More Lightweight, yet Stronger 802.15.4 Security through an Intra-Layer Optimization

Konrad-Felix Krentz[1], Christoph Meinel[1], and Hendrik Graupner[2]

[1] Hasso-Plattner-Institut, University of Potsdam, Germany
{konrad-felix.krentz,christoph.meinel}@hpi.de
[2] Bundesdruckerei, Germany
hendrik.graupner@bdr.de

Abstract. 802.15.4 security protects against the replay, injection, and eavesdropping of 802.15.4 frames. A core concept of 802.15.4 security is the use of frame counters for both nonce generation and anti-replay protection. While being functional, frame counters (i) cause an increased energy consumption as they incur a per-frame overhead of 4 bytes and (ii) only provide sequential freshness. The Last Bits (LB) optimization does reduce the per-frame overhead of frame counters, yet at the cost of an increased RAM consumption and occasional energy- and time-consuming resynchronization actions. Alternatively, the timeslotted channel hopping (TSCH) media access control (MAC) protocol of 802.15.4 avoids the drawbacks of frame counters by replacing them with timeslot indices, but findings of Yang et al. question the security of TSCH in general. In this paper, we assume the use of ContikiMAC, which is a popular asynchronous MAC protocol for 802.15.4 networks. Under this assumption, we propose an Intra-Layer Optimization for 802.15.4 Security (ILOS), which intertwines 802.15.4 security and ContikiMAC. In effect, ILOS reduces the security-related per-frame overhead even more than the LB optimization, as well as achieves strong freshness. Furthermore, unlike the LB optimization, ILOS neither incurs an increased RAM consumption nor requires resynchronization actions. Beyond that, ILOS integrates with and advances other security supplements to ContikiMAC. We implemented ILOS using OpenMotes and the Contiki operating system.

1 Introduction

The major features of the 802.15.4 radio standard are low energy consumption, cheap transceivers, sub-GHz and 2.4-GHz support, and reliable communication thanks to meshing [1]. These features are suitable for implementing wireless sensor and actuator networks. Furthermore, with the advent of 6LoWPAN, an adaption layer for conveying IPv6 packets over 802.15.4 links [14], 802.15.4 is becoming a main choice for implementing Internet of things (IoT) applications.

As for wireless security, many 802.15.4 networks make use of 802.15.4 security. Essentially, 802.15.4 security filters out injected and replayed 802.15.4 frames, and optionally encrypts the payload of 802.15.4 frames. Specifically, to filter out injected 802.15.4 frames, 802.15.4 security ensures that incoming 802.15.4

frames contain authentic message integrity codes (MICs). Both, for generating MICs and for encrypting payloads, 802.15.4 security employs a tweaked version of Counter with CBC-MAC (CCM) [23]. CCM, in turn, requires a nonce for generating a MIC and encrypting data. 802.15.4 security generates CCM nonces based on incrementing 4-byte frame counters, which 802.15.4 security adds to 802.15.4 frames. Besides, to filter out replayed 802.15.4 frames, 802.15.4 security ascertains that the frame counter of an incoming 802.15.4 frame is greater than that of the last authentic 802.15.4 frame from the sender, thereby providing sequential freshness [20].

Yet, the use of frame counters in 802.15.4 security ensues two drawbacks. First, as a frame counter is added to each 802.15.4 frame, frame transmissions and receptions become more energy consuming as a result. Moreover, since frame counters cut down the maximum payload of 802.15.4 frames, IPv6 packets need to be fragmented at the 6LoWPAN adaption layer more often, thus necessitating additional frame transmissions and receptions. Altogether, frame counters reduce the lifetime of battery-powered 802.15.4 nodes. Second, while frame counters provide sequential freshness, upper layers may require strong freshness. Strong freshness is provided if a receiver can ensure that an incoming 802.15.4 frame was sent within a limited time span prior to its reception [20]. This is particularly desirable if readings from sensors or commands to actuators lose their meaning when being delayed and hence should not be considered fresh.

In order to reduce the per-frame overhead of frame counters, Krentz et al. tailored the Last Bits (LB) optimization to 802.15.4 security [10, 15, 18]. In their version, senders only add the 8 least significant bits (LSBs) of frame counters to outgoing 802.15.4 frames. Nevertheless, receivers can restore higher-order bits using their anti-replay data. On the other hand, to enable receivers to restore higher order bits, each node needs to use a separate frame counter for broadcast frames, as well as separate frame counters for unicast frames for each of its neighbors. Also, every node has to keep track of both the unicast and broadcast frame counter of each of its neighbors. Hence, the LB optimization consumes more RAM than the original anti-replay protection of 802.15.4 security. Moreover, if a node A misses 2^8 unicast or broadcast frames in a row from a neighbor B, A can no longer restore B's unicast or broadcast frame counters, respectively. To this end, Krentz et al. propose an "`UPDATE-UPDATEACK` exchange" for resynchronizing frame counters. Unfortunately, this exchange entails sending two unicast frames and is not triggered immediately upon desynchronization, but only delayed.

A seemingly better solution appeared as part of the timeslotted channel hopping (TSCH) media access control (MAC) protocol of 802.15.4 [1]. TSCH nodes wake up in certain timeslots so as to receive or transmit data on a certain channel as governed by a schedule. Consequently, TSCH requires network-wide time synchronization. Further, since TSCH requires network-wide time synchronization anyway, TSCH uses implicitly known timeslot indices in lieu of frame counters. This elegantly avoids both mentioned drawbacks with frame counters through an intra-layer optimization. However, Yang et al. outlined various attacks on TSCH's mechanisms for time synchronization [25], thus questioning the security

of TSCH in general. After all, TSCH's intra-layer optimization is inapplicable to asynchronous MAC protocols, which work without network-wide time synchronization [13]. A popular asynchronous MAC protocol is, e.g., ContikiMAC [6].

This paper's main contribution is an Intra-Layer Optimization for 802.15.4 Security (ILOS). ILOS solves both mentioned drawbacks with frame counters by intertwining 802.15.4 security and ContikiMAC. In fact, ILOS reduces the security-related per-frame overhead even more than the LB optimization, as well as achieves strong freshness. Furthermore, unlike the LB optimization, ILOS neither consumes more RAM nor needs resynchronization actions. Beyond that, ILOS integrates with and advances other security supplements to ContikiMAC, namely Krentz et al.'s Adaptive Key Establishment Scheme (AKES) [15], as well as their Practical On-The-fly Rejection (POTR) [17].

The rest of this paper is structured as follows. Section 2 introduces ContikiMAC, as well as security supplements to it. Section 3 specifies the design of ILOS. Section 4 outlines our implementation of ILOS. Section 5 gives an evaluation of ILOS. Lastly, Section 6 concludes and suggests topics for future research.

2 Background and Related Work

In ContikiMAC, receivers wake up periodically and perform two clear channel assessments (CCAs). If one of these CCAs indicates a busy channel, receivers stay in receive mode until a frame is received or a timeout occurs, whatever comes first. Senders, on the other hand, repeatedly transmit each frame for a whole wake-up interval, plus once to cover corner cases. This behavior is often called strobing. In the case of unicast frames, senders may stop strobing prematurely if an acknowledgement frame is received in between two consecutive unicast frame transmissions, as shown in Fig. 1. Additionally, to further reduce the time that senders spend in transmit mode, ContikiMAC's phase-lock optimization schedules the start of a strobe of unicast frames right before the intended receiver wakes up. For this, ContikiMAC's phase-lock optimization exploits that if an acknowledgement frame is received, the next to last strobed unicast frame must have been transmitted while the receiver woke up. Hence, the time when the transmission of the next to last acknowledged unicast frame began can serve to estimate when the receiver will wake up next. Furthermore, once the wake-up time of a receiver is known, ContikiMAC's phase-lock optimization no longer strobes unicast frames to that receiver for a whole wake-up interval plus once if no acknowledgement frame returns, but only for a shorter time span plus once. Yet, to account for clock drift, ContikiMAC's phase-lock optimization relearns the wake-up time of a receiver if unicast transmissions to the receiver tend to fail. This fallback mechanism renders ContikiMAC's phase-lock optimization susceptible to collision attacks, which provoke longer strobes via jamming [16].

Since its publication, ContikiMAC was improved to support opportunistic routing [7], burst forwarding [8], and channel hopping [2]. In the following, we will however restrict ourselves to introducing security supplements to ContikiMAC since they are fundamental to ILOS [15–17].

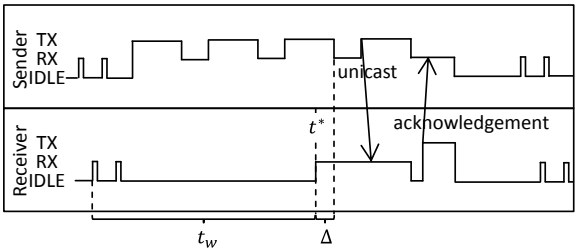

Fig. 1. Operation of a unicast transmission in ContikiMAC

An essential security supplement to ContikiMAC is AKES, which establishes group or pairwise session keys for use in 802.15.4 security [15]. Fig. 2 shows the operation of AKES when configured to establish group session keys. A node A initiates session key establishment by broadcasting a `HELLO`, containing a cryptographic random number R_A. Any receiver B that has not yet established session keys with A also generates a cryptographic random number R_B and, after a random back off period, replies with a `HELLOACK`. The `HELLOACK` carries R_B, B's group session key $K_{B,*}$ encrypted, as well as a MIC. For encrypting $K_{B,*}$ and generating the MIC, B derives a temporary pairwise key $K'_{A,B}$ from a predistributed shared secret $K_{A,B}$ between A and B, as well as the two cryptographic random numbers R_A and R_B. Upon receipt of B's `HELLOACK`, A decrypts $K_{B,*}$ and checks the MIC by deriving $K'_{A,B}$ analogously. If successful, A acknowledges with an `ACK`, which includes A's group session key $K_{A,*}$ encrypted and a MIC. Again, the temporary pairwise key $K'_{A,B}$ serves to generate the `ACK`'s MIC, as well as to encrypt A's group session key.

Apart from establishing session keys, AKES also deletes neighbors that got out of range. Concretely, if a neighbor sent no fresh authentic frame for a critical period of time, AKES checks if the neighbor is still in range by sending an `UPDATE` to him, as shown in Fig. 2. If no fresh authentic `UPDATEACK` returns after a configurable number of retransmissions, AKES deletes that neighbor. Otherwise, if an `UPDATEACK` returns, AKES extends that neighbor's expiration time. In this regard, ILOS obviates the need for sending `UPDATEACK`s, which saves energy. Furthermore, thanks to ILOS, AKES no longer needs to keep track of the expiration times of neighboring nodes, which saves RAM.

Despite AKES, 802.15.4 security remains incomplete in the sense that ContikiMAC stays vulnerable to many ding-dong ditching attacks. Ding-dong ditching attacks belong to the larger group of denial-of-sleep attacks, which generally cause an increased energy consumption on victim nodes [4]. Ding-dong ditching attacks, in particular, mislead ContikiMAC nodes into staying more time in receive mode [16]. For example, broadcast and unicast attacks are ding-dong ditching attacks, where an attacker injects or replays broadcast and unicast frames, respectively [4, 16]. Though 802.15.4 security rejects injected and replayed frames, they are still fully received before being rejected, which consumes much energy. As another example of a ding-dong ditching attack, droplet attacks

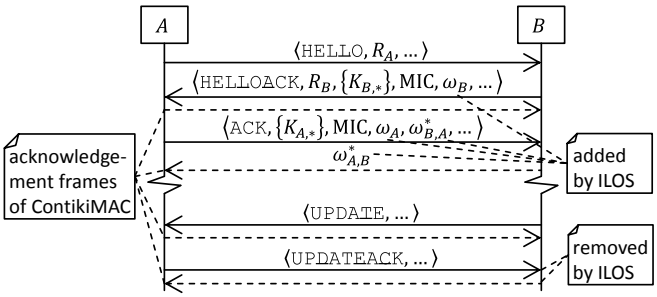

Fig. 2. Adaptive Key Establishment Scheme (AKES) and adaptations to it

exploit that it suffices to transmit an 802.15.4-compliant synchronization header (SHR) plus a few header bytes to keep victim nodes in receive mode for long [11].

A countermeasure against unicast, broadcast, as well as droplet attacks is Krentz et al.'s POTR [17]. The approach of POTR is to cancel the reception of injected and replayed frames early on, similar to what was proposed in related efforts [3, 5, 9, 11, 12, 24]. For this, POTR adapts the headers of 802.15.4 frames like shown in Fig. 3. The Frame Type field encodes the frame's type. Possible frame types are listed in Table 1. Then, the Source Address field states the sender's address. Thereafter, the Frame Counter field includes the sender's 4-byte frame counter. Acknowledgement frames, on the other hand, always just include the 8 LSBs of the frame counter of the frame whose receipt is being acknowledged. Above all, the OTP field contains a one-time password (OTP). POTR validates OTPs during reception by (i) parsing the Frame Type, Source Address, and Frame Counter fields, (ii) deriving an OTP therefrom, and (iii) checking if the derived OTP matches the received one. If they do not match, POTR usually disables the receive mode immediately, which greatly reduces the time that victim nodes stay in receive mode under unicast, broadcast, as well as droplet attacks. Finally, the purpose of the Sequence Number field is to avoid accepting retransmitted frames twice, as we will elaborate on in Section 3.4.

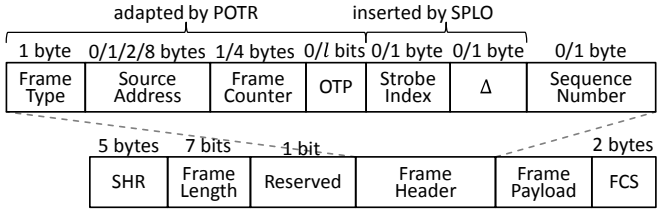

Fig. 3. 802.15.4 frame format as adapted by POTR, SPLO, and ILOS

To shorten POTR's Frame Counter field and to accelerate POTR's rejection speed, Krentz et al. suggested using the LB optimization [17]. The LB optimization accelerates POTR's rejection speed because if the Frame Counter field gets

Table 1. POTR's Frame Types and Fields Present therein

Frame Type	Source Address	Frame Counter	OTP	Strobe Index	Δ Sequence Number
unicast data	√	√ (× if using ILOS)	√	√	× √
broadcast data	√	√ (× if using ILOS)	√	×	× ×
acknowledgement	×	√ (× if using SPLO or ILOS)	×	×	√ ×
`HELLO`	√	√ (× if using ILOS)	√	×	× ×
`HELLOACK`	√	√ (× if using ILOS)	√	√	× ×
`ACK`	√	√ (× if using ILOS)	√	√	× ×
unicast command	√	√ (× if using ILOS)	√	√	× √
broadcast command	√	√ (× if using ILOS)	√	×	× ×

shorter, receivers can validate OTPs earlier. However, as mentioned already, if a node misses a lot of frames, it needs to perform an `UPDATE-UPDATEACK` exchange. Moreover, in POTR, if a node loses track of the unicast frame counter of a neighbor, both nodes have to establish new session keys with each other for subtle reasons [17]. Such resynchronization actions may also become necessary if an attacker guesses an OTP right, irrespective of using the LB optimization or not [17]. By contrast, ILOS never needs resynchronization actions, even if an attacker guesses an OTP right. Beyond that, ILOS accelerates POTR's rejection speed slightly more than the LB optimization, as we will show in Section 5.

Lastly, the Secure Phase-Lock Optimization (SPLO) protects ContikiMAC's phase-lock optimization from pulse-delay and collision attacks [16]. Both of these attacks are denial-of-sleep attacks that mislead ContikiMAC's phase-lock optimization into staying more time in transmit mode. To resist pulse-delay attacks, SPLO ensures the authenticity and timeliness of acknowledgement frames, mainly by adding MICs to acknowledgement frames. This involves inserting the Strobe Index field into unicast frames like shown in Fig. 3. This field indicates how often ContikiMAC strobed a unicast frame already and is incorporated into the CCM nonce of unicast frames, as well as into the CCM nonce of their corresponding acknowledgement frames. To mitigate collision attacks, on the other hand, SPLO limits the maximum duration of a strobe of unicast frames according to the current uncertainty about the wake-up time of the intended receiver. Crucially, unlike ContikiMAC's original phase-lock optimization, SPLO does not relearn the wake-up time of a receiver if unicast transmissions to the receiver tend to fail. Furthermore, to keep the maximum duration of a strobe of unicast frames below a configurable threshold, SPLO instructs AKES to send an `UPDATE` to a neighbor that sent no fresh authentic acknowledgement frame for a critical period of time, as well as to delete that neighbor if no `UPDATEACK` returns. Also, to obtain more precise estimations of wake-up times, SPLO piggybacks Δ on acknowledgement frames. Δ is calculated like shown in Fig. 1 and enables senders of unicast frames to calculate the last wake-up time t^* of a receiver [19].

3 ILOS: Intra-Layer Optimization for 802.15.4 Security

The main idea of ILOS is to replace frame counters with what we call wake-up counters. This necessitates changes to (i) the generation of CCM nonces, (ii) POTR, (iii) anti-replay protection, and (iv) AKES. In the following, we will specify each of these changes. Throughout, we focus on the case of using group session keys, rather than pairwise session keys.

3.1 Notations

Let A and B be adjacent nodes. We denote by:

- t_w ContikiMAC's wake-up interval, as shown in Fig. 1
- ω_A the wake-up counter of A - A increments ω_A at the rate of t_w in lockstep with ContikiMAC's two regular CCAs. If A skips over doing two CCAs, e.g., due to sending at that time, A must increment ω_A anyway.
- $t^*_{A,B}$ what A stores as the last wake-up time of B - SPLO initializes $t^*_{A,B}$ in parallel to establishing group session keys and updates $t^*_{A,B}$ upon receipt of a timely authentic acknowledgement frame from B.
- $\omega^*_{A,B}$ the wake-up counter of B at time $t^*_{A,B}$ - Likewise, ILOS initializes $\omega^*_{A,B}$ in the course of establishing session keys and updates $t^*_{A,B}$ upon receipt of a timely authentic acknowledgement frame from B.
- ID_A A's MAC address
- $K_{A,*}$ A's group session key
- α a field in the CCM nonces generated by ILOS
- λ_A A's current strobe index - each time a unicast frame is transmitted or retransmitted, λ_A starts over from zero
- KDF a key derivation function
- K_n a predistributed network-wide key for use by POTR

3.2 Adapting CCM Nonces

In order for CCM nonces to be secure, they must never reoccur in conjunction with the same key. ILOS achieves this via two complementary techniques. On the one hand, ILOS uses a common base format and the field α to avoid collisions among CCM nonces of different types of frames. On the other hand, ILOS uses wake-up counters and MAC addresses to avoid collisions among CCM nonces of the same frame type. Concretely, ILOS derives CCM nonces from wake-up counters like shown in Table 2.

Unicast Frames As for a `HELLOACK` or `ACK` from a node A to a node B, ILOS generates the CCM nonce by concatenating ID_A, $\alpha = 0$, λ_A, and ω_A, where ω_A is the wake-up counter of A as A begins to strobe the `HELLOACK` or `ACK`. Thus, B needs ω_A to restore the CCM nonce of the `HELLOACK` or `ACK`. Therefore, ILOS adds ω_A to the payload of the `HELLOACK` or `ACK`, as shown in Fig. 2. It

Table 2. CCM Inputs as per ILOS

Frame Types	CCM Nonce	Key
`HELLOACK` or `ACK` from A to B	$ID_{\mathrm{A}} \| \alpha \| \lambda_A \| \omega_A$	$K'_{A,B}$
unicast data or command frame from A to B	$ID_{\mathrm{A}} \| \alpha \| \lambda_A \| \omega_B$	$K_{B,*}$
acknowledgement frame from B to A	same as the corresponding unicast frame except that $\alpha = 2$	same as the corresponding unicast frame
broadcast data, broadcast command, or `HELLO` frame from A	$ID_{\mathrm{A}} \| \alpha \| 0 \| \omega_A + 1$	$K_{A,*}$

will become apparent that adding wake-up counters to frames is only required during session key establishment. The LB optimization is no different in this respect as it requires exchanging certain frame counters in full during session key establishment, too [15]. We also note that ILOS depends on that the back-off period for retransmissions is greater or equal than t_w so that ω_A increments in the meantime. Otherwise, such CCM nonces may reoccur.

As for a unicast data or command frame from a node A to a node B, ILOS generates the CCM nonce by concatenating ID_{A}, $\alpha = 1$, λ_A, and ω_B, where ω_B is B's wake-up counter as B receives the unicast frame. Thus, A needs ω_B. However, using $t^*_{A,B}$ and $\omega^*_{A,B}$, A can predict ω_B as $\omega^*_{A,B} + \left\lceil \frac{t_{\mathrm{sched}} - t^*_{A,B}}{t_w} \right\rceil$, where t_{sched} is the time when SPLO schedules the transmission of the unicast frame. This prediction is correct if SPLO keeps the absolute uncertainty about the wake-up time of B below $\frac{t_w}{2}$, which SPLO achieves by default.

Acknowledgement Frames An acknowledgement frame uses the same CCM nonce as the unicast frame whose receipt is being acknowledged except that α is set to 2.

Broadcast Frames The CCM nonce of a broadcast frame from a node A is generated by concatenating ID_{A}, $\alpha = 3$, 0, and $\omega_A + 1$. Here, ω_A is A's wake-up counter as A begins to strobe. To aid receivers in restoring $\omega_A + 1$, $\omega_A + 1$ needs to be even and A must begin to strobe at $t - \frac{t_w}{2}$, where t is when A increments ω_A next. Thus, A may need to defer the transmission of the broadcast frame until both conditions are met. Yet, this way, a receiver B can restore $\omega_A + 1$ by rounding $\omega^*_{B,A} + \frac{t_{\mathrm{awoke}} - t^*_{B,A}}{t_w}$ to the next even value, where t_{awoke} is when B awoke for doing ContikiMAC's two CCAs that led to receiving the broadcast frame. This restoration of $\omega_A + 1$ is correct (i) if SPLO keeps the absolute uncertainty about the wake-up time of A below $\frac{t_w}{2}$ and (ii) if B wakes up during the interval $[t - \frac{t_w}{2}, t + \frac{t_w}{2}]$. As a side effect of delaying consecutive broadcast transmissions by at least $2t_w$, CCM nonces of broadcast frames can not coincide.

3.3 Adapting the Practical On-the-fly Rejection (POTR)

As POTR derives OTPs of `HELLOACK`s and `ACK`s without frame counters already, ILOS leaves these OTPs unchanged. By contrast, POTR derives the OTPs of data, command, and `HELLO` frames from K_n, the sender's group session key, the receiver's address, and the sender's frame counter. To avoid frame counters there too, ILOS calculates the OTP of a unicast data or command frame from A to B as $\mathrm{KDF}(K_n \oplus K_{B,*}, ID_A \| \alpha \| \omega_B)$, where $\alpha = 1$ and ω_B is B's wake-up counter when receiving the frame. The purpose of XORing K_n and $K_{B,*}$ is to prevent related-key attacks [17]. Likewise, ILOS calculates the OTP of a broadcast data, broadcast command, or `HELLO` frame from A as $\mathrm{KDF}(K_n \oplus K_{A,*}, ID_A \| \alpha \| \omega_A + 1)$, where $\alpha = 3$ and ω_A is A's wake-up counter as A begins to strobe. As the inputs to KDF resemble the CCM nonces of ILOS, the same methods for predicting, or rather restoring wake-up counters apply.

3.4 Adapting Anti-Replay Protection

ILOS provides anti-replay protection as follows.

Unicast Frames To filter out replayed `HELLAOCK`s and `ACK`s, ILOS retains POTR's methods [17].

As for unicast data and command frames, anti-replay protection comes almost as a side effect of generating CCM nonces like ILOS does. This is because a receiver B increments ω_B when waking up. Thus, if B receives a replayed unicast frame, B will assume a CCM nonce that differs from the CCM nonce that was used to secure the replayed unicast frame. Hence, B will reject the replayed unicast frame due to an invalid OTP during reception. Also, even if the OTP of a replayed unicast frame is valid by chance, B will eventually reject the replayed unicast frame due to an inauthentic MIC. However, a subtlety is that the sender A of a unicast data or command frame may miss an acknowledgement frame. In this case, A may retransmit and hence, the receiver B, may accept the same frame twice. This issue also arises in TSCH and POTR, where it may be solved by adding 8-bit sequence numbers to frames [1]. ILOS adopts this solution and adds sequence numbers to unicast data and command frames, as shown in Figure 3. These sequence numbers are incremented on a per neighbor basis. B discards a unicast data or command frame from A if the contained sequence number matches the one of the previously accepted unicast data or command frame from A. However, although duplicated unicast data and command frames could already be discarded during reception, we opted to fully receive and acknowledge them so as to avoid self-imposed collision attacks.

Acknowledgement Frames As for acknowledgement frames, anti-replay protection comes indeed by itself. If a sender A receives a replayed acknowledgement frame, A will use a different CCM nonce to check if the MIC of the replayed acknowledgement frame is authentic. Consequently, A will consider the MIC of the

replayed acknowledgement frame inauthentic and reject the replayed acknowledgement frame.

Broadcast Frames As for broadcast frames, anti-replay protection does not come automatically in two occasions. First, it may happen that a receiver B receives a broadcast frame from a sender A and, during the next wake up of B, an attacker replays that same broadcast frame. In this case, B may assume the same CCM nonce, thus causing B to consider both the OTP and the MIC of the replayed broadcast frame valid. Only if there is one wake up in between, B will definitely assume a different CCM nonce and hence reject the replayed broadcast frame due to an invalid OTP or an inauthentic MIC. This also holds true if B updates $t^*_{B,A}$ and $\omega^*_{B,A}$ in between since, in this case, $\omega^*_{B,A}$ is raised, which causes B to restore a different CCM nonce, too. Altogether, ILOS merely needs to take care of not accepting a broadcast frame if, during the last wake up, a broadcast frame from the same sender was accepted already. Like POTR, ILOS does so already during reception to counter ding-dong ditching. Second, when a sender retransmits a broadcast frame because of collision avoidance (CA), a receiver may receive that frame twice. This issue is specific to ContikiMAC since, in TSCH, CA is only done before transmitting, whereas ContikiMAC also does CA in between strobed frames. Neither does this issue arise in POTR since POTR simply does not increment the frame counter when retransmitting broadcast frames, causing duplicated frames to be considered as replayed and hence rejected. To solve this issue, ILOS requires ContikiMAC to only do CA before strobing a broadcast frame. While this only offers intra-network CA, it has the security benefit that jamming during broadcast transmissions no longer induces retransmissions, fixing an open denial-of-sleep vulnerability.

3.5 Adapting the Adaptive Key Establishment Scheme (AKES)

Let A and B be adjacent nodes. To initialize $\omega^*_{B,A}$ and $\omega^*_{A,B}$ while A and B establish session keys, ILOS adds additional data to `ACK`s, as well as to acknowledgement frames that are sent in response to `ACK`s, as shown in Fig. 2. Specifically, to initialize $\omega^*_{B,A}$, A reports back on its wake-up counter at time of receiving B's `HELLOACK` since SPLO initializes $t^*_{B,A}$ to the time when A woke up for receiving B's `HELLOACK`. Likewise, B reports back on its wake-up counter when receiving A's `ACK` since SPLO initializes $t^*_{A,B}$ to when B woke up for receiving A's `ACK`.

In addition, ILOS applies two tweaks to AKES' transmission of `UPDATE`s. First, rather than sending dedicated `UPDATEACK`s, ILOS relies on SPLO's authenticated acknowledgement frames, which are sent in response to `UPDATE`s anyway. Second, it is also somewhat redundant that SPLO stores the last known wake-up time of each neighboring node while AKES additionally keeps track of each neighbor's expiration time. Hence, `ILOS` solely relies on SPLO to schedule the transmission of `UPDATE`s, thereby freeing AKES from storing expiration times.

4 Implementation

Our implementation of ILOS advances Krentz et al.'s implementation of 802.15.4 security, AKES, POTR, ContikiMAC, as well as SPLO for the Contiki operating system [15–17]. We preserved their design and inserted the changes of ILOS surrounded by conditional preprocessor directives. This enables us to switch between frame counters and wake-up counters at compilation time. Within our conditional code, we only let information flow downwards, following the terminology introduced in [21]. That is, upper layers retrieve additional information, such as the current wake-up counter, from lower layers. Nevertheless, ILOS is not a real cross-layer optimization because ILOS only affects the MAC layer.

5 Evaluation

Below, we (i) argue that ILOS achieves strong freshness, (ii) quantify the reduction of the security-related per-frame overhead due to ILOS, (iii) demonstrate the resulting reduction in energy consumption, (iv) show that ILOS accelerates POTR's rejection speed, and (v) give insight to the RAM footprint of ILOS.

5.1 Freshness Guarantees

Recall that strong freshness is provided if a receiver can ensure that an incoming 802.15.4 frame was sent within a limited time span prior to its reception [20]. In the case of unicast data and command frames, an upper bound that ILOS achieves is t_w. This is because, if a unicast data or command frame is delayed by $\geq t_w$, receivers will use a different CCM nonce to verify its MIC, which results in the rejection of the frame. In the case of acknowledgement frames, this upper bound is even lower because an acknowledgement frame is only considered timely if it belongs to the unicast frame that was just strobed. Additionally, SPLO only accepts acknowledgement frames within a short window after sending a unicast frame. This way, SPLO ensures that acknowledgement frames are not delayed by more than 0.122ms by default [16]. In the case of broadcast frames, ILOS achieves an upper bound of $2t_w$ since receivers will definitely assume a different CCM nonce when a broadcast frame is delayed by $\geq 2t_w$. On the other hand, delayed `HELLOACK`s and `ACK`s get accepted, but session key establishment ultimately fails as a result since SPLO aborts session key establishment if `HELLOACK`s and `ACK`s are not being acknowledged in a timely manner [16].

5.2 Security-Related Per-Frame Overhead

Table 3 compares the security-related per-frame overhead of various frame formats. According to the original frame format of 802.15.4, a secured 802.15.4 frame comprises a 1-byte Security Control field, a 4-byte Frame Counter field, an optional 1-byte Key Index field, an optional Key Source field of up to 8 bytes, as well as an m-bit CCM-MIC [1]. In TSCH networks, the Frame Counter field

Table 3. Security-Related Per-Frame Overhead

Frame Format	Overhead (in bytes)
802.15.4 [1]	$[5, 13] + \frac{m}{8}$
TSCH [1]	$[1, 9] + \frac{m}{8}$
802.15.4+AKES+LB [15]	$1 + \frac{m}{8}$
POTR+AKES [17]	$[4, 5] + \frac{l+m}{8}$
POTR+AKES+LB [17]	$[1, 2] + \frac{l+m}{8}$
POTR+AKES+SPLO+LB [16]	$[1, 3] + \frac{l+m}{8}$
POTR+AKES+SPLO+ILOS	$[0, 2] + \frac{l+m}{8}$

becomes unnecessary, as described in the introduction [1]. In ContikiMAC networks, using AKES obviates the need for the Key Index, as well as the Key Source field [15]. In addition, the LB optimization reduces the overhead of frame counters to 1 byte. On the other hand, POTR requires adding an l-bit OTP field to non-acknowledgement frames and a 1-byte sequence number to unicast frames. Moreover, SPLO requires adding the 1-byte Strobe Index field to unicast frames. ILOS reduces the security-related per-frame overhead again by dispensing with frame counters altogether.

5.3 Energy Efficiency

To demonstrate that ILOS reduces the energy consumption of sending unicast data frames, the following experiment was conducted. An OpenMote A sent a unicast data frame with 50 bytes of payload to another OpenMote B [22]. A and B were recently synchronized so that A only strobed twice. While A strobed and subsequently received B's acknowledgement frame, the current draw of A was measured by connecting A, a μCurrent Gold, and a Rigol DS1000E oscilloscope in series, as is further detailed in [22]. The current draw over time was then converted into the actual energy consumption under the assumption of a constant supply voltage of 3V. This was repeated using three different configurations, namely (i) with the LB optimization, as well as ILOS disabled, (ii) with the LB optimization enabled, and (iii) with ILOS enabled. For each of the three configurations, 100 samples were obtained. POTR, SPLO, and 8-byte addresses were used throughout.

Fig. 4a depicts the results as boxplots. Expectably, the most energy-consuming configuration is to use neither the LB optimization nor ILOS. This is because, in this configuration, frame counters are transmitted uncompressed. Enabling the LB optimization saves energy since the security-related per-frame overhead decreases by 3 bytes. Another byte can be saved by enabling ILOS instead, yielding a slightly lower energy consumption compared to using the LB optimization.

To also demonstrate that ILOS reduces the energy consumption of receiving unicast data frames, the above experiment was repeated with two differences.

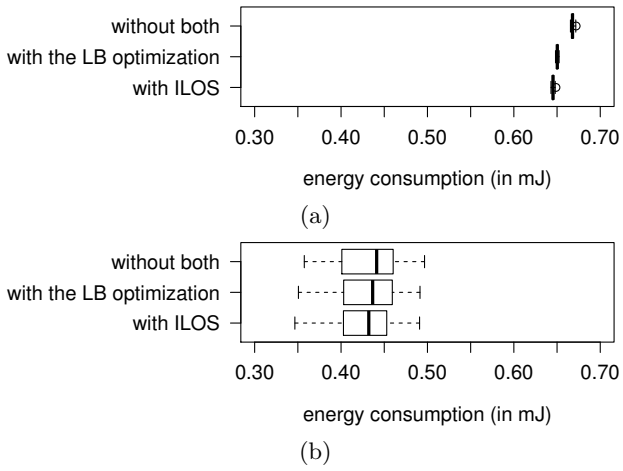

Fig. 4. Energy consumption per (a) transmission and (b) reception of a unicast data frame with 50 bytes of payload

First, B's energy consumption while receiving A's unicast data frames and acknowledging them was measured. Second, to avoid bias, B woke up at randomized times. Throughout, the dozing optimization for ContikiMAC was switched on [16].

Fig. 4b shows the results. This time, the variation in the data is much higher because the energy consumption per frame reception highly depends on how long a receiver waits until the next unicast frame is being strobed. Apart from that, the results are similar. While ILOS constitutes the most energy-efficient configuration, using neither the LB optimization nor ILOS constitutes the least energy-efficient configuration.

The above results apply to unicast command and data frames alike since they are treated equally. Broadcast receptions should also consume less energy when using the LB optimization or ILOS. Broadcast transmissions, by contrast, will not become more energy efficient since ContikiMAC strobes broadcast frames for a whole wake-up interval anyway. Additionally, we note that ILOS may also reduce the frequency of fragmentation, which then saves further energy.

5.4 Rejection Speed

To compare the rejection speed of POTR in different configurations, the following experiment was conducted. An OpenMote A sent a unicast data frame with an invalid OTP to another OpenMote B [22]. Upon receipt, B stopped the time between detecting the frame's SHR and the rejection of the frame. This was repeated using three different configurations, namely (i) with the LB optimization, as well as ILOS disabled, (ii) with the LB optimization enabled, and (iii) with ILOS enabled. For each of the three configurations, 100 samples were obtained. Throughout, 8-byte addresses were used.

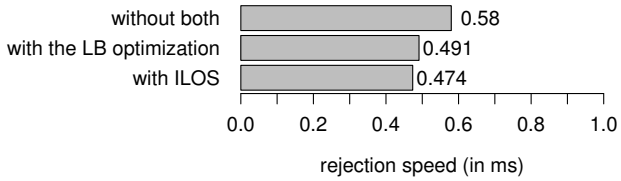

Fig. 5. Mean rejection speed of POTR

Fig. 5 shows that the LB optimization accelerates POTR's rejection speed noticeably. This was expected, as explained in Section 2. ILOS, by comparison, accelerates POTR's rejection speed a bit more since the validation of OTPs begins even earlier. In effect, ILOS further reduces the time spent in receive mode under ding-dong ditching.

5.5 RAM Footprint

To measure the RAM footprint, the tool `arm-none-eabi-size` was used. As a baseline for comparison, the RAM footprint when disabling 802.15.4 security altogether was measured. Based on this measure, the overhead in RAM was then determined when (i) using neither the LB optimization nor ILOS, (ii) enabling the LB optimization, and (iii) enabling ILOS. Also, the RAM footprint was determined when configuring the Contiki operating system to use 0, 5, 10, 15, 20, or 25 neighbor slots.

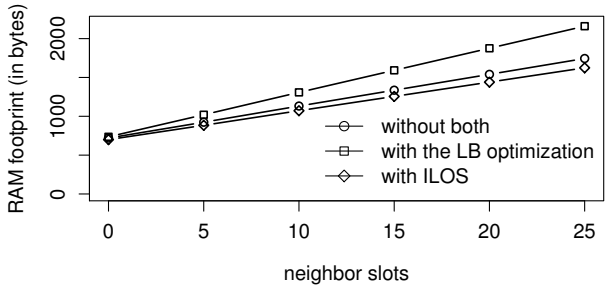

Fig. 6. RAM footprint

Fig. 6 shows the results. Surprisingly, ILOS consumes even less RAM than if using neither the LB optimization nor ILOS. There are two main reasons for this. First, ILOS reuses the wake-up times that are stored by SPLO anyway. Second, ILOS frees AKES from storing expiration times. Conversely, the LB optimization incurs a high RAM overhead, as conjectured in the introduction. This is problematic since 802.15.4 nodes usually have just a few kilobytes of RAM.

6 Conclusions and Future Work

Using frame counters incurs drawbacks in terms of energy efficiency and freshness guarantees. TSCH avoids these drawbacks by replacing frame counters with timeslot indices. However, TSCH's mechanisms for time synchronization are vulnerable to a range of attacks. ContikiMAC, on the other hand, avoids many of TSCH's vulnerabilities in the first place since ContikiMAC works asynchronously. Yet, as far as ContikiMAC is concerned, only the LB optimization is currently available for alleviating the drawbacks of frame counters. To address the limitations of the LB optimization, we have proposed ILOS. According to our evaluation, ILOS outperforms the LB optimization in terms of security-related per-frame overhead, energy efficiency, rejection speed, as well as RAM footprint. Additionally, ILOS has three major advantages over the LB optimization. First, ILOS achieves strong freshness. Second, ILOS avoids resynchronization actions. Third, ILOS simplifies AKES. The only drawback of ILOS seems to be that ILOS intertwines ContikiMAC and 802.15.4 security. From a software engineer's perspective, we would actually like to decouple the implementation of 802.15.4 security and ContikiMAC so that we can change one without affecting the other. Future work may generalize ILOS to other MAC protocols or formally assess the correctness of ILOS, e.g., with a protocol verification tool.

References

1. IEEE Standard 802.15.4 (2015)
2. Al Nahas, B., Duquennoy, S., Iyer, V., Voigt, T.: Low-power listening goes multi-channel. In: Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS). pp. 2–9. IEEE (2014)
3. Aljareh, S., Kavoukis, A.: Efficient time synchronized one-time password scheme to provide secure wake-up authentication on wireless sensor networks. International Journal of Advanced Smart Sensor Network Systems (IJASSN) 3 (2013)
4. Brownfield, M., Gupta, Y., Davis, N.: Wireless sensor network denial of sleep attack. In: Proceedings of the Sixth Annual IEEE SMC Information Assurance Workshop (IAW '05). pp. 356–364. IEEE (2005)
5. Capossele, A.T., Cervo, V., Petrioli, C., Spenza, D.: Counteracting denial-of-sleep attacks in wake-up-based sensing systems. In: Proceedings of the IEEE International Conference on Sensing, Communication and Networking (SECON 2016). pp. 1–9. IEEE (2016)
6. Dunkels, A.: The ContikiMAC radio duty cycling protocol. Tech. Rep. T2011:13, Swedish Institute of Computer Science (2011)
7. Duquennoy, S., Landsiedel, O., Voigt, T.: Let the tree bloom: scalable opportunistic routing with ORPL. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13). pp. 2:1–2:14. ACM (2013)
8. Duquennoy, S., Österlind, F., Dunkels, A.: Lossy links, low power, high throughput. In: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11). pp. 12–25. ACM (2011)
9. Falk, R., Hof, H.J.: Fighting insomnia: a secure wake-up scheme for wireless sensor networks. In: Proceedings of the Third International Conference on Emerging

Security Information, Systems and Technologies (SECURWARE '09). pp. 191–196 (2009)
10. Gouda, M.G., ri Choi, Y., Arora, A.: Antireplay protocols for sensor networks. In: Wu, J. (ed.) Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, pp. 561–574. CRC (2005)
11. He, Z., Voigt, T.: Droplet: a new denial-of-service attack on low power wireless sensor networks. In: Proceedings of the 2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2013). pp. 542–550. IEEE (2013)
12. Hsueh, C.T., Wen, C.Y., Ouyang, Y.C.: A secure scheme against power exhausting attacks in hierarchical wireless sensor networks. IEEE Sensors Journal 15(6), 3590–3602 (2015)
13. Huang, P., Xiao, L., Soltani, S., Mutka, M., Xi, N.: The evolution of MAC protocols in wireless sensor networks: a survey. IEEE Communications Surveys Tutorials 15(1), 101–120 (2013)
14. Hui, J., Thubert, P.: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282 (2011), updates RFC 4944
15. Krentz, K.F., Meinel, Ch.: Handling reboots and mobility in 802.15.4 security. In: Proceedings of the 31st Annual Computer Security Applications Conference (ACSAC '15). pp. 121–130. ACM (2015)
16. Krentz, K.F., Meinel, Ch., Graupner, H.: Countering three denial-of-sleep attacks on ContikiMAC. In: Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN 2017). Junction (2017)
17. Krentz, K.F., Meinel, Ch., Schnjakin, M.: POTR: practical on-the-fly rejection of injected and replayed 802.15.4 frames. In: Proceedings of the International Conference on Availability, Reliability and Security (ARES 2016). IEEE (2016)
18. Luk, M., Mezzour, G., Perrig, A., Gligor, V.: MiniSec: a secure sensor network communication architecture. In: Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07). pp. 479–488. ACM (2007)
19. Michel, M., Voigt, T., Mottola, L., Tsiftes, N., Quoitin, B.: Predictable MAC-level performance in low-power wireless under interference. In: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN '16). pp. 13–22. Junction (2016)
20. Raymond, D.R., Marchany, R.C., Midkiff, S.F.: Scalable, cluster-based anti-replay protection for wireless sensor networks. In: Proceedings of the IEEE SMC Information Assurance and Security Workshop (IAW '07). pp. 127–134. IEEE (2007)
21. Srivastava, V., Motani, M.: Cross-layer design: a survey and the road ahead. IEEE Communications Magazine 43(12), 112–119 (2005)
22. Vilajosana, X., Tuset, P., Watteyne, T., Pister, K.: OpenMote: open-source prototyping platform for the industrial IoT. In: Ad Hoc Networks. vol. 155, pp. 211–222. Springer (2015)
23. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM). RFC 3610 (2003)
24. Wood, A., Stankovic, J., Zhou, G.: DEEJAM: defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks. In: Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07). pp. 60–69. IEEE (2007)
25. Yang, W., Wang, Q., Qi, Y., Sun, S.: Time synchronization attacks in IEEE802.15.4e networks. In: Proceedings of the International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI 2014). pp. 166–169. IEEE (2014)