

Obfuscating IEEE 802.15.4 Communication Using Secret Spreading Codes

Björn Muntwyler
ETH Zurich, Switzerland
bjoernm@ee.ethz.ch

Vincent Lenders
armasuisse, Switzerland
vincent.lenders@armasuisse.ch

Franck Legendre
ETH Zurich, Switzerland
legendre@tik.ee.ethz.ch

Bernhard Plattner
ETH Zurich, Switzerland
plattner@tik.ee.ethz.ch

Abstract—The IEEE 802.15.4 standard specifies an M-ary spread spectrum system with public and fixed spreading sequences. We propose instead to use secret and dynamic, random spreading sequences to obfuscate communications. Through theory and experiments with a prototype IEEE 802.15.4 transmitter and receiver that have been adapted to use random and dynamic codes, we quantify the benefits and performance implications of our approach. We first give a theoretical approximation of the performance loss when secret random codes are used instead of the IEEE 802.15.4 codes. We identify then three transitional regions for IEEE 802.15.4 communication above thermal noise i.e., when the incoming power level at the attacker is above thermal noise. We derive the obfuscation gain for signal detection and signal interception attacks respectively in these regions. Our implementation shows the feasibility of our approach. It is also used to evaluate the performance loss and time requirements to perform attacks on the system under realistic wireless conditions.

I. INTRODUCTION

Wireless on demand communication is increasingly used by activists in countries where repressive regimes exert control over the communication infrastructure. Technologies such as IEEE 802.15.4 or IEEE 802.11 offer cheap and rapidly deployable independent communication services to users in local areas. However, as the communication with these technologies is easily detectable and interpretable, repressive forces have an easy task at pinpointing, jamming or shutting down these ISM band technologies. To prevent censorship, one would ideally require “shadow” networks that remain undetected.

A common technique to preserve privacy in wireless networks is datalink-layer encryption (e.g. WPA2), which is well suited to protect the data content against eavesdroppers. However, even when the entire MAC layer including the headers is encrypted, the traffic activities may be recovered by analyzing side channel information. Studies [1], [2] show that online activities of wireless users may accurately be recovered by looking only at packet size, timing and direction of the packets.

A more suitable technique to protect from wireless detection and interception is spread spectrum (SS) communication. SS is a relatively old idea that consists of spreading a narrow band information signal to a larger spectral band in order to limit the spectral density and hide its presence. SS communication is generally implemented as direct sequence spread spectrum (DSSS) or as frequency hopping spread spectrum (FHSS). SS communication was originally conceived for the military domain in order to reduce the probability of signal interception and the impact of jamming but has now found its use in standards like IEEE 802.15.4 or IEEE 802.11. The key difference between the military and the commercial SS systems is that the commercial systems do not use SS for

security but for performance and robustness reasons. As a result, the spreading codes are public and users cannot profit from the obfuscation properties or the jamming gain of SS.

This paper proposes to replace the public spreading codes of IEEE 802.15.4 with secret and pairwise-dynamic, random codes. While the theory of military SS communication is well understood in the literature, the practical implications of using secret and random codes in commercial standards has to the best of our knowledge not been explored so far. Understanding these implications is however of great interest to parties which may not have access to military SS equipment and wish to obfuscate on demand communication using commercial off-the-shelf devices. This idea raises three particular questions that we tackle in this paper:

- How much performance is lost by using random codes instead of the IEEE 802.15.4 pseudo-orthogonal ones?
- How much obfuscation do we gain by using secret and dynamic, random codes in IEEE 802.15.4?
- How can we efficiently synchronize these codes in IEEE 802.15.4 networks?

To address these questions, we use theory and experiments with a prototype IEEE 802.15.4 transmitter and receiver that we have implemented on the GNU Radio software defined radio that uses randomized spreading codes. We give an approximation of the performance loss when using secret random codes with optimal IEEE 802.15.4 transceivers (IV). Then, we identify three transitional regions for IEEE 802.15.4 communication above thermal noise, i.e., when the incoming power level at the attacker is above thermal noise. We derive the obfuscation gains for the detection and interception attacks respectively in these regions (V). The theoretical results are complemented with a practical implementation of IEEE 802.15.4 with random codes (VI). The implementation is used to evaluate the performance loss and time requirements to perform attacks on the system under realistic wireless conditions (VII). We discuss the results (VIII) before concluding (IX). But before, we start by reviewing related work (II) and describing DSSS and how it is used in IEEE 802.15.4 (III).

II. RELATED WORK

The theory of DSSS systems using secret spreading codes has been intensively studied in the military literature in the seventies and eighties [3], [4]. These DSSS systems are however rarely M-ary DSSS systems and they operate below thermal noise at a much larger spreading factor than IEEE 802.15.4 (one hundred to up to more than one thousand). This paper focuses on the obfuscation of spread signals above thermal noise. A more recent design is presented in [5] where the authors focus on CDMA systems using DSSS and also propose

the use of dynamic and secret spreading sequences changing at regular intervals. In contrast, we show the feasibility and provide an experimental study with actual implementation and measurements using software defined radios.

There are many privacy dimensions in wireless networks such as device identification, traffic analysis to determine used applications, protocols and communication relationships, and location privacy. In [6], Gruteser et. al. propose to increase privacy by the frequent disposal of interface identifiers. But in [7], they argue that this is not sufficient and propose an identifier-free link layer protocol, in which the source and destination addresses are replaced by AES encrypted integers, defining packet numbers. However, [8] shows that traffic analysis still remains possible. For location privacy there are also interesting proposals such as [9], which tried to achieve location privacy by frequently changing pseudo identifiers. This is similar to [6]. Here the concept of mix zones [10] gets fully exploited. In wireless networks, each node can however be detected when emitting by energy detection techniques and privacy must therefore also be addressed at the physical layer.

Bernaile et. al. introduce attacks in [2] that enable the identification of applications even if they are using upper layer encryption only by observing the size of the first few packets of an SSL connection. In [11], authors discuss the issue of traffic analysis on the temporal patterns like arrival timings of packets to infer communication relationships. A countermeasure is proposed to prevent such attacks by maximizing the entropy of the packet inter-arrivals by adding exponential random delays. [12] proposes using dummy traffic to anonymize and pad the real traffic. The authors of [13] apply steganography techniques in IEEE 802.15.4 wireless communication to embed additional information in existing packets. Steganography only allows to hide information when the existing packets are not sensitive to privacy issues.

III. DSSS COMMUNICATION PRIMER

This section introduces the basics of DSSS communication and how DSSS is implemented in IEEE 802.15.4.

A. DSSS Functional Overview

The basic operation of a DSSS communication system is shown in Figure 1. At the transmitter, a low bit-rate information signal of bandwidth B_i is first modulated with a carrier signal and then spread by multiplying the signal with a higher bit-rate spreading chip sequence. The nature of the higher-rate spreading signal causes the frequency spectrum of the output signal to spread evenly over a wider frequency range B_x than the original information signal. Since the output signal power gets distributed over this extended range, the amount of power transmitted within the information bandwidth of the signal (i.e., its bandwidth before it was spread) is reduced by the processing gain $G = \frac{B_x}{B_i}$.

At the receiver, a signal demodulator despreads the spread signal with a synchronized replica of the same chip sequence that was applied by the transmitter. For the despreading to be successful, it is important that chip sequences are synchronized. When this occurs, the received signal is collapsed back to the carrier-modulated information signal of bandwidth

B_i , recreating the signal that was input to the spreading modulator in the transmitter. This signal is then demodulated from the carrier to reproduce the information transmitted. Note that the properties remain the same when the spreading and information demodulator are inverted [14].

The evolution of the power spectral density from the transmitter to the receiver is sketched in Figure 1. At the transmitter, the power spectral density of the information signal is first reduced by the processing gain G and then transmitted over the air. The transmitted signal will experience propagation losses on its way to the receiver and the signal power will hence be attenuated by the path loss. At the receiver, the transmitted signal is despread and the signal power spectral density is thus increased by the processing gain G .

B. Adversarial Considerations

When the spreading sequences are unknown, an attacker trying to intercept the transmitted signal cannot collapse the signal back and must, therefore, deal with the low power density of the spread signal with bandwidth B_x . In military DSSS systems, the signals are spread such that the power spectral density is very low. In fact, most systems are designed such that this power level is lower than the thermal noise. The thermal noise refers to the inherent noise that resides in an ideal receiver from the agitation of the electrons given a particular temperature. The thermal noise is defined as kTB , where k is Boltzmann's constant ($1.38 \cdot 10^{-23}$ Joule / K), T is the operating temperature in degrees Kelvin, and B is the effective receiver bandwidth ($B = B_x$ in this case). Since the thermal noise is directly proportional to the bandwidth, spreading will eventually render the signal power lower than the noise floor from an ideal receiver.

C. DSSS in IEEE 802.15.4

The IEEE 802.15.4 standard [15] specifies the wireless medium access control (MAC) and physical layer (PHY) for low-rate wireless sensor networks. While IEEE 802.15.4 supports three frequency bands at 2450 MHz, 915 MHz and 868 MHz, our focus is on the most popular 2450 MHz band with O-QPSK modulation. At this band, 16 channels of 2 MHz bandwidth are supported offering a data throughput of 250 kb/s each. A 16-ary quasi-orthogonal DSSS modulation technique is used. The spreading is done by mapping a low-rate (250 kb/s) symbol of 4 information bits (e.g., 0101) to a corresponding higher-rate (2 Mchip/s) chip-sequence of 32 chips (e.g., 01111011100011001001011000000111). These mappings are shown in Figure 2 for illustration.

While IEEE 802.15.4 is a DSSS system, the DSSS scheme is primarily used for performance reasons and differs from traditional low probability of intercept signals in mainly four relevant aspects:

- A 16-ary quasi-orthogonal DSSS modulation technique is used compared to conventional DSSS systems, that use a single pseudo random binary sequence for spreading.
- The spreading sequences and modulation parameters are defined publically in the standard. An adversary may hence look up the spreading sequences by simply referring to mapping tables in the standard.

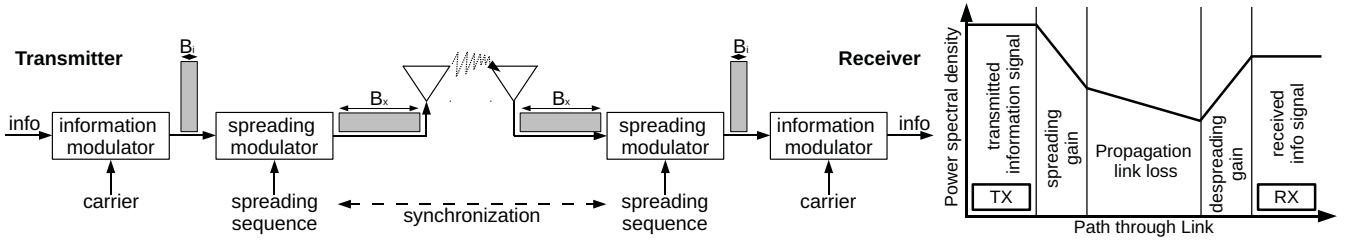


Fig. 1. Left: General DSSS modulation and demodulation design. The signal width of the information signal B_i is spread by the processing gain G to B_x . Right: Evolution of the power spectral density of a DSSS signal from the transmitter to the receiver.

Data symbol (decimal)	Data symbol (binary) (b_1, b_2, b_3, b_4)	Chip values ($c_0, c_1, \dots, c_{30}, c_{31}$)
0	0000	11011001110000110101001000101110
1	1000	11101101100111000011010100100010
...
7	1110	10011100001101010010001011101101
...
10	0101	01111011100011001001011000000111
...
15	1111	11001001011000000111011110111000

Fig. 2. Symbol to chips mapping defining a code (consisting of 16 chip sequences), as publicly available and fixed in IEEE 802.15.4 [15].

- The spreading sequences are fixed. The mappings of symbols to chip-sequences never change.
- The used mappings correspond to a spreading factor of 32 chips / 4 symbols = 8. Military DSSS systems use typically higher spreading factors ranging from 10 to 1000 [14], [4]. As a result, the power spectral density of the spread signals generally occur to be above the thermal noise floor.

IV. APPLYING SECRET CODES IN IEEE 802.15.4

To obfuscate the IEEE 802.15.4 signals against detection and interception, we propose to change the known mappings from Figure 2 of symbols to chips with dynamic and random secret mappings, which we refer to as code.

For best performance, codes should always exhibit the highest hamming distance. However, when using random codes, there is no guarantee that the hamming distance between different chip sequences will have a particular distance. The random nature of the codes is however important as any dependence property between chip sequences may be used by an attacker to reduce the search space when trying to recover the code being used in a communication. The best codes in terms of security are therefore the ones with independent random chip sequences for the different symbols which is the solution we adopt in our work.

A. Theoretical Implications on Performance

To understand the theoretical implications of replacing the spreading code of IEEE 802.15.4 with random codes, we consider the receiver sensitivity which defines the minimal signal power at the antenna to achieve a specified error performance. The sensitivity limit P_{min} of IEEE 802.15.4 in AWGN channels can be characterized by considering the

thermal noise, the receiver noise figure (N_f) and the required signal to noise ratio in the following way [16]:

$$P_{min} = kTB \cdot N_f \cdot SNR_{min} \quad (1)$$

where k is Boltzmann's constant, T is absolute temperature, B is the communication bandwidth (2 MHz in IEEE 802.15.4), and SNR_{min} is the minimum baseband signal power to noise power ratio at the demodulator. SNR_{min} is given by

$$SNR_{min} = \frac{(E_b/N_0)_{min}}{C \cdot G} \quad (2)$$

where $(E_b/N_0)_{min}$ is the minimum energy per bit to noise ratio required for the O-QPSK modulation scheme in IEEE 802.15.4, C is the coding gain, and G is the processing gain.

The $(E_b/N_0)_{min}$ value depends on the required error rate tolerated. For example, assuming 26 bytes packets, a required packet error rate below 1%, and coherent detection of O-QPSK as required by the standard, the required $(E_b/N_0)_{min}$ is 8.8 dB [16]. The processing gain G is equal to the spreading ratio which is given by the ratio of the number of chips transmitted per bit of information. For IEEE 802.15.4, the processing gain is $G = 32$ chips / 4 symbols = 9 dB.

To determine the coding gain C available from the symbols, the degree of orthogonality for the codes must be understood. When one code word, r_1 , is compared to the set of template code words $R = \{r_1, \dots, r_M\}$, the number of chip flips required to change r_1 to any r_i is called the Hamming distance between the code words. For a DSSS code, the coding gain of R is calculated by finding the mean Hamming distance of the code words, \bar{d} . Given the code for 802.15.4, $R_{15.4}$, the maximum, mean and minimum distances are 20, 17, and 12 respectively. The distance cumulative distribution function (CDF) across all 16 code words from this set is shown in Figure 3. For DSSS code sequences, an approximate expression for coding gain is

$$C \approx K \left(\frac{\bar{d}}{n} - \frac{\ln 2}{E_b/N_0} \right) \quad (3)$$

where K is the number of bits per symbol and n is the length of the code words [17]. For a 1% packet error rate and $R_{15.4}$, the coding gain is approximately $C_{15.4} = 2.5$ dB which directly reduces the required E_b/N_0 . The value of C in Equation (3) is an average across all r_i in $R_{15.4}$, meaning that some symbols will have better or worse properties according to their Hamming distance to other r_i and the errors that occur.

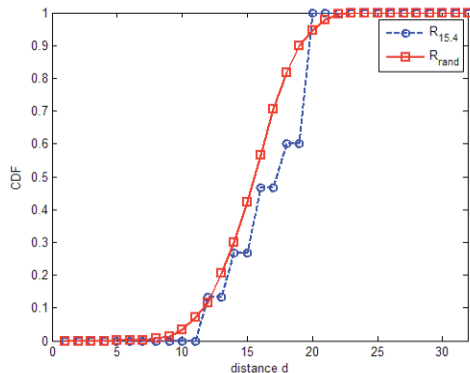


Fig. 3. CDF of Hamming distance between all 16 codes in $R_{15.4}$ and all possible codes in R_{rand} .

Finally, SNR_{min} for IEEE 802.15.4 can be calculated from the application of equation (2):

$$SNR_{min}^{R_{15.4}} \approx \frac{(E_b/No)_{min}}{K(\frac{\bar{d}}{n} - \frac{\ln 2}{E_b/No}) \cdot G} = -2.7 \text{ dB}. \quad (4)$$

Using the sensitivity equation in (1), the minimum sensitivity is given by

$$P_{min}^{R_{15.4}} \approx kTB \cdot N_f \cdot \frac{(E_b/No)_{min}}{K(\frac{\bar{d}}{n} - \frac{\ln 2}{E_b/No}) \cdot G} = N_f - 113.7 \text{ dBm}. \quad (5)$$

When changing the code sequences as we propose, the code distances vary and hence the coding gain as well as the minimum sensitivity. Given a random code of 16 symbols of length 32 chips R_{rand} , the maximum, mean and minimum distances are 32, 16, and 0 respectively. The distance cumulative distribution function (CDF) for R_{rand} and uniform randomly selected code words is shown in Figure 3. In contrast to $R_{15.4}$, code distances may now be smaller than 12.

To derive the new sensitivity when applying the random code R_{rand} , we use again Equation (3). The average code distance \bar{d} is now 16, resulting in a coding gain of $C_{rand} = 2.1 \text{ dB}$. Compared to the coding gain of $C_{15.4} = 2.5 \text{ dB}$, this results in an average performance loss of approximately 0.4 dB. The minimal signal to noise ratio is therefore this time

$$SNR_{min}^{R_{rand}} \approx \frac{(E_b/No)_{min}}{K(\frac{\bar{d}}{n} - \frac{\ln 2}{E_b/No}) \cdot G} = -2.3 \text{ dB}, \quad (6)$$

and the sensitivity limit:

$$P_{min}^{R_{rand}} \approx kTB \cdot N_f \cdot \frac{(E_b/No)_{min}}{K(\frac{\bar{d}}{n} - \frac{\ln 2}{E_b/No}) \cdot G} = N_f - 113.3 \text{ dBm}. \quad (7)$$

This concludes that by applying a random code R_{rand} instead of the IEEE 802.15.4 code $R_{15.4}$, the theoretical performance loss is on average only a marginal 0.4 dB. The gain in terms of obfuscation is however more significant as we discuss in the next section.

V. OBFUSCATION ANALYSIS

Now that the code is unknown to an adversary, simple eavesdropping is no longer possible. However an adversary

may try to attack the system. We address two classes of radio-equipped passive adversaries. The attacker model I aims at detecting the presence of the obfuscated IEEE 802.15.4 signals without attempting to recover the codes, while attacker model II targets interception by attempting to recover the codes from eavesdropped communications. This section introduces the attacker models and provides necessary conditions in order to launch these attacks. We further derive the obfuscation gains.

A. Threat Models

Attacker model I (detection): With this model, an attacker attempts to detect the presence of IEEE 802.15.4 communication by listening to energy variations on the wireless channel. Without loss of generality, we assume a passive attacker that listens to one IEEE 802.15.4 channel (multiple channels can be monitored simultaneously by implementing multiple receivers) and that does not know the spreading codes. The attacker samples the channel of width $B = 2 \text{ MHz}$ and implements a detector according to a power threshold detection scheme $\frac{P_a}{N_a} > \delta$. When this threshold δ is exceeded, the attacker declares detection. Note that this simple energy detection scheme assumes an environment where all communication is of interest to an attacker. In mixed environments with regular IEEE 802.15.4 traffic or other active networks operating on the same frequency band, the model must be extended with context information to suppress detection of regular transmissions that are not of interest to the attacker.

Attacker model II (interception): In the second attacker model, the attacker attempts to recover the information signal. We assume a passive attacker, which (i) knows all parameters used for communication except the secret chip sequences, (ii) can distinguish between different nodes sending and (iii) is already synchronized during the attack (and as such knows the starting position of the chip sequences). Furthermore, we assume that the attacker has no side channel information of the actual symbols being sent over the air. This might not be true when fixed symbols are used for preamble or headers. However, it is possible to implement a random symbol assignment scheme as we propose in Section VI to assure that an attacker cannot recover the chip sequence to symbol mapping by exploiting expected symbols (e.g. the start frame delimiter). The strategy of this attacker is to digitally sample incoming signals inside the channel of width B_x . Knowing the modulation parameters he can convert the collected chips in a list of chip sequences. With the given assumptions, the attacker is then required to try all possible combinations of mappings from chip sequences to symbols until the frame check sum is correct to reveal the entire code.

B. Transitional Regions

Whether these two attacks may be launched depends for each attack on the incoming signal power at the attacker P_a . In the following, we differentiate four necessary conditions that characterize the attacker's ability to perform attacks I and II depending on the incoming signal power P_a at the attacker:

- 1) $P_a \leq kTB$: In this case, the received signal power at the attacker is lower than the thermal noise. With $B = 2 \text{ MHz}$ as in IEEE 802.15.4 and a typical room

Region #	Incoming power level at attacker	Model I	Model II
1)	$P_a \leq -111$ dBm	no	no
2)	-111 dBm $< P_a \leq -111$ dBm $+ N_a$	no	no
3)	-111 dBm $+ N_a < P_a \leq -104.5$ dBm $+ N_a$	yes	no
4)	$P_a > -104.5$ dBm $+ N_a$	yes	yes

TABLE I

FEASIBILITY TO LAUNCH ATTACK MODELS I AND II. (ASSUMPTIONS: $T = 290$ K, PACKET SIZE = 26 BYTES, ATTACK SUCCESS PROBABILITY $P_{II} = 1\%$, OPTIMAL COHERENT O-QPSK DEMODULATION)

temperature of 290 K, the thermal noise results in $kTB = -111$ dBm. Under this regime, attacks I and II will both fail since the thermal noise at the attacker is stronger than the received signal. This is the most desired state in terms of obfuscation since even with an ideal receiver system, the attacker will not be able to differentiate the signal from noise. This state is generally referred to as hiding the information signal under the noise floor. However, this state is generally not achieved in IEEE 802.15.4 since the processing and coding gains are not substantial enough to compensate the path loss that arises at typical communication distances. The transmitters are therefore forced to use transmit powers that significantly exceed the thermal noise.

- 2) $kTB < P_a \leq kTB \cdot N_a$: In this case, the received signal power is higher than the thermal noise but lower than the thermal noise plus the receiver system noise N_a of the attacker. The receiver system noise figure results from non-ideal receiver components such as cables, analog-to-digital converters, etc. Under this condition, attacks I and II will hence fail as before as the power of the noise is stronger than the power of the incoming signal. In order to perform attacks I or II, the attacker must either reduce its noise figure (which basically comes out to build a more sensitive and expensive device) or increase the received signal power by for example moving closer to the transmitter.
- 3) $kTB \cdot N_a < P_a \leq kTB \cdot N_a \cdot (E_b/N_0)_{a,min}$: The incoming signal power is here higher than attacker's system noise figure plus thermal noise but not sufficient for an attacker to correctly demodulate the signal in the absence of the secret spreading codes with a sufficiently low packet error rate ($SNR_{a,min} = (E_b/N_0)_{a,min}$ since $G=C=1$). The attacker will now be able to eavesdrop the channel and perform attack I. However, since the incoming signal power is lower than the required signal strength for demodulation, he will not be able to make sense of the spread signal and hence perform attack II.
- 4) $P_a > kTB \cdot N_a \cdot (E_b/N_0)_{a,min}$: Here, the incoming signal power is larger than the required signal power to demodulate signals even in the absence of the secret spreading codes. In addition to attack I, the attacker may now attempt to recover the spread signal using attack II.

The four regions with the incoming power ranges and the feasibility to launch attacks I and II are summarized in Table I.

C. Obfuscation Gain

While it might not be possible to prevent attacks I and II in all cases, using secret codes in IEEE 802.15.4 still raises

the barrier to be able to launch these attacks. To quantify this barrier, we introduce the obfuscation gain γ as the ratio between the minimal sensitivity required for an attacker a to perform an attack and the minimal sensitivity required by a legitimate receiver r to correctly receive a transmitted signal:

$$\text{Definition of obfuscation gain : } \gamma = \frac{P_{min}^a}{P_{min}^r} \quad (8)$$

Detection gain (attack model I): The minimal sensitivity required by an adversary to perform attack I is given by the thermal noise (kTB), its noise figure (N_a) and the detection threshold (δ):

$$P_{min}^a = kTB \cdot N_a \cdot \delta \quad (9)$$

A legitimate node requires a sensitivity of

$$P_{min}^r = kTB \cdot N_r \cdot SNR_{r,min}^{R,rand} \quad (10)$$

where $SNR_{r,min}^{R,rand} \approx -2.3$ dB (see Section V). The obfuscation gain is therefore

$$\gamma_I = \frac{N_a \delta}{N_r SNR_{r,min}^{R,rand}} \approx \frac{N_a}{N_r} \cdot \delta + 2.3 \text{ dB}. \quad (11)$$

At equal noise figure for the legitimate receiver and the attacker ($N_a = N_r$), the obfuscation gain becomes $\delta + 2.3$ dB.

Interception gain (attack model II): The minimal required sensitivity for attacker model II is

$$P_{min}^a = kTB \cdot N_a \cdot SNR_{a,min}^{R,rand}. \quad (12)$$

To determine $SNR_{a,min}^{R,rand}$, we need to account for the success probability of the attack. In order to successfully match all possible chip sequences to symbols allocations against the frame checksum, there may be no chip errors in a frame. Otherwise, there is no unique mapping. The probability that there are no chip errors in a frame and hence that the attack is successful is $P_{II} = (1 - CER)^{p \cdot n}$, where CER is the chip error rate, p is the number of symbols in a frame and n is the number of chips per symbol ($n = 32$ with IEEE 802.15.4). Assuming a frame length of 26 bytes, an attack success probability of $P_{II} = 1\%$, and coherent detection of O-QPSK, we get a $SNR_{a,min}^{R,rand} = (E_b/N_0)_{a,min} = 6.5$ dB [16], [17], since $G = 1$ and $C = 1$ (see Section V).

A legitimate node requires in contrast a lower sensitivity as given in equation 10, where $SNR_{r,min}^{R,rand} \approx -2.3$ dB since it knows the codes. The obfuscation gain is therefore

$$\gamma_{II} = \frac{N_a \cdot SNR_{a,min}^{R,rand}}{N_r \cdot SNR_{r,min}^{R,rand}} \approx \frac{N_a}{N_r} + 8.8 \text{ dB}. \quad (13)$$

D. Attack Time Requirements

Attack I may be launched almost instantly after a few samples of the measured incoming signal power. However, attack II requires a certain number of transmitted symbols in order to recover all chip sequences. For attacker model II, an attacker may attempt to record the incoming signals and recover the spreading sequences. An attacker may for example sniff the air, synchronize to the transmissions and try to recover the codes being used in legitimate transmissions by learning the sequences being sent. To break the spreading codes of

an M-ary system, [18] have proposed an attack, recording intercepted chip sequences from the channel and applying a k-means clustering algorithm to cluster these chip sequences (partly corrupted by noise) into M clusters. If the centroids of the clusters do not correspond to the true chip sequences used by the sender, the algorithm continues gathering more information from the channel and starts over.

We analyze the asymptotic behavior for a $SNR \rightarrow \infty$ (no chip errors). This asymptotic behavior can be determined using the double dixie cup problem [19]. For no chip errors and assuming that all $m = 16$ chip sequences are drawn uniformly, the expected number $E_m[z]$ of intercepted chip sequences required to get each sequence at least z times is given by

$$E_m[z] = z \cdot \int_0^\infty \left[1 - \left(1 - \left(\sum_{k < m} \frac{t^k}{k!} \right) \cdot e^{-t} \right)^z \right] dt. \quad (14)$$

Without chip errors (which is very unlikely to occur), an attacker needs to record each sequence at least once. Hence, for $E_{16}[1] = 54$ an attacker needs to intercept at least $\frac{54 \cdot 4}{8} = 27$ bytes. Assuming an average packet size of 26 bytes [15], implies changing the code for roughly every packet¹. When chip errors occur, the attacker will require longer observation periods. The success rate with chip errors is evaluated using our implementation under real-world conditions in Section VII.

VI. IMPLEMENTATION

A. Hardware and Software Platform

We have implemented a IEEE 802.15.4 transmitter and receiver with random codes on software defined radio using the GNU Radio platform. The GNU Radio platform runs on the USRP2 hardware with XCVR2450 transceiver daughterboards and omnidirectional antennas. The XCVR2450 is used in the 2.4 GHz band of IEEE 802.15.4 with O-QPSK modulation. We rely on the ZigBee code from UCLA [20] for our implementation, in which the spreading and despreading is achieved in software on the host. Hence, we adapt the transmitters and receivers to work with random spreading codes by changing the spreading code tables in user space.

B. Code Synchronization

DSSS systems that rely on secret codes traditionally rely on pseudo random binary sequence generators that are implemented using shift registers [14]. The generated binary sequences are then synchronized and multiplied at the transmitter and receiver, respectively. In principle, we could have such an approach with IEEE 802.15.4 in which 16 code generators spread the 16 different symbols. However, the problem with this approach is that synchronization of the shift registers at both end-points is very time consuming [17]. Hence, this approach is not well suited for sporadic communication like IEEE 802.15.4 where nodes might form mobile ad hoc networks and have time-limited communication sessions only when two nodes happen to be in vicinity. To address

¹To intercept the payload of the communication, an attacker would still need to map the chip sequences to their corresponding 4-bit symbols, if the chip sequences would have been found.

this problem, we have designed and implemented PSCHP a pairwise code synchronization protocol on top of IEEE 802.15.4 that minimizes the code synchronization time.

With PSCHP, sender and receiver hop pairwise between pseudo-randomly generated, secret spreading codes in a synchronized manner. The pairwise-dynamic code-hopping will prevent an attacker from recording enough information to compromise the secrecy of the chip sequences of a code, before the nodes hop to the next code. To further increase the resilience of side-channel attacks, the preamble and SFD symbols are also selected randomly from the 16 secret chip sequences in a code and a physical layer header obfuscation algorithm is applied.

For the generation of a pairwise-shared secret, we rely on a secure key-exchange protocol (e.g., Elliptic-Curve Diffie-Hellman [21], [22]). Bootstrapping PSCHP can either use Uncorrelated DSSS [23] or pre-shared keys. As soon as this first initialization phase has completed, each node will generate two sets of codes per neighboring node using their established shared secret. One set of codes is used for sending and the other set for receiving. Each of these codes defines a symbol-to-chips mapping consisting of 16 pseudo-random chip sequences. Eventually, nodes hop from one code to the next within a code-set, in a pairwise synchronized manner.

When sending a packet, the node spreads it with the first (active) code in the sending-code-set dedicated to the intended receiver. If more than Code-Life-Time (CLT) Bytes have been sent with this active code, it is deleted and the next code within this sending-code-set becomes the new active code.

When receiving packets, a node needs to first compare on-the-fly the received preamble sequence with all the preamble sequences in the stored receiving-code-sets, in order to identify the sending node and to load the corresponding code for despreading. If no active code matches, but another code within a code-set, a code-hop has been detected and the latter code replaces the active code of that receiving-code-set.

In order to prevent shared-secrets from being disclosed by brute-force attacks, our protocol will renew the pairwise-shared secret by repeating the key-exchange protocol after a fixed number of code-hops. For the benefit of privacy, broadcast communication is implemented as separate unicast transmissions per receiver using the pairwise code-sets.

VII. EXPERIMENTAL EVALUATION

A. Performance Implications

To validate the theoretical performance loss of 0.4 dB one should expect when using random codes instead of the IEEE 802.15.4 code (see Section V), we compare the measured packet error rate in our implementation with standard and random chip sequences. We differentiate two setups: a cable and a wireless channel series of measurements. The cable experiment can be viewed as a AWGN channel as assumed previously for our modelling while the wireless channel may have reflections and multipath components. The transmitter sends 113 bytes packets at a rate of around 246 packets per second. In the cable experiment both USRP2 are connected with a cable and a 60 dB attenuator. To observe the packet error rate under various signal to noise ratios (SNR), we

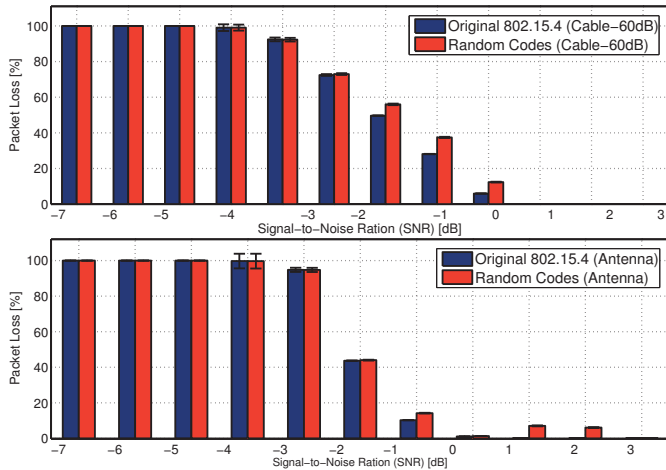


Fig. 4. Packet error rate (PER) vs. signal to noise ratio (SNR) [dB] with the 95% confidence interval (no confidence interval was measured for SNR values between -7 and -4 dB). Top: cable with attenuator, Bottom: air (wireless).

adjust the power level of the transmitter from the maximum value to the minimum value that the hardware supports. In the wireless channel experiment, both USRP2 communicate over an antenna and different SNR values are obtained by moving both USRP2 further apart from each other. The results of both experiments are shown in Figure 4. The horizontal axis represents the SNR and the vertical axis the packet error rate (PER). For each pair of bars, the blue leftmost represents the use of the original chip sequences and the red rightmost bar represents the use of the random codes. Each value represents an average over multiple measurement runs. The confidence bars indicate the 95% confidence intervals.

For the cable experiments (AWGN channel), we see that the performance loss is less than 1 dB across the entire range. This is consistent with the predicted 0.4 dB performance loss derived in Section V. In the wireless experiments, the performance loss depends on different channel characteristics as before since we move the USRP2s apart and hence change the channel response. Still, the loss is at most 2 dB across the entire range of SNR values and generally below 1 dB.

B. Obfuscation

To evaluate the performance of an attacker model II under real world conditions (i.e., when chip errors occur), we have further implemented attack model II on the USRP2. In practice, a $SNR \rightarrow \infty$ at the attacker never occurs, and an attacker will experience chip errors. To understand the performance under this condition, we have recorded chip sequences from real channel measurements using our implementation of IEEE 802.15.4 on GNU Radio and simulated such an attack to test its performance. The algorithm of the attack is shown in Algorithm 1. At first, the attacker starts recording every signal on the channel. It will then cluster the recorded chip sequences, which may be corrupted by chip errors, into $M = 16$ clusters using the k-means clustering algorithm [24]. The k-means clustering algorithm computes the centroids as a mean of the hamming distance of each chip sequence within the corresponding cluster. These centroids are then compared to

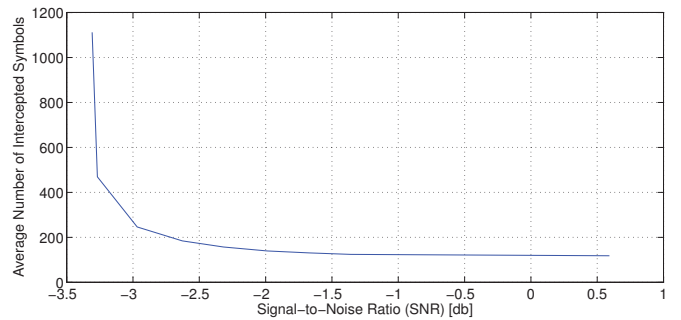


Fig. 5. Averaged number of intercepted chip sequences needed to successfully perform the attack described in Algorithm 1.

the real chip sequences used by the legitimate nodes. If not all chip sequences have been found, the attacker needs to add more of them to perform the attack.

Algorithm 1 Attack Algorithm.

- 1: Record signal from wireless communication channel
- 2: $RecSeqs \leftarrow$ partitioned signal into chip sequences
- 3: Initial chip sequence pool: $SeqPool = \{\}$
- 4: Number of used chip sequences: $z = 0$
- 5: List of found chip sequences: $foundSeqs = \{\}$
- 6: $SeqPool$ increased by y chip sequences at each iteration
- 7: **while** not all chip sequences found
- 8: $z = z + y$
- 9: Append y chip sequences from $RecSeqs$ to $SeqPool$
- 10: $centroids \leftarrow$ k-means($SeqPool$)
- 11: compare centroids with $RealChipSequences$
- 12: **if** new chip sequences have been found
- 13: append found chip sequences to $foundSeqs$
- 14: **return** z

The results of Algorithm 1 based on the empirical measurements are plotted in Figure 5. It shows how many chip sequences were required in total to find all real chip sequences of a code for different SNR values. We see that the average number of required chip sequences is rather low (i.e., between 100 and 200) for low SNR values and starts to increase exponentially around -3 dB. The number of chip sequences the attacker requires depends on the SNR: the lower the SNR, the higher the chip error rate, the more chip sequences are required. Below -3 dB, the chip error rate becomes so high that for a given chip sequence several erroneous chip sequences are received before an error-free one. Above this value, even if chip errors occur, they are very seldom.

C. Code Synchronization Time

We define the link setup time as the time between node discovery and until PSCHP has synchronized all codes and is ready for transmission. The average link setup time with our implementation is 110 ms. The protocol overhead of PSCHP to maintain the synchronization is approximately 0.6% of the transferred payload. More details on PSCHP and its evaluation can be found in [25].

VIII. DISCUSSION ON FURTHER IMPROVEMENTS

We have shown that using secret spreading codes produces an obfuscation gain while only slightly decreasing the performance of IEEE 802.15.4. Performance loss was on average

less than 1 dB. The detection gain is $\frac{N_a}{N_r} \cdot \delta + 2.3$ dB and the interception gain is $\frac{N_a}{N_r} + 8.8$ dB. Whether these gains are sufficient depends on the scenario of interest and the attacker's capabilities. Depending on the resulting ratio of the noise figure of the attacker to the noise figure of the legitimate receivers and the incoming power level at the attacker, the attacker will operate in one of four transitional regions. For example, when decreasing its distance to the transmitter, the attacker may increase its power level and move into a more favorable region to perform detection or interception. The transmitter will need to decrease its transmit power. However if the receiver is too far away, there is a point when the receiver will no longer be able to correctly demodulate and decode the packets. In this case, a higher obfuscation gain is required.

A possible way to further increase the obfuscation gain is by increasing the processing and/or coding gain. This can be achieved by increasing the signal width and the code sequence length, respectively. For example, the 2.4 GHz ISM band used by IEEE 802.15.4 foresees 16 channels of 2 MHz width with carrier frequency separation by 5 MHz. In principle the signal could further be spread over the whole 16 channels offering a theoretical processing gain increase of more than 15 dB and hence also a higher obfuscation gain. Increasing the spreading factor comes however at the cost of higher energy consumption at the receiver and might also require more complex transceivers in multipath environments. Furthermore, commercial off-the-shelf IEEE 802.15.4 hardware might not be able to deal with different signal widths and spreading code lengths, requiring proprietary and presumably more expensive hardware platforms. We leave such regards up to future work.

A further improvement that comes along with random codes is jamming resistance. It is well known that a jammer will have the highest impact by jamming with a signal that has the same frequency components as the signal to be jammed. When the codes are known by the attacker, jamming with a modulated signal that uses the same spreading codes as the legitimate traffic will achieve this goal. When the codes are unknown, a jammer can at best use a modulated signal using random codes. However, as these random codes will be different than the codes used for communication, the effect of the jamming signal will be less severe than when the codes are identical. Although we did not touch jamming issues in this paper, we consider this property as another motivation to use random spreading codes in adversarial settings with IEEE 802.15.4.

IX. CONCLUSIONS

We proposed to use secret and pairwise-dynamic, random spreading codes in IEEE 802.15.4 instead of fixed and public ones in order to obfuscate the communication at the physical layer. Our results show that the approach improves obfuscation at the cost of a marginal performance degradation in terms of packet error rate. While this paper has focused on IEEE 802.15.4 networks, the idea could presumably be applied to other existing commercial standards like IEEE 802.11 that also specify a SS modulation. Although many applications may not require obfuscation properties in real-life, we believe that this kind of technique is useful and should be considered by standardizing bodies for commercial technologies. Secret

spreading codes could be provided as an optional feature in future technologies in addition to higher layer encryption as an additional line of defense against emerging threats.

ACKNOWLEDGMENT

This work was partially funded by armasuisse under the communication and cyberspace research program and the European Commission under the SCAMPI (FP7 – 258414) FIRE Project. It represents the views of the authors.

REFERENCES

- [1] F. Zhang, W. He, X. Liu, and P. Bridges, "Inferring Users Online Activities through Traffic Analysis," in *ACM Wisec 2011*.
- [2] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," *Passive and Active Network Measurement*, pp. 165–175, 2007.
- [3] R. A. Poisel, *Modern communications jamming principles and techniques*. Artech House Publishers, 2004.
- [4] D. L. Adamy, *EW 102: A Second Course in Electronic Warfare*. Artech House, 2004.
- [5] F. Hermanns, "Cryptographic CDMA code hopping (CH-CDMA) for signal security and anti-jamming," in *EMPS 2004*. ESA/ESTEC.
- [6] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis," *Mobile Networks and Applications*, vol. 10, no. 3, p. 325, 2005.
- [7] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall, "Improving wireless privacy with an identifier-free link layer protocol," in *MobiSys 2008*.
- [8] K. Bauer, D. McCoy, B. Greenstein, D. Grunwald, and D. Sicker, "Performing traffic analysis on a wireless identifier-free link layer," in *ACM TAPIA 2009*.
- [9] T. Jiang, H. Wang, and Y. Hu, "Preserving location privacy in wireless LANs," in *ACM MobiSys 2007*.
- [10] A. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *PERCOM Workshops 2004*.
- [11] P. Kamat, W. Xu, W. Trappe, and Y. Zhang, "Temporal privacy in wireless sensor networks: Theory and practice," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, pp. 1–24, 2009.
- [12] W. Shbair, A. Bashandy, and S. Shaheen, "A New Security Mechanism to Perform Traffic Anonymity with Dummy Traffic Synthesis," in *IEEE CSE 2009*.
- [13] A. M. Mehta, S. Lanzisera, and K. S. J. Pister, "Steganography in IEEE 802.15.4 Wireless Communication," in *ANTS 2008*.
- [14] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2004.
- [15] IEEE, "Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans), iee standard 802.15.4-2006," Tech. Rep., 2005.
- [16] S. Lanzisera and K. S. J. Pister, "Theoretical and Practical Limits to Sensitivity in IEEE 802.15.4 Receivers," in *ICECS 2007*.
- [17] J. Proakis, *Digital Communications*, 3rd ed. McGraw Hill, 2001.
- [18] H. Wang, J. Guo, and Z. Wang, "Cluster-based Blind Estimation of Mary DSSS Signals," *ICC 2008*.
- [19] D. Newman, "The double dixie cup problem," *American Mathematical Monthly*, vol. 67, no. 1, pp. 58–61, 1960.
- [20] T. Schmid, "Gnu radio 802.15.4 en- and decoding," *UCLA Technical Report*, 2006.
- [21] E. Rescorla, "Diffie-Hellman Key Agreement Method," RFC 2631 (Proposed Standard), Internet Engineering Task Force, 1999. [Online]. Available: <http://www.ietf.org/rfc/rfc2631.txt>
- [22] SECG, "SEC 1: Elliptic Curve Cryptography," Standards for Efficient Cryptography Group, Tech. Rep., 2000. [Online]. Available: http://www.secg.org/collateral/sec1_final.pdf
- [23] C. Popper, M. Strasser, and S. Capkun, "Anti-jamming broadcast communication using uncoordinated spread spectrum techniques," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 5, pp. 703–715, 2010.
- [24] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [25] B. Muntwyler, V. Lenders, F. Legendre, and B. Plattner, "Physical Layer Security: Pushing Encryption Down the Stack," ETH Zurich, ETHZ-TR-2011-10, Tech. Rep., 2011. [Online]. Available: <http://people.ee.ethz.ch/lfranck/TR-PSCHP.pdf>