COKE
Crypto-less Over-the-air Key Establishment

Roberto Di Pietro [†], Gabriele Oligeri [*]
[†] Dipartimento di Matematica, Università di Roma Tre, Roma, Italy
[*] Dipartimento di Ingegneria e Scienza dell'Informazione, Università di Trento, Trento, Italy

***Abstract*—In this paper we present a novel probabilistic protocol (COKE) to allow two wireless communicating parties to commit over-the-air (OTA) on a shared secret, even in the presence of a globally eavesdropping adversary. The proposed solution leverages no crypto but just plaintext message exchange. Indeed, the security of the solution relies on the difficulty for the adversary to correctly identify, for each one-bit transmission, the sender of that bit—not its value, which is indeed exchanged in cleartext. Due to the low requirements of COKE (essentially, the capability to send a few wireless messages), it is particularly suited to resource constrained wireless devices (e.g. WNSs, wireless embedded systems), as well as for those scenarios where just energy saving is at premium, such as smartphones.**

For instance—under a security challenging setting—it is possible for the two parties to commit, with a probability greater than $(1 - 2^{-48})$, on a 128 secret bits key exchanging as few as 1,300 one-bit messages. Under less stressing conditions, establishing a 128 secret bits key with the same probability requires just as few as 300 one-bit messages. The security parameters (that is, secret-key length and its establishment probability) are adjustable: they can be agreed between the two parties for each key establishment session. A thorough analysis characterizing the security of the protocol, its robustness to packet loss, and energy consumption is provided. Finally, extensive simulations results support the quality and viability of the proposal.

I. Introduction

Secure key establishment between two parties can be addressed when public key infrastructure (PKI) or an online trusted third party (TTP) is available. If not, another option could be to use the Diffie-Hellman solution [1]. Unfortunately, these solutions cannot always be applied to resource-constrained devices operating in pervasive environments because of both the lack of either a PKI or a TTP, and the high computation and bandwidth overhead required by asymmetric cryptography.

A preliminary solution addressing the key establishment issue was provided in [2], [3]. The authors resorted to a kinetic procedure: they collect accelerometers data during the shaking of the two devices coupled together: shared secret comes from the accelerometers data that authors proved to be correlated. Unfortunately, such an approach is effective for devices that can be shaken together only. Establishing a new secret without pre-configured information and avoiding asymmetric cryptography is a challenging topic that has been previously undertaken in mainly two ways: leveraging anonymous channels [4] or adopting received signal strength (RSS-based) key establishment protocols [5], [6]. This technique is based on extracting shared secrets from the observation of the RSS values at both the peers. Some issues still have to be resolved, such as asymmetric effects introduced by the multipath fading and the need of a very dynamic environment to generate keys with sufficient entropy [5].

Crypto-less key establishment protocols based on anonymous channels have been introduced for the first time by [4] and subsequently improved by [7]. The main idea relies on establishing a secret key between two peers without using crypto functions but leveraging source indistinguishability of anonymous channels [8]. Anonymous channels guarantee that an adversary cannot identify the actual signal source even if it is able to eavesdrop all the transmitted messages. In such a scenario, two peers can exchange the bits of a secret message but it is not possible for an adversary to associate the current transmitted bit to the actual source. Such crypto-less algorithms are particularly useful in *pairing* of resource-constrained devices due to the fact that they do not rely on battery expensive computations.

The security of these approaches strong relies on the spatial and temporal indistinguishability [8]. However, secret key establishment still remains a real challenge due to the intrinsic lack of security that characterizes the wireless channel.

Contribution This paper presents COKE, a novel probabilistic crypto-less over-the-air key establishment protocol that guarantees secret key establishment in presence of a global eavesdropper without relying on cryptographic primitives. We present a detailed analysis about the adversarial capabilities to discover the exchanged secret key bits. We highlight how the major threat to the security of our proposal is the adversarial distance with respect to the two communicating parties, and we show how the transmission power of cooperating parties can be tuned in order to mitigate the adversarial capabilities. Our theoretical results showing the security, effectiveness, and efficiency of COKE as a crypto-less key establishment algorithm are supported by extensive simulations. Finally, note that other than being computationally efficient, our protocol introduces just a small transmission overhead when compared against other key-establishment solutions, such as the Diffie-Hellman key-exchange protocol. Hence, our proposal is also an ideal candidate for those devices where computing capabilities are not a constraint, but energy consumption is, such as smartphones.

Organization Next section surveys related literature while Section III introduces the adversarial model and a brief overview of our solution. Section IV presents the details of our crypto-less on-the-air key establishment protocol and Section V shows its security analysis. Simulation results and discussion follow in Section VI and VII, respectively. Section VIII introduces practical considerations on the usage of our protocol. Finally, some concluding remarks are reported in Section IX.

II. Related work

Secret key establishment between resource constrained devices realized without making use of crypto functions has been undertaken in mainly two different ways: extracting secret bits by observing the same physical phenomenon or exchanging secret bits anonymously without using crypto functions.

Extracting a shared sequence of bits by observing the same physical phenomenon has been addressed in several papers. In [9][6][10][11] authors prove that received signal power can be used to extract shared secrets between two peers. Multipath fading may introduce asymmetries on the received power but they proposed a few algorithms to recover the errors and, eventually, agree on a shared secret key. In [12] authors proposed to use the accelerometer to verify if the two peers are carried by the same user; while in [2][13] authors proposed to extract shared secrets by observing accelerometers readings, i.e. the proposed algorithms involve to put together the peers, shake them collecting values from the accelerometer, and finally, make a key agreement on the collected values.

Key exchange using key-less cryptography has been early proposed by [4]: authors proposed to generate and distribute a secret key by only hiding the identity of the source (transmitter) and not the content of the packet. Each packet carries only one bit of the secret key and the packet source field is hidden; in this way, the adversary knows the value of the secret bit but she does not know the sender of that bit. In [8], authors proposed a pairing algorithm based on exchanging packets with the source field chosen as a function of the secret bit to share, in this way, only the sender and the receiver know the actual value of the transmitted bit. This approach relies on the anonymity of both the sender and the receiver obtained by shaking the devices, i.e. the signal source cannot be identified by the adversary because randomized by the shaken procedure.

As stated in [8], crypto-less protocols security relies on two main factor: temporal and spatial indistinguishability. While the former can be easily achieved, the latter is more difficult due to the *power analysis* an adversary can perform. While [8] proposes the "shaking" as an effective solution to such analysis attack, in this work we show how random transmission power can definitely improve crypto-less protocols performance even for static devices.

We observe that other techniques have been proposed in order to generate secrets from random sources. An early solution to secret key generation from biometrics and noisy data comes from [14][11]: authors proved how fuzzy extractors can be effectively used in order to achieve a noise-robust secret key generation algorithm. In [15] authors propose a solution for secure key deployment and storage of keys in sensor networks and RFID systems based on the use of Physical Unclonable Functions (PUFs). In [16], authors proposed a generic construction for fuzzy extractors from noisy continuous sources and analyzed the privacy properties of the scheme.

Nevertheless, we want to point out that, although key-generation algorithms are strictly related, secret-key establishment protocols deal with a different issue. While the goal of the former is to robustly generate a secret from a random source, the goal of the latter is to generate a *shared* secret between two peers.

III. System Model

Our scenario is constituted by a couple of fixed radio-equipped devices, hereafter $\mathcal{A}$ and $\mathcal{B}$, and an adversary—hereafter $\mathcal{ADV}$. No assumptions are made on the communication protocols, i.e. IEEE 802.11, IEEE 802.15.4, GSM, Bluetooth.

A. Adversarial model

As for the adversarial *radio eavesdropping capabilities*, she can perform global or local eavesdropping. Actually, the eavesdropping scale does not rely on the adversary resource power, but only on physical constraints: while a global eavesdropper is able to receive packets from all the devices but cannot selectively recognize the signal source (the transmitter is identified by the source field in the packet), the local eavesdropper is able to focus on just one device and be selective despite the other communications. In this paper we deal with a global eavesdropper adversary provided with an *omni-directional antenna*. For the $\mathcal{ADV}$ it is fundamental to know the relative distance against $\mathcal{A}$ and $\mathcal{B}$. Assuming $\mathcal{ADV}$ is aware of the sender transmission power (T), she could compute the distance from the sender resorting to the following formula [17]:

$$d = \frac{\lambda}{(4\pi)}\sqrt{G\frac{T}{R}} \tag{1}$$

where λ is the wave length, G takes into account the characteristics of the transmitter and the receiver antennas, and R is the received power. Equation (1) represents the signal path loss, i.e. the difference between the actual transmitted and received power, in a free space environment. Actually, more suitable models have been proposed to estimate the path loss for clear-outdoor [18], urban [19], and indoor environments [20] taking into account noise effects due to multipath fading. As it will be clear in the sequel, noise effects do help the establishment of a secret key-bit. However, we take a conservative stance on security, and we will not take into consideration signal propagation issues, while assuming a worst case scenario, that is the free space waves propagation—this model representing the best operating condition for our types of adversary. Further, we will also abstract from physical antenna issues—such as physical constraints due to gain, direction, and polarization features—[21], but we consider the ideal omni-directional (isotropic) antenna characterized by a uniform radiation power over one plane. Finally, in order to reduce the error the adversary could incur in when estimating the distance from the sender (due to lack of knowledge of the exact power transmission), we give $\mathcal{ADV}$ the further advantage of being aware of the distances between her and $\mathcal{A}$ (d_{EA}), and between her and $\mathcal{B}$ (d_{EB}).

Yet, we want to stress that the envisaged adversary is able to eavesdrop both all the communications and the distance between itself and the transmitter, while $\mathcal{ADV}$ has to perform an educated guess as for the direction of the signal propagation. In fact, this latter information can be retrieved with certainty only using a directive antenna, but such a kind of configuration is out of the scope of this paper.

B. Our solution in brief

We propose a probabilistic key establishment protocol (COKE) which does not rely either on complex crypto functions or heavy computations, while requiring only a few cleartext messages to be exchanged between the parties. These features make it an ideal candidate for low computational power devices. Moreover, it works under a powerful adversary model; that is, a global eavesdropper.

We illustrate how COKE protocol works focusing on the establishment of a single secret bit. Let us assume $\mathcal{A}$ and $\mathcal{B}$ randomly generate one bit each: $\mathcal{A}$:$\{0\}$ and $\mathcal{B}$:$\{1\}$, respectively. In a time slot $\mathcal{A}$ and $\mathcal{B}$ will try to send this bit to the other party via a single message (randomizing the source and the receiver addresses of the exchanged packets, should this information be necessary).

We assume that peers are loosely time-synchronized and time is divided into slots [22][23], and in each slot a device can perform (at most) one transmission. Further, to make the slot contention fair, at the beginning of each slot each peer waits for a random time before trying to send its bit.

Let us assume $\mathcal{B}$ is the party able to send its bit (0). This because $\mathcal{B}$ generated a random wait less than the random wait experienced by $\mathcal{A}$. Now, $\mathcal{A}$ is well aware of the fact that the received bit was sent by $\mathcal{B}$. Hence, that bit will be stored by both the peers and will constitute the established one secret bit. Note that if the sender would have been $\mathcal{A}$, the bit would have been *complemented* and later stored. Details are exposed in Algorithm 1.

For the adversary it is difficult to guess how the exchanged bit (that we assume she is able to eavesdrop) will be stored by both the peers. Indeed, the value corresponding to the bit has been stored as transmitted (or complemented) uniquely depending on the ID of the sender. The rationale behind our solution is that the adversary is not able to discriminate between the identities of the two parties participating to the protocol, while each party is aware about the identity of the transmitter. Hence, after having iterated this procedure K times, $\mathcal{ADV}$ could have some 2^K combinations to explore to find the key. Unfortunately, leveraging knowledge of the distance from each of the peers, $\mathcal{ADV}$ is able to perform an educated guess on the ID of the sender (given the received signal power). Therefore, some of the K bits locally stored by the peers as shared secret will be known to $\mathcal{ADV}$ as well.

In the sequel of the paper we will detail the COKE protocol for secret key establishment and will provide a thorough analysis characterizing the chances for the adversary to compromise the secrecy of the established shared key. Further, we will derive the number of transmissions needed to amplify the uncertainty of the adversary up to the point we can establish at least k secret bits with a desired probability. Simulation results will confirm the effectiveness and efficiency of the proposed protocol.

C. On channel anonymity

The security of COKE strictly depends on one factor: *the channel anonymity*. In fact, for each correct guess of the message source $\mathcal{ADV}$ discloses one secret key bit. There are two main attacks that can be performed against channel anonymity: *position guessing*, by measuring the received signal strength, and *radio source identification* by means of physical layer analysis. Both the previous attacks aim at discriminating the peers involved in the key-establishment protocol, i.e., given an eavesdropped message, the challenge is constituted by guessing the actual source.

While we postpone the analysis of the *position guessing* issue to the following sections, in the following we argue why radio source identification in wireless scenarios is not feasible—according to the current literature. Radio source identification, by the analysis of the physical layer, turned out to be feasible and effective for HF RFID [24] and Ethernet cards [25]. Conversely, wireless radio devices features, e.g. 802.11b [26] [27], appeared to be easily reproducible. In fact, source identification is mainly performed by observing two radio features: the transient and the modulation. While the transient-based techniques consist of observing unique features during the transient phase when the radio is turned on, modulation-based techniques rely on imperfections in the modulator of the radio transceiver, such as frequency and constellation symbol deviations. Authors in [26] showed how both modulation-based and transient identifications can be impersonated with a high accuracy (close to 100%) by simply modifying and replaying the used features with a universal software radio peripheral (USRP).

In this work, given the above evidence, we assume that the identities of both $\mathcal{A}$ and $\mathcal{B}$ cannot be detected by simply analyzing their physical radio behaviors: in fact, we consider both $\mathcal{A}$ and $\mathcal{B}$ as provided with USRPs; therefore, they can randomly change their radio features, avoiding any radio fingerprinting by malicious eavesdroppers.

Finally, we notice that an adversary provided with directional antennas can violate the secret key established with COKE. Indeed, $\mathcal{ADV}$ may easily guess the secret key by targeting $\mathcal{A}$ with one directive antenna, and $\mathcal{B}$ with a second one. Although this attack is effective for two peers, it becomes not feasible when the number of deployed nodes increases—the number of directive antennas increases proportionally with the number of targets.

IV. Secret key establishment

In this work we assume all the communication packets are anonymized, i.e. the source and the destination addresses (if needed) does not reveal the real sender. For instance, a sender could randomly decide whether to use its own ID or the receiver ID to fill in the sender field. However, the actual choice is out of the scope of this paper. Further, we can assume the peers $\mathcal{A}$ and $\mathcal{B}$ are able to estimate the minimum transmission power p_m which allows them to communicate each others—this could be easily achieved leveraging a dicothomic search based on varying the transmission power.

Key establishment protocol is constituted by two algorithms: Algorithm 1 and 2 show the sender and receive side protocols instanced for $\mathcal{A}$. Each node chooses a random power level T to transmit the secret bit. The transmission power T is chosen at random in the range $\{p_m, \ldots, p_M\}$, i.e. each node chooses a random transmission power between the minimum (guaranteeing the peer communication) and the maximum. We want to highlight that the protocol can be enriched in order to account for different sessions and (within a session) for packet loss, e.g., each exchanged packet could carry a session ID and a sequence number.

Algorithm 1: Secret key establishment between $\mathcal{A}$ and $\mathcal{B}$: Secret bits transmission — $\mathcal{A}$ side.

```
let $H(\cdot)$ be a cryptographic hash function.
let $K_A \xleftarrow{\$} [0,1]^K$.
let $T$ be the current transmission power.
let $p_m$ be the minimum transmission power which allows
communication between the peers.
let $p_m, \ldots, p_M$ be the power transmission levels.
/* $\mathcal{A}$ ($\mathcal{B}$) randomly chooses the transmission power
    */
$T \xleftarrow{\$} [p_m, \ldots, p_M]$;
for $i \leftarrow 1$ to $K$ do
    /* Select a random waiting time within the
       slot: Elapsed is True when such time is
       elapsed, False otherwise.                    */
    while not Received AND not Elapsed do
        skip
    end
    if not Received then
        /* Extract the $i^{th}$ bit                 */
        $b \leftarrow K_A[i]$;   /* if $\mathcal{B}$ : $b \leftarrow K_B[i]$;   */
        $K_s[i] = b$;
        Send $< \neg b >$;   /* if $\mathcal{B}$ : Send $< b >$;   */
    end
    /* Wait for the current slot to expire.         */
    $Elapsed = FALSE$; $Received = FALSE$ ;
end
/* Shared secret key generation                     */
$K_s$ = H ($K_s$);
```

Algorithm 2: Secret key establishment between $\mathcal{A}$ and $\mathcal{B}$: Secret bits reception — $\mathcal{A}$ side.

```
let $K_s$ be the shared secret key.
/* $\mathcal{A}$ ($\mathcal{B}$) receives the secret bit $b$      */
Receive $< b >$;
$Received = TRUE$ ;
$K_s[i] = b$;   /* if $\mathcal{B}$ : $K_s[i] = \neg b$;      */
```

The rationale of the protocol is to start from two different random keys, K_A and K_B, to converge to a shared (partially) secret key K_s such that $K_s[i]$ is either $\neg K_A[i]$ or $K_B[i]$—depending on the random wait experienced within the time slot. Each loop (i) of Algorithm 1 enables the exchange of a secret bit—stored in $K_s[i]$. As for Algorithm 2, receive procedure is asynchronously invoked by the physical layer.

The security relies on the fact that for each eavesdropped bit, $\mathcal{ADV}$ can only make a probabilistic choice about the originator of that bit. However, $\mathcal{ADV}$ could correctly impute the eavesdropped bit to the correct sender, and hence deriving the value of $K_s[i]$. As it will be clear in the following, although the probability to guess the actual transmitter is high, there is still a not negligible probability for $\mathcal{ADV}$ not to guess the sender. Repeated communications serve to amplify the number of not corrected guesses performed by $\mathcal{ADV}$, that is, to increase the security of the shared key. We prove that the generated key can be considered secure for a wide range of system parameters.

Finally, the shared secret key that will be used to secure following communications is obtained by hashing the previously established shared secret key K_s. Hashing is needed just to reduce the key-length, while not sacrificing security. Indeed, to establish a k bits secret shared key, $\mathcal{A}$ and $\mathcal{B}$ are required to exchange more than k bits, given the fact that a fraction of the exchanged bits will be likely guessed by $\mathcal{ADV}$—a tight estimation of the number of bits to be exchanged in order to achieve K secret bits is provided in the following.

Note that COKE needs just one hash computation for each shared secret key; such a computation introduces very low overhead [28][29][30].

It is worth noticing that Algorithm 1 allows to exchange K (possibly) secret bit between the parties. However, as it will be clear in the following, $\mathcal{ADV}$ could have a not negligible probability to derive an exchanged secret bit. Hence, the number of transmissions needed to actually exchange the required number of secret bits, will be greater than K—as detailed later on.

Finally, looking at Algorithm 1, it could appear more efficient to exchange more than one bit for each communication. However, note that for a given transmission $\mathcal{ADV}$ does not hold the same key-bit the two parties commit on if $\mathcal{ADV}$ does not guess the identity of the transmitter. Indeed, note that the key-bit the two parties commit on is dependent on the identity of the transmitter, but independent from the actual value of the transmitted bit—this also accounts for the fact that the bit is transmitted in clear-text. Therefore, even sending multiple bits with the same message, this action would not increase the contribution—with respect to single bit transmission—to the establishment of the secret key.

Dealing with packet loss: Wireless communication channels are prone to packet loss, and this may prevent secret establishment in COKE. In particular, if one packet (bit) gets lost due to a wireless channel impairment, $\mathcal{A}$ and $\mathcal{B}$ loose their synchronization, and this will prevent the generation of the shared secret K_s. In order to make COKE robust to packet loss, we suggest to protect the keys with an erasure code such as Reed-Solomon (RS) [31]. RS can correct up to $\frac{n-k}{2}$ wrong symbols, where k and n are the data-word and codeword length in symbols, respectively, and finally, $n = 2^m - 1$ where m is the symbol depth in bits.

Typical values for (n, k) are $(255, 223)$: a data-word (K_a or K_b in our case) of up to 1784 bits is encoded in a code-word of 2040 bits robust to the lost of up to 32 symbols (256 bits).

Nevertheless, RS could not be feasible in particularly noisy environments with high packet loss, where more computationally efficient erasure codes are available [32].

V. Analysis

Figure 1 shows $\mathcal{A}$ and $\mathcal{B}$ positions with their radio coverages, i.e. a single dot dashed line is associated to the minimum transmission power (p_m), while a double-dot dashed line is associated to the maximum transmission power (p_M). In the following, we assume that the minimum transmission power p_m allows $\mathcal{A}$ ($\mathcal{B}$) to communicate with $\mathcal{B}$ ($\mathcal{A}$). Let $C(p_m, \mathcal{A})$ and $C(p_M, \mathcal{A})$ be the coverage areas associated to the minimum and maximum transmission powers generated by $\mathcal{A}$, respectively. Let also $C(p_m, \mathcal{B})$ and $C(p_M, \mathcal{B})$ be the coverage areas associated to the minimum and maximum transmission powers generated by $\mathcal{B}$, respectively. Let $A : \{x_a, y_a\}$ and $B : \{x_b, y_b\}$ be the coordinates of $\mathcal{A}$ and $\mathcal{B}$, respectively. To ease exposition, we only analyze the half-plane with $x \leq 0$, and we divide it in 5 different regions:

$$\begin{aligned}
a_5 &: \{(x,y) \text{ s.t. } x \leq 0, x \geq x_a, (x,y) \in C(p_m, \mathcal{A})\} \\
a_4 &: \{(x,y) \text{ s.t. } x \leq 0, (x,y) \in (C(p_m, \mathcal{A}) - C(p_m, \mathcal{B}))\} \\
a_3 &: \{(x,y) \text{ s.t. } x \leq 0, (x,y) \in (C(p_M, \mathcal{B}) - a_4)\} \\
a_2 &: \{(x,y) \text{ s.t. } x \leq 0, (x,y) \in (C(p_M, \mathcal{A}) - (C(p_M, \mathcal{B})\} \\
a_1 &: \{(x,y) \text{ s.t. } x \leq 0, (x,y) \notin a_2\}
\end{aligned}$$

Table I shows the different configurations $\mathcal{ADV}$ has to deal with when taking into account her position belonging in scenario of Fig. 1. Recalling that both $\mathcal{A}$ and $\mathcal{B}$ have a minimum and a maximum transmission power p_m and p_M, respectively, $\mathcal{ADV}$ receive $\mathcal{A}$ and $\mathcal{B}$'s signals as a function of her distance from the signal source. When $\mathcal{ADV}$ belongs to the a_1 region the communication is secure, (a_1 : [NOT, NOT]), i.e. $\mathcal{ADV}$ does not receive any signals from any sources; if $\mathcal{ADV}$ belongs to a_2 region she could *possibly* receive a signal from $\mathcal{A}$ (P in the table) but not from $\mathcal{B}$; eventually, when $\mathcal{ADV}$ gets closer to the axis origin (region a_5) she receives the signals from both the sources ($\mathcal{A}$ and $\mathcal{B}$). Let us define R_A and R_B the powers estimated by $\mathcal{ADV}$ on the signals received by $\mathcal{A}$ and $\mathcal{B}$, respectively [17]; yielding:

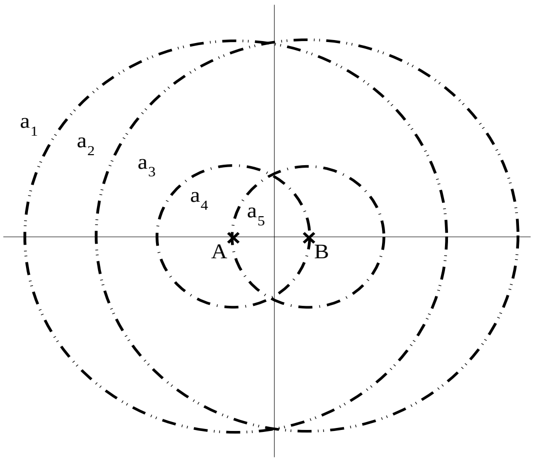


Fig. 1. Minimum and maximum transmission powers.

TABLE I
Areas and eavesdropping capabilities

	$\mathcal{A}$	$\mathcal{B}$
a_1	NOT	NOT
a_2	P	NOT
a_3	P	P
a_4	YES	P
a_5	YES	YES

$$R_A = G\frac{T_A}{d_{EA}^2}, \quad R_B = G\frac{T_B}{d_{EB}^2} \tag{2}$$

where T_A and T_B are the transmission powers of $\mathcal{A}$ and $\mathcal{B}$, respectively, G takes into account antenna gains and physical constants, and finally d_{EA} (d_{EB}) represents the distance between $\mathcal{ADV}$ and $\mathcal{A}$ ($\mathcal{B}$).

Adversarial strategy: In the following, we assume $\mathcal{ADV}$ is aware of her relative distance from $\mathcal{A}$ (d_{EA}) and $\mathcal{B}$ (d_{EB}). Node $\mathcal{A}$ ($\mathcal{B}$) needs to avoid to be identified by $\mathcal{ADV}$ as the sender of the message on the basis of a simple source identification based on the analysis of the received power. Hence, $\mathcal{A}$ ($\mathcal{B}$) selects T_A (T_B) uniformly at random between p_m and p_M, that is $p_m \leq T_A(T_B) \leq p_M$. We highlight that the communication between $\mathcal{A}$ and $\mathcal{B}$ is still guaranteed because the minimum transmission power is p_m—enough to allow both peers to communicate to each other. Let $\overline{T} = E[T_A] = E[T_B]$ be the mean transmission power, i.e. $\overline{T} = \frac{P_M + P_m}{2}$. The mean power received by $\mathcal{ADV}$ follows:

$$\overline{R}_A = G\frac{P_M + P_m}{2d_{EA}^2} \quad \overline{R}_B = G\frac{P_M + P_m}{2d_{EB}^2}$$

We assume $\mathcal{ADV}$ selects the *strategy* that maximizes her chances to guess the source of the communication. To this goal, the decision is taken as "the source is $\mathcal{A}$" when the current received power by $\mathcal{ADV}$ is $R < R_{th}$, while "the source is $\mathcal{B}$" when $R > R_{th}$, where R_{th} is defined as the log-average between $\overline{R}_A$ and $\overline{R}_B$, that is:

$$R_{th} = 10^{\frac{\log \overline{R}_A + \log \overline{R}_B}{2}} = \frac{G}{2} \frac{p_M + p_m}{d_{EA} \cdot d_{EB}} \tag{3}$$

Figure 2 resumes $\mathcal{ADV}$ guessing strategy: $\mathcal{ADV}$ knows the mean received powers $\overline{R}_A$ and $\overline{R}_B$ from $\mathcal{A}$ and $\mathcal{B}$, respectively; therefore, she splits the received powers into two sets: if the current received power belongs to the left side ($R < R_{th}$), $\mathcal{ADV}$ takes the decision "the current signal source is from $\mathcal{B}$"; on the contrary, if the current received power belongs to the right side ($R > R_{th}$), $\mathcal{ADV}$ takes the decision "the current signal source is from $\mathcal{A}$".

Definition. We define *probability to experience a secret one-bit transmission*, hereafter P_{sb}, as the error probability related to the educated guess that $\mathcal{ADV}$ performs in deciding the transmitter identity.

Let S be the actual signal source and $S_{\mathcal{ADV}}$ the purported signal source by $\mathcal{ADV}$. The secret-bit transmission probabilities between $\mathcal{A}$ and $\mathcal{B}$ can be computed as follows:

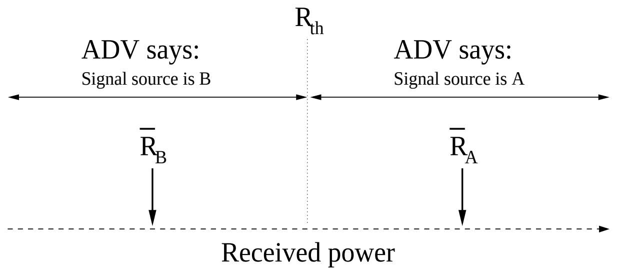


Fig. 2. Signal source guessing depends on the current received power and R_{th}.

$$\begin{aligned}
P_{sb}(\mathcal{A} \to \mathcal{B}) &= P(S_{\mathcal{ADV}} = \mathcal{B} \mid S = \mathcal{A}) \cdot P(S = \mathcal{A}) \\
&= P(\Phi_A) \cdot P(S = \mathcal{A}) \\
P_{sb}(\mathcal{B} \to \mathcal{A}) &= P(S_{\mathcal{ADV}} = \mathcal{A} \mid S = \mathcal{B}) \cdot P(S = \mathcal{B}) \\
&= P(\Phi_B) \cdot P(S = \mathcal{B})
\end{aligned}$$

where $P(\Phi_A)$ ($P(\Phi_B)$) is the probability that $\mathcal{ADV}$ does not correctly guess the source as $\mathcal{A}$ ($\mathcal{B}$).

Finally, assuming both the peers have a fair channel contention, i.e $P(S = \mathcal{A}) = P(S = \mathcal{B}) = \frac{1}{2}$, the secret-bit transmission probability can be written as:

$$P_{sb} = P_{sb}(\mathcal{A} \to \mathcal{B}) + P_{sb}(\mathcal{B} \to \mathcal{A}) = \frac{1}{2} \cdot P(\Phi_A) + \frac{1}{2} \cdot P(\Phi_B) \quad (4)$$

Recalling $\mathcal{ADV}$ strategy and Fig. 2, the probability $P(\Phi_A)$ of secret-bit transmission between $\mathcal{A}$ and $\mathcal{B}$, assuming $\mathcal{A}$ as the current source and R_A the current received power by $\mathcal{ADV}$, can be rewritten as:

$$P(\Phi_A) = P(R_A < R_{th})$$

Recalling Eq. (2), $P(\Phi_A)$ can be rewritten as:

$$P(\Phi_A) = P\left(G\frac{T_A}{d_{EA}^2} < R_{th}\right) = P\left(T_A < \frac{d_{EA}^2}{G} R_{th}\right)$$

Now, let $T'_A = \frac{d_{EA}^2}{G} R_{th}$ and recalling T_A is uniformly distributed between p_m and p_M, i.e. $p_m \leq T_A \leq p_M$, it follows:

$$P(\Phi_A) = \frac{T'_A - p_m}{p_M - p_m} \quad (5)$$

Yet, recalling Eq. (3), it follows:

$$T'_A = \frac{d_{EA}^2}{G} R_{th} = \frac{d_{EA}}{d_{EB}} \frac{p_M + p_m}{2} \quad (6)$$

Finally, recalling from Eq. (5) that $T'_A \geq p_m$, and therefore $\frac{d_{EA}}{d_{EB}} \geq \frac{2p_m}{p_M + p_m}$, the final expression of $P(\Phi_A)$ can be drawn combining Eq. (5) and Eq. (6), yielding:

$$P(\Phi_A) = \begin{cases} \frac{1}{2} \cdot \frac{d_{EA}}{d_{EB}} \cdot \frac{p_M + p_m}{p_M - p_m} - \frac{p_m}{p_M - p_m} & \text{if } \frac{d_{EA}}{d_{EB}} > \frac{2p_m}{p_M + p_m}, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The previous procedure can also be applied to estimate $P(\Phi_B)$. Recalling that:

$$P(\Phi_B) = P(R_B > R_{th})$$

it can be rewritten as:

$$P(\Phi_B) = \frac{p_M - T'_B}{p_M - p_m} \quad (8)$$

TABLE II

BREAKPOINTS: bp_A AND bp_B

p_M	$2 \cdot p_m$	$5 \cdot p_m$	$10 \cdot p_m$	$100 \cdot p_m$
bp_A	0.66	0.33	0.18	0.02
bp_B	0.75	0.6	0.55	0.50

where:

$$T'_B = \frac{d_{EB}^2}{G} R_{th} = \frac{d_{EB}}{d_{EA}} \frac{p_M + p_m}{2} \quad (9)$$

Finally, combining Eq. (8) and Eq. (9) it yields:

$$P(\Phi_B) = \begin{cases} \frac{p_M}{p_M - p_m} - \frac{1}{2} \cdot \frac{d_{EB}}{d_{EA}} \cdot \frac{p_M + p_m}{p_M - p_m} & \text{if } \frac{d_{EB}}{d_{EA}} < \frac{2p_m}{p_M + p_m}, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

We highlight that $\frac{d_{EA}}{d_{EB}} = 1$ means that $\mathcal{ADV}$ is either at equal distance from both peers (that is, on the y axis of Fig. 1) or quite far from both peers $\left(\frac{d_{EA}}{d_{EB}} \approx 1\right)$.

Figure 3 shows $P(\Phi_A)$, $P(\Phi_B)$, and P_{sb} computed using Eq. (7), Eq. (10), and Eq. (4), respectively. In particular, recalling the scenario introduced by Fig. 1, we have that Fig. 3(a), Fig. 3(b), Fig. 3(c), and Fig. 3(d) show the theoretical trends computed considering $p_M = 2 \cdot p_m, 5 \cdot p_m, 10 \cdot p_m$, and $100 \cdot p_m$, respectively, while p_m has been chosen in order to enable $\mathcal{A}$ and $\mathcal{B}$ to communicate to each other.

It is worth noticing how P_{sb} is characterized by two breakpoints: $\frac{d_{EA}}{d_{EB}} = bp_A$, i.e. $P(\Phi_A)$ becomes greater than 0, and $\frac{d_{EA}}{d_{EB}} = bp_B$, i.e. $P(\Phi_B)$ becomes greater than 0. The first breakpoint can be computed enforcing $P(\Phi_A) = 0$ in Eq. (7), yielding:

$$bp_A = \frac{2p_m}{p_M + p_m}$$

Analogously, bp_B can be computed enforcing $P(\Phi_B) = 0$ in Eq. (10), yielding:

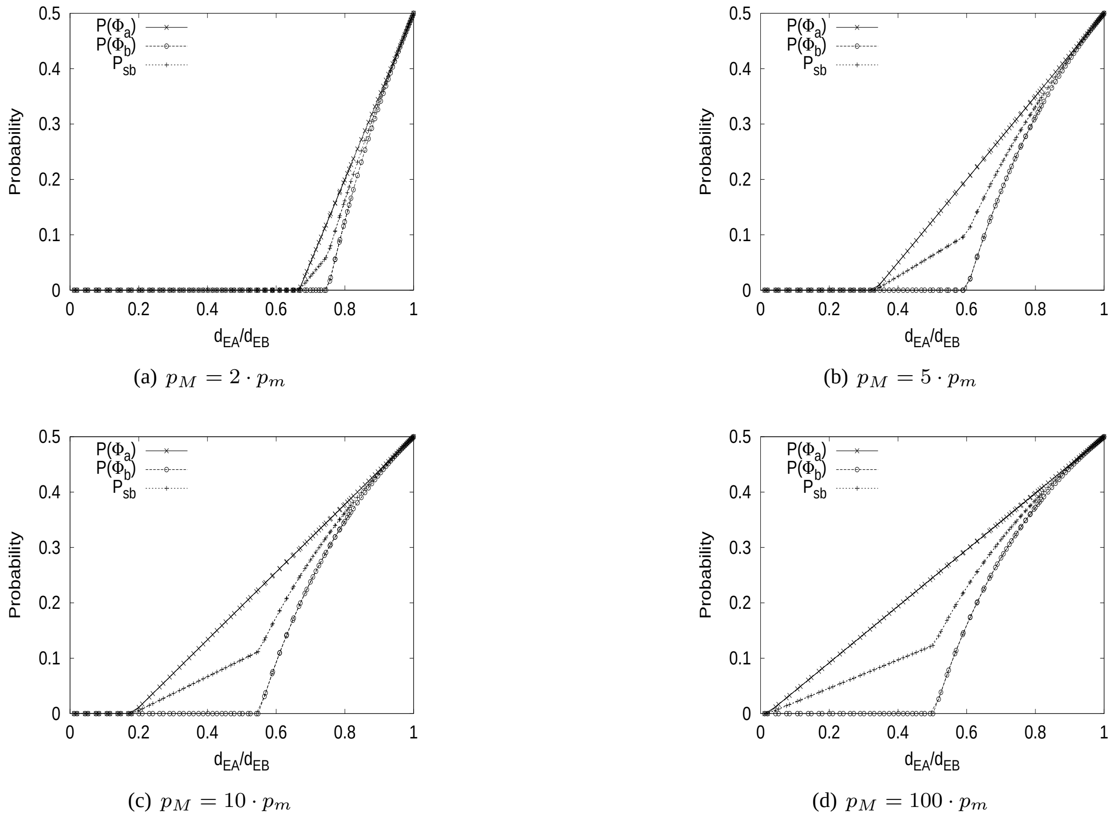
$$bp_B = \frac{p_M + p_m}{2p_M}$$

For example, recalling Fig. 3, where $p_M = 5 \cdot p_m$, $b_A \approx 0.33$ and $bp_B = 0.6$. Table II shows all the breakpoints computed for all the maximum transmission powers we are considering in this work.

We want to highlight the importance of the two values bp_A and bp_B: recalling Fig. 3, it can be noticed how increasing the maximum transmission power p_M has a positive effect on the secret-bit transmission probability—mainly due to the left-shift of both the bp_A and the bp_B values.

VI. Simulation results

In this section we present the simulation results for a randomly placed $\mathcal{ADV}$ in each of the areas of Fig. 1. In particular, for each maximum transmission power $p_M = 2 \cdot p_m, 5 \cdot p_m, 10 \cdot p_m, 100 \cdot p_m$ we simulated 200 $\mathcal{ADV}$'s positions and 5000 random source transmissions for each of them. Note that maximum transmission power $p_M = 100 \cdot p_m$ results on a power variation of almost 20dBm which is currently achievable by the modern communication technologies such as GSM/UMTS/3G networks [33].



(a) $p_M = 2 \cdot p_m$

(b) $p_M = 5 \cdot p_m$

(c) $p_M = 10 \cdot p_m$

(d) $p_M = 100 \cdot p_m$

Fig. 3. Theoretical trends associated to the secret-bit transmission.

For all the simulations, we deployed the two peers at the coordinates $\mathcal{A}$:[-50, 0], $\mathcal{B}$:[50, 0] of a flat grid (centered at [0,0]), and we fixed the minimum transmission power p_m to enable $\mathcal{A}$ and $\mathcal{B}$ to communicate to each other. In the following we present simulation results when $\mathcal{ADV}$ is positioned in the areas $a_1, \ldots a_5$ as *per* Fig. 1.

A. Area 5 and 4

We observe that areas a_5 and a_4 are characterized by $0 \leq \frac{d_{EA}}{d_{EB}} \leq 1$, i.e. $\mathcal{ADV}$ can overlap $\mathcal{A}$'s position ($\frac{d_{EA}}{d_{EB}} = 0$) and $\mathcal{ADV}$ can reach the y axis ($\frac{d_{EA}}{d_{EB}} = 1$). Figure 4 shows the theoretical and simulated P_{sb} when $\mathcal{ADV}$ belongs to either area a_5 or area a_4 of Fig. 1. In particular, theoretical curves are computed according to Eq. (4).

B. Area 3

We observe that area a_3 is characterized by $0.5 \leq \frac{d_{EA}}{d_{EB}} \leq 1$, i.e. $\mathcal{ADV}$ can reach the y axes ($\frac{d_{EA}}{d_{EB}} = 1$) but the minimum value for the ratio $\frac{d_{EA}}{d_{EB}}$ turns out to be $\frac{p_m}{2p_m} = 0.5$. In such case $\mathcal{ADV}$ stays on the x axis on the border between areas a_3 and a_4. Figure 5 shows the simulation results related to the secret-bit transmission probability when $\mathcal{ADV}$ belongs to the area a_3 and $p_M = 2 \cdot p_m, 5 \cdot p_m, 10 \cdot p_m, 100 \cdot p_m$.

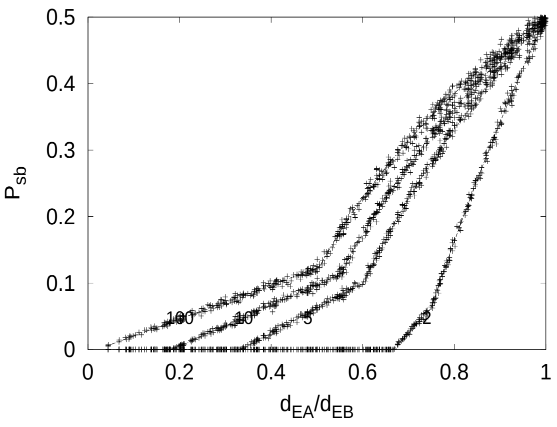


Fig. 4. Secret-bit transmission probability when $\mathcal{ADV} \in \{a_5, a_4\}$ with $p_M \in [2, 5, 10, 100] \cdot p_m$.

C. Area 2

We observe that when $\mathcal{ADV} \in a_2$, she can reach the y axis ($\frac{d_{EA}}{d_{EB}} = 1$) but the minimum value for $\frac{d_{EA}}{d_{EB}}$ ratio, hereafter x_L, depends on the maximum transmission power p_M: if $p_M = 2p_m$ than $x_L = \frac{p_m}{2p_m} = 0.5$ but when $p_M = [5, 10, 100] \cdot p_m$ than $x_L = \frac{p_M - p_m}{p_M}$. Table III shows the numerical values for x_L as a function of the maximum transmission power p_M. Fig. 6 shows the simulated results related to the secret-bit transmission probability when $\mathcal{ADV}$ belongs to the a_2 area.

D. Area 1

When $\mathcal{ADV}$ belongs to area a_1, she cannot receive any signal from either $\mathcal{A}$ or $\mathcal{B}$. Therefore, all the communications

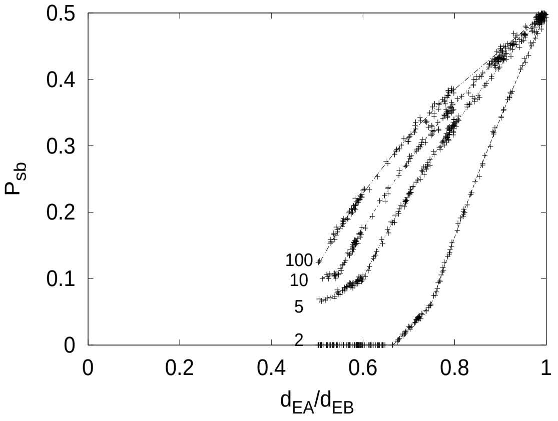


Fig. 5. Secret-bit transmission probability when $\mathcal{ADV} \in a_3$ with $p_M \in [2, 5, 10, 100] \cdot p_m$.

TABLE III

x_L VALUES AS FUNCTION OF p_M

p_M	$2 \cdot p_m$	$5 \cdot p_m$	$10 \cdot p_m$	$100 \cdot p_m$
x_L	0.5	0.8	0.9	0.99

between $\mathcal{A}$ and $\mathcal{B}$ are secret. The secret bit communication probability yields from Eq. (4): $P_{sb} = \frac{1}{2}$.

VII. Discussion

Simulation results presented in Section VI perfectly fit the theoretical analysis resumed by Fig. 3. Yet, we want to highlight two critical parameters:

- The ratio d_{EA}/d_{EB}
  The proposed protocol is based on $\mathcal{ADV}$'s inability to guess, from the received signal power, the actual signal source. Therefore, d_{EA} and d_{EB} are critical parameters. In particular, we analyzed the ratio d_{EA}/d_{EB}, and we observed that secret key establishment has maximum performance when $d_{EA}/d_{EB} \approx 1$ (maximum uncertainty). This occurs when either $\mathcal{ADV}$ stays at the same distance from both peers or she is quite far from both the peers, i.e. $d_{EA} \approx d_{EB}$. However, key establishment security decreases as $\mathcal{ADV}$ gets closer to one of the two peer, i.e. $d_{EA}/d_{EB} \approx 0$.
- The maximum power p_M
  Maximum transmission power can be leveraged to increase the probability of secret-bit transmission: even if $\mathcal{ADV}$ gets closer to one of the two peers, increasing p_M can (probabilistically) hide the actual signal source. For instance, recalling Fig. 3, fixing $\frac{d_{EA}}{d_{EB}} = 0.6$, we observe that $P_{sb} = 0$ when $p_M = 2 \cdot p_m$, $P_{sb} = 0.1$ when $p_M = 5 \cdot p_m$, and finally $P_{sb} = 0.25$ when $p_M = 100 \cdot p_m$.

In the following, we discuss the parameters affecting P_{sb} when $\mathcal{ADV}$ belongs to a specific area. In particular, when $\mathcal{ADV}$ belongs to a_1 there are no parameters relevant to our analysis: $\mathcal{ADV}$ is not aware of any communication. When $\mathcal{ADV}$ belongs to a_2 area, both $\mathcal{ADV}$'s position and maximum transmission power P_M can affect the P_{sb}. It is worth noticing how P_M bounds adversarial proximity to both the peers (Fig. 1). Therefore, increasing P_M compensates for the difference between d_{EA} and d_{EB}, helping to reach maximum anonymity; in other words, increasing P_M increases x_L in Table III. Finally, when $\mathcal{ADV}$ belongs to either a_3, or a_4, or a_5, the probability of secret-bit transmission P_{sb} depends only on the positioning of $\mathcal{ADV}$ with respect to the positioning of $\mathcal{A}$ and $\mathcal{B}$: $0.5 \leq \frac{d_{EA}}{d_{EB}} \leq 1$ when $\mathcal{ADV}$ belongs to a_3, and $0 \leq \frac{d_{EA}}{d_{EB}} \leq 1$ when $\mathcal{ADV}$ belongs to either a_4 or a_5.

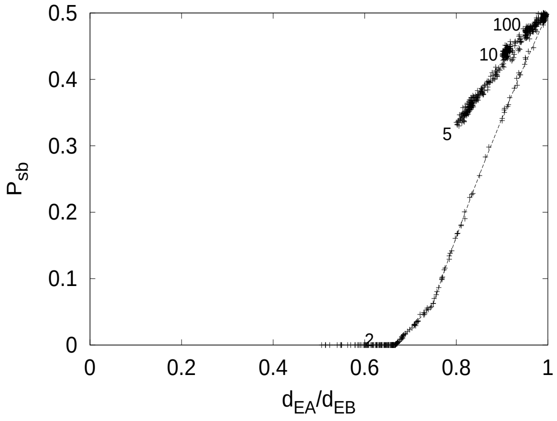


Fig. 6. Secret-bit transmission probability when $\mathcal{ADV} \in a_2$ with $p_M \in [2, 5, 10, 100] \cdot p_m$.

VIII. OTA secret key establishment in practice

In the following we provide the relation between the secret-bit transmission probability and the number of expected transmissions (K) to commit on a shared key K_s; where K_s enjoys at least q secret bits (that is, bits unknown to $\mathcal{ADV}$) with a probability greater than $1 - \epsilon$, for a given ϵ chosen by the parties. In the following we will set $\epsilon = 2^{-48}$.

Let X_i be the r.v. that takes on the value 1 if the i^{th} bit is secret, and 0 otherwise. Then, $X = \sum_{i=1}^{K} X_i$ is the r.v. counting the secret transmissions.

We are interested in setting the value K s.t. we can obtain $X \geq q$ (X being the number of bits unknown to $\mathcal{ADV}$ the secret key is made of) with probability greater than $(1 - \epsilon)$.

By the linearity of expectation, we have:

$$E[X] = \mu = K \cdot P_{sb} \tag{11}$$

Since the X_i are independent and identically distributed (i.i.d) random variables we can leverage the central limit theorem and assume that X can be approximated via the normal distribution, that is $X \sim \mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are the mean and the variance of the r.v. X, respectively. Under such hypothesis, we leverage the three-sigma rule [34] in order to bound the number of transmissions needed to commit on a secret shared key. A simple calculation shows that the sigma-variation to be taken into account to satisfy the required probability turns out to be eight, yielding:

$$P(\mu - 8\sigma \leq X \leq \mu + 8\sigma) = erf\left(\frac{6}{\sqrt{2}}\right) > 1 - 2^{-48}$$

Figure 7 shows the mean number of transmissions in order to commit on a 64, 96, and 128 secret-bit key length, respectively,

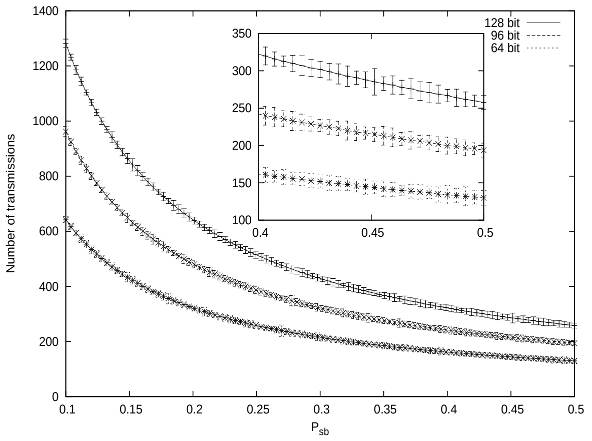


Fig. 7. Number of transmissions to commit on a 64, 96, and 128 secret-bit key length, respectively, as function of the P_{sb}.

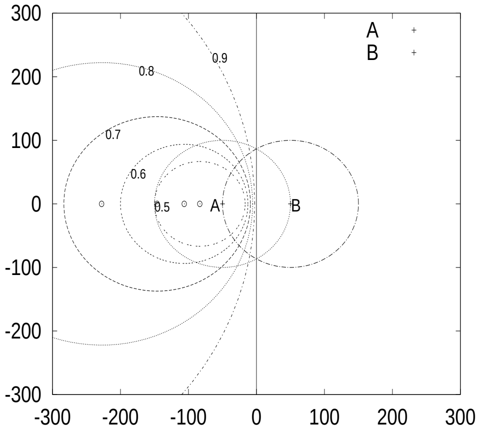


Fig. 8. Adversarial positions and the ratio $\frac{d_{EA}}{d_{EB}}$.

as function of the P_{sb}. In detail, errorbars in Fig. 7 show the eight-sigma variation ($\pm 8\sigma$) around the mean value. This means that with probability at least $(1 - 2^{-48})$ the r.v. X assumes values within $\mu \pm 8\sigma$. We also present a magnification for the interval $0.4 \leq \frac{d_{EA}}{d_{EB}} \leq 0.5$. We observe that errorbars are tight to the mean values and simulated mean number of transmissions perfectly follow Eq. (11) represented by the solid lines.

A. Adversarial positions

In this section we provide a qualitative analysis about the relation between the adversarial position and the ratio $\frac{d_{EA}}{d_{EB}}$. Such relation will be used in the following to provide an estimation of the P_{sb}, and consequently, the minimum number of transmissions needed to commit on a shared secret key.
We observe that $\frac{d_{EA}}{d_{EB}} = \overline{d}$ is the equation of a circumference with center:

$$C_x = \frac{x_a + x_b\overline{d}^2}{1 - \overline{d}^2}; \quad C_y = \frac{y_a + y_b\overline{d}^2}{1 - \overline{d}^2}$$

and radius equal to $\sqrt{C_x^2 + C_y^2 - c}$, where:

$$c = \frac{x_a^2 + y_a^2 - \overline{d}^2(x_b^2 + y_b^2)}{1 - \overline{d}^2}$$

and $[x_a, y_a]$ and $[x_b, y_b]$ are the coordinates of $\mathcal{A}$ and $\mathcal{B}$, respectively.
Figure 8 shows the two peers $\mathcal{A}$ and $\mathcal{B}$ deployed on a [600 × 600] grid with the circles corresponding to their minimum transmission powers, and a set of circles $\frac{d_{EA}}{d_{EB}} = \overline{d}$, with $\overline{d} \in [0.5, 0.6, 0.7, 0.8, 0.9]$. For instance, if $\mathcal{ADV}$ is placed at position [-200, 250], we observe that $0.8 < \overline{d} < 0.9$, and therefore recalling Fig. 3, $P_{sb} \approx 0.4$, when $p_M = 5p_m$. Yet, recalling Fig. 7 we observe that a 64 secret-key length needs about 160 transmissions for the given P_{sb} and transmission power $p_M = 5 \cdot p_m$.

B. Wrap-up

Secret-key generation depends on P_{sb}, that is $\mathcal{ADV}$'s position and maximum transmission power p_M. Recalling Fig. 7, we can consider $P_{sb} = 0.1$ as a lower bound for the key generation: for such value we need about 1280, 960, and 640 transmissions to establish secret-keys of length 128, 96, and 64 bits, respectively. Recalling Fig. 3, it can be observed as $\frac{d_{EA}}{d_{EB}} \approx 0.76, 0.6,$ and 0.4 when $P_{sb} = 0.1$ and $p_M = 2 \cdot p_m, 5 \cdot p_m$, and $100 \cdot p_m$, respectively. Finally, recalling Fig. 8 it can be seen which are the limits for the $\mathcal{ADV}$'s positions for such $\frac{d_{EA}}{d_{EB}}$ values. It is worth noticing how the boundary limit equal to 0.4 has a radius less than the coverage area of $\mathcal{A}$ and $\mathcal{B}$ devices: to achieve such *privacy region* each peer has to be provided with a radio able to transmit with $p_M = 100 \cdot p_m$.
Higher values of P_{sb} increase the protocol performance. When $P_{sb} = 0.4$, then 320, 240, and 160 transmissions are sufficient to establish a secret-key of length 128, 96, and 64, respectively (with probability at least $1 - 2^{-48}$). From Fig. 3 we observe that $\frac{d_{EA}}{d_{EB}} > 0.8$ when $P_{sb} = 0.4$ independently of the maximum transmission power p_M. The *privacy region* correspondent to $\frac{d_{EA}}{d_{EB}} > 0.8$ is shown in Fig. 8: we can consider as safe all the positions outside the circle labeled with 0.8.

C. Energy Consumption

In the following we provide a qualitative comparison, as for energy consumption, between COKE and the standard Diffie-Hellman (DH) key-establishment protocol [1].
Using DH to generate a shared key that could be considered comparable to a symmetric key composed of 80 random bits [35], the prime module p has to be roughly 1000 bits long, and the secret exponent each party randomly generates has to be roughly 160 bits long. Then (assuming that the generator g over the Galois field p has been shared before deployment), each party—we focus on A—needs: to compute one modular exponentiation ($g^{S_a} \pmod{p}$), to send to the other party a message (g^{S_a}) of (roughly) the same length of the modulus, and to compute one modular multiplication ($g^{S_a} g^{S_b} \pmod{p}$) [36]. The most demanding computation is the modular exponentiation that, considering it is performed

according to the Montgomery algorithm [37], in our setting requires about 10^8 elementary computations.

The energy consumption associated to the reception of one bit can be roughly compared to the execution of 100 (10^2) elementary computations in WSN devices (ATMEL 90LS8535) [38]. Hence, DH requires for each party an energy cost associated to transmission equivalent to $10^3 \cdot 10^2 = 10^5$ computations. Therefore, the overall DH energy consumption can be roughly summarized to $2(10^8 + 2 \cdot 10^5) > 2 \cdot 10^8$ computations.

As for COKE, it needs—under stringent security setting—an overall of only 1000 (10^3) transmissions in order to establish a 96 bit secret key. As for computations, they are just associated to the generation of K_a, K_b, and the invocation of a hash functions. Assuming to use a PRNG [39] for the generation of each key, the overall computing cost of the PRNG and the hash function is negligible with respect to the transmission cost. Hence, assuming a packet header length of 7 bytes (56 bits) [8], the overall energy consumption is equivalent to less than $56 \cdot 10^3 \cdot 10^2 \approx 10^7$ computations. Therefore, we observe that COKE is roughly more than 100 times cheaper than DH from the energy consumption perspective—under conservative hypothesis for COKE.

Moreover, COKE is even less demanding than elliptic curve Diffie-Hellman key-establishment (ECDH). In particular, we consider an ECDH implementation based on a 192-bits prime field—providing a security equivalent to our 96 bits symmetric key [40]. ECDH consumes about 57 mJ on ATMEL MICAz architectures [41] considering both computational and communication overheads. The cost for COKE can be summarized by leveraging the transmission and reception per-bit-cost provided by [30]. In detail, assuming a cost of 185 nJ/bit for transmitting and 133 nJ/bit for receiving, COKE needs 17 mJ in order to establish a 96 bits secret key.

Finally, we compare COKE with three crypto-less key-establishment algorithms that leverage received signal strength readings in order to establish a secret key: SHAKE [42], ARUBE [10], and Radio-Telepathy [43]. SHAKE has been proposed as an efficient RSS-based key establishment. Although SHAKE needs only about 345 transmissions in order to establish a secret key of 96 bits, one of the two peers experiences a high computational burden, i.e., more than 10^5 hash computations. ARUBE [10] needs about 1200 transmissions in order to establish a 96 bits key and the computational complexity linearly increases with the length of the key. Finally, Radio-Telepathy [43] needs about 1400 transmissions for establishing a 96 bits length key with a constant computational complexity.

To wrap up, note that the number of messages required by COKE ranges (roughly) between 300 and 1300, according to the desired security requirements, while the computational cost amounts to a single hash computation. Therefore, under all the considered metrics, COKE outperforms competing solutions.

IX. Conclusion

In this work we have introduced COKE, a crypto-less over-the-air key establishment algorithm that allows two peers to commit on a shared secret key without using pre-constituted secrets or asymmetric crypto-functions. COKE requires to exchange a few one-bit messages between the parties for the shared secret to be built and requires only one hash computation for each generated secret key. Given its efficiency, it is particularly suited to resource constrained wireless devices, as well as for those scenarios where energy saving is at premium, such as smartphones. A thorough analysis of the security of the proposed protocol is provided. Further, extensive simulations confirm our findings.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments. Roberto Di Pietro has been partially supported by a Chair of Excellence from University Carlos III, Madrid, Spain. Gabriele Oligeri has been supported by the project Autonomous security, sponsored by the Italian Ministry of Research under the PRIN 2008 Programme. Simulations have been made possible by a HPC 2012 grant from CASPUR.

References

[1] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, November 1976.

[2] D. Kirovski, M. Sinclair, and D. Wilson, “The martini synch: Joint fuzzy hashing via error correction,” in *Security and Privacy in Ad-hoc and Sensor Networks*, ser. Lecture Notes in Computer Science, vol. 4572. Springer Berlin / Heidelberg, 2007, pp. 16–30.

[3] R. Mayrhofer and H. Gellersen, “Shake well before use: Intuitive and secure pairing of mobile devices,” *Mobile Computing, IEEE Transactions on*, vol. 8, no. 6, pp. 792–806, june 2009.

[4] B. Alpern and F. B. Schneider, “Key exchange using keyless cryptography,” Tech. Rep., 1982.

[5] 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 2009*, 2009, pp. 321–332.

[6] M. Wilhelm, I. Martinovic, and J. B. Schmitt, “Secret keys from entangled sensor motes: implementation and analysis,” in *WiSec ’10*. ACM, 2010, pp. 139–144.

[7] M. Young, “A secure and useful keyless cryptosystem,” *Information processing letter*, pp. 35–38, 1985.

[8] C. Castelluccia and P. Mutaf, “Shake them up!: a movement-based pairing protocol for cpu-constrained devices,” in *MobiSys ’05*. ACM, 2005, pp. 51–64.

[9] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, “Exploiting multiple-antenna diversity for shared secret key generation in wireless networks,” ser. INFOCOM’10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 1837–1845.

[10] J. Croft, N. Patwari, and S. K. Kasera, “Robust uncorrelated bit extraction methodologies for wireless sensors,” in *IPSN ’10*. ACM, 2010, pp. 70–81.

[11] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, “Robust key generation from signal envelopes in wireless networks,” ser. CCS ’07. ACM, 2007, pp. 401–410.

[12] J. Lester, B. Hannaford, and G. Borriello, “Are you with me? - using accelerometers to determine if two devices are carried by the same person,” in *Book Series Lecture Notes in Computer Science*, vol. 3001/2004. Publisher Springer Berlin / Heidelberg, pp. 33–50.

[13] 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,” in *UbiComp ’01*. London, UK: Springer-Verlag, 2001, pp. 116–122.

[14] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, Mar. 2008.

[15] J. Guajardo, B. Škorić, P. Tuyls, S. S. Kumar, T. Bel, A. H. Blom, and G.-J. Schrijen, “Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions,” *Information Systems Frontiers*, vol. 11, no. 1, pp. 19–41, Mar. 2009.

[16] E. A. Verbitskiy, P. Tuyls, C. Obi, B. Schoenmakers, and B. Škorić, “Key extraction from general nondiscrete signals,” *Trans. Info. For. Sec.*, vol. 5, no. 2, pp. 269–279, June 2010.

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

[18] P. Barsocchi, G. Oligeri, and F. Potorti, “Measurement-based frame error model for simulating outdoor wi-fi networks,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 3, pp. 1154 –1158, march 2009.

[19] M. Hata, “Empirical formula for propagation loss in land mobile radio services,” *Vehicular Technology, IEEE Transactions on*, vol. 29, no. 3, pp. 317 – 325, aug 1980.

[20] B. Bandemer, C. Oestges, N. Czink, and A. Paulraj, “Physically motivated fast-fading model for indoor peer-to-peer channels,” *Electronics Letters*, vol. 45, no. 10, pp. 515 –517, may 2009.

[21] T. A. Milligan, *Modern antenna design*. McGraw-Hill, Inc., 1985.

[22] J. Elson and D. Estrin, “Time synchronization for wireless sensor networks,” ser. IPDPS ’01. IEEE Computer Society, 2001, pp. 186–.

[23] S. Ganeriwal, C. Pöpper, S. Capkun, and M. B. Srivastava, “Secure time synchronization in sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 11, pp. 23–35, July 2008.

[24] B. Danev, S. Capkun, R. Jayaram Masti, and T. S. Benjamin, “Towards practical identification of hf rfid devices,” *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 2, pp. 7:1–7:24, July 2012.

[25] R. Gerdes, M. Mina, S. Russell, and T. Daniels, “Physical-layer identification of wired ethernet devices,” *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 4, pp. 1339 –1353, aug. 2012.

[26] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, “Attacks on physical-layer identification,” ser. WiSec ’10. ACM, 2010, pp. 89–98.

[27] E. Matthew and Y. Blent, “Active attacks against modulation-based radiometric identification,” Rensselaer Polytechnic Institute, Department of Computer Science, Tech. Rep., 2009.

[28] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Tapiador, and A. Ribagorda, “Information security applications.” Berlin, Heidelberg: Springer-Verlag, 2009, ch. Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol, pp. 56–68.

[29] G. de Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, “On the energy cost of communication and cryptography in wireless sensor networks,” ser. WIMOB ’08. IEEE Computer Society, 2008, pp. 580–585.

[30] D. Singelee, S. Seys, L. Batina, and I. Verbauwhede, “The communication and computation cost of wireless security: extended abstract,” ser. WiSec ’11. ACM, 2011, pp. 1–4.

[31] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Indust. Appl. Math*, vol. 8, pp. 300–304, 1960.

[32] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data,” *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 56–67, Oct. 1998.

[33] C. Joumaa, A. Caminada, and S. Lamrous, “Mobility simulation for the evaluation of umts power control algorithms,” in *New Technologies, Mobility and Security, 2008. NTMS ’08.*, November 2008, pp. 1 –5.

[34] F. Pukelsheim, “The three sigma rule,” in *The American Statistician*, vol. 48, 1994.

[35] H. Orman and P. Hoffman, “Determining strengths for public keys used for exchanging symmetric keys,” United States, 2004.

[36] E. Rescorla, “An introduction to openssl programming,” *Linux J.*, vol. 2001, no. 89, pp. 3–, Sept. 2001.

[37] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1996.

[38] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” *SIGPLAN Not.*, vol. 35, no. 11, pp. 93–104, Nov. 2000.

[39] J. Lan, W. L. Goh, Z. H. Kong, and K. S. Yeo, “A random number generator for low power cryptographic application,” in *SoC Design Conference (ISOCC), 2010 International*, nov. 2010, pp. 328 –331.

[40] A. K. Lenstra and E. R. Verheul, “Selecting cryptographic key sizes,” *Journal of Cryptology*, vol. 14, pp. 255–293, 1999.

[41] C. Lederer, R. Mader, M. Koschuch, J. Grosschädl, A. Szekely, and S. Tillich, “Energy-efficient implementation of ecdh key exchange for wireless sensor networks,” ser. WISTP ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 112–127.

[42] B. Paolo, O. Gabriele, and S. Claudio, “Shake: Single hash key establishment for resource constrained devices,” *Ad Hoc Networks*, 2012.

[43] S. Mathur, N. M, C. Ye, and A. Reznik, “Radio-telepathy: extracting a secret key from an unauthenticated wireless channel,” in *MobiCom*, 2008, pp. 128–139.

Roberto Di Pietro is an Assistant Professor at University of Roma Tre, Mathematics Dept.. He received the laurea degree in computer science from the University of Pisa, Italy, in 1994. He received the Ph.D. in Computer Science at the Sapienza University in 2004. In 2005 he was presented with the ERCIM (European Research Consortium for Informatics and Mathematics) fellowship. In 2008 he spent the period Feb-Sep visiting the UNESCO Chair in Data Privacy at the University Rovira i Virgili (URV). He was appointed Chair of Excellence from University Carlos III, Madrid, for the Academic Year 2011/2012. He has been serving as reviewer for the EU Commission (6th and 7th FP), research-support National Agencies (France, Poland, Cyprus, Singapore), and the ERC. His main research interests include security and privacy of wireless systems, cloud computing virtualization and security, computer forensics, computer and network security, and role mining.

Gabriele Oligeri is a Post-doc at the University of Trento. He received the laurea degree in Computer Engineering from University of Pisa in 2005, and the Ph.D. in Information Engineering from the Engineering Ph.D. school “Leonardo da Vinci” of the same University in 2010. He has been Post-doc at CNR-ISTI for the period 2010-2011. His research interests include security and privacy in distributed systems.