R-TSCH: Proactive Jamming Attack Protection for IEEE 802.15.4-TSCH Networks

Dimitrios Zorbas[1], Panayiotis Kotzanikolaou[1,2], and Christos Douligeris[1,2]
[[1]NetLab, [2]SecLab], Department of Informatics, University of Piraeus, Greece
{dzorbas, pkotzani, cdoulig}@unipi.gr

***Abstract*—Time Slotted Channel Hopping (TSCH) has been proposed in various wireless protocols as a solution to combat external interference, path-loss fading and static jamming attacks. However, since TSCH algorithms generate a deterministic and periodic pattern of channel hops, they are still subject to jamming attacks. Proactive randomization of the channel generation process could provide a good solution against jamming attacks, however due to the strict time constraints of the timeslots, practical solutions should be very efficient. In this paper, we propose R-TSCH, a randomized radio channel generation algorithm that can be used to *proactively* protect wireless nodes from jamming attacks. Based on a cryptographic hash function and a secret key, R-TSCH produces a new pseudo-random channel sequence, which looks as truly random to anyone who has no access to the key. Our simulation results show that the attacked links of the TSCH network enhanced with the proposed mechanism can achieve an over 90% Packet Reception Rate (PRR) in presence of multiple jammers.**

I. Introduction

The Time Slotted Channel Hopping technique has been adopted by many Internet of Things (IoT) protocols such as the WirelessHART [1], the ISA100.11a [2], and the IEEE 802.15.4-TSCH [3]. Channel hopping has been introduced to cope with external interferences and path-loss effects since successive transmissions are carried out using different radio channels. Time slotted communications aim to facilitate the access to a shared medium by allowing the nodes of the network to transmit data one after the other each using its own timeslot. The combination of time division with channel hopping communications reduces the energy consumption of the nodes and can achieve extremely high packet delivery ratios [4].

A TSCH channel generation formula produces a different radio channel for each communicating node-pair, at each new timeslot. Thus, TSCH-based protocols provide enhanced protection against “static” jamming attacks, i.e. attacks that constantly jam the same channel. However, channel generation in TSCH is periodic, which means that the channel hopping sequence (CHS) follows a specific repeated behavior (pattern). This deterministic CHS pattern can be easily surmised or learned by an attacker equipped with a cognitive radio. Thus, TSCH networks are still vulnerable to special jamming attacks, which in turn can be combined with other types of attacks such as eavesdropping and injection attacks [5]. Current TSCH standards do not incorporate any security mechanism against cognitive jamming attacks to protect the availability of the communications. Although they propose cryptographic techniques such as end-to-end encryption, these can only protect the confidentiality and the integrity of the communication and cannot mitigate jamming attacks.

In this paper, we propose Randomized Time Slotted Channel Hopping (R-TSCH), a local radio channel generation algorithm to protect the IEEE 802.15.4-TSCH networks from dynamic jamming attacks. In contrast to older solutions that apply channel hoping reactively, *i.e.* nodes change their channel only if they experience interference or jamming, our algorithm is proactive; it continuously applies channel hopping proactively, to prevent from selectively jamming attacks against targeted nodes. The proposed approach changes the default CHS with one that follows a random and unpredictable channel sequence. Our security mechanism can work above any known scheduling approach (distributed or centralized) found in the literature. Obviously efficiency is also very important, given the strict time constraints.

The rest of the paper is organized as follows. In Section II we briefly review the related work. In Section III we describe R-TSCH and we analyze the randomized channel generation process. In Section IV we evaluate the efficiency of our algorithm in terms of packet reception rate. Finally Section V concludes this paper and presents ideas for future enhancements.

II. Related work

In [6], Nevda *et al.* propose a mechanism that protects 802.11 networks from jamming attacks, using (pseudo)randomized channel hopping. To achieve this, the legitimate nodes are able to generate and exchange a pseudo-random channel sequence, using some cryptographic randomization technique. For any outsider having no access to some secret cryptographic keying material (e.g. a seed), the channel sequence looks as truly random. Since no mechanism can protect from an adversary that constantly floods all the channels, while at the same time flooding attacks require specialized hardware and are generally easy to detect, such attacks are not considered in [6]. Thus, an adversary can follow no better strategy that randomly jamming one channel per slot. We will also follow a similar channel hopping randomization technique but in 802.15.4-based networks.

Wood *et al.* present DEEJAM [7], a MAC-layer protocol that aims to defeat jamming attacks in 802.15.4, based on four defensive mechanisms to hide the communication from a jammer, evade its search and reduce its impact. DEEJAM is examined against various jamming attacks such as interrupt jamming, activity jamming, scan jamming and pulse jamming. The protocol of [7] uses four complementary solutions. The first one is *frame masking*; the sender and the receiver use a secret pseudo-random sequence to agree on the Start Frame Delimeter (SFD) of each packet. For *channel hopping*, the parties use a shared channel key which is used to generate a pseudo-random channel sequence. In addition, *packet fragmentation* and *redundant encoding* are also applied. DEEJAM provides complementary anti-jamming protection techniques and achieves a very high packet delivery ratio, up to 88% in the presence of a pulse jammer, according to simulation results.

Note that solutions like [6], [7] are designed for protocols that use "static" channel assignment; nodes are assigned to a single channel and channel hoping is used only if the nodes experience interference and jamming attacks during their communication. In such cases, the channel hopping algorithm is not subject to hard time constraints. A similar argument holds for algorithms based on Frequency Hopping Spread Spectrum, such as [8], [9].

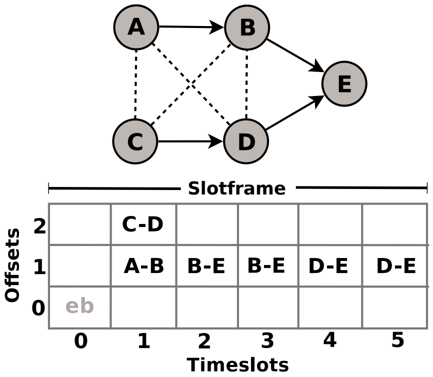
In contrast, in Time Slotted Channel Hopping protocols (as it is the case in our algorithm), nodes constantly change their communication channel at each new timeslot. Since a timeslot takes 10 ms and most of the time is spent on transmission and acknowledgement reception (about 6 ms) [10], the channel generation process must be completed in less that 3 or 4 ms. Thus the algorithm must be efficient enough for the legitimate nodes since a non efficient algorithm will not be practical. Finally, although our solution only applies one of the four techniques that are combined in [7], it is possible to combine our algorithm with techniques such as frame masking, packet fragmentation and redundant encoding.

Chang *et al.* [11] propose Tri-CH, a jamming resistance channel hopping scheme for cognitive radio networks. Tri-CH adopts a random channel pattern and a reception stay mode (i.e. nodes stay at a channel for receiving packets only). The main goal of the protocol is to avoid interference between communications of unlicensed users, e.g., those that use idle licensed channels, with the communications of licensed users that use primary channels. Thus the protocol cannot be directly applied to TSCH-based protocols. One of the most interesting features of Tri-CH is that it does not require from the nodes to have pre-shared secret keys.

III. R-TSCH: An algorithm for randomized Channel generation

A. TSCH preliminaries

Du and Roussos presented a technique for adaptive Time Slotted Channel Hopping [12]. In TSCH-adopted protocols the time is divided into slotframes of equal length. Each slotframe consists of a number of timeslots of equal size. At each timeslot, a node may transmit a frame, receive a frame, or turn to sleep mode to save energy. A node may participate in multiple timeslots per slotframe depending on the number of generated packets and its position in the network. Each timeslot is labeled with an Absolute Sequence Number (ASN), which counts the number of timeslots since the network was established. Moreover, a single frame transmission is allowed per timeslot, followed by an acknowledgement of reception.



Offsets						
2		C-D				
1		A-B	B-E	B-E	D-E	D-E
0	eb					
	0	1	2	3	4	5

Fig. 1. A TSCH-based scheduler: A-B stands for *'A transmits to B'*, while a shared cell is used for advertisement (eb). Dashed lines represent physical neighboring nodes.

A TSCH scheduler is responsible to organize the transmissions and to assign one or more channel offsets per node per link. A TSCH scheduling example is presented in Fig. 1. The first timeslot is used for advertisement purposes, while the rest slots are dedicated to data transmissions. Slot 1 includes two parallel transmissions with different channel offsets to avoid collisions.

As far as the routes to the sink have been established and a schedule has been built, the data communication process can begin. At the beginning of a timeslot, a pair of nodes wakes up and generates a physical radio channel as follows:

$$CH = \texttt{map}\left[ASN + OFFSET\right] \ \texttt{mod} \ nFreq, \qquad (1)$$

where $OFFSET$ is the channel offset of the current cell (assigned by the scheduler), $nFreq$ is the number of available channels and $\texttt{map}()$ a bijective function mapping an integer ranging between 1 and $nFreq$ into a physical channel [10].

B. Attacks on TSCH-based scheduling

TSCH techniques provide a level of defense against jamming attacks by an attacker that constantly jams the same channel, since the nodes proactively change their communication channel at each timeslot. However, since in TSCH the channel generation is deterministic and channel hopping follows a periodic pattern, they are still subject to jamming attacks by nodes equipped with cognitive radios. A sophisticated attacker may use spectrum sensing techniques to identify the channel hop pattern. If the hop pattern is revealed, the attacker will "proactively" jump to the next channel (as the legitimate nodes will) and will eventually succeed to continuously jam the communications of a target pair. Note that the same holds not only for jamming attacks, but also for other attacks that may target to eavesdrop or to alter the communications. All these attacks require that the adversary is able to continuously identify the slot being used by a specific target.

1) Assumptions and attack model: We assume IEEE 802.15.4 communication medium, which provides 16 non-interfering radio channels, separated by 5 MHz each, that are available for dynamic selection by software. We assume that legitimate nodes are pre-deployed (or can efficiently exchange) a secret key, say K. Key exchange is out of the scope of the proposed algorithm, since the proposed algorithm may use either pre-deployed or dynamically generated keys. A simple solution would be to pre-deploy the secret key to the legitimate nodes. A more dynamic solution would require a secure key bootstrapping phase, that is executed once for each new node. Finally, we assume a time synchronization service, which is necessary for channel hopping.

A jamming attack is a Denial of Service attack at the physical layer, during which the attacker intentionally interferes to one or more target channel(s), in order to disrupt the legitimate communication at the target channel(s).

Our threat model involves active jamming attacks triggered by adversaries having similar equipment, communication and computation capabilities with the legitimate nodes. We assume that the channel hopping protocol is known to the adversary. We also assume an adversary that is capable to jam some, but not all the communication channels at the same time. The goal of the adversary is to continuously jam targeted communicating pair(s), by continuously jamming their selected channel(s) at each slot. At the same time, the goal of the attacker is to avoid detection of the attack by the legitimate nodes.

We exclude from our threat model more powerful attacks against *all* the communication channels, similarly to [6]. Although it is indeed feasible for a more powerful attacker with specialized radio equipment to concurrently jam all 16 channels, such attacks are easy to detect and thus to identify the jamming node. In addition, such attacks are power consuming.

C. Randomized channel generation process

The proposed radio channel generation process is essentially a randomized proactive channel hopping algorithm, and it is described in Algorithm 1. The algorithm replaces the execution of Eq. (1) and it is executed in a per pair basis for a given ASN. We assume that a channel offset (or multiple channel offsets) has already been assigned by a scheduler. It also requires that the secret key K is known to all the legitimate nodes in the network. We denote as $OFFSET_j$ the offset assigned by the scheduler to a pair of communicating nodes j. Note that both nodes of the pair need to execute the algorithm using the same ASN and channel offset. The algorithm will generate the same radio channel for both nodes.

Let $Hash$ be a cryptographic hash function (such as SHA-2). Let $msb_i(X)$ denote a function that outputs the i most significant bits of the input sting X[1]. Let $bin(X)$ denote a function that outputs the binary representation of X. Finally, for ease of reading, we use small letters with indexes to denote the binary bit at the indexed position and we use the symbol '|' as a bit delimiter.

Algorithm 1: R-TSCH – A randomized channel generation algorithm to protect nodes from jamming attacks

```
  require: $ASN$; $K$ secret key; $OFFSET_j$ unique 4-bit
           channel offset for pair $j$
1 $X_{ASN} = Hash(ASN, K)$;
2 $x_0|x_1|x_2|x_3 = msb_4(X_{ASN})$;
3 $OFFSET_j{}' = (OFFSET_j + ASN)\%16$;
4 $b_0|b_1|b_2|b_3 = bin(OFFSET_j{}')$ ;    // decimal to
    binary conversion
5 $CH_j = x_0|x_1|x_2|x_3$ XOR $b_0|b_1|b_2|b_3$;
6 return $CH_j$;
```

The algorithm hashes the concatenation of the current ASN with the secret key K and then outputs the 4 most significant bits of the hash value ($x_0|x_1|x_2|x_3$). Then, the offset $OFFSET_j$ assigned to each pair running the algorithm (say pair j) is added to the current ASN and a modulo 16 operation is performed, to output a 4-bit string $OFFSET_j{}'$ (recall that the maxium number of pairs allowed at each slot in a neighborhood is 16 pairs). Let $b_0|b_1|b_2|b_3$ be the binary representation of $OFFSET_j{}'$. Then the pair j will compute its unique random channel number for the current slot (ASN), by XOR-ing $x_0|x_1|x_2|x_3$ (which is common for all pairs at a given slot) with the unique bitstring $b_0|b_1|b_2|b_3$, to produce a unique channel number CH_j.

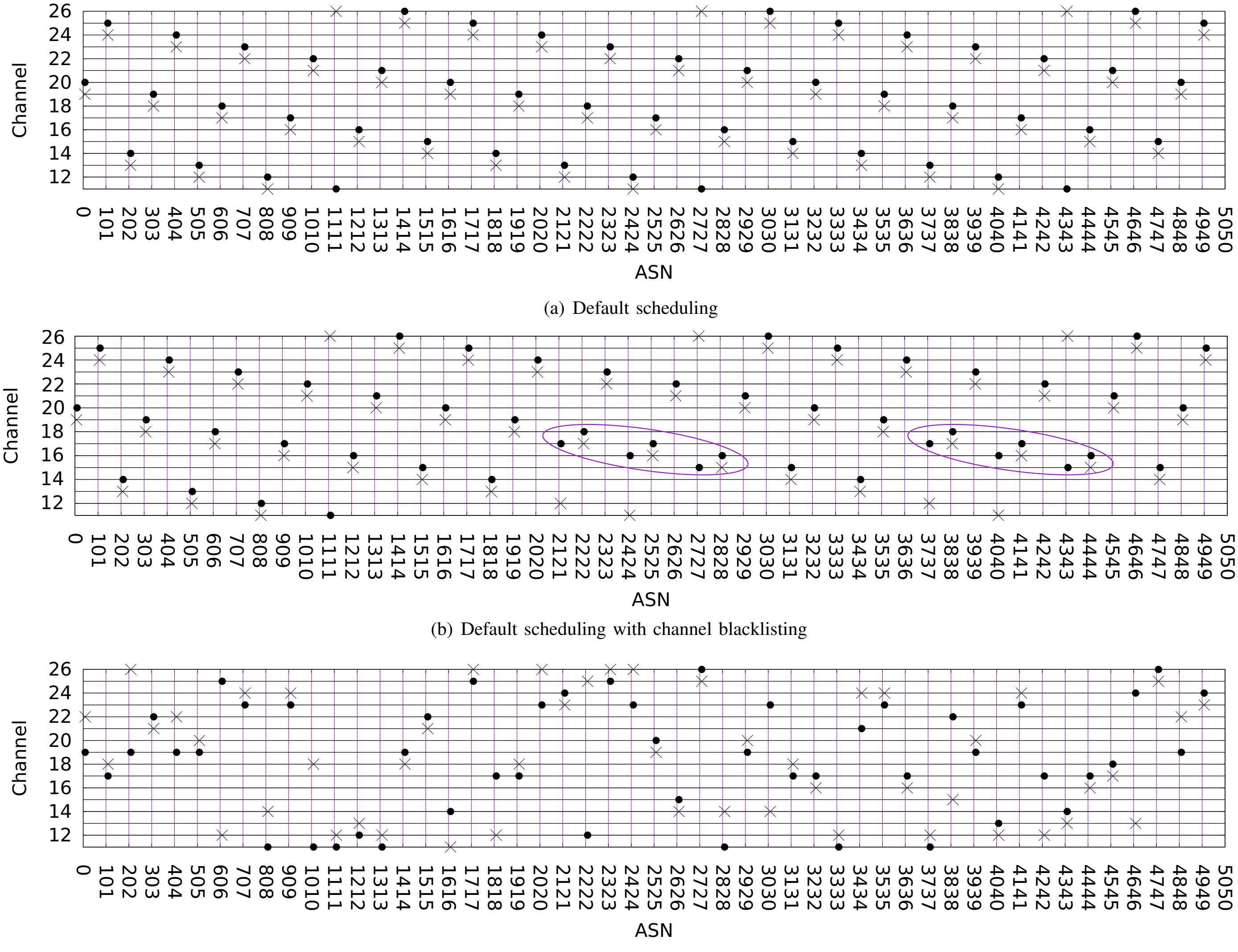
D. Security analysis

It is easy to see that the CHS produced by Algorithm 1 is pattern-free and non-repeated. $Hash$ is a cryptographically secure hash function (such as SHA-2) and the secret key K has sufficient entropy (say 128 bit length). Under these conditions, $Hash(ASN, K)$ is essentially used as a keyed cryptographic hash function. Since a keyed cryptographic hash function produces a pseudo-random output that looks as truly random for anyone not having the secret input, the output X_{ASN} (and therefore $x_0|x_1|x_2|x_3$) will be random bit strings. The channel number for each pair is produced in each timeslot, by XOR-ing the random string $x_0|x_1|x_2|x_3$ with the bitstring $b_0|b_1|b_2|b_3$ that is deterministically produced based on $OFFSET_j$. Since one of the strings is randomly produced, it holds that CH_j produced in line 5 will be a random 4-bit integer and thus the channel hopping sequence is pattern-free. Our experimental analysis of the channel hopping sequence presented in Section IV-B verify this property.

Moreover the CHS cannot be computed by an outsider. Provided that the key K remains secret, an outsider has only negligible success probability, other than random guessing, to find X_{ASN} and therefore $x_0|x_1|x_2|x_3$.

Finally, it is easy to see that Algorithm 1 is collision-free. Since each pair j has been assigned by the scheduler to a unique integer $OFFSET_j$ in the range $[0, 15]$, then the value $OFFSET_j{}'$ produced at line 3 of the algorithm will also be a unique integer in the same range. Since at each timeslot $x_0|x_1|x_2|x_3$ is common for all node pairs, each node pair

[1] If X is not binary, it is first converted to its binary representation



(a) Default scheduling

(b) Default scheduling with channel blacklisting

(c) Scheduling with R-TSCH

Fig. 2. Radio channel scheduling for two parallel transmissions: With the default scheduling (cases a and b) radio channels follow a deterministic pattern, in contrast to R-TSCH (case c) that is pattern-free. The advertisement slots have been omitted.

computes a unique bitstring $b_0|b_1|b_2|b_3$, and CH_j is computed as shown in line 5, it holds that $CH_j \neq CH_i,\ \forall\ i \neq j$. Collision freeness is also a property that is verified by the simulation results.

E. Computation cost & energy consumption

R-TSCH computation and energy cost mainly comes from the computation of the hash function. This computation may be slow but most of the modern IoT platforms are equipped with a SHA-2 Hardware Encryption Engine which accelerates the computation of the hash function. For example, the Texas Instruments CC2538 ARM Cortex-M3 chip can lower the cost of a SHA-2 digest down to 0.45ms [13]. This time is enough to carry out more than one SHA-2 computation during a timeslot. Moreover, the energy cost has been experimentally computed at $12.27\mu J$ [13] which is less than 1% of the total energy cost of a transmitting node during a timeslot[2].

F. Radio channel blacklisting

In channel blacklisting, specific radio channels can be excluded from the channel selection pool if they repeatedly present a *bad* behavior (e.g., if their PRR falls below 0.9 within some slotframes). Since the external interference affects some specific channels, the nodes can use blacklisting to further increase the reliability of some links. R-TSCH can be combined with any local blacklisting method such as the one proposed by Gomes *et al.* [14].

IV. Evaluation & discussion of the results

A. Setup

In this section, we evaluate the proposed mechanism by conducting a set of Monte Carlo simulations. We compare R-

[2] we consider Zolertia RE-Mote Revision B hardware: https://github.com/Zolertia/Resources/wiki/RE-Mote

TSCH to the default IEEE 802.15.4-TSCH channel generation process. We use the LOST algorithm [15] to assign timeslots and channel offsets to the links. However, any other scheduling algorithm may also be used. We consider a scenario with 50 nodes randomly scattered on a square terrain of 50×50 m^2 size and a communication range of $10m$. Each node generates one packet per slotframe. We vary the number of jammers from 1 to 10 and we measure the average packet delivery ratio over 25 different topologies. The simulation time is set to 100 slotframes. We assume that the attacker is located close to the targeting link and he can successfully jam transmissions with a random probability between 0.85 to 0.95. We assume that at the beginning of the simulation, the attacher has already learned the default CHS. In the case of R-TSCH, a jammer attacks a random radio channel.

B. Channel hopping sequence

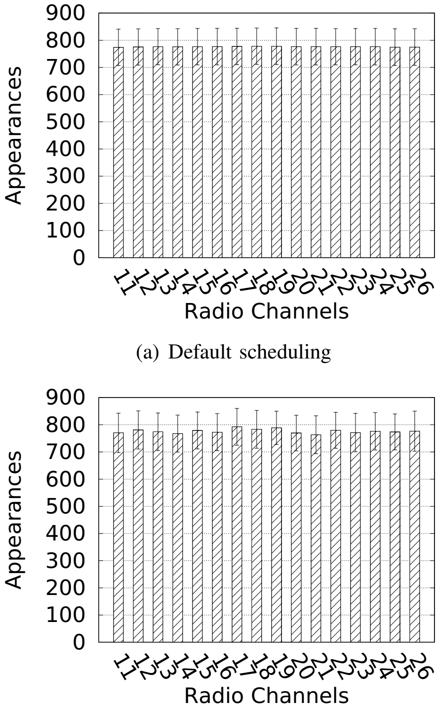
Fig. 2 depicts the CHS using (a) the traditional channel generation formula of Eq. (1), (b) the traditional channel generation formula enhanced with a blacklisting mechanism (we will talk about this in the next subsection), and (c) the proposed security mechanism. Two arbitrary parallel transmissions were used in this figure, denoted with a "×" and a "•" respectively. We can observe that the default scheduling follows a specific pattern where every transmission has a 5-channel difference with the previous transmission. In the second subfigure, we manually blacklist radio channels 11, 12, and 13 for the second pair. The CHS is modified when the frequency generation process generates one of the blacklisted channels. Finally, using the proposed channel generation algorithm, the CHS follows a random non-trackable and non-repeated pattern without causing collisions between the two transmissions.

Fig. 3 presents the number of appearances per radio channel. We can observe that R-TSCH does not promote the selection of any specific channel since all the radio channels have similar number of appearances. As it was expected the default generation process exhibits equal number of appearances per channel.

C. Packet delivery ratio

Fig. 4a illustrates the overall packet delivery ratio considering various number of jammers. Each jammer targets a specific radio link throughout the experiment. However, since parallel transmissions are carried out, more than one link may be affected at the same timeslot. We must mention here that only lost packets due to the jamming attacks are taken into account. We can observe that the PDR decreases for both approaches as the number of jammers increases. The default TSCH behavior presents a high PDR drop as more jammers are placed in the network. This happens due to the low packet reception rate (PRR) of the attacked links as we can see from Figure 4b.

As it is depicted in Figure 4c,d the performance of the attacked links can be improved by considering channel blacklisting. The simulation results capture a considerable improvement of the overall PDR and the PRR of the attacked links.



(a) Default scheduling

(b) Scheduling with R-TSCH

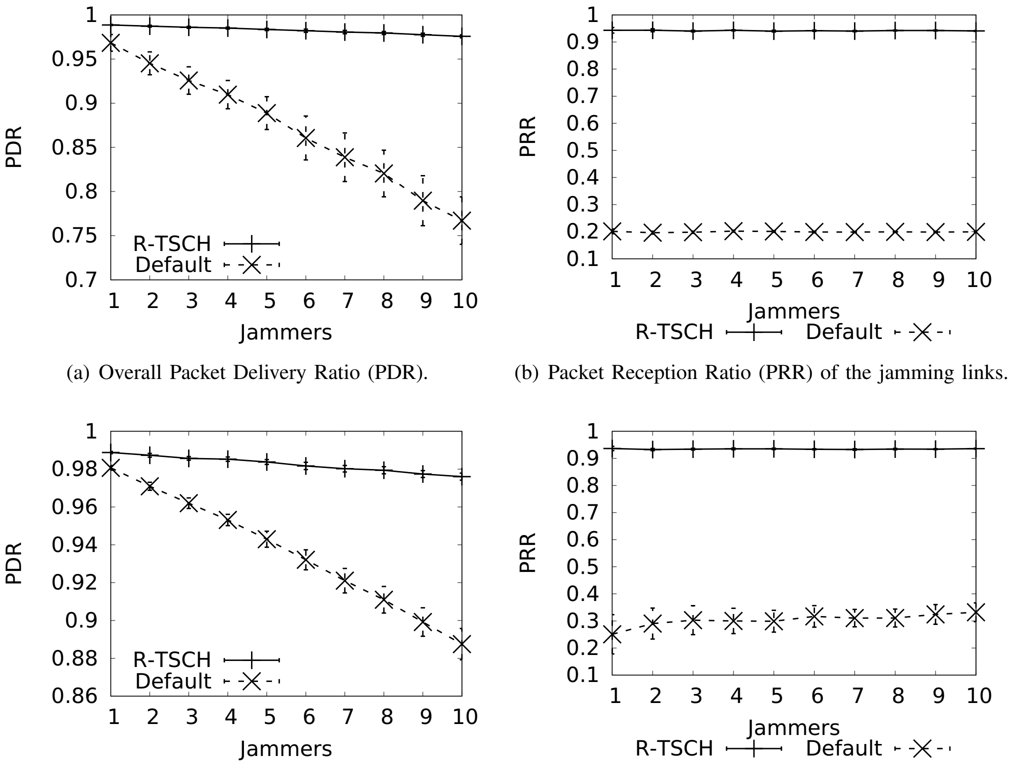
Fig. 3. Number of appearances per radio channel using the default generation process and R-TSCH.

In these figures we assume that the attacker is not aware of the *new* CHS. However, as we can see from Fig. 2b even in the case of blacklisting, the new CHS still follows a predictable pattern. Thus, an attacker could easily adapt its behavior according to the new CHS after some slotframes.

V. Conclusion & Future work

In this paper we dealt with the problem of proactively preventing or mitigating packet collisions caused by the malicious activity of jammers in IEEE802.15.4-TSCH networks. We consider the case where the attacker can learn the default CHS of specific links and, thus, jam the corresponding transmissions. We propose R-TSCH; a new pseudo-randomized channel generation algorithm based on a known to the network-nodes key, a hash function, and an offset assigned by the scheduler. R-TSCH is capable of generating collision-free radio channels as well as non-trackable and non-repeatable channel hopping sequences. We compared R-TSCH to the default TSCH approach, and the evaluation results showed a huge improvement of the packet reception rate for the attacked links. As a consequence, we recorded a considerable improvement for the overall packet delivery ratio, especially when multiple jammers are placed in the network.

As explained in Section III-B1, our algorithm does not define, and is independent from, the key exchange mechanism. However, despite whether the keys are pre-deployed or dynamically generated, an interesting and open challenge is to deal with insider attacks, i.e. jamming attacks coming from nodes that already know the secret key. In our future work we plan



(a) Overall Packet Delivery Ratio (PDR).

(b) Packet Reception Ratio (PRR) of the jamming links.

(c) Overall Packet Delivery Ratio (PDR) with blacklisting.

(d) Packet Reception Ratio (PRR) of the jamming links with blacklisting.

Fig. 4. Overall Packet Delivery Ratio (PDR) and Packet Reception Ratio (PRR) of R-TSCH and the default TSCH generation method with and without radio channel blacklisting.

to search for mechanisms that thwart insider jamming attacks, for example by identifying and reactively updating the secret key used for channel randomization.

ACKNOWLEDGMENTS

This work was carried out within the action "Strengthening Post Doctoral Research" of the "Human Resources Development Program, Education and Lifelong Learning", 2014-2020, which is being implemented from IKY and is co-financed by the European Social Fund – ESF and the Greek government.

It has been also partially supported by the University of Piraeus Research Center and the Greek General Secretariat for Research and Technology under its matching funds programme.

REFERENCES

[1] "WirelessHART Specification 75: TDMA Data-Link Layer," *HART Communication Foundation Std., Rev. 1.1,*, vol. HCF SPEC-75, 2008.

[2] ISA-100.11a-2011:, "Wireless Systems for Industrial Automation:Process Control and Related Applications," *International Society of Automation (ISA) Std.*, vol. 1, May 2011.

[3] "IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), April 2016.

[4] S. Duquennoy, A. Elsts, A. Nahas, and G. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2015)*, Ottawa, Canada, Nov. 2017.

[5] J. Xu, L. Duan, and R. Zhang, "Proactive eavesdropping via cognitive jamming in fading channels," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2790–2806, May 2017.

[6] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE.* IEEE, 2007, pp. 2526–2530.

[7] A. D. Wood, J. A. Stankovic, and G. Zhou, "Deejam: Defeating energy-efficient jamming in ieee 802.15.4-based wireless networks," in *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2007, pp. 60–69.

[8] E.-K. Lee, S. Y. Oh, and M. Gerla, "Randomized channel hopping scheme for anti-jamming communication," in *Wireless Days (WD), 2010 IFIP.* IEEE, 2010, pp. 1–5.

[9] E. Achuthan and R. Kishore, "A novel anti jamming technique for wireless sensor networks," in *Communications and Signal Processing (ICCSP), 2014 International Conference on.* IEEE, 2014, pp. 920–924.

[10] T. Watteyne, M. Palattella, and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement," RFC 7554, 2015.

[11] G.-Y. Chang, S.-Y. Wang, and Y.-X. Liu, "A jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6712–6725, 2017.

[12] P. Du and G. Roussos, "Adaptive time slotted channel hopping for wireless sensor networks," in *Computer Science and Electronic Engineering Conference (CEEC)*, Sept 2012, pp. 29–34.

[13] D. Sitenkov, "Access control in the internet of things," Master's Thesis, 2014.

[14] P. H. Gomes, T. Watteyne, and B. Krishnamachari, "MABO-TSCH: Multi-hop And Blacklist-based Optimized Time Synchronized Channel Hopping," *Transactions on Emerging Telecommunications Technologies*, vol. e3223, pp. 1–20, 2017.

[15] D. Zorbas, V. Kotsiou, F. Theoleyre, G. Z. Papadopoulos, and C. Douligeris, "Lost: Localized blacklisting aware scheduling algorithm for ieee 802.15.4-tsch networks," in *10th IFIP Wireless Days conference*. IEEE, April 2018.