DRAKE: Distributed Relay-Assisted Key Establishment

Savio Sciancalepore*, Roberto Di Pietro*
* Division of Information and Computing Technology (ICT)
College of Science and Engineering (CSE), Hamad Bin Khalifa University (HBKU), Doha, Qatar
{ssciancalepore, rdipietro}@hbku.edu.qa

***Abstract*—In this paper we propose DRAKE, a distributed relay-assisted key establishment protocol working at the physical layer of a wireless network. DRAKE leverages the superposition of the signals emitted by dedicated relays to provide a symmetric key to a remote constrained device, by requiring zero transmissions from this device. Requiring zero transmissions from the remote device, DRAKE emerges as a unidirectional and radio stealthy solution, suitable for a number of applications and scenarios, such as tactical communications, stealthiness, and for devices with a limited transmission range, to name a few. Thanks to the combination of a repetition encoder and further error correction at the relays, DRAKE is able to establish a key with a very low error probability. We discuss the security of the DRAKE protocol against a passive eavesdropper, under different realistic assumptions. Finally, we provide the results of a preliminary performance assessment of the protocol, showing outstanding performance both in terms of resilience to random noise and security against adversaries equipped with multiple omnidirectional antennas.**

I. Introduction

With the emergence of the Internet of Things (IoT), billions of tiny smart objects will invade the commercial market and our homes, thanks to their ability to be wirelessly and remotely controlled to monitor the surroundings and perform fundamental tasks. In the last years, the IoT has even evolved to integrate new emerging technologies, including Unmanned Aerial Vehicles (UAV) and Wireless Body Area Networks (WBAN), to name a few [1].

On the one hand, being very small and constrained, most of the smart devices are equipped with limited memory and computational capabilities. On the other hand, the even higher sensitivity of managed data requires the application of strong and reliable security techniques. This issue emerges especially in the context of key establishment; while this is traditionally achieved through well established public-key cryptography techniques, often these are too computationally and storage expensive to be applied on smart objects [2].

Physical-Layer Security is a research area that tackles the above issues. Specifically, key establishment protocols working at the physical layer aim at providing constrained devices with shared secrets to use for further communications, without leveraging either pre-shared secrets or computationally expensive public-key cryptography solutions [3]. To this aim, they exploit the features of the wireless channel, and leverage physical parameters such as the Received Signal Strength (RSS), the Channel Impulse Response (CIR) and the Time of Arrival (ToA), to name a few (see Sec. II).

However, all these schemes are based on mutual communications between involved devices. These contributory key establishment solutions could be particularly hard to achieve in specific operational conditions involving constrained devices, or they could even be undesired properties. For instance, if one of the devices would like to be stealthy, it could not run the above protocols to obtain a dynamic key. Radio stealthiness is often required in tactical UAV communications, where an UAV would like to keep radio silence, transmitting the minimum number of messages. Moreover, there are scenarios where one of the devices has a limited transmission range with respect to the other party, thus not being able to run an interactive key establishment protocol with the remote party. Similarly, there are scenarios in which a constrained sensor, equipped only with a receiving antenna, needs to acquire dynamically a key to securely store data for different owners. Given that the device does not have any transmission capability, running the above protocols would be impossible. All the scenarios described above require a non-interactive unidirectional key establishment protocol between constrained devices, in which one of the remote parties could obtain a secret key without being involved neither in transmission operations nor in computationally heavy techniques. To the best of our knowledge, such a solution is still not available in the literature.

To solve the above issues, we hereby propose DRAKE, a distributed relay-assisted key establishment protocol working at the physical layer of a wireless network. DRAKE allows a remote constrained device to obtain a symmetric key of the desired size by relying on the superposition of the waveforms generated by dedicated wireless nodes, namely *relays*. The key establishment is achieved only by leveraging waveforms transmitted by the relays, the remote device being totally passive, thus being suitable for the scenarios discussed above. The scheme is enriched with a repetition scheme and further error correction codes added at the relays, achieving a very high reliability.

The security of DRAKE has been evaluated in the presence of a passive eavesdropper, equipped with multiple omnidirectional antennas and unaware of the deployment of the relays. Under these assumptions, the security of DRAKE lies in the impossibility for the eavesdropper to reconstruct the profile of the amplitude of the signal experienced by the remote node, because of the hardness to solve the well-

known Blind Source Separation (BSS) problem. DRAKE emerges also as a post-quantum solution, given that improvements in the computational capabilities due to quantum computing have no effect in solving the BSS issue [4].

The rest of the paper is organized as follows. Sec. II reviews the related work, Sec. III introduces the system and the adversary model, while Sec. IV details all the aspects of DRAKE. Sec. V provides some security considerations, while Sec. VI includes an extensive performance assessment. Finally, Sec. VII tightens conclusions and draws future directions.

II. Related Work

In the recent literature it is possible to find many approaches using the RSS of signals to generate a symmetric key. E.g., the authors in [5] propose a RSS-based key establishment protocol using the distance between two moving objects. However, it requires both of the nodes to perform many messages exchanges, thus allowing an eavesdropper to easily detect the presence of a target device. RSSI reciprocity is used in many works, e.g. [6] and [7], with reference to WBAN and IoT, leveraging the temporal and spatial fluctuations of the wireless channel, antenna diversity and multiple frequencies. With reference to these valuable contributions, DRAKE does not require an interactive real-time communication between involved devices. Moreover, these approaches assume totally flat channels, where the random fluctuations of the wireless medium are completely controlled. DRAKE, instead, is also able to tame the noise, thanks to the addition of the repetition encoder and the further error correction at the relays (see Sec. IV-B).

Many approaches in the literature considered position-based techniques to provide key establishment. These protocols combine physical and cryptographic properties and enable a verifying party to determine an upper-bound on its distance toward a prover, who claims to be within a certain range. A recent example is [8], proposing a position-based key establishment protocol based on time-of-arrival, leveraging messages broadcast by satellites and assuming instantaneous processing time of the messages. However, these approaches are not suitable for our scenario, where the remote device cannot leverage Public Key Infrastructure (PKI)-based solutions. Some recent contributions also proposed non-interactive key establishment protocols [9], [10]. However, they are too computationally expensive to be integrated in constrained devices.

Given that the security of DRAKE lies in the hardness in the resolution of the BSS problem, it is worth mentioning a few contributions that dealt with this issue with reference to RF signals, such as [11] and [12]. These contributions focused on specific conditions where the generic BSS problem can be relaxed, e.g. supposing multivariate signals and instantaneous mixtures of delayed sources, allowing frequency domain identification of emitting sources. DRAKE, instead, assumes the most generic scenario, with instantaneous superposition of indistinguishable sources, where the BSS issue is still an unsolved problem. In this context, it is worth mentioning a recent contribution [13], demonstrating the feasibility of constructive interference schemes in IEEE 802.15.4 networks.

To sum up, to the best of our knowledge, a solution that is able to achieve key establishment for a constrained device, requiring zero transmissions on this device, and a limited computational burden, is still missing, and it is exactly what is provided by DRAKE.

III. System Model and Assumptions

A. System and Adversary Model

The scenario considered in this contribution is depicted in Fig 1.

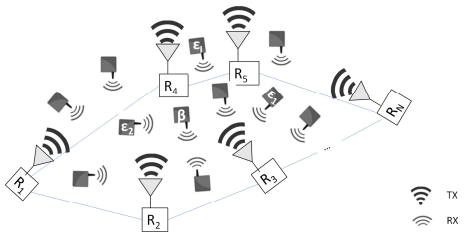

Figure 1. Reference Scenario.

The nodes labeled as $R_1; :::; R_N$ are relay nodes, that are controlled by a central coordinator node. They are dedicated nodes in the network, able to transmit a wireless signal in a given time slot with a pre-defined tunable peak amplitude. We define with R_n the generic n th relay nodes, $n = 1; 2; :::; N$. Each relay node is able to transmit a wireless signal with an instantaneous amplitude level a_n, within the interval fa_{MIN} $a_{MAX}g$ Volts. Without loss of generality, relays can be considered as powerful wireless nodes, without any constraint regarding energy or computational capabilities.

The wireless signals are modulated by the relays using the same carrier frequency, and they are receivable (i.e., clearly distinguishable from the surrounding noise) in a given area, that we define as the *Region of Interest*. The relays are generally supposed to be synchronized (details on the synchronization protocol are outside the scope of this contribution, see [14] for a reference example). In addition, the relays emit the respective signals with the desired initial phase, but using the same modulation frequency. We suppose that it is impossible to identify a specific source of a message from the waveform retrieved over the wireless channel, i.e., any radio fingerprinting technique is meaningless (see Sec. V).

We assume a generic wireless receiver, namely B, positioned in a specific point that lies in the region of interest. To the extent of this paper, B could be: a constrained device, not equipped with the necessary computational capabilities to agree on a shared secret by using PKI-based solutions; or, with a transmission range that is not sufficient—in the scope of contributory key establishment—to reach the other parties of the key establishment protocol; or, it could want to keep temporary radio silence, in order to keep staying

stealthy. B receives a radio signal whose amplitude $c_B(t)$ (directly connected to the RSS) is the superposition of the contributions of the waveforms emitted by the relays, as in the following Eq. 1:

$$c_B(t) = \sum_{n=1}^{N} {}_n a_n(t) + e(t); \quad (1)$$

being ${}_n$ an attenuation factor and $e(t)$ a noise component, whose features depends on the physical environment.

When B wishes to generate a new secret key with a peer, it agrees with the relays on a sequence of time instants and, in case it is a moving node, a series of positions, where it should turn its radio on to listen to the level of incoming signal. Note that this agreement process can happen also offline.

When online and at the agreed positions, the level of the incoming signal heard by B, subtracting the noise, will be controlled by the relays and will translate to a shared secret information known only by the relays and B.

The key establishment procedure happens in the presence of a generic eavesdropper, namely E, whose aim is to reconstruct the level of the signal experienced by B, using a number of antennas spread in the area of interest. It is assumed that E listens to transmissions over the wireless channel and it also has full access to the public channel over which legitimate nodes exchange information. Thus, E is passive, and does not send signals to interfere with legitimate transmissions. The key generation process is not covert, i.e., E knows that there is a key agreement process going on. However, the amount of information she is able to recover is limited, as will be showed in Sec. VI. Without loss of generality, E is assumed to be located at least ${}_s$=2 meters away from the target device. Consistently with the widely accepted principle of spatial de-correlation, over distances of more than half of a wavelength, wireless channel gains decorrelate in multipath fading environments, hence leading to independent observations at the eavesdropper and the legitimate node [15]. Moreover, it is assumed that E could deploy multiple antennas in the area, with the aim of capturing as most samples of power level per round as possible and thus increase its chances to guess on the value of the RSS in the region. In the same way, multiple adversaries can collude by sharing information acquired through the respective antennas. Finally, we assume that E is equipped only with omnidirectional antennas, while she does not resort to directional antennas. We will further discuss in Sec. V the implications and possible drawbacks of using directional antennas to guess the key negotiated through the proposed scheme.

B. Assumptions

We assume a slowly varying channel, holding the same statistical profile during the key establishment procedure. Given the relatively short duration of the whole protocol (few msec, depending on the configuration) and the short distances involved, this is a reasonable assumption, usual in the vast majority of contributions in the literature tackling physical-layer security [5], [16], [17]. Despite the received bit-key information is related to the amplitude of the received signal, DRAKE does not assume the use of any specific modulation scheme. In fact, it is not necessary for B to reconstruct the original information, but only to log the same received signal level (within given bounds) when the relays transmit with a given (fixed) combination of power levels. In addition, the wireless propagation model is intentionally general, and can be easily adapted to the preferred environment [15]. Indeed, for the whole protocol to work, it is only necessary to correctly characterize the model of the signal propagation in the area, consistently with the most part of contributions dealing with physical layer security [17]. Finally, we remark that DRAKE does not deal with authentication. Indeed, the authentication of communicating peers is decoupled from the key establishment, and it can be provided through additional mechanisms, in line with the other contributions working on physical-layer security [5], [18].

IV. The DRAKE protocol

A. DRAKE in a nutshell

DRAKE allows a computational constrained device to obtain a shared key of size K thanks to more powerful remote parties (relays), by leveraging the received signal strength. In addition, DRAKE does not require any transmissions by the remote party during the key establishment phase, thus being suitable for unidirectional applications, where the remote device needs to be stealthy or, it has very limited transmission capabilities, or requires secure data storage.

DRAKE can be triggered by the remote node B, by the relay nodes, on a time base fashion, a combination of these methods, or other forms (e.g., event-based triggering). However, DRAKE is agnostic to any particular solution. Hence, hereby we only assume that a correct trigger has been launched.

On the transmitters side, each relay R_n is equipped with a symmetric key s_n. Each relay knows also the keys $s_1; s_2; \ldots; s_N$ assigned to other relays. By using a Hashed Message Authentication Code (HMAC) function $H(\,)$ over a set of K transmission instants $\mathbf{t_n}$, using the key s_n, the generic n-th relay extracts a sequence of pseudo-random transmission amplitudes $\mathbf{a_n}$, used to transmit over the wireless channel. To provide robustness against random errors on the wireless transmission medium, a repetition code with a factor J is applied (more details on the choice of J will be provided in Sec. IV-E). Thus, the n-th relay performs a total of $M = K \cdot J$ transmissions for each DRAKE instance. On the receiver side, the remote node B simply logs the overall level of the received amplitude, and map this amplitude level on a particular bit. After M rounds, it can decode the received bit-string and obtain a K-bit secret, namely s_B. Thus, it is not necessary for B to reconstruct the level of the signal, but only to log a consistent value, corresponding to a given set of transmission amplitudes used by the relays.

Later on, when B would like to transmit a message or the data owner wants to read the stored data, it unveils

the locally constructed key s_B by computing a verification code $v_B = H(ID_B; s_B)$, using the key s_B and its unique identifier ID_B as the input message. Receiving or reading this message, each relay can compute the effective key s_B in possession of the remote node B, given that it knows the keys of the other relays s_n, $n = [1; 2; :::; N]$ and the set of transmission instants $\mathbf{t_n}$ used for transmissions. Thus, the one hand the n-th relay can verify the correctness of the received verification code v_B. On the other hand, in case up to G bits in s_B have been flipped (due to the noise), each relay can easily detect the occurrence of this situation. This is possible by considering all the possible strings that are obtained by changing up to G bits in the expected key. Note that this operation takes only $= \sum_{g=1}^{G} \binom{K}{g}$ comparisons. Assuming $K = 128$ bits and $G = 2$, the above leads to only 8256 comparisons, that are indeed tolerable for a standard not constrained wireless device. If the relays identify the key effectively recovered by B, this is assumed to be the key s_B associated to B. Otherwise, the relays can run a new instance of the DRAKE scheme.

Algs. 1 and 2 report the pseudo-code of the operations of the Relays and the Remote Node in the DRAKE scheme, respectively.

Input: Receive key generation triggering message.
K, Number of bits in the key to be negotiated;
N, Number of transmitting nodes, i.e., relays ;
J, Number of repetitions of each bit b_k ;
M = $K \cdot J$ total number of transmissions ;
s_n, symmetric key of the n-th node ;
x_n, y_n, x-y coordinates of the n-th node ;
Extract $\mathbf{x_B}$, $\mathbf{y_B}$ and $\mathbf{r_B}$ from the received message ;
Compute the distances vector $\mathbf{d_{n,B}}$;
Compute the set of transmission instants $\mathbf{t_n}$ = { $t_{1,n}$, $t_{2,n}$, ..., $t_{M,n}$ } ;
for *k=1:K* **do**
  for *n=1:N* **do**
    Compute the transmission amplitude $a_{n,k}$ = H($t_{k,n}$; s_n);
    for *j=1:J* **do**
      m = k*J + j ;
      $a^j_{n,k} \leftarrow a_{n,k}$;
      Compute $t^j_{k,n}$;
      Transmit with amplitude $a^j_{n,k}$ at time instant $t_{m,n} = t^j_{k,n}$;
    end
  end
end
Wait for reception of a message containing the verification code v_B ;
if *HMAC (ID_B, s_B) == v_B* **then**
  s_B is the key for B;
end
else
  Try to recover up to $G = 2$ errors by testing $\Delta = \sum_{g=1}^{G} \binom{K}{g}$ combinations ;
  for *p=1:Δ* **do**
    if $v'(p) == v_B$ **then**
      s'(p) is the key for B;
    end
  end
  else
    Restart the DRAKE scheme;
  end
end

Algorithm 1: Pseudo-code of DRAKE - Relay Mode.

Input: Join to the network and request a symmetric key s_B
Receive a *Key Generation Triggering* message, containing the set of reception instants $\mathbf{r_B}$ = { r_1, r_2, ..., r_m, ..., r_M } and the positions vector $\mathbf{x_B}$; $\mathbf{y_B}$;
Repetition Rate J ;
Conversion Map C ;
Key Length K ;
for *m=1:M* **do**
  Evaluate received power at the instant r_m ;
  Map the value of the received power in a bit b_m through the conversion map C;
end
Decode the sequence $\mathbf{eb}$ = { eb_1, eb_2, ..., eb_M } as $\mathbf{s_B}$ = D($\mathbf{eb}$);
Compute verification code $v_B = H(ID_B; \mathbf{s_B})$;
Broadcast verification code when a message needs to be delivered or store it for further access;
Output: $\mathbf{s_B}$

Algorithm 2: Pseudo-code of DRAKE - Remote Mode.

Note that the remote node does not transmit any message during the key establishment phase, thus being completely silent. This feature shows its value in scenarios where the receiver would like to be stealthy, or when the transmission range of the receiver is not enough to reach the other party, in order to have a contributory key establishment *à la* Diffie-Hellman.

B. Detailed description of the DRAKE scheme

DRAKE consists of three phases, that are *Setup*, *Key Generation Triggering* and *Key Delivery*, reported in the following.

Setup. In this phase, executed at the bootstrap of the network, the generic relay R_n, is equipped with a set of materials, stored in the non-volatile memory, to be used in further phases. They include:

- a generic HMAC function $H(msg; key)$ that, given a message msg and a symmetric key key, produces a digest dig, K-bits long;
- a conversion map C, $C(c_m)$! $f0; 1g$, that keeps an amplitude value c_m in volts and outputs a bit, 0 or 1, mapping over through a number L of mapping levels (see Sec. IV-D);
- a matrix S, with N rows, each dedicated to a relay, containing in the columns the symmetric key s_n, K-bits long, dedicated to the n-th relay, and the coordinates x_n,y_n of the n-th relay;
- a repetition factor J, used as the repetition rate of a dedicated repetition encoder E (see Sec. IV-E).
- a correction factor G, that is the maximum number of bits the relays can correct in the final key computed by a remote node (see Sec. IV-C).

Key Generation Triggering. When a new establishment procedure is triggered, B executes the following:

– at the time instant in which it joins the network, B obtains:
  - the current value of the repetition factor J;
  - the conversion map C;
  - the key size K;

We assume that B is already equipped with the code necessary to run the HMAC function $H(\,)$.

– Let us assume DRAKE to be initiated by the relays. One of the relays, e.g., R_n, establishes the following values:

  a series of 2-D positions, $(x_B; y_B)$, that is the set of coordinates a remote node would need to be located at $M = K \quad J$ defined time instants; a set of time instants $\mathbf{r}_B = [r_{1;B}; r_{2;B}; :::; r_{m;B}; :::; r_{M;B}]$, that is the set of time instants in which the remote node will need to log the value of the received signal level.

– Then, the relay R_n broadcasts a *Key Generation Triggering* message, including the position vector $(x_B; y_B)$ and the timestamps vector $\mathbf{r}_B$.

Key Delivery. Both the relays and the remote node receive the broadcast message delivered at the previous step. Let us focus on the generic relay R_n. It performs the following operations:

- R_n computes the vector of euclidean distances from the positions indicated in the position vector, i.e., $\mathbf{d_{n;B}} = f d_{1;n}; d_{2;n}; :::; d_{m;n}; :::; d_{M;n} g$;
- by assuming $k = mod(m; J)$, R_n computes the vector $\mathbf{t_n} = f t_{1;n}; t_{2;n}; :::; t_{k;n}; :::; t_{K;n} g$, where the generic $t_{k;n} = r_{B;k} \quad d_{k;n}=c$ is the k-th transmission time of the n-th relay;
- R_n computes the vector $\mathbf{a_n} = f a_{1;n}; a_{2;n}; :::; a_{k;n}; :::; a_{K;n} g$, where the generic $a_{k;n}$ is the amplitude of the signal transmitted by the n-th relay at the time instant $t_{k;n}$. In detail, the vector $\mathbf{a_n}$ is computed through the following eq. 2:

$$\mathbf{a_n} = mod(H(\mathbf{t_n}; s_n); L); \tag{2}$$

  where $mod(\,)$ refers to the arithmetic modulo operation.
- by taking into account the repetition rate J, R_n computes the matrix t^0_n, as in the following Eq. 3:

$$\mathbf{t^0_n} = \begin{Bmatrix} t^0_{1;n} & t^0_{2;n} & \cdots & t^0_{K;n} \\ \vdots & \vdots & \ddots & \vdots \\ t^{J\ 1}_{1;n} & t^{J\ 1}_{2;n} & & t^{J\ 1}_{K;n} \end{Bmatrix}; \tag{3}$$

  where the element $t^j_{k;n}$ is the transmission instant of the j-th repetition of the k-th signals by n-th relay, with $t^j_{k;n} < t^{j+1}_{k;n} < t^0_{k+1;n}$.
- then, R_n computes the matrix $a^0_{n;B}$, as in the following Eq. 4:

$$\mathbf{a^0_n} = \begin{Bmatrix} a^0_{1;n} & a^0_{2;n} & \cdots & a^0_{K;n} \\ \vdots & \vdots & \ddots & \vdots \\ a^{J\ 1}_{1;n} & a^{J\ 1}_{2;n} & & a^{J\ 1}_{K;n} \end{Bmatrix}; \tag{4}$$

  where the element $a^j_{k;n}$ is the amplitude of the j-th repetition of the k-th signals by n-th relay, with $a^j_{k;n} = a^w_{k;n}$ with $j; w \in [0; \quad ; J \quad 1]$.

– R_n finally transmits the signal at the time instant $t^j_{k;n}$, by using the peak amplitude $a^j_{k;n}$.

– At each time instant $r_{m;B}$, being in the specified position $(x_B; y_B)$, Bturns on its radio and logs the received signal amplitude, that is:

$$c_m = \sum_{n=1}^{N} {}_n a^j_{k;n} \cos(2 \quad f_n r_m + \quad _n) + e_m; \tag{5}$$

  where $m = k \quad J + j$, $_n$ is a signal attenuation factor for the signal transmitted by the n-th relay, f_n is the frequency of the sinusoid, $_n$ is the phase of the sinusoid, while e_m is a generic noise component, that takes into account the unpredictability of the wireless communication medium. For each received signal c_m, B applies the Conversion Map $C(c_m)$ and obtains the corresponding bit eb_m.

– After $M = K \quad J$ receptions, B has in memory the full vector $\mathbf{eb} = [eb_1; eb_2; \quad ; eb_m; \quad ; eb_M]$. To obtain the K-bits key, B applies a repetition code decoder D with the repetition factor J, and obtains as the output a K-bit string, namely s_B, as $D(eb; J) = s_B$. Note that D is a simple majority voting decoder, that chooses the bit that appears the most in the J repetitions (see secs. IV-D and IV-E). At this time, this phase is completed and s_B is the symmetric key obtained by B.

Even if B has obtained the key without any transmission during the *Key Delivery Phase*, despite the repetition code, the effective key computed by B could still be different from the intended one, because of the noise. In the following section we describe the verification process performed by the relays.

C. Communication and Verification of the key

Due to the noise, the key obtained by B can be different from the intended one. Thus, when this key is first used or when the owner of the device would like to access stored data, the relays verify it and, in case of a mismatch, they try the recovering process described below.

When B needs to create a message, or when the data owner would like to access data stored inside, it computes the verification code v_B, as:

$$v_B = H(I D_B; s_B): \tag{6}$$

Then, assuming $mess$ is the plain-text message, B creates the encrypted message by encrypting the plain-text with the key s_B. Then, B appends the verification code to the encrypted message. If it can transmit—frequencies are free—, it broadcasts the message. Otherwise, it simply stores the verification code in a dedicated memory location, to be later retrieved by who is accessing the data.

To obtain the key s_B used to encrypt the information, each relay has all the necessary elements: it knows the key used by other relays s_n, their position, and the

transmission instants. However, because of the noise that is present on the wireless communication channel, the effective key computed by B can be different from the intended one. Indeed, if the number of mismatched bits is contained, the relays can identify them and synchronize with B on the negotiated key. The generic relay R_n executes the following operations:

- starting from the table S containing the keys s_n and the positions x_n, y_n for all $n = [0; 1; :::; n; :::; N]$, the relay R_n computes the key s'_B that should be received by B;
- then, R_n computes the verification code v'_B that should have been computed by B, as $v'_B = H(ID_B; s'_B)$. If v'_B is equal bit by bit to the bit-string v_B received by B in the previous phase, the key is verified and $s'_B = s_B$ is the key associated to B. Thus, the relay can decrypt the message and access the information.
- If $v'_B \neq v_B$, R_n computes a matrix V, with rows and K columns, with $= \sum_{g=1}^{G} \binom{K}{g}$, where the p-th row is obtained from s'_B by changing up to G bits, while the remaining $K - G$ are unchanged.
- For each row of the matrix V, namely $V(p)$, R_n computes a verification code v^p_B, as $v^p_B = H(ID_B; V(p))$ and checks if $v^p_B == v_B$. If $9pjv^p_B == v_B$, then $V(p) = s_B$ is the key associated to B. Thus, the relay can decrypt the message and access the information. If, instead, $\nexists pjv^p_B == v_B$, then the relay R_n discards the message, broadcasts a *Key Agreement Restart* message and triggers a new execution of DRAKE.

D. Details on the Conversion Map

DRAKE leverages a conversion map $C(c_m) \rightarrow \{0; 1\}$ that, given in input a received signal level c_m in volts, outputs a bit b_m. The conversion map uses L quantization levels, in which the whole dynamic range of the relays is divided.

We can define as c_{MIN} and c_{MAX} the minimum and the maximum receivable signal level by a given device, i.e., the sensitivity and the saturation level of the device, respectively. By fixing the number of the quantization levels L, the width of each of them is fixed to a value $w_l = \frac{c_{MAX} - c_{MIN}}{L}$. When a new value c_m is given in input, it falls within a given quantization level A_l, which in turn identifies a corresponding bit b_l. The correspondence between a given quantization level A_l and the corresponding bit b_l is interleaved, i.e., $A_l = : A_{l+1}$. This is planned to maximize the unpredictability of the corresponding bit from the perspective of a passive eavesdropper that would like to guess the exact value of the received bit. On the one hand, the smallest the value of w_l, the higher the unpredictability of the bit b_l at a given distance from the target node. On the other hand, the smallest the value of w_l, the higher the impact of the noise on the correct recovery of the bit b_l. Thus, the value of w_l (i.e., L) must be selected trading off between the error resilience and the protection against passive eavesdroppers.

E. On establishing the Repetition factor

Let us assume K to be the key length, with each bit b_k that is determined according to the DRAKE scheme. Also, we assume that p is the error probability on the single transmission, that is the probability that the noise corrupts (i.e., flips) the single bit received by B. Given that each bit is encoded through a repetition code with a factor J, then the key K is encoded in the matrix eb, with J rows and K columns, for a total number of $K \cdot J$ transmissions, being $eb_{k;j}$ the bit on the k-th row and the j-th column.

Assuming a majority voting for decoding the value of any given bit, the probability that the bit b_k computed by the receiver is flipped is the probability that at least $J/2$ times the corresponding encoded bits $eb_{k;j}$ are flipped. If errors are i.i.d., we can model the bit extraction process as a Bernoulli process with error probability p. By repeating the Bernoulli process over J repetitions, the resulting stochastic process is a Binomial Distribution X. Thus, the error probability is the probability that at least $J/2$ extractions of the single bit b_k are erroneous over a total of J. Then, the probability p_e to compute the wrong bit is given by:

$$p_e = pr(\#errors \geq J/2) = 1 - F_X(p; J/2; J) = v_k; \quad (7)$$

where $F_X(p; J/2; J)$ is the cumulative distribution function of the binomial distribution $X(J/2; J)$ with error probability p, evaluated in the point $J/2$.

In addition, we want our scheme to be able to recover up to G errors after the key delivery phase, as detailed in Sec. IV-B. Thus, the value of G must be chosen in a way to be easily handled by the relays. In general, if we want to correct up to G errors, we need to be able to handle $= \sum_{g=1}^{G} \binom{K}{g}$ comparisons, in order to retrieve the key obtained by the remote node. A reasonable value can be to fix $G = 2$, in a way to handle 8256 comparisons, incurring a tolerable overhead on the relays' side. Now, given the limit of $G = 2$ maximum errors, we need to guarantee that only with a negligible probability DRAKE ends up with more than 2 errors. By mimicking the same procedure as before, given that errors are i.i.d., the probability to make more than $G = 2$ mistakes on the negotiated shared secret is:

$$pr(\#errors \geq 3) = 1 - pr(\#errors < 3) = 1 - pr(max\_2\_errors) = 1 - F_X(v_k; 2; K) \quad ; \quad (8)$$

where determines the global error probability of the DRAKE scheme. Given the error probability p on the single bit, we can determine the value of J that assures a very low , i.e., 2^{-40}. The Fig. 2 shows the profile of the error probability with the number of repetitions considered for each bit of the key. As the error probability on the single bit increases, the error probability on the DRAKE scheme increases, too. Considering a value of $p = 0.01$ and repeating each bit for 7 times it is possible to guarantee an overall error probability less than the threshold of 2^{-40}. This means that the target node will fail in negotiating a key with the system only once every 2^{40} instances. Overall, this makes DRAKE an extremely reliable scheme.

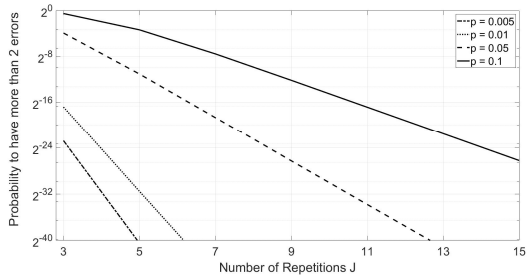

Figure 2. Error probability with different values of p and J.

V. Security Considerations

The security of DRAKE can be discussed in two different system scenarios, depending on the adversary knowledge about the number and positions of the relays.

We recall that DRAKE requires the relays to emit signals using the same frequency. Thus, it is not possible to identify a waveform in which only one of the relays is contributing. We also assume that E does not use radio fingerprinting techniques. Indeed, to the best of the authors' knowledge, recent practical results only achieve identification of classes (i.e., vendors) of devices, while uniquely identifying the emitting device within the same class is still a challenging task [19]. Finally, in the following for ease of discussion we assume that E only uses omnidirectional antennas. However, we notice that using directional antennas to locate the position of the relays does not scale with the size of the network, given that a single antenna should be used for identifying each transmitting relay. In addition, if the relays are very close to each other, even using directional antennas would not be really effective.

Assuming E does not know the position of the relays, the security of DRAKE is directly related with the hardness in the resolution of the well-known BSS problem. In fact, in this case E can collect several measurements of the incoming signal level at different locations, through I omnidirectional antennas placed uniformly in the area. As anticipated in Eq. 5, the instantaneous amplitude of the signal received by the adversary antenna at the time t, namely $c(x_i; y_i; t)$, can be expressed as:

$$c(x_i; y_i; t) = \sum_{n=1}^{N} {}_n a_n (x_i; y_i; t) + e(t); \tag{9}$$

where $e(t)$ is a generic noise sample. Recalling that the instantaneous power of the sinusoid and its amplitude are tied together by the relationship $p(t) = |s(t)|^2$, we can write that:

$$c(x_i; y_i; t) = \sum_{n=1}^{N} \sqrt{\frac{p}{{}_n p_n (x_i; y_i; t)}} + e(t); \tag{10}$$

where $p_n(x_i; y_i; t)$ is the power received by the n-th relay at the position $(x_i; y_i)$. It is possible to express the received power as a function of both the transmission power and the distance between the transmitter and the receiver, by recurring to a particular path loss model. To ease the notation, hereby we leverage the simple Friis free-space path loss model [20]:

$$c(x_i; y_i; t) = \sum_{n=1}^{N} P_n(t - d_n/c) D_n D_i \left(\frac{\lambda_s}{4\pi d_n}\right)^{2\alpha/2} + e(t); \tag{11}$$

where d_n is the distance of the n^{th} relay from the position of the i-th antenna, $P_n(t - d_n/c)$ is the instantaneous power transmitted by the n-th relay at the time instant $t - d_n/c$, D_i and D_n and D_i are the directivity of the transmitter and receiver antennas, and λ_s is the wavelength of the signal. It is possible to express the above Eq. 11 in the matrix form, as:

$$c_i = \begin{bmatrix} a_1 & \cdots & a_N \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} + e_i; \tag{12}$$

where the values $a_n = \sqrt{\frac{D_n D_i \lambda_s^2}{16\pi^2 d_n^{2\alpha}}}$ are the (unknown) mixing coefficients, while x_n includes the instantaneous power transmitted by the n^{th} relay, i.e., P_n. It is worth noting that the adoption of a different propagation model (more suitable for outdoor or indoor scenarios) does not modify the multiplicative relationship between the transmitted and the received power, but only introduces further attenuation factors, already included in the factors a_n. Considering all the I adversary antenna, we have that:

$$\begin{bmatrix} c_1 \\ \vdots \\ c_I \end{bmatrix} = \begin{bmatrix} a_{1;1} & a_{1;2} & \cdots & a_{1;N} \\ & & \ddots & \\ a_{I;1} & a_{I;2} & & a_{I;N} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}^T + \begin{bmatrix} e_1 \\ \vdots \\ e_I \end{bmatrix}; \tag{13}$$

To solve the above equation, E would need two cumbersome information: (i) the position of the transmitting relays and (ii) the maximum number of relays actively participating in the DRAKE scheme. Without such information, the above equation represent a BSS problem in the general form, that is a well-known NP-hard problem [4]. Such a scenario can be the case of a WBAN and an IoT large deployment, in which locating relays could be a challenging task.

Note that another strategy that E could leverage is to deploy its antennas in a circle around B, and try to infer on the value experienced in its position, e.g., by mediating the values logged at the antennas. However, this strategy can be effective only if B is quite far from all the emitting relays. Indeed, if one or more relays are close to B, the values of the received power quickly changes even in a small proximity of a given distance value. To correctly guess the value of the received power, E would need to get closer to B. Indeed, we assumed that the adversary cannot get closer than $\lambda_s/2$ meters from the device.

Finally, in case E knows the maximum number and the position of the relays, it can use a number of antennas at least equal to the number of relays, and use Least Square (LS) techniques to find the transmitted power levels, as:

$$\mathbf{c} = \mathbf{A}\mathbf{x} + \mathbf{n} \Rightarrow \arg\min_{a_{i;j}} (\mathbf{c} - \mathbf{A}\mathbf{x}); \tag{14}$$

However, based on the scenario, deploying an equal (or higher) number of receiving nodes could be very hard, if not impossible to achieve, or because of the high number of potential relays (IoT) or because of the high density of the network (WBAN).

VI. Performance Assessment

In this section we analyze the performance of DRAKE, by considering the number of transmissions required in the *Key Delivery Phase* and the error rate at the legitimate receiver. In addition, we validate its robustness against one of the attacks discussed in Sec. V. The results have been obtained through simulations in Matlab©, by using a Dell XPS15 9560 laptop with 32 GB of RAM and the Windows 10 Operating System. The tests have been performed by assuming an operating frequency $f_s = 433MHz$ and a squared region of interest of 50x50 meters, considering relays that are able to transmit an instantaneous power within the interval $[-20; 20]$ dBm, consistent with the capabilities of modern Software Defined Radios (SDR) commercially available. Finally, all results have been reported along with their 95% confidence interval, estimated through the Gauss statistics.

We first investigated in Fig. 3 the number of transmission required in the *Key Delivery Phase* by each relay to establish a key of a given length with the remote device. This is influenced by the size of the key and by the repetition rate J of DRAKE, established in order to guarantee a high robustness against random errors. The number of transmissions

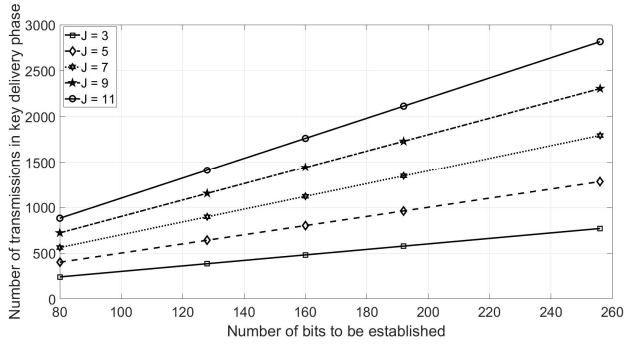

Figure 3. Transmissions required by each relay in the *Key Delivery Phase*.

required by the relays increases by increasing the key length or, in noisier scenarios, by increasing the repetition factor of the code. The overall time required for the completion of DRAKE depends not only on the number of transmissions, but also on the particular technology on which it is deployed and on the scheduling features related to the specific scheme. However, by choosing a key-size of 128 bits and a repetition rate $J = 7$ (in a way to have an error probability $p \quad 0.01$), only 896 transmissions are required, ensuring a completion time that is less than 1 seconds in almost all of todays' wireless technologies. In the worst case, with less than 2,800 messages required, less than 3 seconds would be needed.

We also investigated the performance of DRAKE while varying the number (and thus, the size) of the quantization levels, with different strengths of the surrounding noise. Specifically, we chose an increasing number of the quantization levels (from 216,697 to 6,852,570). Then, by assuming a repetition rate $J = 7$, we have tested the impact of a gaussian noise with zero-mean and increasing variance (from -65 dBm to -40 dBm) on the number of erroneous bits after the application of the repetition code. Fig. 4 reports the error rate with different number of levels and variance of the noise.

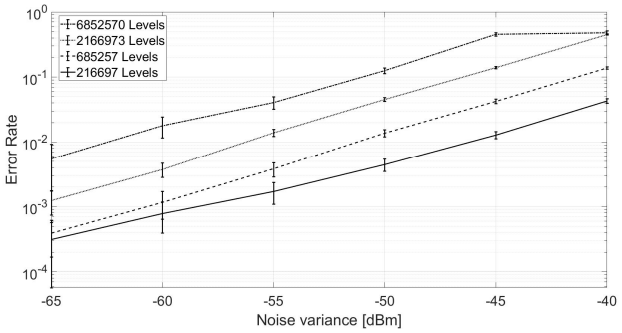

Figure 4. Error rate on $\mathcal{B}$, varying the quantization interval and the noise.

It is possible to reduce the impact of the noise by reducing the number of possible quantization intervals (i.e., by increasing the size of each interval), thus minimizing the error probability on the final recovered bit-string. Note that DRAKE can also correct up to $G = 2$ errors in the final established bit-string, as described in Sec. IV-E. Thus, by performing a preliminary recognition of the noise level in a given scenario, and tuning the number (size) of quantization levels appropriately, DRAKE can guarantee a very high reliability.

Finally, we tested the security of DRAKE against an adversary that does not know the position of the relays, but knows the position of the target remote node, as discussed in Sec. V. Specifically, we assumed $N = 8$ relays and an ideal noise-less environment (the best case from the attacker's perspective). We tested two network topologies, as shown in Fig. 5. In the first one all the relays are far from the target node, while in the second case two relays are closer. We assumed 10 adversarial antennas placed at decreasing

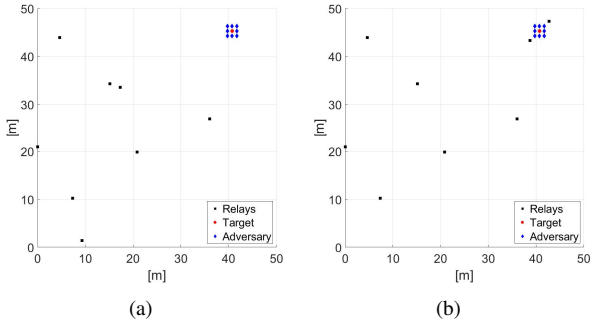

Figure 5. Network topologies: relays a) *far* and b) *close* to the target.

distances from the target node, i.e., 1, $_s$=2 = 0.35 and $_s$=4 = 0.175 meters. Then, we evaluated the rate of the bits guessed by the adversary through mediating the value of the received amplitude of the signal at its antennas. Results reported in Fig. 6 confirm that if the relays are far from the target node (Fig. 5 (a)), an adversary that locates its antenna at a minimum distance d_{MIN} = $_s$=2 meters from the position of the target node can recover almost all of the

bits established by the remote node. In this case, to provide further protection, it is possible to reduce the size of the quantization interval, increasing the effect of the reconstruction error on the adversary side but also (as depicted in Fig. 4) increasing the bit error probability due to noise. Instead, by deploying relays in the proximity of the target node (as in the typical case of a WBAN or an IoT network), also with greater size of quantization levels, it is still possible to achieve an optimal level of security. Hence, the adversary recovers roughly the 50% of the bits, that is approximately the same guessing probability it would achieve by doing a random guess on the single bit. To provide reference

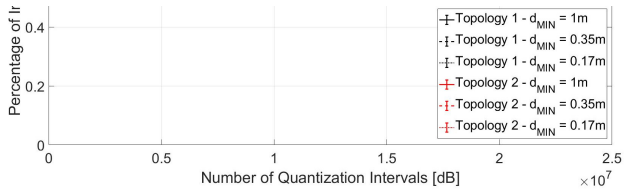

Figure 6. Bits percentage intercepted by $\mathcal{E}$ in the topologies of Fig. 5.

performance, considering the network scenario depicted in Fig. 5 (b), a gaussian noise with a variance 50 dBm, a repetition rate $J = 7$ and 685257 levels, the error rate on the receiver is 0.013 transmissions, meaning that 1.3 (mean) over 100 transmissions are erroneous. However, thanks to the repetition encoder and further error recovery on the relays, they can be indeed corrected. In this case, an adversary located $_S$=2 = 0:35 meters away from the remote node can only guess less than the 50.6% of the bits.

VII. Conclusions

In this paper we presented DRAKE, a distributed physical layer relay-assisted key establishment protocol. DRAKE allows to establish a key of the desired length with a constrained remote device located in a known position, thanks to the overlapping of waveforms delivered by relays distributed in the region of interest. Moreover, DRAKE does not require any transmission by the remote node, and it can be configured appropriately to be highly robust against random errors on the wireless channel. We discussed the security of DRAKE against a passive eavesdropper in different settings. DRAKE can be applied in a variety of scenarios, such as where the receiving device would like to be stealthy, or it has very limited transmission capabilities. Moreover, DRAKE enjoys a few further properties, like limited computational requirements, post-quantum computing robustness, and high flexibility.

Future research directions include the design of a probabilistic model for the quantization levels, the modeling and performance evaluation of the protocol with burst errors, and the experimental evaluation of DRAKE on real devices.

Acknowledgements

This publication was partially supported by awards NPRP-S-11-0109-180242, UREP23-065-1-014, and NPRP X-063-1-014 from the QNRF-Qatar National Research Fund, a member of The Qatar Foundation. The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the QNRF.

References

[1] M. R. Palattella, M. Dohler, A. Grieco, and al., "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," *IEEE Journal on Selected Areas in Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[2] R. T. Tiburski, L. A. Amaral, E. de Matos, and al., "The Role of Lightweight Approaches Towards the Standardization of a Security Architecture for IoT Middleware Systems," *IEEE Commun. Magaz.*, vol. 54, no. 12, pp. 56–62, Dec. 2016.

[3] X. Chen, D. W. K. Ng, W. H. Gerstacker, and al., "A Survey on Multiple-Antenna Techniques for Physical Layer Security," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1027–1053, 2nd quart. 2017.

[4] M. Behr and A. Munk, "Identifiability for Blind Source Separation of Multiple Finite Alphabet Linear Mixtures," *IEEE Trans. on Information Theory*, vol. 63, no. 9, pp. 5506–5517, Sep. 2017.

[5] Z. Li, Q. Pei, I. Markwood, and al., "Secret Key Establishment via RSS Trajectory Matching Between Wearable Devices," *IEEE Trans. on Inf. Forens. and Secur.*, vol. 13, no. 3, pp. 802–817, Mar. 2018.

[6] Z. Li, H. Wang, and H. Fang, "Group-Based Cooperation on Symmetric Key Generation for Wireless Body Area Networks," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1955–1963, Dec. 2017.

[7] X. Li, J. Liu, Q. Yao, and J. Ma, "Efficient and Consistent Key Extraction Based on Received Signal Strength for Vehicular Ad Hoc Networks," *IEEE Access*, vol. 5, pp. 5281–5291, 2017.

[8] S. Dziembowski and M. Zdanowicz, "Position-Based Cryptography from Noisy Channels," in *Progress in Cryptology*, 2014, pp. 300–317.

[9] O. Blazy and C. Chevalier, "Non-Interactive Key Exchange from Identity-Based Encryption," in *Proc. of Int. Conf. on Availab., Reliab. and Secur.*, 2018, pp. 13:1–13:10.

[10] D. Boneh, D. Glass, D. Krashen et al., "Multiparty Non-Interactive Key Exchange and More From Isogenies on Elliptic Curves," *CoRR*, vol. abs/1807.03038, 2018.

[11] H. Imamura, J. Okello, and H. Ochi, "Blind source separation of PSK and amplitude modulated signals," in *Proc. of Int. Symp. on Intelligent Sig. Proc. and Commun. Sys.*, Nov 2004, pp. 343–346.

[12] W. Zhao, Z. Yuan, Y. Shen, and al., "Blind Source Separation of Instantaneous Mixture of Delayed Sources Using High-Order Taylor Approximation," *ETRI Journal*, vol. 37, no. 4, pp. 727–735, 2015.

[13] T. Chang, T. Watteyne, X. Vilajosana and al., "Constructive Interference in 802.15.4: A Tutorial," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2018.

[14] N. Choudhury, R. Matam, M. Mukherjee and al., "Beacon Synchronization and Duty-Cycling in IEEE 802.15.4 Cluster-Tree Networks: A Review," *IEEE IOT J.*, vol. 5, no. 3, pp. 1765–1788, Jun. 2018.

[15] A. Goldsmith, *Wireless Communications*. Cambridge Univ. Press, 2004.

[16] J. Zhang, F. Du, J. Ma, and C. Yang, "Position based key exchange: defnitions and implementations," *Journal of Communications and Information Networks*, vol. 1, no. 4, pp. 33–43, Dec 2016.

[17] J. Zhang, R. Woods, T. Q. Duong, and al., "Experimental Study on Key Generation for Physical Layer Security in Wireless Communications," *IEEE Access*, vol. 4, pp. 4464–4477, 2016.

[18] S. Sciancalepore, G. Oligeri, G. Piro, G. Boggia, R. Di Pietro, "EXCHANge: Securing IoT via channel anonymity," *Computer Communications*, vol. 134, pp. 14 – 29, 2019.

[19] Z. Zhuang, X. Ji, T. Zhang, and al., "FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting," in *Proc. of Asia Conf. on Comput. and Commun. Sec.*, ser. ASIACCS '18, 2018, pp. 261–272.

[20] J. A. Shaw, "Radiometry and the Friis transmission equation," *American Journal of Physics*, vol. 81, no. 1, pp. 33–37, 2013.