrets. While previous results demonstrate the feasibility of generating secret keys from channel state fluctuations and erratic, yet reciprocal signal behavior, they also suffer from either limited key sizes or assumptions of highly mobile devices. This work showed that even static WSNs can generate fresh secrets without requiring mobility, but instead by leveraging dynamics resulting from the physical environment, especially from human movements. Hence, this work contributes to a wide range of scenarios where the sensor networks are deployed as a part of ambient intelligence applications, such as assisted living and building monitoring applications, i.e., where the physical environment is affected by frequent human movements.

We want to stress the extreme importance of disturbance events for the proposed solution: how it has been shown, static environments are not well suited for this approach, the generated keys appears with no entropy and therefore easily to guess, on the contrary noisy scenarios, like a crowded office, produces high, fast and therefore unpredictable fluctuations in the received power that can be leveraged to produce shared secrets.

Using extensive real-world measurements, it has been demonstrated how key generation can be successfully performed in different controlled scenarios, and finally, in a realistic environment such as a crowded office.

To analyze the security of the proposed scheme a new RSS-based attack has been defined and insights on guessing effectiveness under real-world adversary have been provided.

Future works will focus on how the proposed system behaves in a non-Lo-S environment and investigate the relation between the entropy and the key agreement probability. Yet, further analysis on how radio source interferences could influence secret key agreement will be undertaken.

REFERENCES

[1] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y.-F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards," *Computer Communications*, vol. 30, no. 7, pp. 1655–1695, 2007.

[2] M. Girolami, S. Lenzi, F. Furfari, and S. Chessa, "Sail: A sensor abstraction and integration layer for context awareness," in *EUROMICRO-SEAA*, 2008, pp. 374–381.

[3] A. Perrig, J. Stankovic, D. Wagner, and C. Rosenblatt, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, pp. 53–57, 2004.

[4] A. Wyner, "The wire-tap channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[5] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 339–348, 1978.

[6] U. Maurer, "Protocols for Secret Key Agreement by Public Discussion Based on Common Information," in *Advances in Cryptology — CRYPTO '92*, ser. Lecture Notes in Computer Science, vol. 740. Springer-Verlag, Aug. 1993, pp. 461–470.

[7] U. Maurer and S. Wolf, "Secret-Key Agreement Over Unauthenticated Public Channels - Parts I-III," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 822–851, April 2003.

[8] U. Maurer, R. Renner, and S. Wolf, "Unbreakable keys from random noise," in *Security with Noisy Data*, P. Tuyls, B. Skoric, and T. Kevenaar, Eds. Springer-Verlag, 2007, pp. 21–44.

[9] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[10] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel," in *MobiCom '08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2008, pp. 128–139.

[11] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust Key Generation from Signal Envelopes in Wireless Networks," in *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*. New York, NY, USA: ACM, 2007, pp. 401–410.

[12] T. Aono, K. Higuchi, M. Taromaru, T. Ohira, and H. Sasaoka, "Experiments of IEEE 802.15.4 ESPARSKey (Encryption Scheme Parasite Array Radiator Secret Key) - RSSI Interleaving Scheme," in *IEICE Tech. Rep.*, vol. 105, Kyoto, April 2005, pp. 31–36.

[13] R. Wilson, D. Tse, and R. A. Scholtz, "Channel Identification: Secret Sharing using Reciprocity in Ultrawideband Channels," in *ICUWB '07: IEEE International Conference on Ultra-Wideband*, sep 2007, pp. 270–275.

[14] K. Zeng, D. Wu, A. Chan, and P. Mahapatra, "Exploiting Multiple-Antenna Diversity for Shared Secret Key Generation in Wireless Networks," in *Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM)*, San Diego, CA, Mar. 2010.

[15] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2009, pp. 321–332.

[16] M. Wilhelm, I. Martinovic, and J. B. Schmitt, "Secret keys from entangled sensor motes: implementation and analysis," in *Proceedings of the Third ACM Conference on Wireless Network Security (WiSec)*, Mar. 2010, pp. 139–144.

[17] R. B. Lee, D. K. Karig, J. P. McGregor, and Z. Shi, "Enlisting hardware architecture to thwart malicious code injection," in *Security in Pervasive Computing*, ser. Lecture Notes in Computer Science, vol. 2802. Springer Berlin / Heidelberg, 2004, pp. 170–179.

[18] M. Milenkovi, "Hardware support for code integrity in embedded processors," in *in The 2005 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*. ACM Press, 2005, pp. 55–65.

[19] J. Zhang, M. H. Firooz, N. Patwari, and S. K. Kasera, "Advancing wireless link signatures for location distinction," in *MobiCom '08: Proceedings of the 14th ACM international conference on mobile computing and networking*. New York, NY, USA: ACM, 2008, pp. 26–37.

[20] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "Virtual calibration for rssi-based indoor localization with ieee 802.15.4," in *Communications, 2009. ICC '09. IEEE International Conference on*, june 2009, pp. 1 –5.

[21] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 8, pp. 4024 –4036, august 2009.

[22] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 1, pp. 41–52, 2007.

[23] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-Rate Uncorrelated Bit Extraction for Shared Secret Key Generation from Channel Measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.

[24] Crossbow, "http://www.xbow.com."

[25] D. Gay, M. Welsh, P. Levis, E. Brewer, R. von Behren, and D. Culler, "The nesc language: A holistic approach to networked embedded systems," in *In Proceedings of Programming Language Design and Implementation (PLDI*, 2003, pp. 1–11.

[26] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," in *in Ambient Intelligence*. Springer Verlag, 2004.

[27] K. Papagiannaki, M. Yarvis, and W. S. Conner, "Experimental characterization of home wireless networks and design implications," in *In Proc. of Infocom 2006*, 2006.

[28] R. L. Rivest, "The rc5 encryption algorithm." Springer-Verlag, 1995, pp. 86–96.

[29] U.S. National Institute of Standards and Technology (NIST), "Des model of operation." Federal Information Processing Standards Publication 81 (FIPS PUB 81).