PRM - Private Interference Discovery for IEEE 802.15.4 Networks

Dominik Roy George, Savio Sciancalepore
Eindhoven University of Technology, Eindhoven, Netherlands
{d.r.george, s.sciancalepore}@tue.nl

***Abstract*— Due to the mobile and pervasive nature of IoT networks, even more frequently, multiple IoT networks managed by different network administrators share the same spectrum and operate in the same area, leading to packet losses and degradation of the Quality of Service (QoS). Assuming the use of the widespread IEEE 802.15.4 communication technology, the most straightforward solution would be to allow the networks to share the local Radio Scheduling Table (RST) to optimize channel access. However, exchanging the RST can leak several key information, such as the topology of the network, the number of devices, and the channel access patterns.**

To address such problems, we present PRM, the first scheme for discovering in advance potential interferences among IEEE 802.15.4 networks, without exposing the whole RST to untrusted parties. Our solution adapts a protocol for Private Set Intersection, while combining it with an innovative iterative set division algorithm, making the whole solution feasible on constrained devices of the IoT domain. Our experimental performance assessment, carried out on heterogeneous devices, shows that PRM can discover colliding channel assignments in less than 1 sec. on more capable embedded devices (e.g., the Raspberry PI), while also being feasible for more constrained platforms (e.g., the ESPCopter), depending on the amount of used radio resources.

***Index Terms*—Privacy-Enhancing Technologies, IoT Security, Private Set Intersection.**

I. Introduction

Internet of Things (IoT) devices are increasingly used in our daily lives in several use-cases, including Wireless Sensor Networks (WSNs), smart cities, and transportation [1]. According to recent reports, the number of connected IoT devices was 8 billion in 2020, and it is projected to 30 billion by 2025 [2].

In this context, IEEE 802.15.4 is emerging as the leading communication technology for constrained devices, thanks to the capability of minimizing energy consumption [3]. In a IEEE 802.15.4 network, the time is divided into time-slots, and the nodes communicate deterministically on a given frequency and time slot according to a Radio Scheduling Table (RST), established through radio scheduling algorithms [4].

However, scheduling algorithms usually apply within IEEE 802.15.4 networks managed by the same network administrator. Due to the mobile and pervasive nature of modern IoT, often, networks managed by different administrators work on the same spectrum, at the same time. For example, IEEE 802.15.4 networks installed within cars or on-board of drones may need to work in area where another IEEE 802.15.4 network is already deployed. Similarly, two static IoT networks may need to be deployed in proximity, while being managed by different administrators. In such cases, the application of traditional radio scheduling algorithms would require the sharing of the local RST to arrange an optimized communication schedule and prevent interferences, packet losses, and Quality of Service (QoS) degradation. However, sharing the RST might generate significant threats. Indeed, through the RST, a malicious adversary might gain knowledge of the number of devices, the used frequencies, their deployment, and the communication patterns, to name a few. Using such information, the adversary can capture the devices, or launch more advanced attacks, jeopardizing the service provided by the IoT network. Thus, RSTs should remain private, and an interference-free schedule has to be arranged without knowing the details of the scheduling of the remote IoT network.

To the best of our knowledge, no contributions in the scientific literature investigated security and privacy issues in radio scheduling algorithms for IEEE 802.15.4 networks. Indeed, all the solutions currently available focus on the optimization of dedicated metrics, such as the communication overhead [5], the energy consumption [6], and the throughput [7], neglecting security and privacy considerations. At the same time, deploying a solution for *private interference discovery* in IoT networks is challenging. Indeed, solutions for this problem should cope with the intrinsic limitations of IoT devices, in terms of processing capabilities, communication overhead, and most importantly, energy consumption. At the same time, they should discover only *shared* channels, while maintaining the privacy of the whole RST.

Contribution. In this paper, we propose PRM (an acronym for Private RST Matching), a novel protocol enabling private interference discovery among IEEE 802.15.4 networks. PRM aims to allow two nodes, e.g., the gateways of two IoT networks, to identify which radio cells are used by both at the same time, without sharing to the other party the whole RST. Identifying such cells and resolving the conflict beforehand helps reduce interference, while improving reliability and preserving the secrecy of the radio resources assignment. To cope with the limitations of IoT devices, we design two modes of operation of the protocol, one based on the plain execution of Private Set Intersection (PSI) protocols and another one based on an innovative iterative set division strategy that significantly reduces the computational overhead of PSI on constrained

This is a personal copy of the authors. Not for redistribution. The final version of the paper will be available soon through the IEEExplore library.

devices. Besides providing a formal security analysis of PRM, we also carry out an experimental performance assessment on heterogeneous IoT devices, i.e., the Raspberry Pi and the ESPCopter. Our results show that PRM can always discover shared radio cells, reporting an execution time dependent on the mutual allocation of the occupied cells in the RST. Overall, while more powerful gateways (e.g., the Raspberry Pi) can handle a larger amount of occupied cells in less than 1 sec, more constrained devices (e.g., the ESPCopter) can take more time while still taking advantage of the iterative set division approach. To the best of our knowledge, PRM is the first solution in the literature to address and solve the problem of private interference discovery among IoT networks.

Roadmap. The paper is organized as follows. Sec. II presents our scenario and adversary, Sec. III describes PRM, Sec. IV provides the security analysis, Sec. V shows the performance of PRM, Sec. VI wraps-up the discussion, and finally, Sec. VII concludes the paper and outlines future work.

II. Scenario and Adversary Model

In this section, Sec. II-A describes the scenario, while Sec. II-B reports the details of the adversary model.

A. Scenario

In this paper, we assume two independent wireless networks, namely A and B, which are using the IEEE 802.15.4 communication technology to exchange wireless messages between member nodes. As such, they use one or more of the 16 available channels in the bandwidth $[2.4-2.5]$ GHz, and one or more of the available N time-slots. IEEE 802.15.4 provides low-data-rate wireless communication between constrained devices [3]. To communicate, two nodes need to be synchronized, i.e., to have their radios on at the same frequency, at the same time. In IEEE 802.15.4, time is divided into time-slots, lasting $T = 10$ msec by default. Within a time-slot, based on its status, a node can transmit (receive) a data packet and receive (transmit) the corresponding acknowledgement, if required. We denote the combination of a specific time-slot and frequency as a *cell*, as in [3]. Overall, N time-slots and F frequencies are available, where $F = 16$ and, usually, $N = 101$, according to [8].

Each IEEE 802.15.4 network has a *Coordinator*, i.e., a specific node responsible for maintaining and regulating network connectivity among the devices. The coordinator is often also the *Gateway* of the IoT network, responsible for communicating with other IoT networks managed by a different network administrator, as well as the Internet. As such, the *Coordinator* is usually equipped with larger processing, storage, and energy capabilities than the constrained IoT devices, thus supporting the execution of symmetric and asymmetric cryptography techniques. Without loss of generality, we assume the coordinator of each network is in possession of the information regarding the cells used in the network, where at least one packet can be transmitted by any node. The table containing the information described above is the RST, and it can change periodically based on new cells assignment. It is worth noting that the specific radio scheduling algorithm used by the network is not relevant to our work, and it can be either centralized or distributed [4]. We only assume that the information about the possible usage of a cell is available to the coordinator. Also, we do not care if a specific cell is not always used (e.g., because of lack of traffic). If the IoT network plans to use a specific cell, then we consider it as *occupied* by a communication.

We assume B is a static network, installed in a dedicated location. For instance, B can be the monitoring network of a building, where IoT devices equipped with vibration sensors constantly monitor the stability of the overall construction. At the same time, A can be either a static or a mobile network. For instance, A could be a network installed on a transportation means, such as a car, or be another static network to be deployed in the same area of A.

Using the same bandwidth, when in the same reception radius, A and B are very likely to interfere with each other, causing packet losses and throughput reduction, with a consequent degradation of the QoS offered to the end-users. A naive solution to this problem would be that the Coordinators of the two networks exchange the respective RST, to identify colliding time-slots and avoid interferences. However, disclosing the specific cells to be used by the IoT devices to communicate would represent a severe threat for the specific network. Indeed, as detailed in Sec. II-B, the leakage of such information could enable easy disruption by a motivated adversary, e.g., enabling effort-less eavesdropping, traffic analysis, and jamming, to name a few. Therefore, we assume that each network would like to avoid interference with the neighboring one, while not revealing sensitive details about its operations, such as the channel allocation (i.e., the RST) and its logical topology. Our solution, namely PRM, is specifically intended to address and solve this problem (see Sec. III).

Finally, for the readers' convenience, we report in Tab. I the overview of our main notation.

B. Adversary Model

We assume an adversary, namely, ϵ, interested in becoming aware of the RST of a given party. The information contained in the RST can be used for multiple malicious purposes. For instance, by knowing the cells used by any two nodes to communicate, ϵ can stealthily listen to the communication as a passive eavesdropper, and acquire sensitive information. Assuming the communication link is encrypted, such information can still be used by ϵ to perform traffic analysis, or to jam the communication link, preventing any communication to occur and potentially causing more severe harm to the system monitored by the IoT network. With reference to our scenario, we assume ϵ is in line with the well-known Honest-but-Curious (HbC) model. As such, the adversary behaves honestly, by following all the steps of our protocol. At the same time, by using the received information, it tries to gain as much knowledge as possible regarding the cell assignment schedule of the victim.

Table I
NOTATION SUMMARY.

Notation	Description
A, B	Network Identifiers.
R_A, R_B	Radio Scheduling Tables of the networks A and B.
a_j, b_j	j-th elements/cells of the list of A and B.
r_{A_j}	Individual nonce for each element.
c_j	Computed element of a_j.
C_A	List of computed elements.
$p_{A\|\|B}, q_{A\|\|B}$	Safe primes for each party.
$n_{A\|\|B}$	Size of the modular field for each party.
$H(\cdot)$	Generic hash function.
C_B	List of computed elements ($H(x_j)$) on B.
Y	List of elements send to server to compare.
M	Row count of RST.
N	Column count of RST.
$R_{A,i,k}, R_{B,i,k}$	Split part k of RST at round i.
oc_i	Occupied cells at the round i of PRM.
e_{tx}	Energy of sending packets.
e_{rx}	Energy of receiving packets.
$\rho_{i,k}$	Defines the current state of consumed energy.
$\tau_{A\|\|B}$	Energy threshold.
δ_i	Privacy overhead at the round i.
ν	Privacy threshold.

To reach the above-cited objective, we assume ϵ is equipped with both passive and active capabilities. On the one hand, ϵ is a global eavesdropper, passively listening to the messages exchanged by both parties. Furthermore, ϵ is a location-bounded adversary, located in the same area of the target network. On the other hand, ϵ can also feature active capabilities, e.g., inject packets at will, either by replaying existing packets or by inserting ad-hoc messages, to obtain the desired information.

Our protocol, namely PRM, is intended specifically to allow two networks to discover interfering cells in the RST, while preventing the previously-described adversary to gain information about their cell assignment schedule. More details follow in Sec. III.

III. PRM PROTOCOL

This section introduces PRM. Sec. III-A describes the basic cryptographic primitive, Sec. III-B introduces the iterative set division strategy, while the overall PRM scheme is described in Sec. III-C.

A. Private Set Intersection

PSI protocols allow a set of parties to evaluate the intersection of the sets of elements in their possession, by revealing to the other parties only the shared elements [9], [10]. Such protocols have received increasing attention in the last years, since the elements not possessed by both parties are not exposed to the remote party, allowing for enhanced privacy [11], [12].

Many PSI protocols are available in the literature. They differ on several factors, such as in computational overhead, bandwidth, energy consumption, and number of involved parties, to name a few. In this work, we build on the private equality testing approach proposed by Kotzanikolaou et al. [13] because first, both parties become aware of the *shared* input. Hence, both parties can take action to solve the situation. Note that, if only one party has insights regarding the intersecting input, it would be the only one to have such knowledge, enabling several possible threats to emerge (see Sec. II). Second , compared to other schemes in the literature, such a scheme is the most lightweight and efficient. In the IoT context, where processing constraints and energy consumption play a major role, such considerations are crucial. The approach by Kotzanikolaou et al. relies on the well-known prime numbers Factorization Problem (FP) at the basis of the RSA algorithm, and it uses the corresponding security guarantees to build a PSI scheme for discrete, finite, low entropy data. In the following, we describe the adaption of the protocol by [13] to our problem, i.e., considering the comparison of multiple elements in the RSTs of the Coordinators of the IoT networks.

During the setup phase, A and B select two large prime numbers at random, namely, (p_A, q_A) and (p_B, q_B), and use them to compute the values $n_A = p_A \cdot q_A$ and $n_B = p_B \cdot q_B$. Also, for each element in the set, they compute a key pair (e, d), where d is the identifier of the element in the set and $e = \frac{1}{d} \mod \phi(n)$. Figure 1 illustrates the steps required by the protocol, while we provide the details below.

1) Denote with a_j and b_j the generic elements of the set possessed by A and B, respectively. A first extracts a vector of J random nonces, namely, $\mathbf{r_A} = [r_{A,1}, r_{A,2}, \ldots, r_{A,j}, \ldots, r_{A,N_A}]$, where N_A is the number of elements in the set of A. Then, A generates a vector of encrypted elements $\mathbf{C_A}$, where each element c_j is computed according to Eq. 1.
$$c_j = r_{A_j}^{a_j} \mod n_B, \tag{1}$$
A delivers the vector $\mathbf{C_A}$ to B.
2) At message reception, B takes each element in the vector C_A and, for each element in its own set, namely, b_i, it computes an encrypted response x_j, as in Eq. 2.
$$x_j = c_j^{e_{b_i}} \mod n_B. \tag{2}$$

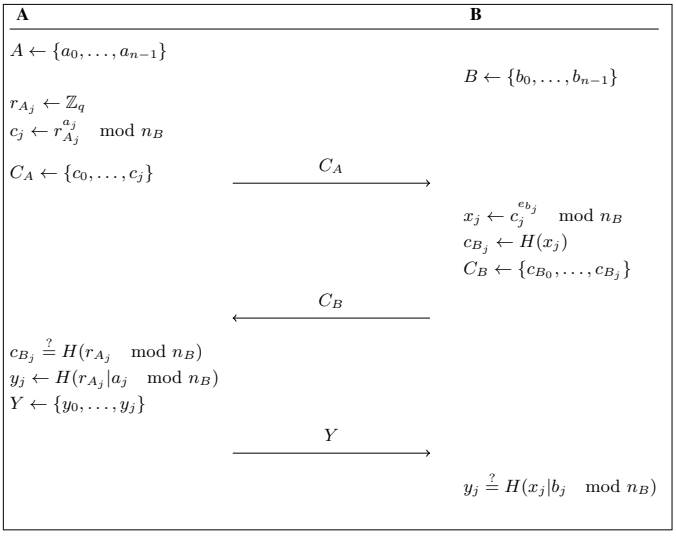

Figure 1. FP-PSI protocol.

Then, B computes the digest of each encrypted response x_j, namely $c_{B,j}$, through the hash function H, as in Eq. 3.

$$c_{B,j} = H(x_j). \tag{3}$$

Then, B delivers to A the vector $\mathbf{C_B} = [c_{B,1}, \ldots, c_{B,j}, \ldots, c_{B,K}]$ (with $K = M * N$).

3) At message reception, A performs the equality check on each element, by applying Eq. 4.

$$c_{B_j} \stackrel{?}{=} H(r_{A_j} \mod n_B). \tag{4}$$

For all the elements satisfying Eq. 4, A computes an encrypted proof, namely y_j, as per Eq. 5.

$$y_j = H(r_{A_j} | a_j \mod n_B). \tag{5}$$

A delivers back to B the vector of encrypted proofs, namely, $\mathbf{Y_A} = y_1, \ldots, y_j, \ldots, y_S$ (where S is the number of elements satisfying Eq. 5).

4) At message reception, B compares the received proofs with its elements, according to Eq. 6.

$$y_j \stackrel{?}{=} H(x_j | b_j \mod n_B), \tag{6}$$

thus confirming the intersecting elements in its list.

In summary, at the end of the protocol, both parties know the elements (in our case, the cells) possessed by both of them, requiring a change to prevent interference issues.

However, we notice that the straightforward application of the scheme in [13] to our problem could result in a computationally-intensive protocol. Overall, the scheme presented above requires as private input the number of occupied cells (namely, *oc*), and such number depends on many factors (no. of nodes, type of traffic, etc.). Considering that IoT nodes are usually energy-constrained, such a scheme is hardly applicable efficiently. In the following, we couple the scheme described above with a new algorithm based on multiple iterative set divisions, thus significantly reducing its overhead.

B. Iterative Set Division Strategy

The rationale behind the iterative set division approach lies in the consideration that, usually, network administrators schedule radio operations in consecutive time-slots and channels, located close each other in the RST. Thus, the core idea of this approach is to identify before-hand collision-prone portions of the RST and to focus the comparison on these portions of the RST, if necessary, instead of comparing each cell singularly. To ease the description of the afore-mentioned approach, we refer to the example in Figure 2.

Denote with R_A and R_B the RSTs of A and B, respectively. First, each of them divides its RST in two halves. We denote the two halves of the RST possessed by the entity A at the iteration $i = 1$ as $R_{A,1,0}, R_{A,1,1}$, i.e., $R_{A,i,k}$ (similarly for B). If the RST in possession of A has at least one slot assigned to a communication link in the specific half of the RST, then A is set to be in possession of the element $R_{A,1,0}$; otherwise, A does not possess the element. Similar considerations apply for B. Then, A runs the PSI protocol described in Sec. III-A to check if any half of its RST is possessed also by B. In the specific example of Fig. 2, we can see that the first half of the RST of both parties is occupied by at least a single communication link (marked as green). However, the second half of the RST is marked red, since B does not have any occupied cells here. Thus, at the end of the scheme, A and B know that only the left-most halves of their RST collide. Therefore, they trigger a new round $i = 2$, splitting the colliding half in two new halves, namely $R_{A,2,0}, R_{A,2,1}$ (resp. $R_{B,2,0}, R_{B,2,1}$). In the example in Fig. 2, at the end of the round $i = 2$, A and B learn that both portions (halves) of their RSTs could have collisions (second round marked on both portions as green on both parties). Thus, they continue the protocol with the round $i = 3$, further splitting each colliding portion of the RST in two new elements. At the end of this iterative scheme, A and B narrow down the comparisons to only the portion of the RST where a collision really exists, finding the colliding slot.

We notice that, in some conditions, the iterative set division approach can reduce the computational overhead on involved parties. Indeed, when the two parties have occupied cells in far-away portions of the RST, the approach immediately discovers the absence of colliding cells, requiring a number $oc + \sum_{i=1}^{log_2(N)} 2^i$ exponentiations. Thus, depending on the location of occupied cells, the iterative set division scheme has a computational advantage over the PSI approach in Sec. III-A. In particular, when the entities have occupied cells in far-away portions, the iterative set division approach takes two exponentiations only for each party, while the PSI requires a number of exponentiations equal to the number of all occupied cells oc of both parties.

However, in general, the iterative set division approach is not always more advantageous than the application of the PSI scheme described in Sec. III-A. Indeed, the convenience of the iterative approach depends on the mutual distribution of the occupied cells among the communicating parties. Thus, if the parties have occupied cells in the same portion of the RST, the iterative set division approach ends up increasing the amount of operations necessary to discover collisions, worsening the computational overhead as compared to the application of the PSI scheme. The integration of such considerations at each

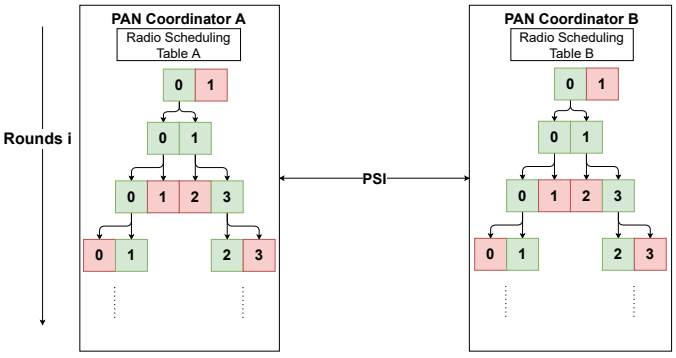

Figure 2. Illustration of the iterative set division approach.

step of the iterative set division algorithm leads to the full PRM scheme, described below.

C. The PRM scheme

Fig. 3 illustrates the whole PRM protocol, while we describe each step below.

1) PRM is triggered by the client, when it realizes to be in the wireless reception radius of the server.
2) Both parties run the iterative set division approach described in Sec. III-B. Thus, the client and the server first split R_A and R_B on the column dimension, into two halves, namely, $R_{A,1,0}, R_{A,1,1}$ and $R_{B,1,0}, R_{B,1,1}$. Then, A applies the step 1) described in Sec. III-A considering as *possessed elements* the cell identifiers $R_{A,i,k}$ where occupied cells exist. Thus, first A delivers the challenge vector $\mathbf{C_A}$ to B, and then, B applies the step 2) described in Sec. III-A, over the *possessed elements* and delivers back the response $\mathbf{C_B}$.
3) At message reception, A identifies the colliding elements, running Eq. 4, and it computes the vector of encrypted proofs at the round $i = 1$, namely, $\mathbf{Y_{A,1}}$ (see step 3) in Sec. III-A). If no colliding elements are found, A delivers $\mathbf{Y_{A,1}}$, and stops the protocol, concluding that no collisions are present.
4) Conversely, if at least one colliding element is found, A checks if it is convenient to continue further with the iterative set division approach, considering both energy and privacy requirements. We define ν as the *privacy threshold* of the protocol, i.e., the maximum privacy leakage the entity A would like to provide through the protocol. Similarly, we define τ [J] as the *energy threshold* of the protocol, i.e., the maximum amount of energy that A would like to spend for the protocol. First, A computes the *privacy overhead* at the current round i, namely $delta_i$, through Eq. 7.

$$\delta_i = \frac{1}{2^{M * \frac{N}{2^i}}}, \tag{7}$$

where M is the number of rows of the RST, N the number of columns of the RST and 2^i the number of elements in the current round. Note that $delta_i$ considers the probability of guessing the *occupied* cells in possession of A from B. A checks if $\delta_i < \nu$.
If such condition is not satisfied, A switches from the iterative set division approach to the full FP-PSI scheme (see Sec. III-A), considering as occupied cells only the ones in the portion of the RST colliding with B. If, instead, the above-stated condition is verified, A computes the *energy overhead* at the current round i, namely ρ_i, as in Eq. 8

$$\rho_i = \sum_{j=1}^{i} (2^i + e_{tx} + e_{rx}) + (oc_{i_{(A)}} + 2 * e_{tx} + e_{rx}), \tag{8}$$

where e_{tx} and e_{rx} are the energy to transmit and receive a packet (in J), respectively, while oc_i is the number of elements possessed by A at the round i. Note that ρ_i considers the overall energy spent by A at this time, as well as the worst-case amount of energy spent by A if all the elements in the next round of the iterative set division approach collide with the elements in possession of B. As before, A checks if $\rho_{i,k} < \tau$. If such condition is not satisfied, A switches from the iterative set division approach to the full FP-PSI scheme (see Sec. III-A), considering as occupied cells only the ones in the portion of the RST colliding with B.
Instead, if this condition is verified, A goes ahead with PRM by applying further the iterative scheme.
5) A applies the same procedure of step 2) for the new iteration of the iterative set division approach. Then, A delivers to B the vector $Y_{A,1}$ from the previous round $i = 1$ and the vector of encrypted challenges of the round $i = 2$, namely, $C_{A,2}$.
6) At message reception, B first verifies the matching of the elements at the round $i = 1$, through the application of Eq. 6 (step 4) in Sec. III-A) on the input vector $Y_{A,1}$., Then, considering the elements colliding with A, B further divides the colliding elements in two halves, and applies the same procedure described at step 3).
7) The procedure described at step 3) is repeated, until either there are no matches between the elements possessed by A and B or they cannot divide further the elements in two (either because the FP-PSI protocol in Sec. III-A was run or the iterative set division approach ended up in portions made up of single columns with occupied cells of the RST). After identifying all the colliding cells, the client and server select new channels and time-slots for the colliding cells, choosing into the non-colliding portions. If necessary, they can launch a new instance of PRM to check if these new cells still collide.

IV. Security Analysis

We provide here the security analysis of PRM, using both ProVerif (Sec. IV-A) and probability theory (Sec. IV-B).

A. Formal Security Analysis via ProVerif

The most relevant security feature of PRM is to guarantee the privacy of the whole RSTs of the participating parties, while allowing to discover interfering cells. Such a property is provided through the integration of the protocol by Kotzanikolaou et al. [13]. In the following, in line with many works in the recent literature [14], [15], [16], [17], we use the automated tool ProVerif to formally verify the security of the single run of our scheme, as well as to confirm that our construction using such a protocol does not break its security.

ProVerif assumes that the underlying cryptographic primitives at the roots of the protocols are robust, and that the adversary is aware of the steps and the public values of the protocol [18]. Based on these assumptions, ProVerif formally verifies the security of the cryptographic protocol against the Dolev-Yao attacker model, i.e., an adversary able to access, modify, delete, and forge new messages on the public communication channel. If ProVerif identifies an

Figure 3. The design of the PRM.

attack during the formal verification, it also lists the steps to achieve to break the protocol. In our context, we applied ProVerif to formally verify the secrecy of the elements possessed by the two communicating entities (the *occupied* cells), namely, *a* and *b*. Recall that ProVerif provides the output *not attacker(elem[]) is true*, when the attacker cannot retrieve the value of *elem*. On the other hand, if the output is *not attacker(elem[]) is false*, then the attacker can retrieve the value of *elem*. Similarly, ProVerif provides the output *weak secret(elem[]) is true* when the attacker cannot execute guessing attacks on the value *elem* or brute-force it, while it provides the outcome *weak secret(elem[]) is false* when the attacker is able to guess or bruteforce the value *elem*. Lst. 1 shows the output provided by Proverif. Interested readers can use the source code we released at [19] to reproduce our results and further extend them in customized use-cases.

Listing 1. Excerpt of the output of the ProVerif tool.

```
Verification summary:
Weak secret a is true.
Weak secret b is true.
Query not attacker(a[]) is true.
Query not attacker(b[]) is true.
Query not attacker(rA[]) is true.
```

The output of ProVerif shows that the private occupied cells *a* and *b* are not exposed to the attacker. In addition, the attacker is not able to guess/brute-force the occupied cells *a* and *b* of the RST of the participating parties. Thus, ProVerif confirms that the single run of PRM protects the confidentiality of the occupied cells of the entities participating in the protocol, while ensuring the identification of interfering cells.

B. Additional Security and Privacy Considerations

In the previous section, we formally showed the security of the single run of PRM. However, our protocol includes multiple rounds and, at each round, the two communicating

entities use as private input element identifiers representative of a reduced number of RST cells.

At each round, PRM leaks some information to the remote party, i.e.: (i) the number of element identifiers possessed by this party, namely k, and (ii) the number of colliding element identifiers. In the following, using sound probability theory, we formally define the guessing probability of the adversary at the round i of PRM, namely $p_c(i)$. We first discuss in Prop. IV.1 the case where the iterative approach stops at the round i because no colliding element identifiers are found.

Proposition IV.1. *Assume that at the round i of PRM A and B do not find any colliding element identifiers in their private set. Then, the probability for an honest-but-curious adversary to guess the $\tilde{k}$ cells possessed by the remote party is $p_c(i) = \frac{\tilde{\kappa}}{F \cdot 2^{(MAX-i)} \cdot (U_i - \xi_i)}$, being ξ_i the number of challenges sent by the adversary, U_i the number of element identifiers at the round i, F the number of rows in the RST and $MAX = \lceil * \rceil log_2(N)$.*

Proof. Assume that at the round i of PRM A and B do not find any colliding element identifiers in their private set. At this round i, we can define the probability for an honest-but-curious adversary to guess the correct element identifiers possessed the remote party (namely, $p_B(i)$) as in Eq. 9.

$$p_B(i) = \frac{\kappa_i}{U_i - \xi_i}, \tag{9}$$

where ξ_i is the number of challenges sent by A to B at the round i, κ_i is the number of responses sent back by B to A at the round i, and U_i is the overall number of available element identifiers at the round i. Note that, per construction, $P_B(i) \in [0, 1]$. Recall that at the round i we consider element identifiers representative of a number 2^{MAX-i} of columns in the RST. Therefore, Eq. 9 can be casted into Eq. 10 to obtain the probability of guessing a single column.

$$p_{col}(i) = \frac{\tilde{\kappa}}{2^{(MAX-i)} \cdot (U_i - \xi_i)}, \tag{10}$$

As each column contains F cells, we can resort to Eq. 11 to obtain the probability $p_c(i)$ of guessing the possessed cell.

$$p_c(i) = \frac{\tilde{\kappa}}{F \cdot 2^{(MAX-i)} \cdot (U_i - \xi_i)}, \tag{11}$$

where F is the number of rows of the RST (i.e., the number of available frequencies), while $\tilde{k}$ represents the number of cells occupied in the RST of the remote party. Note that, as each round of the PRM divides colliding element identifiers in halves, $MAX = \lceil * \rceil log_2(N)$. □

However, PRM could end also because of energy or privacy constraints. In these cases, the following Prop. IV.2 apply.

Proposition IV.2. *Assume that at the round i of PRM A and B find κ_i colliding element identifiers in their private set, but PRM ends. Then, the probability for an honest-but-curious adversary to guess one of the $\tilde{k}$ cells possessed by the remote party is $p_c(i) = \sum_{j=1}^{\kappa_i} \frac{\tilde{k_j}}{F \cdot 2^{(MAX-i)}}$, with $MAX = \lceil * \rceil log_2(N)$ and $\sum_{j=1}^{\kappa_i} \tilde{k_j} = \tilde{k}$.*

Proof. Assume that at the round i of PRM A and B find κ_i colliding element identifiers in their private set, but PRM ends because of a privacy or energy constraint. Recall that the element identifier at the round i of PRM is representative of a group of 2^{MAX-i} columns. Also, consider that each of the κ_i colliding identifiers is representative of a number $\tilde{k_j}$ of *occupied columns*, and that, per definition, $\sum_{j=1}^{\kappa_i} \tilde{k_j} = \tilde{k}$. Thus, the probability $p_{col}(i)$ of guessing an occupied column of the remote party at the round i of PRM is defined by Eq. 12.

$$p_{col}(i) = \sum_{j=1}^{\kappa_i} \frac{\tilde{k_j}}{2^{(MAX-i)}}, \tag{12}$$

As any of the F rows of a column can be occupied, we can derive the probability of guessing an occupied cell as in Eq. 13.

$$p_c(i) = \sum_{j=1}^{\kappa_i} \frac{\tilde{k_j}}{F \cdot 2^{(MAX-i)}}. \tag{13}$$

□

V. Performance Assessment

In this section, we provide the performance assessment of PRM, through combined experimentation and simulations. Sec. V-A describes the setup of our experiments, while Sec. V-B provides the results.

A. Experiments Setting

To evaluate the feasibility and impact of PRM on real IoT devices, we evaluated the overhead of our scheme through combined simulations and experimentation on real hardware devices. As for the hardware, we selected the Raspberry Pi 3 Model B+ and the ESPCopter. The Raspberry Pi Model B+ is equipped with a Cortex-A53 processor running at 1.4 GHz, 1GB of RAM, and 16GB of storage, and it is often used as a tiny gateway for static IoT deployments [20], [21]. The ESPCopter is a wirelessly networkable small-size programmable mini-drone, powered by the microcontroller ESP8266, equipped with 1 MB of Flash memory and 96KB of RAM [22]. In our setup, it models a resource-constrained IoT device. On the Raspberry Pi, we adapted the code provided by the authors in [13], while on the ESPcopter we implemented our solution from scratch, leveraging the support provided by the widespread *mbedTLS* library. As for the simulations, we used Matlab R2021a on a Lenovo Thinkpad P1, equipped with two Intel Core i7-8750H processors running at 2.20 GHz, 32GB of RAM, and 1 TB of SDD Storage. For each experiment presented in the next section, we ran 100 simulations, and we reported the average results and the 95% confidence intervals of our measurements.

Overall, we implemented the operations required by PRM on the afore-mentioned hardware platforms, and we took note of the required time. Then, through simulations, we extracted the average number of protocol instances required in different settings, and we leveraged the previously-noted execution times to derive the overhead of the full scheme. Finally, note that our simulations address the worst case of our scheme,

where PRM is executed until it reaches the maximum round of the iterative set division approach.

B. Results

In this section, we provide some results and insights into the use of our proposed scheme, looking at its performances with different filling profiles of the RSTs on the involved devices, as well as different hardware platforms. We first investigated the scenario where the RST of the network A contains a given fraction of occupied cells, all grouped in the upper-right part, while the RST of the network B has this same profile, but with a reduced number of occupied cells. Note that the afore-cited configurations are a typical choice for network administrator, especially when cells are assigned starting from the first available frequencies and slots. Fig. 4 reports the execution time in such a scenario of both the PSI-based approach and the iterative set division scheme on the Raspberry Pi device, while increasing the number of occupied cells, i.e., cells assigned for radio communications. We report the average results of our experiments, together with the 95% confidence intervals. Overall, we notice that the execution time

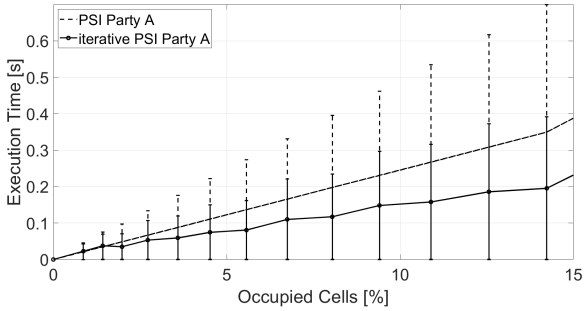

Figure 4. Execution time of party A of the PSI-based approach and the iterative scheme, when A has a RSTs filled in the right-upper part with the 95% confidence interval.

of both schemes increases with the increase of the filling ratio (% of occupied cells) of the right upper part of the RST on both parties, with different behaviors. When the occupied cells exceed 1%, the iterative set division approach is faster, as the peers (specifically, the client) take advantage of the distribution of the cells in the RST to perform a reduced number of exponentiations. Using such an approach, shared occupied cells could be localized quicker, leading to reduced execution times.

Figure 5 provides the results in the same setup of Fig. 4, but using the ESPCopter and focusing on the party A. We notice a similar behavior to Fig. 4, but the ESPcopter takes more time to complete the protocol. When only the 12.5% of the RST is populated, the ESPcopter needs more than one minute to complete PRM (99.6 sec.), which is not acceptable neither feasible. This is the reason why we introduced the *energy threshold* in our protocol. On the one hand, we notice that mobile networks typically have few occupied cells. In these cases, PRM can complete even on a very constrained platform in less than 1 sec. On the other hand, when the RST is more populated, more powerful IoT devices are desirable.

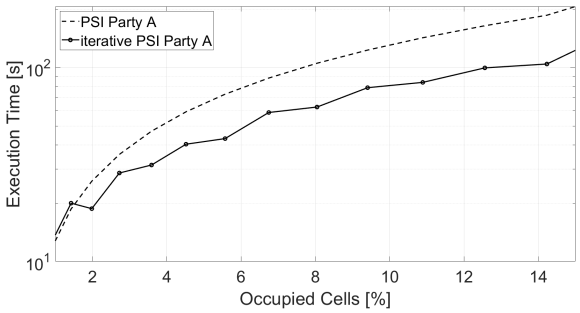

Figure 5. Execution time of the two variants of PRM on the ESPCopter, when A has a RSTs filled with occupied cells in the right-upper part.

The previous results investigated the scenario where occupied cells are localized in a specific part of the RST. Assuming a scenario where the occupied cells are extracted uniformly at random in the whole RSTs, Figure 6 shows the execution time of the two proposed schemes, with an increasing value of the filling ratio. When the occupied cells in the RST are distributed

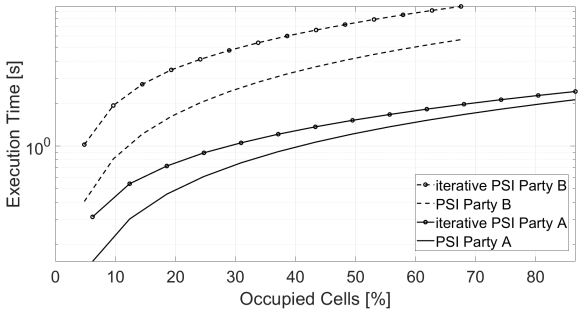

Figure 6. Execution time of the two proposed schemes, with an increasing value of the filling ratio of the RST, where occupied cells are extracted uniformly at random.

uniformly at random, the plain-PSI approach is the quickest. Indeed, as occupied cells are distributed at random in the RST, grouping operations performed by the iterative approach do not lead to the progressive elimination of multiple unused portions of the RST. Therefore, as the rounds of the iterative approach are executed, multiple additional exponentiation operations are performed, leading to higher execution times. Conversely, the plain-PSI approach takes an execution time that is proportional to the amount of occupied cells (not to their distribution), being more efficient in this specific case. We will discuss these results in the next section.

VI. Discussion

In this section, we wrap-up on our results and provide some additional considerations on PRM.

Schemes Selection. As shown by the results in Sec. V-B, the iterative set division approach is the most efficient solution when one of the parties in the protocol has occupied cells localized in a given portion of the RST. Conversely, if the distribution of the cells in the RST is random (see Fig. 6), the

plain-PSI scheme is the best choice. Note that the party initiating the protocol (A) always knows its own cells' distribution. Thus, it can choose the strategy minimizing the overhead.

Set division Strategy. When applying the iterative set division approach, PRM divides the RST progressively into equal-sized portions. In line of principle, multiple different shapes can be chosen when dividing, e.g., rows, squares, or columns. However, the selection of a shape directly affects the privacy leakage of PRM. In our specific case, recall that the number of rows of the RST is $F = 16$, and thus, dividing progressively on rows would cause a significant guessing probability to the remote party. Instead, as the number of columns in the RST is higher ($N = 101$ in our case), dividing over columns only provides enhanced privacy guarantees, always allowing to fulfill privacy constraints (see part below).

Thresholds. As discussed in Sec. III-C, PRM features two thresholds, namely, the privacy threshold ν and the energy threshold τ. We define ν to be in range of $[0, 1]$. When ν is 0, PRM executes the plain-PSI over the entire set of occupied cells. When ν is 0.1, then PRM allows the iterative set division to reach the last possible round, since the highest privacy leakage of such approach is $\frac{1}{2^{16}}$, i.e., the guessing probability of a single cell in one column of the RST. On the one hand, a peer can choose the desired value of ν, having the control on the privacy leakage. On the other hand, the selection of ν impacts on the number of performed exponentiations, and thus, its overhead. Therefore, the peer should always select the privacy threshold trading off between privacy and energy.

VII. Conclusion and Future Work

In this paper, we presented PRM, the first solution for privacy-preserving interference discovery among IEEE 802.15.4 networks. PRM allows the Coordinators of two IEEE 802.15.4 networks to discover shared cells in the Radio Scheduling Tables, causing interference, while keeping the whole RST private. PRM integrates and smartly combines two dedicated algorithms, namely, the plain-PSI and the iterative set division approach, performing PSI on elements representing a distinctive amount of occupied cells in the RSTs. Besides designing the overall protocol and proving its security, we also provided an experimental performance assessment using real heterogeneous Internet of Things devices, i.e., a Raspberry Pi and an ESPCopter. Our results show that, through a smart choice in the allocation of occupied cells in the RST, the iterative set division strategy can allow even constrained devices (such as the ESPCopter) to perform privacy-preserving interference discovery in a few seconds. On more capable gateways (e.g., the Raspberry Pi), when the percentage of occupied cells is below the 20%, PRM always concludes in less than 1 sec., proving to be a feasible solution.

Future work will consider the comparison among different strategies for the iterative set division approach (columns, rows, squares) and a detailed investigation on the energy consumption of PRM.

Acknowledgments

This work has been supported by the INTERSECT project, Grant No. NWA.1162.18.301, funded by Netherlands Organisation for Scientific Research (NWO). The findings reported herein are solely responsibility of the authors.

References

[1] H. Ning, F. Farha, Z. N. Mohammad, and M. Daneshmand, "A survey and tutorial on "connection exploding meets efficient communication" in the Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10 733–10 744, 2020.

[2] Statista, *Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025*, 2020, available at https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/, (Accessed: 31-3-2022).

[3] M. Palattella, N., Nicola, X. Vilajosana, et al., "Standardized Protocol Stack For The Internet Of (Important) Things," *IEEE Commun. Surveys Tuts.*, vol. 15, 01 2013.

[4] R. T. Hermeto, A. Gallais, and F. Theoleyre, "Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey," *Computer Communications*, vol. 114, pp. 84–105, 2017.

[5] A. Karalis, D. Zorbas, and C. Douligeris, "Collision-Free Advertisement Scheduling for IEEE 802.15.4-TSCH Networks," *Sensors*, vol. 19, no. 8, 2019.

[6] S. Jeong, H.-S. Kim, J. Paek, and S. Bahk, "Ost: On-demand tsch scheduling with traffic-awareness," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 69–78.

[7] K. Phung, B. Lemmens, M. Goossens, et al., "Schedule-based multi-channel communication in wireless sensor networks: A complete design and performance evaluation," *Ad Hoc Networks*, vol. 26, pp. 88–102, 2015.

[8] A. Nikoukar, S. Raza, A. Poole, et al., "Low-Power Wireless for the Internet of Things: Standards and Applications," *IEEE Access*, vol. PP, pp. 1–1, 11 2018.

[9] B. Pinkas, T. Schneider, and M. Zohner, "Faster private set intersection based on {OT} extension," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 2014, pp. 797–812.

[10] E. De Cristofaro, J. Kim, and G. Tsudik, "Linear-complexity private set intersection protocols secure in malicious model," vol. 2010, 12 2010, pp. 213–231.

[11] E. De Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity," vol. 6052, 01 2010, pp. 143–159.

[12] A. Cerulli, E. D. Cristofaro, and C. Soriente, "Nothing Refreshes Like a RePSI: Reactive Private Set Intersection," Cryptology ePrint Archive, Paper 2018/344, 2018.

[13] P. Kotzanikolaou, C. Patsakis, E. Magkos, et al., "Lightweight private proximity testing for geospatial social networks," *Computer Communications*, vol. 73, pp. 263–270, 2016.

[14] N. Kobeissi, K. Bhargavan, and B. Blanchet, "Automated Verification for Secure Messaging Protocols and their Implementations: A Symbolic and Computational Approach," in *IEEE EuroS&P'17*, Apr. 2017, pp. 435–450.

[15] B. Blanchet, "Symbolic and Computational Mechanized Verification of the ARINC823 Avionic Protocols," in *IEEE CSF'17*, Aug. 2017, pp. 68–82.

[16] S. Sciancalepore, "PARFAIT: Privacy-preserving, secure, and low-delay service access in fog-enabled IoT ecosystems," *Computer Networks*, vol. 206, p. 108799, 2022.

[17] P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "PPCA - Privacy-Preserving Collision Avoidance for Autonomous Unmanned Aerial Vehicles," *IEEE Trans. on Depend. and Secure Comput.*, pp. 1–1, 2022.

[18] B. Blanchet, "Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif," *Foundations and Trends in Privacy and Security*, vol. 1, no. 1–2, pp. 1–135, Oct. 2016.

[19] Dominik Roy George, "PRM ProVerif Code," https://github.com/DominikRoy/PRM, 2022, (Accessed: 2022-08-10).

[20] R. Morabito, R. Petrolo, V. Loscri, et al., "LEGIoT: a Lightweight Edge Gateway for the Internet of Things," *Fut. Gen. Comp. Systs.*, vol. 81, 10 2017.

[21] B. Kang and H. Choo, "An experimental study of a reliable iot gateway," *ICT Express*, vol. 4, no. 3, pp. 130–133, 2018.

[22] ESPcopter, "ESPcopter: Programmable MiniDrone," https://espcopter.com/, 2021, (Accessed: 2021-06-17).