Jamming for Good: A Fresh Approach to Authentic Communication in WSNs

Ivan Martinovic, Paul Pichota, and Jens B. Schmitt
TU Kaiserslautern
Distributed Computers and Systems Lab
67653 Kaiserslautern, Germany
{martinovic,p_pichota,jschmitt}@informatik.uni-kl.de

ABSTRACT

While properties of wireless communications are often considered as a disadvantage from a security perspective, this work demonstrates how multipath propagation, a broadcast medium, and frequency jamming can be used as valuable security primitives. Instead of conventional message authentication by receiving, verifying, and then discarding fake data, sensor nodes are prevented from receiving fake data at all. The erratic nature of signal propagation distributes the jamming activity over the network which hinders an adversary in predicting jamming nodes and avoids selective battery-depletion attacks. By conducting real-world measurements, we justify the feasibility of such a security design and provide details on implementing it within a realistic wireless sensor network.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Design, Security, Performance

Keywords

Wireless Sensor Networks, Authentication, Jamming

1. MOTIVATION

The conventional approach for protecting computer networks is to rely on cryptographic primitives. Such an approach is considered as beneficial because it abstract from the physical properties of communication and thus supports security design in different networks and scenarios. However, it has evolved from wired medium and point-to-point communication. Consequently, assumptions upon which the current security mechanisms are based, such as similar hardware capability of devices and their energy availability, are often contradictory in nature to those of wireless communication. Wireless devices are heterogeneous to such an extent that even a common key-exchange computation, which may be considered as trivial for some devices, oftentimes presents a high computational burden for others. Another example in which discrepancies between a conventional security design and wireless networks become visible is message authentication. To verify the cryptographic authenticity of a message, a device is forced to receive it (depending on the MAC protocol even an ACK is returned), compute the message digest, and then eventually discard it. For battery-powered devices, e.g., sensors or mobile phones, such tasks are not an advantageous resource investment and inherently allow for selective battery-depletion attacks. Additionally, attacks against stateful protocols, which are usually a precondition for conventional key-exchanges, are especially effective on the shared broadcast medium. The adversary can simply choose to block any message exchange by taking advantage of frequency jamming or launching a number of resource-depletion attacks by flooding with fake requests. This leaves us with mixed feelings – while an adversary takes full advantage of the wireless communication to attack, the security design abstracts from it, even though there is a wide spectrum of features that can be used to strengthen security. Recently, a number of contributions apply properties of wireless communication to extend cryptographic methods (see, e.g., [3, 9, 1, 14, 15, 12, 8, 5]). For example, in [9] authors experimentally show how to derive a cryptographic key from the wireless channel using commodity hardware and as such avoid traditional key-exchange protocols. However, in this work we completely abandoned cryptographic methods and demonstrated how a novel security design can be created relying merely on physical properties of wireless communications.

One important property that can enrich existing protection mechanisms is the ability of frequency jamming. Although it is usually considered as one of the most powerful adversarial tools (and the reason why availability in wireless networks is often downgraded as a security objective), the ability to jam is not an exclusive property of the adversary. In this work, we introduce the concept of *attack cancelation*, a mechanism to prevent legitimate sensor nodes to receive impersonated and unauthenticated transmissions. By turning jamming against the adversary and using signal properties to detect impersonation, legitimate WSN nodes are able to destroy fake frames while still being "in-the-air" and as such avoid useless investment of resources by first receiving and acknowledging, then verifying, and finally rejecting the fake data. Since only fake frames are jammed, correctly received frames can be considered authentic and no further security-related tasks are required. This also implies, that during a normal network operation security mechanisms are not visible (in contrast to "always-on" cryptographic authentication) and additional costs are avoided.

2. WIRELESS SECURITY PRIMITIVES

In the following, we describe some experimental results that demonstrate the unpredictable nature of the signal propagation. The scenario we focus on is an indoor WSN assuming that an adversary

WiSec'09, March 16–18, 2009, Zurich, Switzerland.

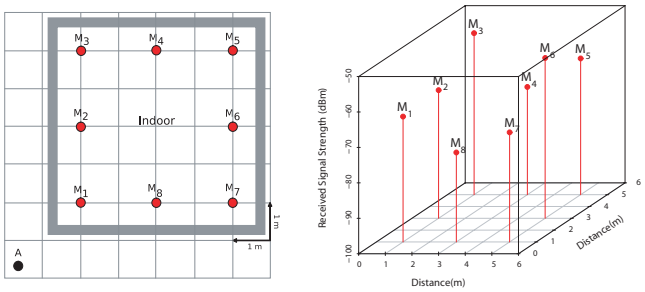

Figure 1: Residential monitoring scenario used as testbed (left) and signal strength measured on indoor nodes from an outdoor sender (right).

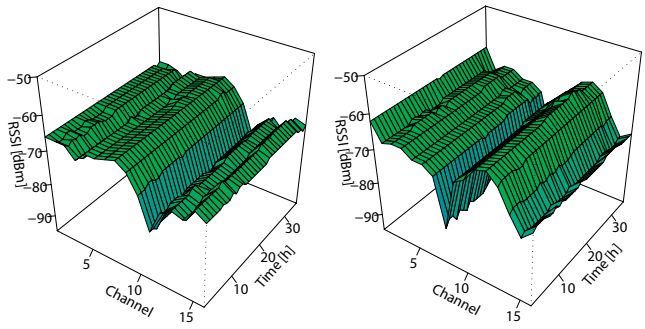

Figure 2: Results of sampling all available wireless channels on three different physical positions (both receivers are $< 2m$ from the sender).

does not have physical access to the network. We also assume that the WSN operation is divided into two phases, the deployment phase where legitimate sensors are positioned and no adversary is present, and the operational phase during which the sensors do not change physical positions and the adversary attempts impersonation and injection attacks.

A corresponding WSN scenario was deployed in our university lab where the sensors were positioned on the ceiling of the lab maintaining a Line-Of-Sight (LOS) connection and the adversary is positioned outside, as depicted in Figure 1 (left) (also described in [6]). The WSN is based on the MicaZ platform with CC2420 radios [13], allowing for 8-bit resolution measurements of received signal strengths. The power measurements are reported in RSSI (Received Signal Strength Indicator), yet the conversion to dBm is easily computed by $P_{dBm} = RSSI_{VAL} + RSSI_{OFFSET}$, where $RSSI_{OFFSET} \approx -45$.

2.1 Radio signal propagation

Let us assume there is a sensor A sending a message m on an arbitrary power level $p \in \mathbb{P}$ and frequency $f \in \mathbb{F}$ into the network. Let us further assume the message can be received by a number of k indoor sensors, denoted as $M_1, \ldots, M_k$.

Upon reception, each sensor individually computes the received signal strength denoted as a function $RSS_{M_i}(m)$ with $i = 1, \ldots, k$. The results are pairwise distinct with high probability due to arising phase shifts, multipath propagation, and distance. Sorting the RSS values of all sensors in decreasing order, we can express the senders signalprint (we use the same terminology as introduced in [4]) over the indoor environment as a totally ordered set S with $\leq$ as underlying relation:

$$S_A = \{RSS_{N_1}, \ldots, RSS_{N_k}\} \text{ subject to } p \in \mathbb{P}, f \in \mathbb{F}$$

Now, in this particular example given in Figure 2, the signalprint $S_A = \{M_3, M_6, M_5, M_2, M_1, M_4, M_7, M_8\}$, since the sensor M_3 has the highest measured RSS ($\approx -55\,dBm$) and sensor M_8 the weakest RSS ($\approx -75\,dBm$).

It should be clear that applying another power level from the set $\mathbb{P}$ does influence the yielded RSS values at each sensor, but the obtained order, i.e., the signalprint remains the same since solely proportionate increases and decreases will arise on all sensors. On the other hand, when the frequency is changed to $f' \in \mathbb{F}$, with $f \neq f'$, the transmitted signal is affected by a different multipath propagation which results in a changed signalprint.

This can be seen in Figure 2, where transmitting on different wireless channel results in different RSS relations among receivers. For example, transmitting on the channels 8, 14, and 11 results in relations $RSS_{M_1} > RSS_{M_2}$, $RSS_{M_1} < RSS_{M_2}$, and $RSS_{M_1} \approx RSS_{M_2}$, respectively. Hence, by simply changing frequencies different signalprints from the same device and physical position are produced.

2.2 Transmission control

Since we assume that an adversary is not limited by its hardware capabilities, its transmitted signal can be several magnitudes stronger than transmissions of the legitimate WSN sensors. To limit the adversary in abusing this hardware advantage, the idea is to apply transmission power control within legitimate WSN which is available on most of the currently popular WSN platforms, e.g., the CC2420 radio supports 32 different transmission power settings with output power ranging from -25 to 0 dBm. By setting the RSS_{min} and RSS_{max} values only frames with $RSS \in [RSS_{min}, RSS_{max}]$ are accepted and processed. This simple countermeasure forces an adversary to chose between two attack vectors: *(i)* attempting to inject fake frame on receivers by adapting its transmission power or, *(ii)* launching a frequency jamming attack by producing $RSS > RSS_{max}$, however in this case its frames are not accepted by the receivers, i.e., this attack can only serve for jamming and not for impersonation. Importantly, in the more serious attack where the adversary is interested in injecting the fake frames into the network, the legitimate sensors are able to produce $RSS \geq RSS_{max}$ which is a precondition for a successful jamming of the attacker's transmissions as we show later in this work. In addition, by adapting the $[RSS_{min}, RSS_{max}]$ interval to different values, depending on the density of the network, the WSN can force the adversary to "speak loud" and hence, its transmission can be sampled by more sensor nodes.

2.3 Jamming for Good

Low-cost WSN devices are, in general, not capable of sending and receiving frames at the same time. For example, in case of CC2420 radios, when switching from receive mode to transmit mode there is an additional resynchronisation delay of 12 symbol periods [13]. Since sensing the transmission is required for analyzing its properties and deciding whether to jam or not, the sensors are not able to analyze the authenticity of a frame and at the same time to jam others from receiving it. Therefore, to support jamming of the fake data frames, the idea is to integrate jamming into the communication protocol. This is done by separating the data exchange into two frames – a small *Data Follows Notification* (DFN) frame, and a *Data Content* (DATA) frame. After the DFN frame is sent, there is a predefined time interval, called *Inter-Frame Time Gap*, after which the DATA frame is expected to arrive. The purpose of this timegap is twofold. First, receivers addressed by the DFN

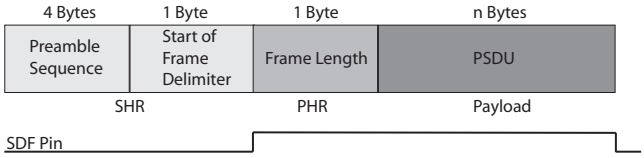

Figure 3: IEEE 802.15.4 Frame Format (Data Frame) and its reception as detected by chip CC2420

compute the point in time when the DATA frame is due to arrive and reject its processing when it is received too soon or too late, i.e., the DFN frame *commits* the sender to a defined transmission period of the DATA frame. Second, other sensors are provided with enough time to analyze the authenticity of the DFN frame, and in case of impersonation detection, they are able to timely schedule a *jamming* (JAM) frame to intercept the DATA frame.

In the following, we focus on the isolated task of controllable and planned jamming, and experimentally analyze its feasibility especially the timing accuracy in scheduling DFN, DATA, and JAM frames. These questions where initially addressed in [2]. However, in this work we extend our analysis and integrate them within a real-world WSN.

3. JAM WHERE IT HURTS

First, we briefly introduce the IEEE 802.15.4 PHY frame reception and identify the points of action to intercept it. However, it is important to note that the overall concept presented in this work is not limited to either a specific medium access protocol (MAC) or physical layer (PHY).

Figure 3 depicts the composition of a typical data frame according to the IEEE 802.15.4 standard, divided in MAC and PHY parts in the scope of the specification. Two blocks are added to the MAC frame, namely the synchronization header (SHR) followed by the physical header (PHR), whereas the latter solely comprises the Frame Length which indicates when a receiver will assume the frame to be complete.

Each transmission is preceded by a Preamble Sequence consisting of 4 consecutive bytes of $0x00$ serving for *symbol* synchronization and frequency offset adjustments at potential receivers. The signal is then followed by the start of frame delimiter (SFD) of one byte containing the value $0x7A$ for the purpose of *byte* synchronization and to indicate the end of this phase.

If a potential receiver could not detect the complete SHR because of interference or selective signal strength, it will ignore the current transmission and immediately start to look for the next preamble. Hence, reception of the following data, i.e. Frame Length and MPDU carrying the payload, is only possible if the preceding synchronization bytes could be received without error.

As long as a sensor is not transmitting or receiving a frame, it continuously scans for a sequences of at least 4 bytes of $0x00$ and expects them to be concluded by the SFD byte. Once this specific byte has been received, the SFD pin becomes active locking the transceiver to the detected transmission until as many bytes as indicated by the length field have been received. Once the transmission has finished the SFD pin is reset and the transceiver senses the channel for the next preamble.

Importantly, if the preamble is correctly detected but the SFD byte does not match the expected value, for instance if a single bit is inverted, then the SFD pin will remain inactive and the current frame will be ignored by the transceiver chip. As a result, it is not necessary to interfere for the complete duration of the transmission, but it is sufficient to destroy a single byte of the SFD frame.

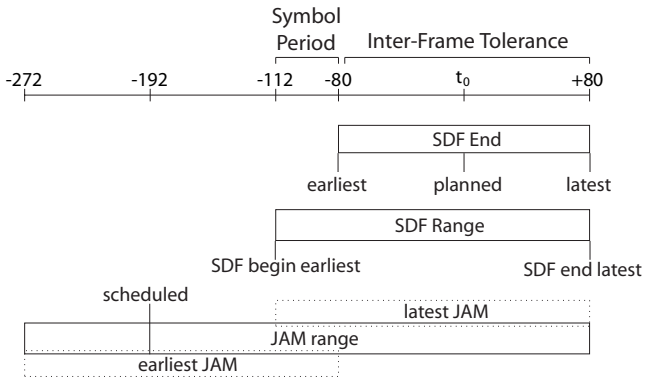

Figure 4: Jamming timing

3.1 Dominant vs. Submissive Jamming

There are two possible outcomes of the successful jamming, either nothing meaningful is received or the JAM frame arrives. Both outcomes are considered as an effective countermeasure as long as the fake DATA frame is not successfully injected and processed by legitimate sensors. To effectively jam the DATA frame, a jammer can *(i)* transmit a signal strong enough to interfere at the moment an SFD field is about to be transmitted, or *(ii)* attempt to "prematurely lock" the receiver to JAM transmission, i.e., if the SFD byte of JAM frame is received correctly, the receiver will not start processing another transmission before the current one is finished. The first approach we call *dominant* since it depends merely on transmission power, and the second we refer to as *submissive* jamming. Clearly, the submissive jamming seems more appropriate since it does not assume that a jammer's signal is necessarily dominant. However, as we experimentally analyzed, there is a pitfall in this approach. The problem lies in a preamble sequence which is supposed to have a certain length, however its actual duration is arbitrary long as it comprises at least 4 bytes and is completed by the the start of frame delimiter. The attacker can harness this detail by extending his preamble and starting the transmission ahead of time. If it yields a stronger signal at the intended receiver, thus drowning all others, the premature lock will fail and the submissive approach will be useless.

Hence, we must consider signal strength as the crucial factor and follow the dominant jamming approach. For this approach it is important to define the transmission power and the resulting RSS which should be strong enough to successfully jam DATA frames. The specification of the CC2420 [13] states a co-channel rejection of $3\,dB$, expressing the least difference that will yield a packet error rate (PER) of less than $1\,\%$. Yet, we are interested in the complementary event, that is the least difference that exhibits a preferably large PER. In [11], the authors introduce and investigate the notion of the relation between the actually received transmission and other interfering ones termed as *signal-to-interference-plus-noise-ratio* (SINR). From the contribution of [11], we can draw the conclusion that the jamming signal should be as strong as the one we want to interfere with. Using different configurations of an experimental setup, in Section 3.3 we will conduct our own experiments to validate this observation.

3.2 Timing and Jam Duration

We turn now to the timing issues important to successfully intercept and jam the SFD frame field. The factors that constitute delay in switching from receiver to a sender are $128\,\mu s$ required for transceiver recalibration (5 $byte$ of synchronization header at 250

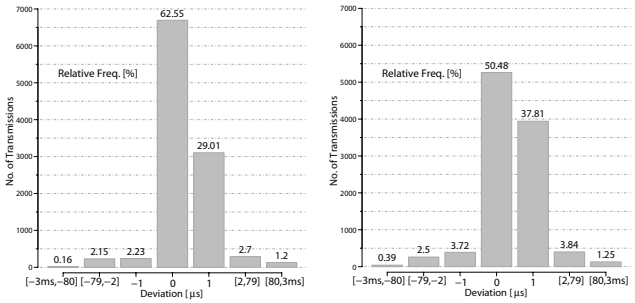

Figure 5: Timing deviations of DATA and JAM frames scheduled 5 *ms* after the DFN frame.

kbps) along with an $\approx 186\,\mu s$ on average for in-system processing, i.e., copying data segments from the microcontroller memory to a transmission buffer. Hence, the mean delay itself is not that crucial, since it can be compensated by a time offset. More important is the variance and granularity of available timers.

TinyOS supports two timer abstractions referred to as *Timer*, which offers a precision in milliseconds and runs synchronously within the task context, and *Alarm*, which supports a granularity of microseconds and runs asynchronously within the interrupt context. To find out if it is possible to take advantage of the Inter-Frame Time Gap, i.e., of a sender's time commitment, the following experiment was run using MicaZ sensors. The sender transmits the DFN-DATA frames and the receiver measures the precision of arrivals, where the Inter-Frame Time Gap was set to 5 *ms*, i.e., after it receives DFN, ideally after exactly 5 *ms* the corresponding DATA should arrive[1]. The response variable measured was the deviation from the 5 *ms* of Inter-Frame Time Gap for both, senders and jammers. The results are depicted in Figure 5 as the cumulative and relative frequency over more then 6000 samples transmitted every 100 ms, for both DFN-DATA and DFN-JAM transmissions. As it can be seen, more than 98% of samples gathered within an interval of $[-79\mu s, 79\mu s]$ in both scenarios.

This timing accuracy of legitimate sensors allows for the following jamming scheme. First, we neglect the propagation delay in the wireless medium and set the transmit delay to a value of $32\,\mu s$ per Byte which results from the maximum data rate of $250\,kbps$ that is specified in the standard. Further, for the sake of simplicity, we will assume $80\,\mu s$ instead of $79\,\mu s$. Let time t_0 denote the exact point in time when the reception of the SFD byte is expected to be complete (refer to Figure 4). Since this cannot be precisely achieved in a large number of cases due to the limited timing accuracy that was observed, a so called target-zone of $80\,\mu s$ is defined around t_0, such that messages arriving within $[t_0 - 80; t_0 + 80]$ are accepted. Data messages outside the interval will not be processed and simply rejected. With respect to the transmission delay given as $32\,\mu s$, this results in an interval from $t_0 - 112$ to $t_0 + 80[\mu s]$ that needs to be covered by the jam frame, because this comprises the complete spread of the SFD transmission. Since the jammer itself is subject to limited accuracy (every WSN node can be a potential jammer), the range has to reflect its uncertainty as well, that is to say the transmission can commence $80\mu s$ before or after the intended point in time. Thus, if we scheduled the jam message to start at the earliest time the SFD byte can arrive, the actual transmission would begin sometime between $t_0 - 192$ and $t_0 - 32[\mu s]$. In order to eliminate all uncertainties, the jam transmission must be scheduled at $t_0 - 192$ and endure for a time-frame of $[t_0 - 272; t_0 + 80]$. In total, this yields a duration of $352\,\mu s$ which is equivalent to a jam frame comprising 11 bytes in size, including the Physical Header.

[1]We have chosen 5 ms as an adequate value, since it does not significantly impact the throughput of the WSN, and provides enough time to prepare for jamming.

	1 *Jam*	2 *Jams*	3 *Jams*
$RSS(J \to R) < RSS(S \to R)$	0%	78%	80%
$RSS(J \to R) \approx RSS(S \to R)$	94%	98%	99%

Table 1: Experimental results of real-world jamming scenario.

3.3 How Many Concurrent Jammers?

Using the previous jamming scheme, we extended every WSN node with jamming capability and performed an experimental analysis of jamming in a real-world WSN. The test bed was similar to the one introduced at the beginning of this work and the response variable was the number of jammers required to successfully jam a frame. We have set two levels of signal strength relation between the sender and the jammer with respect to the receiver. In one configuration the sender's RSS at the receiver was ≈ 6 dBm stronger than jammer's RSS, and in another configuration both RSS were approximately equal. During experiment we collected ≈ 1000 samples for each configuration using different positions of sensors but keeping the same RSS relation. In Table 1, the results of jamming success for both relations of RSS are shown.

As assumed, in case there is only one jammer and in terms of RSS, a stronger sender, there is no success in jamming. However, if the jammer is at least equally strong, than there is $\approx 93\%$ of jamming success. Further, in case that there are two jammers, even if their RSS is a weaker than the senders (up to 6 dBm as predefined in the experiment) there is $\approx 72\%$ success in jamming, and $\approx 98\%$ if RSS is equal. Finally, for three jammers having equal RSS as the sender, there was $\approx 99\%$ of jamming success, and $\approx 80\%$ if jammer's RSS is weaker. These empirical results provide a feeling for the number of jammers that should be active during a fake transmission. Also, since acceptance intervals limit an adversary's RSS to RSS_{max} during an injection attack, legitimate nodes are able to transmit at least equally strong signal as an adversary.

4. WIRELESS SECURITY DESIGN

The main objective in the security design of the described scenario is data authentication. A WSN should be able to verify whether sensor data originated from legitimate sensors. To fulfill this objective, our protection concept is based on two mechanisms composed from the wireless security primitives:

- *attack detection* and
- *attack cancelation.*

Using the detection mechanism, fake transmissions are identified, while the attack cancelation utilizes jamming to prevent sensors from receiving the fake data. This security design significantly differs from conventional. Rather than receiving data, analyzing it and then rejecting it, the WSN can be sure that transmissions that can be correctly received are also authentic. In [6], we have introduced the concept of attack detection which we briefly describe here to provide a full picture of a security design. However, in this work we focus on the attack cancelation phase and the corresponding jamming activity.

4.1 WSN Deployment and Operation

During the WSN deployment phase, the sensors are manually placed within an indoor environment. We assume that the MAC

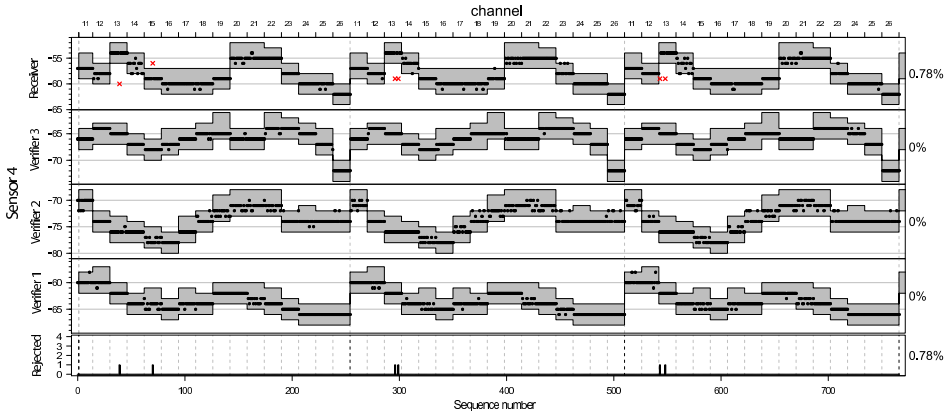

Figure 6: Time-line of legitimate transmissions verified by 3 sensors and 1 receiver. Legitimate sensors dynamically switch between different frequencies and transmission power (gray area are the acceptance intervals, black dots inside intervals are accepted frames)

protocol is given and the nodes are able to create forwarding tables under a common feed-forward topology. During this phase, no adversary is assumed and the nodes use initial measurements to build statistics over the legitimate communication. The resulting acceptance intervals are defined over RSS as $[\mu - k\sigma, \mu + k\sigma]$, where μ is the sample median, σ the standard deviation of a sample, and $k > 1$ is an environment-dependent constant defining the width of the interval, i.e., it describes within how many standard deviations the RSS of the DFN frame is still considered to be legitimate. After initial measurements each nodes keeps a table with the acceptance intervals given a node ID, wireless channel, and the transmission power reported by the sender. Clearly, such intervals can be further improved to be more robust against channel fluctuations to minimize false positives, e.g., [12, 4, 5] demonstrate the feasibility of detecting fake transmissions using statistics over the wireless channel.

4.2 Attack Detection

After defining acceptance intervals, the operational phase of the WSN begins during which the presence of an adversary is assumed. The WSN task is common to a typical multihop WSN scenario; the sensors monitor environmental conditions and periodically transmit sampled data to the next dedicated neighbor in direction of the sink. During this phase all sensors *periodically* change their transmission frequencies and power levels. For example, one configuration would be to transmit on channel 12 with a transmission level 10, another to transmit on channel 22 with a transmission level 15. An example of such operation is shown in Figure 6 which depicts a trace of a real-world implementation (using the WSN throughput of 10 frames per second). The sensors use the same PRNG seed to select the equal sequence of transmission frequencies. They also implement a simple synchronization mechanism based on beacon frame transmitted by the base station to compensate for clock drift and signals the change of transmission parameters. As shown in Figure 6, the sender's transmission is verified by four other nodes. If the frame does not comply to any of acceptance intervals, the sensor considers the frame as fake and initiates the jamming procedure as we discuss later in this section. Hence, to be able to inject frames from another physical position, the adversary must find a configuration of transmission properties that fulfill the acceptance intervals on the receivers and all other nodes which are able to detect its transmission. Importantly, such attack cannot be prepared "offline" as it is the case with a cryptographic brute-force. Active probing for acceptance intervals is an intense and active attack, and WSN can introduce further methods to detect and countermeasure such probing. This is also the reason why the WSN dynamically changes its transmission frequency and the sending power. Even if the adversary finds a physical position and an appropriate transmission configuration, the periodic change of the frequency affects the RSS relation within the legitimate WSN (as demonstrated in Section 2.1) and invalidates the configuration used for the attack. Thus, the signalprint which was "broken" by the adversary is not valid for communication using other transmission configuration.

4.3 Attack Cancelation

After detecting DFN frames which do not comply with acceptance intervals, a legitimate node can decide to schedule the JAM frame to intercept the DATA frame. Hence, in this subsection we are concerned with the question – who should jam and how often? Instead of considering sensors as isolated entities, we are interested in the global network behavior resulting from their local decisions. The protection should not require any control traffic to distribute the knowledge and decisions of the jammers, but solely rely on passively monitoring the wireless channel during a node's awake phase.

The desired network behavior is that the jamming task should be fairly (in the best case, uniformly) distributed over the network, or at least over a part of it, and secondly, the adversary should not be able to exactly predict the jammers of the next fake transmission. This means that even if the jamming countermeasure can be abused by an adversary for battery-exhaustion attacks, i.e., by triggering jamming activity of the network, the adversary cannot selectively attack the legitimate nodes. If it launches a battery-depletion attack, which is possible in every WSN application, for each fake frame an unpredictable set of nodes will concurrently jam. Hence, during a long-term battery exhaustion attack the complete network or at least a significant, yet unpredictable part of the network will simultaneously lose battery power. Such an outcome is more desirable than allowing the attacker to plan and to selectively attack and turn off nodes one after another (which is the case with conventional security design).

A further aspect is to reduce the jamming redundancy by controlling the ratio of jams per fake transmission. The appropriate number for concurrent jammers we take from our experimental measurement as discussed in Section 3.3 which is set to three active jammers per fake frame. We are therefore interested in having large number of *potential* jammers (nodes that detect the injected frame), but keep the number of *active* jammer low.

4.4 Adaptive Jamming

Algorithm 1: Active WSN Node

```
Input: rcvFrame containing fields: sender, rssi, freq, power,
       type
if type == DATA then return ;
if (sender,freq) ∉ reachable_neighbours then return ;
if sender == self || rssi ∉ interval(sender, power, freq) then
    if random() > p then
        prepare and send JAM frame ;
        decrease p ;
    end
    else
        wait until timegap is over ;
        if DATA received then increase p ;
        else decrease p;
    end
end
```

To keep the adversary from guessing the next jammer and to avoid permanent jamming from a single node, each node bears an individual probability p for jamming a detected impersonation attack, i.e., after detecting the fake DFN there is a probability p to schedule the JAM frame. We refrain from introducing control traffic to distribute the knowledge of p over the network and therefore each awake node only uses monitoring of the wireless channel to adapt its jamming probability. Since the successful jam and the number of active jammers cannot be explicitly recognized, the nodes may only distinguish between the following three cases during their monitoring of the channel:

1. $DFN\ impersonated \wedge DATA\ received$
2. $DFN\ impersonated \wedge JAM\ received$
3. $DFN\ impersonated \wedge JAM\ sent$

From the local point of view, the first case implies that there was no jamming activity. By detecting the fake DFN and receiving the DATA frame the node assumes that the attack was successful (there may be another node which accepts the fake DATA). Hence, the node increases its jamming probability p by applying some function $f_1(p)$. The opposite is assumed for the other two options. If the JAM frame is received the jamming activity can be assumed, and the node applies another function $f_2(p)$ to appropriately decrease it. Similarly, the third case also decreases the p since the node itself has sent a JAM frame. More specifically, the behavior of the node that can detect and verify a transmission is listed in Algorithm 1. The if-case where the sender is not contained in the variable of *reachable_neighbours* limits the jamming activity only to those nodes which can mutually reach eachother as discovered during the deployment phase of the network and initial transmissions.

We now turn to choice of the functions $f_1(p)$ and $f_2(p)$, which is crucial for the security, since to manipulate the jamming probability an adversary can first induce jamming and thus, decrease p at jammers, and then attempt to inject frames once the number of jammer is low. The problem we deal here reminds of the famous TCP congestion control, yet rather then avoiding congestion, in this case we want to achieve it, i.e., during an attack there should be enough JAM frames to "congest" the wireless channel, however to avoid jamming redundancy, the network should promptly react to decrease p. Hence, one foreseeable solution would be to turn the TCP's AIMD into the MIAD, i.e, the additive-increase and multiplicative-decrease to use as multiplicative-increase and additive-decrease. This means, we rapidly increase the jamming probability when a successful attack is detected, but slowly decrease the probability during jamming activities. We have experimented with this method and, while attack was successfully inhibited, there were too many redundant jammers. The problem with MIAD was that, after the nodes reached probability near 1, they did not decrease p fast enough and many jammers kept being active. More favorable behavior would be if a node rapidly increases p once it is low, and rapidly decreases p once it is high, but in both cases slows down as it converges to any of these to events. For this reason we chose the LILD - *logarithmic-increase* and *logarithmic-decrease* method. Using LILD the jamming behavior is define as:

$$\text{increase: } f_1 : p \leftarrow p \cdot (1 - log_{10} p)$$

$$\text{decrease: } f_2 : p \leftarrow p \cdot (1 - log_{10} p)^{-1}$$

There are also other interesting and more sophisticated approaches to adapt the jamming behavior, but they are currently in progress and belong to our future work.

5. OVERALL NETWORK ANALYSIS

In this section we are interested in an overall network behavior under a more sophisticated thread model. Since empirical evaluation and search for a physical position to attack the network is very time-consuming and limited by walls and other obstacles, we deploy the simulation using lessons learned from our experiment analysis.

5.1 Evaluation Goals

Following a brief introduction to the attacker's capabilities and the network model itself, we present the wireless channel model and reason against alternatives like free-space and Rayleigh fading [10]. The evaluation itself discusses the impact of an attacker's effort to impersonate a legitimate sensor with the purpose of injecting false information into the network whose outcome is measured as the attack success. This is opposed by the number of sensors that are able to detect the attack, those actually taking countermeasures by interfering with the transmission, and eventually the distribution of individual jamming efforts on a global scale.

5.2 Threat Model

An attacker is not bound to any specific hardware, but he must be able to communicate with the wireless sensor network, specifically in terms of frequency band and communication protocols deployed on the PHY and MAC layer. Besides these very basic preconditions we do not impose any further restrictions concerning the maximum output power level or the location as long as he cannot gain access to the physical position of the legitimate node.

During the simulation, the attacker impersonates the identity of one chosen intermediary sensor I in the network and tries to inject messages on its behalf at the corresponding sensor T. In order to determine the impersonation success he repeatedly emits 100 messages addressed to T for each wireless configuration represented by a tuple $(f, p, p*)$, where f denotes the frequency, p the applied power level and $p*$ the pretended power level. His goal is to maximize the number of messages sent on behalf of I which are correctly received and accepted by T for each frequency by adjusting the parameters p and $p*$. This process is then repeated for each randomly chosen position that is to be sampled.

5.3 Wireless Channel

Our longterm measurements (previously shown in Figure 2) exhibit frequency selective effects on the received signal strength that

Table 2: Simulation Parameters

$RSSI_{min}$	$-85\,dBm$
$RSSI_{max}$	$-75\,dBm$
maximum range	$30m$
interval width	6
# channels	16
# power levels	8
# sensors	28
datarate	$250\,kbps$
timegap target	$5\,ms$
timegap tolerance	$79\,\mu s$
channel model	ground-reflection (with stochastic means)
confidence level	0.95

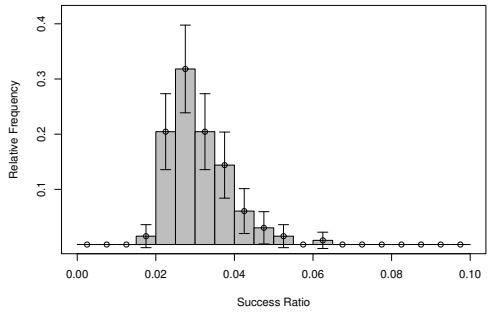

Figure 7: Success ratio of injection attacks.

cannot be modeled using traditional free-space propagation. A majority of research efforts conducted in this area apply Rayleigh fading [10] which models time-variant multipath propagation introduced by, e.g., large buildings during outdoor mobility. These assumptions to not hold in the considered scenario where sensors are statically installed within an enclosed space such that randomly occurring deep fades would not be justified (using Rayleigh fading would help to even further increase the impact of the multipath propagation which could result in too optimistic results). That is why we decided to apply the ground reflection model [10] where the signal comprises two rays. The combination of both rays leads to attenuation and amplification effects which is directly related to the distance and the transmission frequency. To enable the shadowing property of this propagation model we also added stohastic means (more details on this issue can be found in [7]).

5.4 Evaluation Results

The results and discussion focus on the maximum success rate a virtual attacker can achieve, the number of sensors able to overhear malicious transmissions and how many of them eventually take countermeasures. This is concluded by remarks on their distribution and the feasibility of directed battery depletion attacks.

For the course of this evaluation we have chosen the coefficients under the goal to defy at least 5% of all impersonated messages that are transmitted from an individual position. The actual amount is defined as the arithmetic mean across all frequencies considering the highest achievable success ratio for each one, because the network periodically switches channels whereby selectively high penetrability is subsided. The results obtained for 132 randomly chosen positions in close range to the attacked network depicted in Figure 7 exhibit a certain range of possible success ratios starting

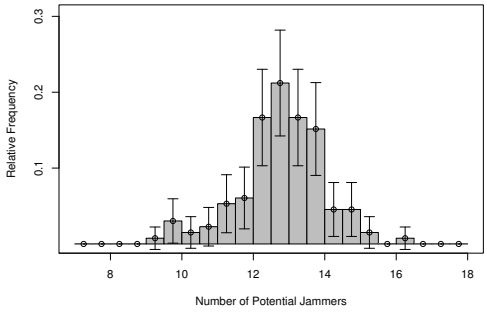

Figure 8: Number of potential jammers, i.e., nodes that detect fake frames before scheduling a JAM frame.

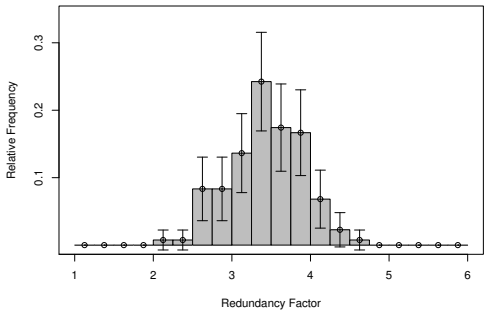

Figure 9: Number of concurrent (active) jammers pro fake frame.

from as little as 1.88% with a mean of 3.1%. We now take a closer look at what is happening within the network itself and investigate the individual states while an attack is carried out. A high level of protection begins with the identification of an impersonation attack which ought to be achieved by a desirably large network partition. The average number of sensors overhearing a malicious transmission for all considered attack scenarios is presented in Figure 8. While a minority of them might not be able to detect the attack because the yielded RSS is contained in the corresponding acceptance interval, there is a vast number of available jammers, ranging from a minimum of 9.5 up to a total of 16.3. This should be considered as an advantage. Instead of allowing each single sensor to take immediate counteraction which would lead to redundancy figures similar to the above numbers, we restrict the waste of energy by minimizing the set of actual jammers in a dynamic and unpredictable manner.

The resulting redundancy values in correspondence to the graph before are depicted in Figure 9. We are able to reduce the previous two-digit redundancy to a value as low as 4.7 in the worst case and even below 3.5 in the average case while the remaining sensors are waiting for their turn.

Restricting the bare number of actual jammers is not sufficient yet, because if jamming is concentrated on only a small subset then some parts of the network will suffer from higher battery depletion than others, leaving the according segment more vulnerable. In Figure 10 we can see that the individual efforts do not vary greatly on a global scale. For each attack configuration we compute the difference between the individual jam ratio and the total of jam

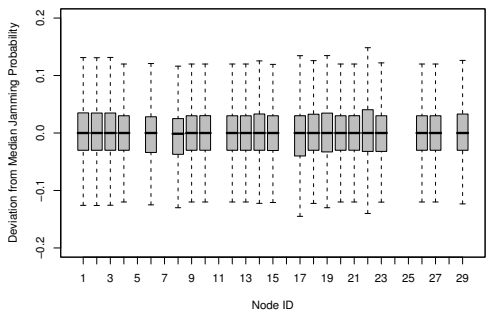

Figure 10: Empirical distribution of jamming activity.

frames sent for a single impersonated message. The depicted boxplot for each sensor derived from the aggregated results reveals a mean value close to 0 indicating very low deviation. While the use of MACs in a cryptographic context can be exploited by directing a large number of messages to single sensor, the proposed approach does not allow for targeted battery depletion attacks.

6. CONCLUSION

Properties of wireless communications evidentially extend the attacker's toolset and provide new attack vectors against all security objectives. The question that arises is: why should legitimate wireless devices abstain from such properties which may be used to strengthen security of wireless networks. Rather than spending battery power to receive, verify, and then discard the data, a wireless device can take advantage of jamming to provide new authentication mechanisms. Moreover, using jamming with other properties of wireless communications such as the unpredictable nature of radio propagation, this work demonstrated that without relying on any secrets, impersonation attacks can be easily detected and avoided.

As a part of our future work we shall further evaluate costs between the jamming approach and a conventional cryptographic authentication. Although jamming transmission is more expensive than computing cryptographic digests, there are two main advantages that can amortize costs of jamming. First, with a single JAM frame, more wireless sensors can be prevented from receiving fake data (moreover, a correctly received fake frame usually has an additional transmission cost for responding with a link-layer ACK, which is, in case of attack cancelation, eliminated). Secondly, jamming task is only activated during an attack which is in contrast to conventional "always-on" cryptographic countermeasures. We also plan to extend our experimental evaluation to more dynamic environments which may result in an increased number of false positives during the authentication procedure.

7. REFERENCES

[1] B. A.-Sadjadi, A. Kiayias, A. Mercado, and B. Yener. Robust key generation from signal envelopes in wireless networks. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 401–410, New York, NY, USA, November 2007. ACM.

[2] A. Bachorek, I. Martinovic, and J. B. Schmitt. Enabling Authentic Transmissions in WSNs – Turning Jamming against the Attacker. In *Proceedings of the IEEE ICNP 4th Annual Workshop on Secure Network Protocols (NPSec 2008)*, Orlando, FL, USA, October 2008.

[3] Y. Chen, W. Trappe, and R. Martin. Detecting and localizing wireless spoofing attacks. In *Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, pages 193–202, May 2007.

[4] D. B. Faria and D. R. Cheriton. Detecting Identity-based Attacks in Wireless Networks using Signalprints. In *WiSe '06: Proceedings of the 5th ACM Workshop on Wireless Security*, pages 43–52, September 2006.

[5] Z. Li, W. Xu, R. Miller, and W. Trappe. Securing Wireless Systems Via Lower Layer Enforcements. In *WiSe '06: Proceedings of the 5th ACM Workshop on Wireless Security*, pages 33–42, September 2006.

[6] I. Martinovic, L. Cappellaro, and J. B. Schmitt. Chaotic Communication Improves Authentication – Protecting WSNs Against Injection Attacks. *Security and Communication Networks Journal (Special Issue on Wireless Sensor Networks)*. Wiley. In press.

[7] I. Martinovic, P. Pichota, and J. B. Schmitt. Jamming for Good: A Fresh Approach to Authentic Communication in WSNs. Technical Report, University of Kaiserslautern, Germany, July 2008.

[8] I. Martinovic, F. Zdarsky, M. Wilhelm, C. Wegmann, and J. B. Schmitt. Wireless Client Puzzles in IEEE 802.11 Networks: Security by Wireless. In *Proc. ACM Conference on Wireless Network Security (WiSec 2008)*, pages 43–52, Alexandria, VA, USA, March 2008.

[9] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-Telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel. In *The 14th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM Press, September 2008.

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

[11] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmissions in wireless sensor networks. In *4th Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[12] L. Song and A. Arora. Spatial Signatures for Lightweight Security in Wireless Sensor Networks. In *27th IEEE International Conference on Computer Communications (INFOCOM 2008)*, Phoenix, AZ, USA, April 2008.

[13] Texas Instruments. *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, mar 2007.

[14] M. Čagalj, S. Čapkun, R. Rengaswamy, I. Tsigkogiannis, M. Srivastava, and J.-P. Hubaux. Integrity (I) Codes: Message Integrity Protection and Authentication Over Insecure Channels. In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 280–294, May 2006.

[15] S. Čapkun, R. Rengaswamy, I. Tsigkogiannis, and M. Srivastava. Implications of Radio Fingerprinting on the Security of Sensor Networks. In *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks (SecureComm07)*, pages 331–340 September 2007.