6LoWPAN Security: Avoiding Hidden Wormholes using Channel Reciprocity

Konrad-Felix Krentz
konrad-felix.krentz@hhi.fraunhofer.de

Gerhard Wunder
gerhard.wunder@hhi.fraunhofer.de

Fraunhofer Heinrich-Hertz-Institut, Einsteinufer 37, 10587 Berlin, Germany

ABSTRACT

6LoWPAN networks are 802.15.4-based wireless sensor networks that seamlessly integrate with IPv6 networks using specially designed protocols. Unfortunately, 6LoWPAN networks are vulnerable to hidden wormholes. To detect and subsequently avoid hidden wormholes, Jain et al. recently proposed two schemes, which exploit the reciprocity of received signal strength indicators (RSSIs). However, threats and attacks can mislead Jain et al.'s schemes into reaching false positives or false negatives. Moreover, their schemes take calibrated RSSIs for granted, which is impractical. In this paper, we propose "Secure Channel REciprocity-based WormholE Detection (SCREWED)", which avoids both false positives and false negatives to a great extend. For this, SCREWED uses channel hopping, randomized transmission powers, message integrity codes, as well as a special replay protection mechanism. Furthermore, SCREWED obviates the need for calibrating RSSIs by using a different channel reciprocity metric. We integrated SCREWED into the link layer of Contiki's 6LoWPAN stack and demonstrate SCREWED's efficacy using TelosB motes.

Categories and Subject Descriptors

C.2.0. [**General**]: Security and protection; C.2.1. [**Network Architecture and Design**]: Wireless communication

Keywords

Wormhole attack, physical layer security, link layer security.

1. INTRODUCTION

The IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) protocol stack is designed for diverse applications, such as smart cities, industrial automation, and precision agriculture [24]. Essentially, the 6LoWPAN protocol stack serves for integrating 802.15.4[2] networks seamlessly with IPv6 networks. Its organization follows the Open Systems Interconnection (OSI) model. At the physical and link layer, 802.15.4 delivers frames to neighboring nodes. In between the link and network layer, the 6LoWPAN adaption layer [16] fits IPv6 packets within frames by fragmenting and compressing them. At the network layer, the Routing Protocol for Low-Power and Lossy Networks (RPL)[44] routes IPv6 packets within 6LoWPAN networks. On top, User Datagram Protocol (UDP)-based application layer protocols, such as the Constrained Application Protocol (CoAP)[35], complete the 6LoWPAN protocol stack.

Unfortunately, the 6LoWPAN protocol stack is vulnerable to both hidden and exposed wormholes [7]. Hidden wormholes are out-of-band links that relay traffic between non-neighboring nodes verbatim. Thus, hidden wormholes trick non-neighboring nodes into believing they were neighbors. An attacker can set up a hidden wormhole, e.g., by (i) placing two transceivers in distant parts of the victim 6LoWPAN network and (ii) using these two transceivers to relay the traffic between these network parts. Exposed wormholes, on the other hand, are out-of-band links between adversarial nodes that do not take any effort to mask their link layer addresses. In contrast to hidden wormholes, exposed wormholes do not violate neighborhood relationships between other than adversarial nodes. For the scope of this paper, we only consider hidden wormholes (hereafter just referred to as wormholes).

Wormholes come in two disruptive incarnations. Selective wormholes, on the one hand, selectively drop single frames or entire IPv6 packets. For a surveillance monitoring system this, e.g., means that alarm messages may not arrive. Selective wormholes also incur problems in the context of intrusion detection. Selective wormholes can, e.g., drop RPL messages[33] or frames for completing reassembly [17]. Consequently, an intrusion detection system may wrongly conclude that a presumed neighbor was compromised. Moreover, selective wormholes are very effective because wormholes in general attract much traffic due to the shortcuts they provide. To counter selective wormholes, frames could be encrypted using 802.15.4 security, leaving no hint on a frame's payload. However, with traffic analysis, selective wormholes will still able to launch purposive attacks. Transient wormholes, on the other hand, are short-lived wormholes, which can, e.g., be used to mount rushing attacks on RPL [15]. For example, by rushing RPL messages for building upward routes through a transient wormhole, an attacker can provoke a reorganization of upward routes. As the transient wormhole goes down, the upward routes no longer work and RPL will repair them with a hysteresis [11]. This hysteresis causes a temporary outage and the repair is energy consuming. Moreover, a transient wormhole can pop up over and over again.

To detect and subsequently avoid wormholes, Jain et al.[19] recently proposed two schemes. Both their schemes leverage received signal strength indicators (RSSIs) provided by off-the-shelf 802.15.4 transceivers. Specifically, the property being leveraged is that when two nodes send frames to each other within a short period of time, those frames will be received with similar, recipro-

TrustED'14, November 3, 2014, Scottsdale, Arizona, USA.

http://dx.doi.org/10.1145/2666141.2666143.

cal RSSIs. By contrast, when two nodes communicate through a wormhole, they will observe uncorrelated RSSI trajectories. This can consequently be used to detect wormholes. However, threats and attacks can mislead Jain et al.'s schemes into reaching false positives or false negatives. A false positive is the event that a wormhole is detected although there is none. A false negative is the event that a wormhole remains undetected. Moreover, their schemes take calibrated RSSIs for granted, which is impractical.

We make two main contributions:

- We propose Secure Channel REciprocity-based WormholE Detection (SCREWED), which avoids both false positives and false negatives to a great extend. For this, SCREWED uses channel hopping, randomized transmission powers, message integrity codes (MICs), as well as a special replay protection mechanism. Furthermore, SCREWED obviates the need for calibrating RSSIs by using a different channel reciprocity metric.

- We integrated SCREWED into the link layer of Contiki's 6LoWPAN stack. There, SCREWED transparently secures all upper-layer protocols, such as the 6LoWPAN adaption layer and RPL, against wormholes.

2. BACKGROUND

In this section, we (i) introduce channel reciprocity, (ii) discuss Jain et al.'s schemes, and (iii) point out threats to and attacks on Jain et al.'s schemes.

2.1 Channel Reciprocity

Channel reciprocity-based wormhole detection is based on three properties of wireless channels [26]:

Reciprocity: When two transceivers transmit identical signals simultaneously to each other, those signals will be received with identical channel characteristics.

Fading: Owing to environmental changes, such as, moving people, moving objects, or movements of the transceivers themselves, propagation paths vary over time. This is known as fading and causes channel characteristics to vary over time.

Spatial decorrelation: An eavesdropper that is more than half a wavelength away from a legitimate receiver, observes different, uncorrelated channel characteristics. This is because the eavesdropper is subject to different fading effects. For example, when using the 802.15.4 channels at 2.4GHz, a distance of 6.25cm suffices.

Accordingly, the RSSIs over the preambles of an 802.15.4 frame and its corresponding acknowledgement frame should be similar. Indeed, previous experiments[19, 28, 43, 45] show that such RSSI pairs have only small discrepancies. These discrepancies have physical and technical sources. Physical sources are the delay in between the two frames and interference, which both deteriorate the reciprocity of such RSSI pairs. Technical sources are, e.g., the following two ones [27]. First, owing to manufacturing variations, each transceiver has a specific offset, which needs to be calibrated and added to its RSSI readings. Second, when both communication partners use different transmission powers, this induces an offset, too. This is especially relevant to a battery-powered node since its transmission power decreases with its battery voltage over time.

2.2 Reactive vs. Proactive vs. Both

The crucial observation made by Jain et al.[19] is that a wormhole in between two nodes breaks channel reciprocity. Based on this observation, Jain et al. proposed a reactive and a proactive wormhole detection scheme. Their reactive scheme detects wormholes at runtime by leveraging frames that are sent by upper-layer protocols anyway. Each such frame and its corresponding acknowledgement frame yield an RSSI pair. After collecting N such RSSI pairs while communicating with a presumed neighbor, the reactive scheme judges whether to keep or drop the presumed neighbor based on a channel reciprocity metric. Jain et al. suggested two channel reciprocity metrics. One of them compares middle-order bits of RSSI pairs and the other one uses an L-bit quantizer [3]. Their proactive scheme, on the other hand, avoids communications through wormholes in the first place. For this, the proactive scheme initiates its own frame exchanges right after discovering a presumed neighbor. After collecting N RSSI pairs, the proactive scheme judges whether to keep or drop the presumed neighbor like the reactive scheme.

It depends on the use case if either the reactive, the proactive scheme, or even both should be used. The advantage of the reactive scheme is that it incurs less additional traffic. However, the reactive scheme may not have enough time to detect transient wormholes. The proactive scheme also detects transient wormholes, but incurs additional traffic and prolongs neighbor discovery. It may even be necessary to repeat wormhole detection periodically. This is because, wormholes may be set up after running the reactive or proactive scheme. For example, when two nodes move apart from each other, an attacker can keep up the link in between them by setting up a wormhole.

2.3 False Positives and False Negatives

Unfortunately, Jain et al.'s schemes are subject to threats and attacks that can lead to false positives and false negatives.

2.3.1 Frame Spoofing

Recall that in both the reactive and the proactive scheme a neighbor is dropped if the channel reciprocity metric is too low. If this decision is based on unauthenticated frames, an attacker can cause a false positive by spoofing frames used for wormhole detection. This is because spoofed frames are sent from different positions and hence deteriorate the employed channel reciprocity metric. In effect, a true neighbor may get dropped.

Therefore, Jain et al. proposed to authenticate each frame. 802.15.4 frames can, e.g., be authenticated using 802.15.4 security, which appends a MIC to each frame. 802.15.4 security does, however, not support authenticated acknowledgement frames [34]. Thus, modifications to 802.15.4 security are needed when implementing the reactive scheme. Another problem is the incurred delay for verifying and generating MICs. This is because delays deteriorate the reciprocity of RSSI pairs. Section 3.2 covers how SCREWED solves these problems. Furthermore, Section 5.2.1 discusses how SCREWED prevents a related attack: replaying frames.

2.3.2 Low Variation

When there is low variation in RSSIs, Jain et al.'s schemes may fail to detect a wormhole. Take for example the channel reciprocity metric based on middle-order bits of RSSIs. If a wormhole has, by chance, no influence on middle order bits of RSSIs and if these bits do not vary due to low variation, the wormhole remains undetected. Similarly, when using an L-bit quantizer, its quantization levels may be chosen unluckily and hence generate similar bit se-

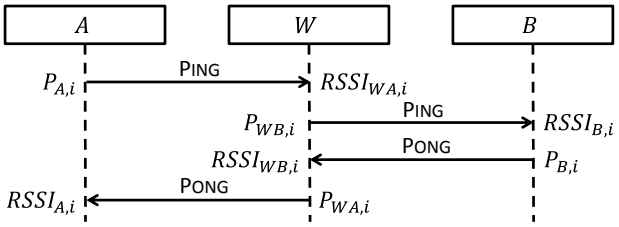

Figure 1: Exchange of the i-th "PING" and "PONG" frame during reactive or proactive wormhole detection between A and B. The adaptive wormhole W tries to remain stealth by adapting its transmission powers $P_{WB,i}$ and $P_{WA,i}$ of the i-th PING and PONG, respectively. Both RSSIs and transmission powers are in dBm.

quences despite a wormhole being in between the presumed neighbors.

Low variation in RSSIs is a real issue in static settings [20]. Therefore, Jain et al. suggested to use other channel characteristics than RSSIs in static settings, such as phase. However, this solution requires special hardware. SCREWED avoids special hardware by using channel hopping, as well as randomizing transmission powers instead.

2.3.3 Adaptive Wormholes

While the above attacks were already recognized by Jain et al., the following attack is presumably unknown at present. We call this attack an *adaptive wormhole*. Consider the situation in Figure 1 and assume $P_{A,i} = P_{B,i}$. To achieve $RSSI_{A,i} - RSSI_{B,i} \approx 0$, the adaptive wormhole W forwards the i-th PONG frame with transmission power $P_{WA,i}$ chosen as:

$$P_{WA,i} = P_{WB,i} + RSSI_{WB,i} - RSSI_{WA,i} \quad (1)$$

This choice is motivated as follows. First, observe that there is a simple relationship between a transmission power P and the respective received power $RSSI$. When $RSSI$ and P are in dBm, it holds that:

$$RSSI = P - L \quad (2)$$

where L is the current loss on the channel in dBm.

Thus:

$$RSSI_{A,i} - RSSI_{B,i} = P_{WA,i} - P_{WB,i} - L_{WA,i} + L_{WB,i} \quad (3)$$

$$RSSI_{WA,i} - RSSI_{WB,i} = P_{A,i} - P_{B,i} - L_{AW,i} + L_{BW,i} \quad (4)$$

where $L_{AW,i}$, $L_{WA,i}$, $L_{WB,i}$, and $L_{BW,i}$ are the losses on the channels $A \to W$, $W \to A$, $W \to B$, and $B \to W$, respectively.

Owing to channel reciprocity, $L_{AW,i} \approx L_{WA,i}$ and $L_{WB,i} \approx L_{BW,i}$ and hence:

$$\begin{aligned} RSSI_{A,i} - RSSI_{B,i} \approx & RSSI_{WA,i} + P_{WA,i} - P_{A,i} \\ & - RSSI_{WB,i} - P_{WB,i} + P_{B,i} \end{aligned} \quad (5)$$

Plugging Equation (1) into Equation (5) and assuming $P_{A,i} = P_{B,i}$ yields $RSSI_{A,i} - RSSI_{B,i} \approx 0$.

Essentially, adaptive wormholes even out discrepancies between $RSSI_{A,i}$ and $RSSI_{B,i}$ for all $i \in \{1, \ldots, N\}$. Thus, when communicating through an adaptive wormholes, Jain et al.'s channel reciprocity metrics show high values, which results in false negatives. Since SCREWED cannot detect adaptive wormholes either, detecting adaptive wormholes is left for future work.

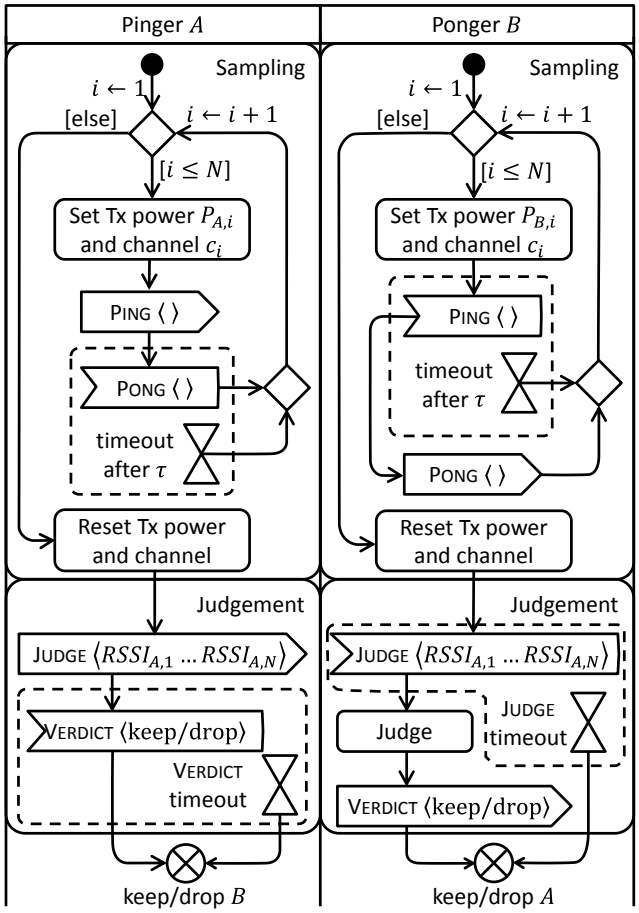

Figure 2: SCREWED

3. SCREWED: SECURE CHANNEL RECIPROCITY-BASED WORMHOLE DETECTION

SCREWED can be run reactively or proactively. When run reactively, SCREWED removes current neighbors that are reached through a wormhole. When run proactively, SCREWED jumps in right after neighbor discovery and immediately filters out neighbors that are reached through a wormhole. However, unlike Jain et al.'s schemes, the reactive and the proactive version of SCREWED is identical. Hence, each node needs only one implementation of SCREWED. For achieving this advantage, neither version of SCREWED leverages upper-layer traffic. Instead, SCREWED always sends designated PING and PONG frames for collecting RSSI pairs. Below, we first outline and then detail SCREWED.

3.1 Protocol Overview

SCREWED operates in two phases, namely sampling and judgement, as shown in Figure 2.

3.1.1 Sampling

In the sampling phase, a node A sends N PINGs to a presumed neighbor B. When B receives a PING, B immediately replies a PONG. Upon receipt of a PONG, A immediately sends the next PING and so on. To avoid a deadlock when a PING or PONG gets lost, SCREWED uses timeouts. A only waits for time τ for a PONG and thereafter sends the next PING. Likewise, B only waits

for time τ for the next PING. The RSSIs of PINGs (denoted by $RSSI_{B,1},\ldots,RSSI_{B,N}$) and PONGs (denoted by $RSSI_{A,1},\ldots,RSSI_{A,N}$) are stored by the ponger B and the pinger A, respectively for use in the judgement phase. RSSIs of missed PINGs and PONGs are designated by a special value denoted by ϵ.

The i-th PING and the i-th PONG are sent with the random transmission powers $P_{A,i}$ and $P_{B,i}$, respectively. $P_{A,1},\ldots,P_{A,N}$ and $P_{B,1},\ldots,P_{B,N}$ are agreed upon before running SCREWED. They are drawn such that $P_{A,i}-P_{B,i}$ is uniformly distributed. Note that SCREWED does not require A and B to exactly send with these transmission powers, but only on a best-effort basis. SCREWED's judgement phase accounts for battery-powered nodes, as well as inaccuracies.

In addition, each PING-PONG is sent on a different channel c_i. Channel hopping is a means to generate variance in RSSIs in static settings [43, 45]. The generated variation stems from different fading effects at different frequencies. 802.15.4 has 16 channels available for channel hopping in the 2.4GHz band. These are indexed 11 through 26. However, only some pairs of 802.15.4 channels, such as 11 and 26, are uncorrelated. Nearby 802.15.4 channels, on the other hand, have similar propagation characteristics and hence similar RSSIs. To work around this problem, decorrelation in time can be exploited in addition. That is, when sampling nearby channels only after some time, there is a better chance of environmental changes occurring. Thus, an ideal channel hopping pattern should sample distant channels consecutively and nearby channels only after some time. For 802.15.4, Yao et al.[45] have found the following channel hopping pattern to be ideal in this respect:

$$c_i = ((c_{i-1} - 11 + 7) \mod 16) + 11 \tag{6}$$

which produces the sequence $\ldots, 26, 17, 24, 15, 22, 13, \ldots$. Hence, SCREWED adopts this channel hopping pattern.

3.1.2 Judgement

After sampling, the pinger A sends a JUDGE message to B containing $RSSI_{A,1},\ldots,RSSI_{A,N}$. Upon receipt of this JUDGE message, B's task is to decide if the pinger A shall be kept as a neighbor in a four-step process. First, if the number of received PONGs N_{rec} is below some threshold N_{min}, i.e. if $N_{rec} < N_{min}$, A is dropped right away. Second, B discards the $N_{\text{rec}} - N_{\text{min}}$ $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs with the highest discrepancies. Discarding a fraction of pairs is necessary to counter certain replay attacks, as will be detailed in Section 5.2.1. Third, B calculates the sample correlation coefficient $r(P_A - P_B, RSSI_B - RSSI_A)$ over the retained N_{min} $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs. Forth, B suspects a wormhole and drops B iff $r(P_A - P_B, RSSI_B - RSSI_A) < \rho$, where ρ is a configurable parameter.

The sample correlation coefficient, or correlation for short, normalizes the sample covariance to a value between -1 an 1. $r(P_A - P_B, RSSI_B - RSSI_A)$ is calculated as:

$$\frac{s(P_A - P_B, RSSI_B - RSSI_A)}{s(P_A - P_B)s(RSSI_B - RSSI_A)} \tag{7}$$

Thereby, $s(P_A - P_B, RSSI_B - RSSI_A)$ is the sample covariance between $P_A - P_B$ and $RSSI_B - RSSI_A$, and $s(P_A - P_B)$ and $s(RSSI_B - RSSI_A)$ are the sample standard deviations of $P_A - P_B$ and $RSSI_B - RSSI_A$, respectively.

The idea behind calculating the correlation between $P_A - P_B$ and $RSSI_B - RSSI_A$ is as follows. Suppose there is no wormhole between A and B. Then:

$$RSSI_{A,i} = P_{B,i} - L_{BA,i} \tag{8}$$

$$RSSI_{B,i} = P_{A,i} - L_{AB,i} \tag{9}$$

Channel reciprocity implies $L_{BA,i} \approx L_{AB,i}$ and hence:

$$RSSI_{A,i} - RSSI_{B,i} \approx P_{B,i} - P_{A,i} \tag{10}$$

That is, the more reciprocal the channel between A and B is, the higher is $r(P_A - P_B, RSSI_B - RSSI_A)$. Furthermore, since correlation is offset and scaling agnostic, calibrating RSSIs is unnecessary. For the same reason, it is unnecessary to account for the decreasing battery voltage on battery-powered nodes.

Choosing $r(RSSI_A + P_A, RSSI_B + P_B)$ as a channel reciprocity metric is also possible, but would complicate the configuration of the parameters ρ, N, and N_{min}. This is because discrepancies affect $r(RSSI_A + P_A, RSSI_B + P_B)$ sometimes more and sometimes less, depending on how strong the variation of the RSSIs on the different channels is. The same holds for $r(RSSI_A, RSSI_B)$, which is the canonical choice when not randomizing transmission powers.

Eventually, B sends a VERDICT message to A containing its decision. If a JUDGE or VERDICT message does not arrive after a timeout, the pinger or ponger is dropped, respectively.

3.2 Protocol Specification

PINGs and PONGs are sent as 802.15.4 command frames with currently reserved command frame identifiers. However, to reduce the round trip time (RTT) of PING-PONGs, PING and PONG command frames have two particularities. First, they do not ask for 802.15.4 acknowledgement frames. Second, PINGs and PONGs do not use 802.15.4 security. Thus, they neither set the security enabled flag nor do they carry an auxiliary security header. Instead, PINGs and PONGs are secured by interfacing with CCM* directly. CCM* is the Counter with Cipher Block Chaining-MIC (CCM)[42] version of 802.15.4 security.

Specifically, securing PINGs and PONGs works as follows. SCREWED appends a CCM*-MIC to each of them. These CCM*-MICs are generated by passing CCM* a nonce that deviates from nonces of 802.15.4 security. The nonce's frame counter field is set to $f_A + i$, where f_A is the frame counter of A when starting SCREWED. The frame counters $f_A + 1,\ldots,f_A + N$ are reserved for use by SCREWED. Thus, both SCREWED and 802.15.4 security can use the same key, without risking a nonce reuse. In addition, the security level field of PONG nonces is modified to differentiate PING nonces from PONG nonces. Furthermore, SCREWED does not pass any data to authenticate or encrypt to CCM*. Only i is authenticated since it is included in the nonce. Altogether, this way of generating CCM*-MICs enables a ponger to precompute the CCM*-MICs of forthcoming PINGs. Thus, verifying a PING comes down to comparing its CCM*-MIC with the precomputed one. Likewise, pongers can also precompute CCM*-MICs of PONGs for sending PONGs faster.

Also JUDGE and VERDICT messages are sent as 802.15.4 command frames. They are, however, secured by using 802.15.4 security "as is" since their RTT is not crucial.

4. IMPLEMENTATION

We integrated SCREWED into the 6LoWPAN stack of the Contiki[1] operating system. Specifically, we integrated SCREWED into the Adaptable Pairwise Key Establishment Scheme (APKES) [25].

[1] `http://contiki-os.org`

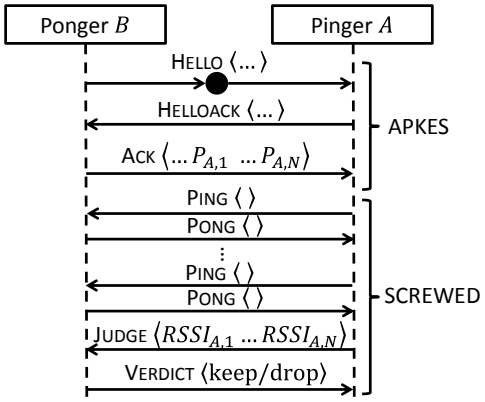

Figure 3: Integration of SCREWED into APKES

APKES is a unified protocol framework for establishing pairwise 802.15.4 keys using key predistribution schemes. Basically, APKES performs a three-way message exchange to discover neighboring nodes and establish pairwise 802.15.4 keys with them. However, APKES also establishes pairwise keys with neighbors that are reached through a wormhole. Therefore, we run SCREWED right after APKES so as to proactively filter out false neighbors.

Our test targets were TelosB motes [30]. TelosB motes are equipped with a CC2420 802.15.4 transceiver, which supports transmission powers between 0 and -25dBm [1]. Despite this, our implementation only uses transmission powers between 0 and -7dBm so as to ensure that most PINGs and PONGs arrive.

More precisely, the drawing of transmission powers is implemented as a two-step process, which is performed by the ponger right before running SCREWED. First, the ponger draws δ_i from the distribution in Table 2 of Appendix B. Second, the ponger draws pairs $(P_{A,i}, P_{B,i})$ with $\delta_i = P_{A,i} - P_{B,i}$ from the respective distribution in Table 3. That matrix balances among the different transmission powers. After drawing N $(P_{A,i}, P_{B,i})$ pairs this way, the transmission powers $P_{B,1}, \ldots, P_{B,N}$ are piggybacked on the final message of APKES, as shown in Figure 3.

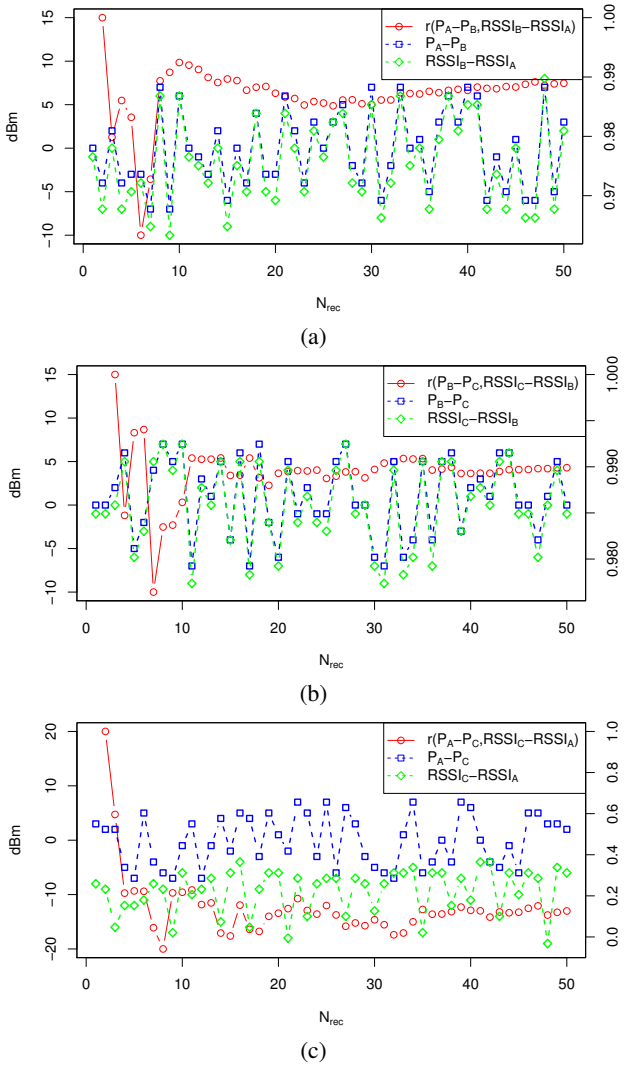

Figure 4: Correlation when running SCREWED between (a) A and B, (a) B and C, and (b) A and C. In (c), B acts as a wormhole.

5. EVALUATION

Using our implementation, we evaluated various aspects of SCREWED. Section 5.1 first demonstrates SCREWED's capability to detect wormholes. Section 5.2 evaluates SCREWED's resilience to threats and attacks. Finally, Section 5.3 benchmarks our implementation. Throughout, we give hints on choosing SCREWED's parameters, namely ρ, N, N_{min}, and τ.

5.1 Wormhole Detection Capability

For demonstrating SCREWED's wormhole detection capability we placed three TelosB motes A, B, C on different floors of a house. The positions were chosen such that B can communicate with both A and C, whereas A and C cannot reach each other. Figure 4a shows a run of SCREWED between A and B. The red dots plot the correlations between $P_A - P_B$ and $RSSI_B - RSSI_A$ over the first N_{rec} completed PING-PONGs. No $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs were discarded before calculating the correlation ($N_{min} = N_{rec}$). The correlations converge to 0.989. Similarly, the correlations between $P_B - P_C$ and $RSSI_C - RSSI_B$ are very high, as shown in Figure 4b. Conversely, when B acts as a hidden wormhole by replaying all frames verbatim, the correlations between $P_A - P_C$ and $RSSI_C - RSSI_A$ are very low, as shown in Figure 4c. Thereby, B imitated the channel hopping of A and C and kept its transmission power at 0dBm.

We reran this setup 10 times. In the adversarial case, the correlations stayed in between -0.41 and 0.62 after $N_{min} = N_{rec} \geq 8$. Of course, when discarding the $N_{rec} - N_{min}$ $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs with the highest discrepancies, higher correlations occur. Table 1 lists the ranges when $N = N_{rec} \geq 13$ for practically relevant fractions. In the nonadversarial case, the correlation always stayed above 0.94 if $N_{min} = N_{rec} \geq 8$. If $N_{min} = N_{rec} < 8$, we observed three times that early outliers dragged the correlation below 0.94.

To ensure that, in the nonadversarial case, correlations are always ≥ 0.94 under the condition that $N_{min} = N_{rec} \geq 8$, we ran SCREWED indoors and outdoors, as well as with and without batteries. The results confirm this condition when $\forall i \leq N : RSSI_{A,i} \leq -60 \wedge RSSI_{B,i} \leq -60$. As RSSIs become greater than -60dBm, which is the case when our TelosB motes are close by, correlations around 0.85 occur. This is caused by saturation (see [27]). Figuring out the exact threshold, as well as devising a workaround, e.g. by lowering transmission powers, is beyond the scope of this paper.

Table 1: Ranges of $r(P_A - P_B, RSSI_B - RSSI_A)$ when running SCREWED through a nonadaptive wormhole and $N = N_{rec} \geq 13$

N_{min}	$r(P_A - P_B, RSSI_B - RSSI_A)$	
	min	max
$\lfloor 0.6N_{rec} \rfloor$	-0.59	0.85
$\lfloor 0.625N_{rec} \rfloor$	-0.59	0.79
$\lfloor 0.65N_{rec} \rfloor$	-0.59	0.78
$\lfloor 0.675N_{rec} \rfloor$	-0.59	0.78
$\lfloor 0.7N_{rec} \rfloor$	-0.49	0.73
$\lfloor 0.725N_{rec} \rfloor$	-0.49	0.73
$\lfloor 0.75N_{rec} \rfloor$	-0.39	0.70
$\lfloor 0.775N_{rec} \rfloor$	-0.39	0.70
$\lfloor 0.8N_{rec} \rfloor$	-0.39	0.69

5.2 False Positives and False Negatives

Now, we evaluate SCREWED's resilience to frame spoofing, low variation, as well as adaptive wormholes.

5.2.1 *Frame Spoofing*

If spoofed or replayed PINGs or PONGs were accepted, this would deteriorate correlation and may cause a false positive. However, SCREWED filters out spoofed PINGs and PONGs by means of CCM*-MICs. Also, SCREWED filters out replayed PINGs and PONGs if they arrived at their intended recipient before. This is because the ponger and the pinger immediately increment i upon receipt of a PING or PONG, respectively. This changes the expected CCM*-MIC of the subsequent PING or PONG and hence replayed PINGs and PONGs are rejected.

However, replayed PINGs and PONGs that did not arrive at their intended recipient do get accepted since i was not incremented by the intended recipient, yet. Thus, an attacker can still replay every PING and PONG and hope that the original one did not arrive. SCREWED prevents this attack by discarding the $N_{\text{rec}} - N_{\text{min}}$ $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs with the highest discrepancies. Hence, N_{min} should be chosen such that $\frac{N_{min}}{N}$ is the fraction of PING-PONGs that always arrive.

During our experiments, we found that $\frac{N_{min}}{N} = 0.68$ is the minimum. Taking into account the results from Section 5.1, neither false positive nor false negatives would have occurred in our experiments if we had set $\frac{N_{min}}{N} = \frac{8}{13} = 0.68$ and $\rho = 0.94$. However, since our sample size is small, our implementation defaults to $\frac{N_{min}}{N} = \frac{10}{16} = 0.625$ and $\rho = 0.93$, which is safer.

Spoofed or replayed JUDGE and VERDICT command frames are filtered out by 802.15.4 security. (Actually, replayed JUDGE and VERDICT command frames that did not already arrive get accepted, but without any repercussions.)

5.2.2 *Low Variation*

SCREWED's main tool to generate variation in RSSIs is channel hopping. To isolate its effect, we placed two TelosB motes on different floors of a house and remotely started SCREWED so that no persons cause variation. Furthermore, we disabled the randomization of transmission powers. Figure 5a shows 50 RSSI pairs in this setting when channel hopping is off. There is almost no variation. By contrast, when enabling channel hopping, variation is generated even in this setting, as shown in Figure 5b. The resulting RSSI patterns do, however, repeat since there is no variation over time in this setting. Thus, low variation is still an issue in static settings.

However, for nonadaptive wormholes to remain undetected, $RSSI_{WA,i}$ should be similar to $RSSI_{WB,i}$ for all i. To see this,

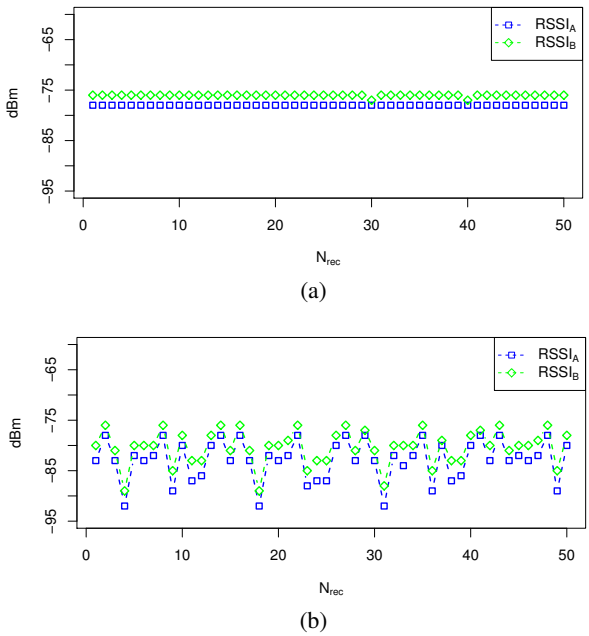

Figure 5: Variation of 50 RSSI pairs in a static indoor setting (a) when neither randomizing transmission powers nor hopping channels (b) when not randomizing transmission powers, but only hopping channels

assume $RSSI_{WA,i} \approx RSSI_{WB,i}$ and rewrite Equation (5) as:

$$P_{A,i} - P_{B,i} \approx RSSI_{B,i} - RSSI_{A,i} + P_{WA,i} - P_{WB,i} \quad (11)$$

Since $P_{WA,i} - P_{WB,i}$ is static for nonadaptive wormholes, SCREWED's channel reciprocity metric would yield a high value.

Fortunately, $RSSI_{WA,i} - RSSI_{WB,i} \approx 0$ for all i is very unlikely since:

$$RSSI_{WA,i} - RSSI_{WB,i} \approx P_{A,i} - P_{B,i} - L_{AW,i} + L_{BW,i} \quad (12)$$

where $P_{A,i} - P_{B,i}$ is uniformly distributed, and $L_{AW,i}$ and $L_{BW,i}$ are independent random variables. Therefore, SCREWED is virtually resistant to low variation.

5.2.3 *Adaptive Wormholes*

Plugging the strategy in Equation (1) into Equation (5) yields:

$$RSSI_{A,i} - RSSI_{B,i} \approx -P_{A,i} + P_{B,i} \quad (13)$$

Hence, $r(P_A - P_B, RSSI_B - RSSI_A)$ will be close to 1 when running SCREWED through an adaptive wormhole. Therefore, SCREWED does not detect adaptive wormholes.

5.3 Benchmarks

Below, we benchmark the RTTs and memory efficiency of our implementation.

5.3.1 *Round Trip Times*

The RTT of PING-PONGs is crucial for achieving highly reciprocal $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs. Figure 6 compares the mean RTTs of different options for implementing PING-PONGs. The RTTs were measured on the pinger side by taking time T_1 right before sending a PING and time T_2 within the interrupt routine that is called upon receipt of the corresponding PONG. Figure 6 shows the mean of $T_2 - T_1$ over 90 samples per implementation option.

The fastest RTTs were achieved with hardware acknowledgement frames, which is a standard feature of 802.15.4 transceivers. Using software acknowledgement frames doubles RTTs. (Software acknowledgement frames can be accelerated by sending them within interrupt routines, which was not done here but in [9]). The RTTs deteriorate further when using command frames since command frames have to be passed up to the link layer. Moreover, when securing command frames, their mean RTT raises to 8.23ms. Thereby, we used the hardware-accelerated Advanced Encryption Standard (AES) implementation of the CC2420. However, since our implementation precomputes the CCM*-MICs of both PINGs and PONGs, the mean RTT of PING-PONGs in our implementation is just 3.52ms.

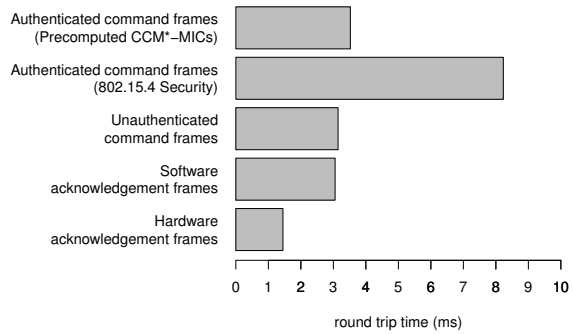

Figure 6: Mean RTTs of different options for implementing PING-PONGs

The overall duration of the sampling phase is exemplified in Figure 7. The jumps stem from timeouts. Throughout our experiments, we set $\tau = 50$ms. τ can be reduced further by implementing the transmission of PINGs and PONGs as real-time tasks.

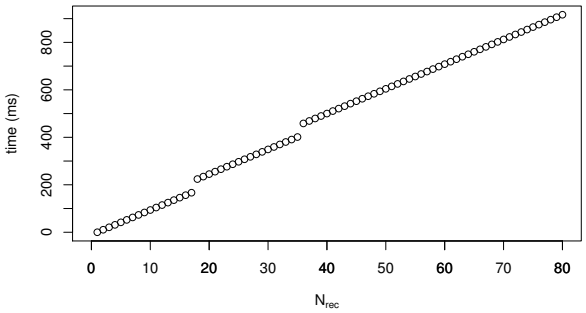

Figure 7: Duration of the sampling phase

5.3.2 *Memory Efficiency*

We measured the program memory and random-access memory (RAM) overhead of SCREWED with the tools `msp430-size` and `msp430-ram-usage`. The result is that SCREWED consumes 2,152 bytes of program memory and 92 bytes of RAM. This corresponds to 4.5% of the program memory and 0.9% of the RAM on a TelosB mote.

6. RELATED WORK

RPL's vulnerability to wormholes was discussed in [38, 40, 23]. These works stress that wormholes may prepare sinkhole attacks. In addition, Tsao et al.[38] note repercussions of selective and transient wormholes. As a preventive measure, Wallgren et al.[40] proposed to use a different group key for link layer security in each network part. However, this approach has several problems. First, group keys do not allow for compromise resilience [25, 29]. Second, group keys require deployment knowledge when predistributing them. Third, wormholes can still be set up within each network part. Last, Wallgren et al. do not specify what link layer keys should be used for communications between different network parts. Another preventive measure was proposed by Khan et al.[23], but they focus on exposed wormholes. SCREWED focusses on hidden wormholes and avoids the problems of Wallgren et al.'s approach.

In the broader context of wireless sensor networks (WSNs), plenty of wormhole detection schemes that use special hardware were proposed [13, 18, 4, 14, 41]. These wormhole detection schemes use directional antennas [13], radio frequency fingerprinting-capable transceivers [18, 4], or Global Positioning System (GPS) receivers [14, 41]. SCREWED avoids special hardware by relying on RSSIs. These are readily-available on off-the-shelf transceivers.

Other directions to avoid special hardware are detecting anomalies in RTTs[7, 10, 31, 37, 39] or routing topologies [36, 12]. Unfortunately, most RTT-based schemes[7, 10, 31, 37] cannot detect low-latency wormholes, which, e.g., use cut-through forwarding. Distance bounding protocols[39] also detect low-latency wormholes, but require special hardware [32]. Both Statistical Wormhole Apprehension using Neighbors (SWAN)[36] and SECUre NeighborhooD (SECUND)[12] detect wormholes based on anomalies in routing topologies. However, while SWAN[36] cannot detect selective wormholes, SECUND[12] cannot detect short wormholes, which bridge only short distances. SCREWED detects low-latency, selective, as well as short wormholes.

RSSI-based wormhole detection schemes were proposed by Chen et al.[6] and Jain et al.[19]. Chen et al.[6] augment an RSSI-based localization protocol with a wormhole detection rule set. However, they neither consider replay attacks nor low-latency wormholes. Such attacks can mislead Chen et al.'s rule set, whereas SCREWED resists such attacks. SCREWED is based on Jain et al.'s channel reciprocity-based wormhole detection schemes [19]. We pointed out threats and attacks that can mislead Jain et al.'s schemes into reaching false positives or false negatives. Moreover, Jain et al.'s channel reciprocity metrics assume calibrated RSSIs. SCREWED avoids false positives and false negatives to a great extent and obviates the need for calibrating RSSIs.

Unfortunately, neither SCREWED nor Jain et al.'s schemes can detect exposed wormholes. Several wormhole detection schemes target both exposed and hidden wormholes [21, 22, 5, 7, 31, 37]. However, in contrast to SCREWED, many of them[21, 22, 5] can neither detect transient wormholes nor work proactively. Moreover, each of them has its specific limitations. Both LITEWORP[21] and MOBIWORP[22] require nodes to overhear each other's traffic. This is energy consuming and incompatible with energy-efficient media access control (MAC) protocols, such as ContikiMAC[8]. Chen et al.'s scheme [5], on the other hand, does not detect selective wormholes. The limitations of [7, 31, 37] were already mentioned.

7. CONCLUSIONS AND FUTURE WORK

Wormholes can paralyze large parts of 6LoWPAN networks. To prevent this, we have proposed SCREWED, which uses channel reciprocity to detect and subsequently avoid wormholes. SCREWED detects selective, transient, low-latency, as well as short wormholes. Thereby, SCREWED forgoes special hardware, avoids calibration issues, supports battery-powered nodes, and operates localized. Our future work will evolve around three aspects. First, we will try to detect adaptive wormholes with different channel

characteristics. Second, we plan to reduce the energy consumption of SCREWED by, e.g., shortening PING and PONG frames, reducing transmission powers, minimizing τ, or leaving transceivers off when possible. Third, to avoid that SCREWED is run again and again with dropped neighbors, we want to blacklist dropped neighbors in our implementation for some time.

8. ACKNOWLEDGMENTS

This work was carried out within the project "Providing Physical Layer Security for the Internet of Things (PROPHYLAXE)" funded by the Federal Ministry of Education and Research (BMBF), grant number 16KIS0005K. We would like to thank the anonymous reviewers for their excellent suggestions.

9. REFERENCES

[1] *2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B)*. `http://www.ti.com/lit/ds/symlink/cc2420.pdf`.

[2] IEEE Standard 802.15.4, 2006. `http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf`.

[3] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener. Robust key generation from signal envelopes in wireless networks. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pages 401–410. ACM, 2007.

[4] K. Bonne Rasmussen and S. Capkun. Implications of radio fingerprinting on the security of sensor networks. In *Proceedings of the Third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm 2007)*, pages 331–340. IEEE, 2007.

[5] H. Chen, W. Chen, Z. Wang, Z. Wang, and Y. Li. Mobile beacon based wormhole attackers detection and positioning in wireless sensor networks. *Distributed Sensor Networks*, 2014, 2014.

[6] H. Chen, W. Lou, and Z. Wang. On providing wormhole-attack-resistant localization using conflicting sets. *Wireless Communications and Mobile Computing*, 2014.

[7] H. S. Chiu and K.-S. Lui. DelPHI: wormhole detection mechanism for ad hoc wireless networks. In *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, pages 6–11. IEEE, 2006.

[8] A. Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report T2011:13, Swedish Institute of Computer Science, 2011.

[9] S. Duquennoy, O. Landsiedel, and T. Voigt. Let the tree bloom: scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*, pages 2:1–2:14. ACM, 2013.

[10] J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos. TrueLink: a practical countermeasure to the wormhole attack in wireless networks. In *Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP'06)*, pages 75–84. IEEE, 2006.

[11] O. Gnawali and P. Levis. The Minimum Rank with Hysteresis Objective Function. RFC 6719, 2012.

[12] T. Hayajneh, P. Krishnamurthy, D. Tipper, and A. Le. Secure neighborhood creation in wireless ad hoc networks using hop count discrepancies. *Mobile Networks and Applications*, 17(3):415–430, 2012.

[13] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *Proeedings of the Network and Distributed System Security Symposium (NDSS 2004)*, 2004.

[14] Y.-C. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003)*, volume 3, pages 1976–1986. IEEE, 2003.

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2Nd ACM Workshop on Wireless Security (WiSe '03)*, pages 30–40. ACM, 2003.

[16] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, 2011. Updates RFC 4944.

[17] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle. 6LoWPAN fragmentation attacks and mitigation mechanisms. In *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '13)*, pages 55–66. ACM, 2013.

[18] S. Jain and J. Baras. Preventing wormhole attacks using physical layer authentication. In *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2712–2717, 2012.

[19] S. Jain, T. Ta, and J. Baras. Wormhole detection using channel characteristics. In *Proceedings of the 2012 IEEE International Conference on Communications (ICC 2012)*, pages 6699–6704. IEEE, 2012.

[20] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom '09)*, pages 321–332. ACM, 2009.

[21] I. Khalil, S. Bagchi, and N. Shroff. LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN 2005)*, pages 612–621. IEEE, 2005.

[22] I. Khalil, S. Bagchi, and N. B. Shroff. MobiWorp: mitigation of the wormhole attack in mobile multihop wireless networks. *Ad Hoc Networks*, 6(3):344–362, 2008.

[23] F. Khan, T. Shon, T. Lee, and K. Kim. Wormhole attack prevention mechanism for RPL based LLN network. In *Proceedings of the Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 149–154. IEEE, 2013.

[24] E. Kim, D. Kaspar, and J. Vasseur. Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). RFC 6568, 2012.

[25] K.-F. Krentz, H. Rafiee, and C. Meinel. 6LoWPAN security: adding compromise resilience to the 802.15.4 security sublayer. In *Proceedings of the International Workshop on Adaptive Security & Privacy Management for the Internet of Things (ASPI '13)*, pages 1:1–1:10. ACM, 2013.

[26] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, pages 128–139. ACM, 2008.

[27] N. Patwari and P. Agrawal. Calibration and measurement of signal strength for sensor localization. In G. Mao and

B. Fidan, editors, *Localization Algorithms and Strategies for Wireless Sensor Networks*, pages 122–145. IGI Global, 2009.

[28] N. Patwari, J. Croft, S. Jana, and S. Kasera. High-rate uncorrelated bit extraction for shared secret key generation from channel measurements. *IEEE Transactions on Mobile Computing*, 9(1):17–30, 2010.

[29] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA CryptoBytes*, 5(2), 2002.

[30] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pages 364–369. IEEE, 2005.

[31] V. K. Raju and K. V. Kumar. A simple and efficient mechanism to detect and avoid wormhole attacks in mobile ad hoc networks. In *Proceedings of 2012 International Conference on Computing Sciences (ICCS)*, pages 271–275. IEEE, 2012.

[32] K. B. Rasmussen and S. Čapkun. Realization of RF distance bounding. In *Proceedings of the 19th USENIX Conference on Security*, pages 25–25. USENIX, 2010.

[33] S. Raza, L. Wallgren, and T. Voigt. SVELTE: real-time intrusion detection in the Internet of things. *Ad Hoc Networks*, 2013.

[34] N. Sastry and D. Wagner. Security considerations for IEEE 802.15.4 networks. In *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe '04)*, pages 32–42. ACM, 2004.

[35] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252, 2014.

[36] S. Song, H. Wu, and B.-Y. Choi. Statistical wormhole detection for mobile sensor networks. In *Proceedings of the Fourth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 322–327. IEEE, 2012.

[37] P. V. Tran, L. X. Hung, Y.-K. Lee, S. Lee, and H. Lee. TTM: an efficient mechanism to detect wormhole attacks in wireless ad-hoc networks. In *Proceedings of 4th IEEE Consumer Communications and Networking Conference (CCNC 2007)*, pages 593–598. IEEE, 2007.

[38] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson. A security threat analysis for routing protocol for low-power and lossy networks (RPL). Internet-Draft, August 2014.

[39] S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, pages 21–32. ACM, 2003.

[40] L. Wallgren, S. Raza, and T. Voigt. Routing attacks and countermeasures in the RPL-based Internet of things. *Distributed Sensor Networks*, 2013, 2013.

[41] X. Wang and J. Wong. An end-to-end detection of wormhole attack in wireless ad-hoc networks. In *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, volume 1, pages 39–48. IEEE, 2007.

[42] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). RFC 3610, 2003.

[43] M. Wilhelm, I. Martinovic, and J. Schmitt. Secure key generation in sensor networks based on frequency-selective channels. *IEEE Journal on Selected Areas in Communications*, 2013.

[44] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, 2012.

[45] L. Yao, S. Ali, V. Sivaraman, and D. Ostry. Decorrelating secret bit extraction via channel hopping in body area networks. In *Proceedings of the 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 1454–1459. IEEE, 2012.

APPENDIX

A. NOTATIONS

Symbol	Meaning
A, B, and W	Node A, node B, and wormhole W
N	Total number of PINGs
N_{rec}	Number of received PONGs
N_{min}	Number of $(P_{A,i} - P_{B,i}, RSSI_{B,i} - RSSI_{A,i})$ pairs that are used in the judgement phase for calculating the correlation
$RSSI_{A,i}$	i-th RSSI measured by A
$P_{A,i}$	i-th transmission power of A
ϵ	RSSI of a missed PING or PONG
ρ	Threshold when a channel is considered a wormhole
τ	Timeout before sending the next PING, or rather the waiting period for the next PING
$s(X)$	Sample standard deviation of X
$s(X,Y)$	Sample covariance of X and Y
$r(X,Y)$	Sample correlation coefficient (or correlation for short) between X and Y
c_i	i-th channel
f_A	Frame counter of A

B. DISTRIBUTIONS

Table 2: Distribution of $P_{A,i} - P_{B,i}$

δ_i in dBm	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
$\Pr(P_{A,i} - P_{B,i} = \delta_i)$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{15}$

Table 3: On the diagonals are the distribution of pairs $(P_{A,i}, P_{B,i})$ with $P_{A,i} - P_{B,i} = \delta_i$ for each $\delta_i \in \{-7, \ldots, 7\}$

$P_{A,i}$ \ $P_{B,i}$	0	-1	-2	-3	-4	-5	-6	-7
0	0.00271	0.00654	0.01575	0.03797	0.09150	0.22052	0.50000	1.00000
-1	0.00654	0.01727	0.04163	0.10033	0.24179	0.40850	0.55895	0.50000
-2	0.01575	0.04163	0.10680	0.25739	0.38392	0.44049	0.40850	0.22052
-3	0.03797	0.10033	0.25739	0.37321	0.38890	0.38392	0.24179	0.09150
-4	0.09150	0.24179	0.38392	0.38890	0.37321	0.25739	0.10033	0.03797
-5	0.22052	0.40850	0.44049	0.38392	0.25739	0.10680	0.04163	0.01575
-6	0.50000	0.55895	0.40850	0.24179	0.10033	0.04163	0.01727	0.00654
-7	1.00000	0.50000	0.22052	0.09150	0.03797	0.01575	0.00654	0.00271