HARPAGON: An energy management framework for attacks in IoT networks

Emilie Bout, Valeria Loscrì, Antoine Gallais

HARPAGON: An energy management framework for attacks in IoT networks

Emilie Bout, Valeria Loscri, Antoine Gallais

***Abstract*—The Internet of Things (IoT) represents an eclectic paradigm that is still growing in popularity. However, security aspects are a major concern for IoT devices due to their applications and the amount of sensitive data they provide. Simultaneously, the energy constraint in IoT networks remains a significant issue due to their limited resources. To reduce their energy consumption, several IoT protocols have integrated the energy-saving mode which offers four operating modes. On the basis of these four states, we derive an analysis framework, named HARPAGON, allowing an attacker to maximize his attack efficiency and minimize his impact in terms of energy consumption. Indeed, the effectiveness of many attacks depends principally on the state of the attacker and the victim at the same time. HARPAGON with the help of Markov Chains Theory allows to model the interaction between the attacker and its victims. In this paper, we demonstrate the effectiveness of the framework coupled to a jamming attack by comparing it to other types of jamming attacks. Experimental results reveal HARPAGON combined with jamming attack drastically reduces the performance of the network, with an impact on the Packet Error Rate (PER), which is around 13% higher than the reactive attack and with a reduced energy budget in respect of other two well-known "green" jamming attacks.**

***Index Terms*—Green attacks, Energy-Effective, Wireless Networks, Markov Chain Theory, Internet of Things (IoT)**

I. Introduction

MANY wireless protocols have been widely proposed to interconnect and develop IoT networks in recent years. However, energy consumption in IoT networks remains a critical issue and developing new systems to reduce this consumption is essential. Devices alternate between four operating modes (transmitting, receiving, idle, sleep) to reduce the expenditure in energy in several IoT protocols. Each mode is employed for diverse purposes and provides different energy consumption levels. However, this mechanism can turn into a new security vulnerability. Several attacks like jamming or replay attacks aim to force the victim node to stay in the most consuming states. In parallel, this mechanism can also assist an attacker to consume less energy by predicting the optimal time of its attack.

Based on this process and Markov-Chain theory, it is possible to represent the interaction between an attacker and its victim and predict the moment of the attack. In this paper, we take the point of view of the attacker. We consider an aware wireless attacker node, able to switch between transmitting, receiving, idle, and sleep state in order to implement an effective attack. In [1], [2], [3] and [4], authors show the effectiveness of the neighbour discovery process based on the alternating/switching states of the wireless nodes in inquiry, scan and sleep state. We derive a similar theoretical framework specific for modelling an attacker node. Since the attacker is considered as a conventional node of the network, this framework is also available for representing a transmitter node. The main objective of our work is to derive an analytical framework based on Markov Chain Theory in order to characterise the different states of a typical wireless node. The derived analytical framework allows the attacker node to compute the probability of staying in each state in order to achieve the following objectives: a) Maximisation of the attack effectiveness as the probability that attack occurs in a timely manner (i.e., during the time slots used by the communicating victim node) by minimising the energy expenditure; b) Given a certain limitation cost, the maximisation of the probability that the attack is occurring in a certain time interval. This framework, named HARPAGON, can be applied to various types of attacks where their effectiveness essentially depends on the respective states of both the attacker and the victim. In attacks against the privacy as Secondary Usage (SU) [5], data are collected for different usage than the one initially consented to by data owners. For SU attacks, an attacker should spend its listening time in an effective way, in order to collect the maximum information while spending as little energy as possible. Similarly, in the case of jamming attacks, it is paramount that the attacker transmits in the same time as the victim, while spending the minimum energy. In this paper, we coupled HARPAGON with a jamming attack and evaluated its performance against different strategies of jamming attacks.

Our main contributions are summarised as follows:

- We develop a new framework to perform several types of attacks while minimising their energy and maximising their effectiveness.
- We detail a security analysis based on a formal framework IoTSAT [6]. IoTSAT is designed to automatically reveal a complex chain of attack vectors related to the predefined adversary goal. Based on this, we show the end goal of the attacker and the different attack vectors that the framework can exploit.
- After a concise mathematical analysis and estimation on Mathematica, we evaluate our framework with two types of attack in a real testbed. We prove their effectiveness

E. Bout and V. Loscri are with the FUN - Self-organizing Future Ubiquitous Network, Inria Lille-Nord Europe, Villeneuve-d'Ascq, France, e-mail: firstname.lastname@inria.fr.

A.Gallais is with the Laboratory of Industrial and Human Automation control, Mechanical engineering and Computer Science, CNRS, UMR 8201, Université Polytechnique Hauts-de-France, INSA Hauts-de-France, 59313 Valenciennes, France, e-mail: antoine.gallais@uphf.fr

This work was partially supported by a grant from CPER DATA and by the General Armament Direction, France and the Defense Innovation Agency, France

by comparing their impact and energy consumption with other existing strategies of attack.
- We discuss a real use case of this framework. We demonstrate that this type of framework has consequences on the performance of a network (delayed reception) but also repercussions on the energy consumption of the victims.

The rest of the paper is organised as follows. After a brief discussion about the related works in Section II, we describe the analytical framework modelling the attack process in Section III. In Section IV, we explain the performance of the system, and we analyse the theoretical results based on the attack probability and the cost factor. After this theoretical evaluation, we present the results obtained with HARPAGON in a real testbed, in Section V. Section VI describes a real use case with a WiFi system under different jamming attacks and provides our concluding remarks.

II. Related Works

The issue of security in the context of IoT is not recent. Many researchers are trying to improve this point by creating both security systems [7] and new attacks.

Indeed, taking the point of view of an attacker allows the discovery of new vulnerabilities and therefore the improvement of defence methods. More and more works concerning the invention of attacks are emerging. Like in [8], where the authors implement a new periodic jamming attack and prove it is possible to bypass classical detection mechanisms. Moreover, works on the use of machine learning for the implementation of attacks have multiplied in recent years [9].

However, in these numerous works, the consideration of the attacker's energy consumption is not taken into account. Designing attacks without taking action this metric can sometimes be unrealistic. Energy efficiency has become an important parameter to consider both from the point of view of the user and the attacker. This is why, studies on "green attacks" have emerged. In [10], [11], the authors focus on creating energy-efficient jamming attacks. In [10], authors create a model where the attacker estimates the distribution of the transmission period in the learning phase, and schedules its jamming attack according to this metric. Moreover, they attempt to optimise the learning and the attacking duration with the energy constraint. A new metric for evaluating the effectiveness of an attack relative to its energy consumed is introduced in [11]. Authors set a strategy of attack to listen and jam the network while minimising their energy consumption. They model their problem under an optimisation problem and try to define several parameters such as the attack mode, the eavesdropping rate and the optimal jamming power with a certain energy cost. However, these works focus on creating a jamming attack with the ability to minimise its listening time in order to collect information and choose the optimal attack parameters.

In this paper, we focus on choosing the optimal time to jam communication without requiring much information about the network and listening time. In addition, we propose a framework that can adapt to several types of attacks and wireless protocols. Compared to previous works, this model was tested in a real testbed.

III. Proposed HARPAGON Framework

Generally, the success of an attack depends on its interaction with its victim. This is the case with eavesdropping attacks which record communications on networks. Indeed, recording communications requires the attacker to be in listening mode at the same time as the target is sending a packet. Likewise, a jamming attack aimed at corrupting packets can only be successful if the attacker transmits a signal at the same time as the communication is taking place. HARPAGON is based on the interaction between the attacker and the transmitter. We name this process: Interaction Attacker Transmitter Model (IATM), it corresponds to the relation between the two nodes. For the rest of this article, we describe the HARPAGON model combined with a jamming attack.

Before describing the IATM, we need to describe the Attacker Node Model (ANM). The ANM represents the states the attacker node can be in, at a certain time t. Specifically, we associate four different macro states to an attacker node: Attack/Transmitting, Listening/Receiving, Idle and Sleep states. These diverse states reflect the different modes involved in existing power-saving mechanisms that are implemented in IoT communication protocols (e.g., IEEE 802.11, Bluetooth). The transmitter node Tx is characterised with the same macro states as the attacker node: Transmitting, Receiving, Idle and Sleep.

We now provide an overview of our HARPAGON framework. The workflow of this framework is described in Fig 1. As previously explained, HARPAGON is based on the four operating states present in several IoT communication protocols. Consequently, the first step of the attacker is to identify the IoT Communication protocol employed by their victim. The attacker can be equipped of several network interfaces corresponding to several communication protocols, which use the four states. It switches between these interfaces until it finds the communication protocol used by its victim. If the communication protocol is found, the attacker then chooses the attack to execute (e.g., jamming, eavesdropping, replay). The next step corresponds to the choice of the parameter that the attacker wishes to optimise. Indeed, the goal of HARPAGON framework is to find the optimal trade-off between the energy consumption employed by the attacker and the effectiveness of the attack. Consequently, the attacker can decide to either maximize its percentage of success (by entering these data) or minimize its energy consumption according to a defined probability success threshold. For example, in the former case, the attacker wants to perform a jamming attack with WiFi protocol and defines that the latter will have a success rate of 70%. Thanks to the information provided during these three steps, the IATM model allows HARPAGON to calculate the percentage of time that the attack spends in each state. A full explanation of this calculation is given in the next section. In the final step, the attacker is able to execute its attack, by varying the state of its attacking nodes according to the probabilities obtained previously. Consequently, before proceeding to calculations, the framework needs to know the communication protocol used. Moreover, as the attacker is equipped with several communication interfaces, it has the

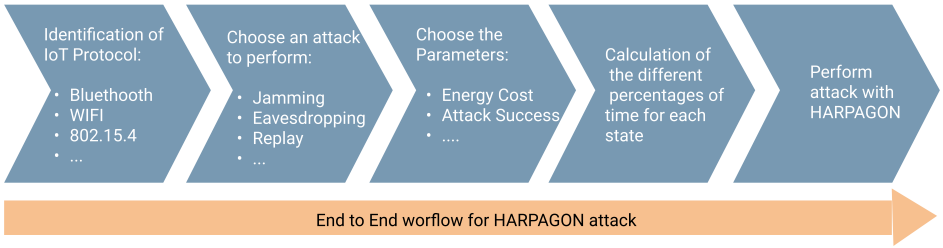


Fig. 1: HARPAGON system flow.

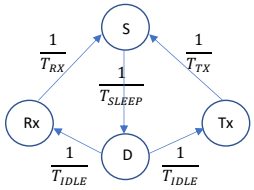


Fig. 2: Markov chain representation of an attacker node.

possibility of attacking multiple devices in a given area.

Before deriving the attacker process as per interaction between the attacker node and the transmitter node, we now describe the attacker/transmission model.

A. Attack/Transmission Node Model

This model concerns a single node. As already outlined, even though the objectives are different, the states for an attacker node and a transmitter node are as follows:

- Transmit(Tx)/Attack: the attacker sends packets in specific time slots in order to collide with the packets sent by a transmitter node. The transmitter sends packets to communicate with a receiver;
- Receive(Rx)/Listening: the attacker listens to acquire information of the communication system. The transmitter is in receiving state in order to receive the acknowledge packets and to establish the occupancy of the channel.
- Idle(D): the node is not transmitting/attacking neither receiving/listening but it keeps consuming some amount of energy since it can switch to Tx/Rx mode;
- Sleep (S): the attacker/transmitter switches to this state to reduce the power consumption. The amount of energy consumption associated to this state is assumed to be null in this work.

In order to model the Attacker/Transmitter Node, we consider Markov Chain Theory and each of the previous states is assumed as a Macro-State (Fig. 2). In order the attack to be effective, the jamming node has to transmit in the same time slot as the transmitter . Let consider M as the number of time slots the attacker uses for transmitting; its current state at any instant time t can be T_{xi}, R_{xi}, D or S. In practice, they represent the state of the Markov chain in which the node attacks (T_{xi}), listens (R_{xi}), is idle or sleeps.

Let us introduce the space state of the attack node F^J, the attacker node model state can be expressed as:

$$S_c^{(J)}(t) \in F^{(J)} = \{R_{x_1}, \ldots, R_{x_M}, T_{x_1}, \ldots, T_{x_M}, D, S\} \quad (1)$$

By considering Q^A as the transition rate matrix, we can represent the transition frequency from a state to another as the generic element $[Q^J]_{S_1,S_2} \in Q^{(J)}$.

By indicating with π^J the steady-state probability vector of the AN model, we can evaluate it by solving the linear system of equations:

$$\begin{cases} \pi^{(J)} \times Q^{(J)} = 0 \\ \sum_{s \in F^{(J)}} [\pi^J]_s = 1 \end{cases} \quad (2)$$

where $[\pi^A]_{[s]}$ is the generic element of the vector $\pi^{(A)}$, namely the probability P to be in a certain state s:

$$[\pi^{(J)}]_{[s]} = P\{S_c^{(J)}(t) = s\} \quad (3)$$

The transition rate matrix can be represented as:

$$Q_J = \begin{pmatrix} -\frac{1}{T_s} & \frac{1}{T_s} & 0 & 0 \\ 0 & -\frac{2}{T_{idle}} & \frac{1}{T_{idle}} & \frac{1}{T_{idle}} \\ \frac{1}{T_{Rx}} & 0 & -\frac{1}{T_{Rx}} & 0 \\ \frac{1}{T_{Tx}} & 0 & 0 & \frac{1}{T_{Tx}} \end{pmatrix} \quad (4)$$

B. Attack Process Model

As already outlined, the Interaction Attacker Transmitter Model (IATM) is based on the interaction between the attacker node and the transmitter node. We consider the attacker and the transmitter alternate between the four different states.

Let us consider that the attacker node starts its attack at time $t \geq 0$. Thus, the jamming node J and the transmitter node Tx are in the coverage area of each other. The attacker process state is represented as:

$$S^{(IATM)}(t) = (S^{(J)}(t), S^{(Tx)}(t)) \in F^{(J)} \quad (5)$$

The state space of the attack process $F^{(IATM)}$ is given by the Cartesian product of the two spaces of the single nodes, namely:

$$F^{(IATM)} = F^{(J)} \times F^{(Tx)} \quad (6)$$

We consider an array $\pi^{(IATM)}$, where a generic element $\pi^{(IATM)}(t)_{[S_1,S_2]}$ represents the probability that the attack process at the time t is $[S_1, S_2]$:

$$\pi^{(IATM)}(t)_{[S_1,S_2]} = P\{S^{(IATM)}(t) = (S_1, S_2)\} \quad (7)$$

and $\pi^{(IATM)}(t)$ can be computed as:

$$[\pi^{(IATM)}(t)] = \pi^{(IATM)}(0) \times e^{Q(IATM)*} \quad (8)$$

where $\pi^{(IATM)}(0)$ represents the state probabilities array at the time $t = t_0$ when the jamming node starts the attack.

Before the time t_0, we consider the two processes associated with the attacker and transmitter nodes are independent and identically distributed (i.i.d.) and then the $\pi^{(IATM)}(0)$ can be computed as the Cartesian product of the steady states probability of each process, that is:

$$\pi^{(IATM)}(0) = \pi^{(J)} \times \pi^{(Tx)} \quad (9)$$

where each factor can be calculated by solving the equations system as in 2. In particular, $Q^{(IATM)}$ represents the transition matrix of the rates for the attack process.

The generic element of the matrix $Q^{(IATM)}$ is:

$$\begin{cases} [Q^{(IATM)}]_{(S_1^{'},S_2^{'})}, (S_1^{''}, S_2^{'}) = \{Q^{(J)}_{[S_1^{'},S_1^{''}]} if(S_1^{''} = S_2^{''}) \\ Q^{(Tx)}_{(S_2^{'},S_2^{''})} if(S_1^{'} = S_1^{''}) \\ -\sum_{(S_1,S_2)\in F^{(IATM)}-\{(S_1^{'},S_2^{'})[Q^{IATM}]_{[(S_1^{'},S_2^{'}),(S_1,S_2)]}}, \\ if(S_1^{'}, S_2^{'} = (S_1)^{''}, S_2^{''})\} \end{cases} \quad (10)$$

The attacker will be able to interfere with the transmitter node if and only if the Tx node is transmitting and the attacker is also sending a packet in the same time slot.

If at the time t the transmitter is emitting on a slot m, the state of the transmitter node will be $S^{(Tx)}(t) = Tx_m$ and in the same time slot t, the jamming node starts the transmission on the same slot, then its state will be $S^{(J)}(t) = Tx_m$ and then it results that:

$$\begin{cases} S^{(Tx)}(t) = Tx_m \\ \lim\limits_{\tau \to t^-}(\tau) \neq S^{(J)}(t) = Tx_m \end{cases} \quad (11)$$

The attack occurs at time t if and only if the conditions expressed by the equations 11 are satisfied.

The array rates of attack Λ^{attack} and more specifically the generic element $[\Lambda^{attack}]_{S_1,S2}$ representing the attack rate when the state of the attack process is (S_1, S_2) can be defined in respect of the number of slots M as:

$$\begin{cases} [\Lambda^{(attack)}]_{[S_1,S_2]} = \sum_{m=1}^{M}([Q^{IATM}])_{[(S_1,S_2),(Tx,Tx)]} \\ 0, otherwise \end{cases} \quad (12)$$

$if(S_1, S_2) \neq (Tx, Tx)$.

The total rate of attack is then computed as:

$$\rho(t) = \pi^{IATM}(t) * \Lambda^{(attack)*} \quad (13)$$

where $\Lambda^{(attack)*}$ is the transposed vector of $\Lambda^{(attack)}$.

C. No-attack case

Let us indicate with $\pi^{(attack)'}_{T\geq t}(t)$ the array where the generic element $\pi^{(attack)'}_{T\geq t}(t)_{(S_1,S_2)}$ represents the probability that the state of the attacker in respect of the transmitter node is (S_1, S_2) by considering that there is no attack:

$$[\pi^{(attack)'}_{T\geq t}(t)] = P\{S^{IATM}(t) = (S_1, S_2) \mid T \geq t\} \quad (14)$$

In order to evaluate $\Pi^{(attack)'}_{T\geq t}(t)$ we need to consider the $Q^{IATM}_{noattack}$ matrix of states transition rates, since no attack occurs. The matrix $Q^{IATM}_{noattack}$ can be computed by Q^{IATM}. Let us consider that:

$$Q^{IATM}_{noattack} = Q^{IATM} \quad (15)$$

The elements of the matrix for which the attacker tries to attack are set to zero. Thus, for each pair of the states of the attack process $[(S_1^{'}, S_2^{'}), (S_1^{''} S_2^{'}]) \in \{F^{(IATM)} \times F^{(IATM)}\}$, satisfying one of the following conditions:

$$\begin{cases} S_1^{'} \neq S_1^{''} = Tx_m and S_2^{'} = S_2^{''} = Tx_m \forall m \leq M \\ S_1^{'} = S_1^{''} = Tx_m and S_2^{'} \neq S_2^{''} = Tx_m \forall m \leq M \end{cases} \quad (16)$$

we put:

$$[Q^{IATM}_{noattack}]_{[(S_1^{'},S_2^{'}),((S_1^{''},S_2^{'})]} = 0 \quad (17)$$

then we can compute:

$$\pi^{IATM}_{T\geq t}(t) = \pi^{(IATM)}(0) \times e^{Q^{IATM}_{noattack} * t} \quad (18)$$

On the basis of the equation $\rho(t) = \pi^{(IATM)}(t) * \Lambda^{(IATM)*}$ characterizing the total frequency of attack, we can derive:

$$\rho_{T\geq t}(t) = \pi^{IATM}_{T\geq t} \times \Lambda^{(attack)*} \quad (19)$$

Let us consider now the attack process represented as a Markov Chain with four states. The matrix of the state transitions rate $Q^{(IATM)}$ is a matrix 16×16, where the generic element can be computed as:

$[Q^{(IATM)}]_{(S_1^{'},S_2^{'}),(S_1^{''},S_2^{'})} =$

$$\begin{cases} \frac{1}{T_{IDLE}} \\ if(S_1^{'} = S_1^{''} and S_2^{'} = IDLE \neq S_2^{''}) \\ or(S_1^{'} = IDLE \neq S_1^{''} and S_2^{'} = S_2^{''}) \\ \frac{1}{T_{SLEEP}} \\ if(S_1^{'} = S_1^{''} and S_2^{'} = SLEEP \neq S_2^{''}) \\ or(S_1^{'} = SLEEP \neq S_1^{''} and S_2^{'} = S_2^{''}) \\ \frac{1}{T_{RX}} \\ if(S_1^{'} = S_1^{''} and S_2^{'} = RX \neq S_2^{''}) \\ or(S_1^{'} = RX \neq S_1^{''} and S_2^{'} = S_2^{''}) \\ \frac{1}{T_{TX}} \\ if(S_1^{'} = S_1^{''} and S_2^{'} = TX \neq S_2^{''}) \\ or(S_1^{'} = TX \neq S_1^{''} and S_2^{'} = S_2^{''}) \\ -\sum\limits_{\substack{(S_1^{''},S_2^{''})\neq(S_1^{'},S_2^{'} \\ (S_1^{''},S_2^{''})\in F^{(AP)}}} [Q^{(AP)}]_{[(S_1^{'},S_2^{'}),(S_1,S_2)]}, \\ if(S_1^{'}, S_2^{'} = (S_1)^{''}, S_2^{''}) \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases} \quad (20)$$

The attack will be effective if at time t the jamming node and the transmitter node will be both in the transmitting state.

IV. Theoretical Evaluation

In this section, we characterize the performance of the system presented in the previous section. Results have been obtained with Mathematica. We evaluate the effectiveness of an attacker node in respect of costs constraints and maximization of the probability that the attack occurs in a certain time. In order the attack to be effective, the attacker has to send a packet in the same time slot when the transmitter node is transmitting. The complete code of the algorithm to generate the HARPAGON attack is available in [12].

We then introduce the concept of cycle time as a time interval between two sleep states of the jamming node. The average time of a cycle time can be defined as the sum of the different times a node spends in one of the previously defined states:

$$T_{CYCLE} = T_{IDLE} + T_{RX} + T_{TX} + T_{SLEEP} \tag{21}$$

Based on the cycle time, we can calculate the probability of our attacker to be in one of the four states:

$$\begin{cases} P_{IDLE} = \dfrac{T_{IDLE}}{T_{CYCLE}} \\ P_{RX} = \dfrac{T_{RX}}{T_{CYCLE}} \\ P_{TX} = \dfrac{T_{TX}}{T_{CYCLE}} \\ P_{SLEEP} = \dfrac{T_{SLEEP}}{T_{CYCLE}} \end{cases} \tag{22}$$

and

$$P_{IDLE} + P_{RX} + P_{TX} + P_{SLEEP} = 1 \tag{23}$$

A. Energy Cost

The different states associated with the nodes are characterised with different energy consumption and the energy spent by the jamming node depends on the amount of time spent by the node in each of the four states.

For the sake of clarity, we assume that the energy consumption associated to the sleep state is negligible and we evaluate then the total cost based on the time spent for transmitting (TX state), for listening (RX state) and when the node is idle (IDLE state). The cost can be derived as:

$$cost = PW_{TX} \times [\pi^{(AN)}]_{[TX]} + PW_{RX} \times [\pi^{(AN)}]_{[RX]} + PW_{IDLE} \times [\pi^{(AN)}]_{[IDLE]} \tag{24}$$

where PW_{TX}, PW_{RX} and PW_{IDLE} represent the power consumption of the jamming node for each state, i.e., transmitting, listening and idle respectively.

Of course, the cost variable is comprised in the $[0, \max(PW_{TX}, PW_{RX}, PW_{IDLE})]$ interval.

As observed in most wireless devices (e.g., WiFi and Bluetooth as considered here), the largest proportion of power consumption is associated with the transmitting state. We thus derive that $0 < cost < PW_{TX}$.

B. Attack Probability Distribution Function

Let us consider T as the time of the jamming attack to be successful, the probability distribution function can be computed as in [13]:

$$F_T(t) = 1 - k \times e^{-A(t)} \tag{25}$$

where

$$k = 1 - ([\pi^{AP}]_{(TX,TX)}) \tag{26}$$

and

$$A(t) = \int_0^t \rho_{T \geq t}(\tau) d\tau \tag{27}$$

where $\tau_{T \geq t}(\tau)$ is defined as in equation (19) as total frequency for the attack.

C. Theoretical results analysis

Based on the developed framework, we consider two different objectives:

- Maximization of the attack probability in a certain time interval t which corresponds to $\max[F(t)]$, with an energy consumption lower than a certain threshold, meaning $cost \leq c$;
- Minimization of the energy consumption by imposing the attack should occur in a certain time interval, namely $F(t) \geq t$.

1) Objective 1: Maximization of the probability that the attack occurs in a certain time interval: This objective can be translated as the need to compute the parameters of the attack node model in terms of listening probability (P_{RX}), attacking probability (P_{TX}), idle probability (P_{IDLE}) and sleep probability (P_{SLEEP}).

The cost function can be expressed as:

$$cost = PW_{RX} \times P_{RX} + PW_{TX} \times P_{TX} + PW_{IDLE} \times P_{IDLE} \leq c \tag{28}$$

The quadruple (P_{RX}, P_TX, P_{IDLE}, P_{SLEEP}) with the optimal energy cost c^* satisfies the following equations:

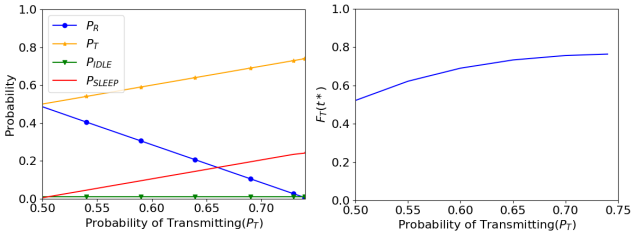
$$\begin{cases} 0 \leq P_{TX} \leq P_{TX}^{max} \\ P_{IDLE} = k \\ P_{RX} = \frac{cost - PW_{IDLE} \times P_{IDLE} - PW_{TX} \times P_{TX}}{PW_{RX}} \\ P_{SLEEP} = 1 - P_{RX} - P_{IDLE} - P_{TX} \end{cases} \tag{29}$$

without less of generality, we arbitrarily assign the value of P_{IDLE} as comprised in $0 < k \leq 1$. The value of P_{RX}^{max} can be calculated by the inequalities:

$$0 \leq P_{RX} \leq 1, 0 \leq P_{TX} \leq 1 \tag{30}$$

First, we have assigned different power values for each state which are respectively: $PW_{Rx} = 0.34W$, $PW_{Tx} = 0.67W$, $PW_{Idle} = 0.30W$ and $PW_{Sleep} = 0.01W$. With these different power values, we computed the combined values of

the different probabilities with various associated costs representing a constraint, namely $cost = 0.35(W)$, $cost = 0.4(W)$, $cost = 0.5(W)$ and $cost = 0.6(W)$. We here present the results obtained with a $cost = 0.5(W)$ (Fig. 3a).



(a) (b)

Fig. 3: a) P_{RX}, P_{TX}, P_{IDLE} and P_{SLEEP} quadruples with $cost = 0.5$. b) Values of $F(t)$ corresponding to energy cost equal to c= 0.5W and $t/T = 2$.

The probability distribution function $F(t)$ does not depend on the specific values of t and T, but rather on the t/T ratio. Consequently, in the following study of this scenario, the ratio will be considered as a system parameter instead of t and T separately. For each of the quadruples computed in Fig. 3a, it is possible to evaluate the $F(t)$ value according to P_{TX} and a t/T ratio, by using the developed framework. The results for a $cost = 0.5$ and $t/T = 2$ are shown in Fig. 3b. Combining the results shown in Fig. 3a and 3b, and assuming $t/T = 2$, the maximum value of $F(t)$ can be obtained (noted as $F(t)_{max}$). When the cost constraint is fixed to c = 0.5W, $F(t)_{max}$ = 0.763712 (Fig. 3b), the values of the quadruple are:

$$\begin{cases} P_{RX} = 0.005, P_{TX} = 0.74, P_{IDLE} = 0.01, P_{SLEEP} = 0.25 \\ \text{if } (c = 0.5W) \end{cases}$$

2) Objective 2: By imposing a threshold in terms of probability the attack occurs in a certain interval time, we minimize the associated cost: For this second objective, we consider that the attack has to occur in a certain interval time with a certain probability and we need to minimize the energy consumption associated to these constraints. As for the first objective, for reasons of space, we will present here only the results obtained for $F(t) = 0.7$ and t/T=2.

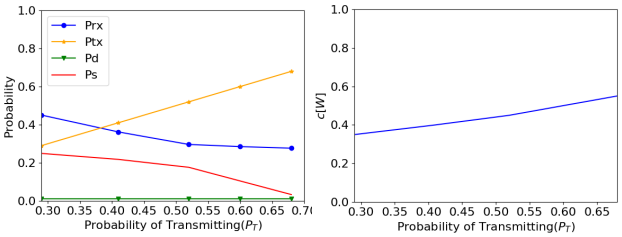
For this parameters, the combined values of the probabilities P_{RX}, P_{TX}, P_{IDLE} and P_{SLEEP} are shown in Fig. 4a.

Based on the quadruples values, we evaluate the cost variation in respect of the attack probability (see Fig. 4b).

For higher values of attack probability the associated cost is higher as well. Based on Fig. 4a and Fig. 4b, we can obtain the minimal cost C_{MIN} in respect of the attack probability $F(t) = 0.7$ and deduce the values for each state. When $F(t)_{max}$ = 0.7, the values of C_{MIN} = 0.35 and the values of the quadruple are:

$$\begin{cases} P_{RX} = 0.45, P_{TX} = 0.29, P_{IDLE} = 0.01, P_{SLEEP} = 0.24 \\ \text{if } (\frac{t}{T} = 2) \end{cases}$$

According to the procedure presented in this section and by imposing any threshold in terms of probability the attack occurs in a given interval time, it is possible to minimize the associated cost and deduce the values associated for each state.



(a) (b)

Fig. 4: a) P_{RX}, P_{TX}, P_{IDLE} and P_{SLEEP} quadruples with $F(t) = 0.7$ and t/T=2. b) Values of c corresponding to $F(t)$ = 0.7 and t/T =2.

D. Formal Security Analysis

In this section, a security analysis is given with the help of a framework IoTSAT developed in [6]. This framework models the behavior of IoT networks, thus allowing to identify potential threat vectors and the adversary's specific technique. It is composed of four main elements: i) IoT Topology Model, ii) Interactions Constraint Model, iii) The Attack Constraints, iv) Threat Model. To facilitate the explanation of this security analysis, we rely on a basic two-node network: one sensor (transmitter) periodically sending temperature data to one controller (receiver). Let $Rep(X_{S,C})$ denote the reported temperature X, as sensed by S and further reported at the Controller C. If $Rep(X_{S,C})$ exceeds a threshold δ, the controller generates an action which is to stop the operation of a machine $Cmd(STOP)$. Consequently, the actuator's function for this example is:

$$R_{1,1} = \{(Rep(X_{S,C}) \vee X > \delta), Cmd(STOP)$$

The attack constraints are subdivided into two objectives: the adversary's capabilities and the adversary's goal. Consequently, the attack model developed for the HARPAGON framework can be modelled as follows:

$$\textit{Attacker Goal} : Act(N_1) \wedge Max(Impact) \wedge Min(Energy)$$
$$\textit{Capabilities} = CN = 1 \wedge CL = 1 \wedge EC \leq c$$

where *Act()* corresponds to the Action of Attack, *Max()* the maximum function and *Min()* the minimum function. *CN* represents the maximun of victim node and *CL* the number of link. To finish, *EC* is equivalent to the energy consumption. Consequently, in this context, the goal of the attacker is to attack one node *N* and one link while maximising its impact and minimising its energy consumption. The energy consumption is lower or equal to the cost c. IoTSAT classifies IoT threats as interlinked threat vectors, where injection of one vector by the attacker can trigger a chain reaction, impacting multiple IoT entities. Basing our analysis on the threat model of the IoTSAT models, three main threat vectors can be exploited by HARPAGON if sensor devices are used. Indeed, IoTSAT framework includes five threats for the sensor context: denied, incomplete, inconsistent, tailored and fabricated contexts.

HARPAGON is based on the interaction between the attacker and the victim. The goal for the attacker is to intercept a packet in order to carry out an attack. Therefore, with this framework an attacker can intercept and delay, block or corrupt a packet. Moreover, by intercepting a packet, it also has the possibility of reading and modifying it. Consequently, with HARPAGON, the context can be denied (DC) if the information is blocked at the node level or at the network level. In practice, the context can be incomplete (IpC) if the information is delayed or not fully delivered (e.g., corrupted packet, fragmented packet but part is missing). Finally, the context can be tailored (TC) if the information has been modified directly at the level of the node or during the sending. As explained in [6], the formal representation of the threat for the sensor is defined below:

$$DC^t = \neg Obs(X^t_{a,b}) \vee \neg Reach^t(S_b, C_d)$$
$$\text{IpC}^t = Obs(X^{t+1}_{a,b}) \vee \neg Reach^t(S_b, C_d) \vee Reach^{t+1}(b, d)$$
$$\text{TC}^t = Obs(X^t_{a,b}) \wedge Rep(\overline{X}_{b,d}) \wedge (X \neq \overline{X})$$

where $Obs(X^t_{a,b})$ corresponds to the observation of an event a effectuated by the Sensor b at the moment t with the X(int) value. The function $Reach^t(S_b, C_d)$ returns true, if a valid communication connection exists from Sensor b to a Controller d. Eventually, $Rep(X_{b,d})$ is the predicate returning true if the value X(int) of the event a is received by the Controller d and reported by the Sensor b. These three threats present in a sensor device and directly exploitable by HARPAGON, impact the trigger function of IoT devices. If the context is denied (DC), the trigger function is blocked (BT). Indeed, the decision process, carried out by function $F()$, which takes the parameter a transmitted by the sensor, can remain blocked until a has been provided. If the context is incomplete, the received value is inadequate with the decision process and the trigger is also incomplete (IT). To finish, with a tailored context, the decision process is altered by the received value a, the trigger is false (FT). We define the threats for the triggers function as follows:

$$BT^t = DC(S_b, C_d)$$
$$IT^t = \neg MCF(C_d, F_a)$$
$$FT^t = \textstyle\sum_{\forall S_b \in S^{ad}} [TC(S_b, C_d)] \geq v_{ad}$$

where *MCF* is the Minimal Context Fusion which corresponds to the sum of the Reach(S_b, C_d).

$$MCF(C_d, F_a) = \textstyle\sum_{\forall S_b \in S^{ad}} [Reach(S_b, C_d)] \geq w_a d$$

Indeed for certain scenarios, several values of sensor may be needed to compute the return value of the trigger function F_a. Consequently, if the connection link between a Sensor S_b is lower than the number of data required $w_a d$ to compute the trigger function, the trigger is incomplete. If the trigger function can be computed, but returns a false result (FT), it means that the sum of the tailored context is greater than the number of ratio values needed to compromise the result v_{ad}. To finish, in an IoT system, most of the trigger functions conduct to an action. Consequently, the anomalies in trigger function cause consequences in the actuation function. Always based on the IoTSAT framework, we see that an incomplete or blocked trigger function conducts to the denied or delayed action (DA). Moreover, an False Trigger (FT) causes an incorrect actuation (IA). These two consequences are formulated as follows:

$$DA(A_f, E_i) = IT(C_d, F_a) \wedge BT(C_d, F_a)$$
$$\text{IA}(\text{A}_f) = FT(C_d, F_a) \vee Reach(C_d, A_f)$$

where A_f is the f actuator and E_i is equivalent to the i Service. Based on the IoTSAT framework, we identified the threat vectors exploited by the HARPAGON attack. Indeed, a blocked or modified data causes incorrect or delayed action. In addition, in the most important cases, the action will not be executed. In our example, if the data sent by the sensor is blocked or delayed, the controller never stops the machine in time, even if the temperature is higher than the warning threshold. If the data is modified, the controller can take the decision to stop or no-stop the machine in inappropriate moment.

Several attacks are based on these threat vectors. In the physical layer, jamming attacks aim to block or delay data in transition. The modification of a packet can be realised with a replay attack. Therefore, HARPAGON can help the attacker to perform jamming, replay, modification and eavesdropping attacks while minimising their energy and maximising their impact.

V. Experiment Performance Evaluation for HARPAGON combined with Jamming attack

Following our theoretical analysis, we evaluated this framework on a testbed. In a jamming attack, the attacker has the possibility of intentionally occupying the channel for a long time or causing collisions in order to corrupt the packet. Therefore, one of the major difficulties is transmitting a signal at a specific time. This is why, in this section, we evaluated HARPAGON combined at this type of attack.

A. Experiment details

We evaluated HARPAGON on a real testbed, composed of two legitimate nodes and one attacker, as we can see in Fig. 5.

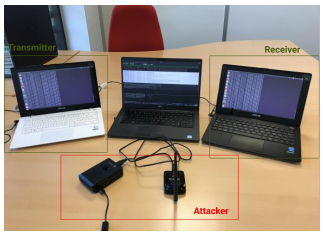


Fig. 5: Testbed composed of three nodes: one attacker (Raspberry Pi), two legitimate nodes and one access point.

The legitimate nodes of the network are assumed to be homogeneous in order to reduce the probability of side-effects. Hence, the sender and the receiver are two classic and identical laptops with the same type of network adapter: a Qualcomm Atheros AR9485 Wireless Network Adapter. Note that a multi-interface attacker could target heterogeneous victims in a given area, provided that the used communication protocols are

known from the attacker, which is the prerequisite of this framework. However, we here choose two identical nodes to simplify the measurement (same characteristics of the network card, etc.). These two nodes are connected to an access point with the IEEE 802.11 protocol and a classical TCP/IP stack on top. The transmitter scans the channel before each transmission to detect the status of the latter. Indeed, if the channel is perceived as busy, it restricts the sending of packets. Otherwise, if the channel is detected as unoccupied, data will be transmitted. Moreover, once an acknowledgement is not received after a certain waiting time, it has the possibility to re-transmit the packet once.

HARPAGON was implemented in a *Raspberry-Pi* equipped with Alfa AWUS036h and Realtek RTL8187L device including the wireless chip *ath9k*. We voluntarily chose this equipment because the driver and firmware are open-source. We have modified the driver of the wireless chip to get direct control over MAC layer parameters, following the work of [14]. Furthermore, this device contains four operating states, as defined in Section III-A. Each state has a different energy consumption. We used the available information for a chip operating at 2.4 GHz [15].

Two types of jamming attacks in addition to the HARPAGON have been implemented in the Rasperry-Pi. We compared our framework with two classical jamming approaches that can be found in the literature:

- **Constant Jamming:** The strategy consists of continuously sending packets on the channel. The jammer does not scan the communication medium and transmits regardless of the state of the latter. Its principal objective is to occupy the transmission channel for a certain time.
- **Reactive Jamming:** This type of attack aims to minimise the risk of being detected. The attacker jams the channel only when a transmission occurs, in order to cause a collision.

In this experimentation, the reactive attack has been implemented to jam only "data" type packets. This implies that the attacker must be in a listening state in order to detect the victim frame, this corresponds to the detection time t_{detect}. Once the frame has been spotted, the attacker switches to transmission mode to inject a dummy frame in order to create a collision: the attack phase t_{jam}. So in theory, this kind of attack is only effective if it satisfies this formula:

$$t_{transmission} < t_{detect} + t_{jam} \tag{31}$$

where $t_{transmission}$ corresponds to the entire time of the transmission of the legitimate frame. The transmission time depends in part on the size of the packet. Therefore the success of the reactive attack is also based on the frame size. Based on these regards, during our experiments, we have considered different sizes of packets emitted by the transmitting node. Indeed, as the size of the packets may vary in a real environment, the forged packets have a random size ranging from 50 to 1400 bytes. These values correspond to the average size of a small packet (e.g., ACK packets) or a slightly larger packet (e.g., Data Packet). The duration of each experience equals four minutes. After thirty seconds of experimentation, the attacker begins to operate for a duration of two minutes. The main parameters of the experiments are listed in Table I.

Parameter Name	Parameter used
Receiver-Transmitter distance	1 meter
Size of transmitted packets	Between 50 and 1400 bytes
Size of jamming packet	50 bytes
Number of retransmission allowed	1
Waiting time for acknowledgment	1 second
Detection threshold	70%
P(W) of Sleep / Idle / Tx / Rx states	0.0001 / 0.3 / 0.67 / 0.34

TABLE I: Experimentation parameters.

B. Experiments Results

In this section, we present the performance of HARPAGON with the parameters evaluated in Section IV-C2. Therefore, $F(t)$, representing the success probability of the attack is fixed to 0.7 and the value of the maximum energy cost is equal to 0.35. Several metrics that we detail below were used to evaluate the different attacks.

*1) **Packet Delivery Ratio and Detection Time**:* One of the first metrics we used to estimate the effectiveness of the attack is the Packet Delivery Ratio (PDR). This metric is equal to the ratio of the total number of successfully received packets to the total number of sent packets. During our experiment, the global PDR of the network is refreshed after each transmission.

Based on this metric, statistical detection was implemented. This solution is based on the behaviour of the network without attack and has been fully explained in [16]. Indeed, on the basis of a network, we can determine the average of the PDR and hence define a detection threshold. Consequently, an attack is identified, if the PDR decreases below the detection threshold. As mentioned in Table I, we defined a detection threshold at 70%. In addition, to avoid the false positive alert, we set a number of observations at 5.

Fig. 6 represents the PDR measurements for each type of offensive. As we can discern, the PDR for a constant attack decreases significantly from the start of the attack. This is justified to the fact that the sender and the receiver are disconnected from the access point a few seconds after the start of the attack. Indeed, the constant attack occupies the entire communication channel, consequently the access point is no longer able to send management frames (beacon) to ensure synchronisation with these clients.

In addition, if we contrast the measurement of the PDR of the HARPAGON attack to that of the reactive attack, we can notice that the latter decreases gradually. Therefore, if we compare the detection time of each attack, as summarised in Table II, the HARPAGON attack is less detectable than the other types of jamming attacks. This type of attack is detected 6.36 seconds later compared to a constant attack and 5.67 seconds compared to a reactive attack.

Type of attack	Detection time (in seconds)
Constant	18.80s
Reactive	19.49
HARPAGON	25.16

TABLE II: Detection time for each attack.

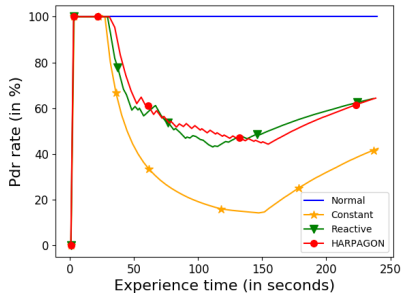


Fig. 6: Packet Delivery ratio for each type of attack.

*2) **Packet Error Rate and Number of Re-transmissions**:* In order to assess the number of corrupted data packets, we also calculated the packet error rate (PER) on the receiver side. The PER metric corresponds to the number of packets received with error divided by the total number of packets received. Fig. 7 presents the PER measurements for each type of attack. One of the first observations, we can perform is that the HARPAGON attack produces a more important error rate than the other type of attacks. Indeed at the end of the attack, the PER for the attack based on our framework is around 20%, against 9.5% for the reactive attack and 0% for the constant attack.

This can be explained by the fact that reactive attack is more likely to occupy the channel than to corrupt a packet when the latter is small. As mentioned in Table I, during these experiments, we vary the sizes of packets between 50 and 1400 bytes. Consequently, as formulated in 31, if the transmission time is inferior to the reaction time added to the jamming time, the reactive attack will not be able to corrupt the packet. Therefore, the jamming signal will be transmitted but will result in partial occupation of the channel.

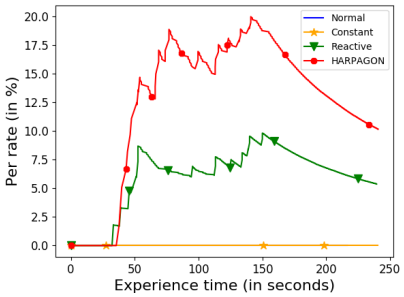


Fig. 7: Packet Error ratio for each type of attack.

The receipt of corrupted packets results in re-transmission in most cases. This is why, we also report the number of each re-transmission for each type of attack in Table III. We can observe that the HARPAGON attack leads to a higher number of re-transmissions than the other two types of attack. Indeed, for a network without a detection system, the number of re-transmissions for the attack based on the framework is 48 against 29 for reactive and 0 for constant attack. In addition, if we base our analysis on a network including a detection method described just above, the number of re-transmissions for the attack created is greater before detection than the other type of attack. Indeed, the number of re-transmissions before detection for the HARPAGON attack is 11, i.e., 1.8 times more than for the reactive attack.

Type of attack	Without detection system	With detection system
Constant	0	0
Reactive	26	6
HARPAGON	48	11

TABLE III: Number of re-transmissions for each attack.

*3) **Received Signal Strength**:* The received signal strength is also a well-known metric for detecting jamming attack. However, as already observed ([17]), depending on the type of jamming, RSS metric cannot be used to prove that an attack has taken place. We have evaluated this metric, shown in Fig. 8. The RSS fluctuates less with the HARPAGON attack than with the other attacks, while approaching the normal behaviour of the network. Therefore, for this type of attack, based on the RSS metric, detection will not be feasible.

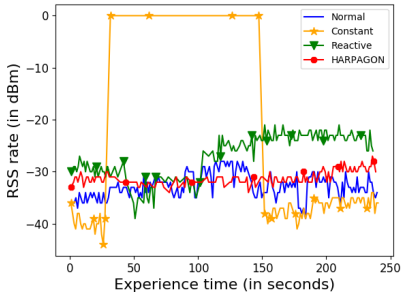


Fig. 8: Received Signal Strength for each type of attack.

*4) **Attacker Energy Efficiency**:* One of the main goals of the model is to reduce the power consumption of the attacker by switching the between different states of the attacker. To evaluate the energy consumption, we used the formula (24) defined in Section IV-A. As observed in Table IV, our HARPAGON attack spends less energy than the other two attacks. In fact, this new type of attack consumes 12.85 J less than the reactive and 41.4J less than the constant. Including the two idle and sleep states, in addition to those already used in the reactive attack, allows to consume less energy.

In [11], authors develop a new metrics: Attacker Energy efficiency (AEE) to measure the performance of a "green" attack. AEE is defined as a ratio of the impact of the network to the total power consumption of the attacker. In the case of jamming attack as demonstrated above, the impact of the network can be measured by packet error rate. Consequently, the HARPAGON attack has an AEE equal to 51 and the reactive attack to 18. In terms of AEE the HARPAGON attack is twice as effective.

C. Evaluation of HARPAGON

In this section, we compare our framework with other methods for generating green attacks. In [10], authors implement

Type of attack	Tx	Rx	Idle	CSleep	Energy Consumption Total
Constant	80,4 J	0	0	0	80,4 J
Reactive	22,4383 J	29,4134 J	0	0	51,8517 J
HARPAGON	16.2 J	23.31 J	0.003 J	0.288 J	39 J

TABLE IV: Energy consumption for each type of attack.

a new model to create green jamming attack called "LearnJam". In this solution, the attacker alternates between two phases: learning and attacking phase. During learning phase, the jammer keeps listening to the communication between two nodes. It records the time instances of incoming pulses over the wireless channel, which indicate the transmission instances. Based on the information the attacker obtained during its listening time, a time interval is calculated. This interval corresponds to the time between two transmissions. With the help of this metrics and the energy budget available, the jammer computes with an optimisation problem its active period time. Consequently, with a timer system the attacker alternates between two operating modes: sleep and transmission. The jammer wakes up at the beginning of the time transmission and jams the channel during its active period time. After, the jammer remains asleep until the end of the time interval between two transmissions. In [11], the authors also focus on creating a green jamming attack. First, the authors decompose this problem into three sub-problems and then obtain the jointly global-optimal solution by using outcomes of these three sub-problems. The first sub-problem concerns the optimal listening rate, which tries to find the optimal listening time to retrieve a maximum of useful information. The second, the optimal jamming power sub-problem, has been defined to find the optimal power of the sending signal to perform the attack. To finish, the last sub-problem concerns the optimal mode selection. The goal is to define the optimal time of jamming and eavesdropping. The jointly global-optimal solution combines the three results and tries to maximise the Attacker Energy Efficiency metric with a power constraint.

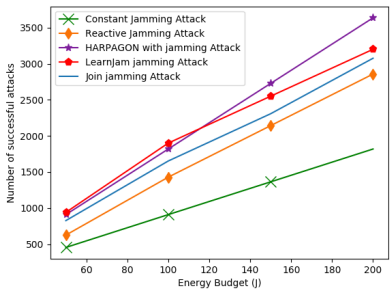


Fig. 9: Jamming strategies under different energy budgets.

Fig 9 represents the comparison of these two previous methods and HARPAGON combined with jamming attack according to the energy budget. We also compare this performance with the two classic jamming attacks: constant and reactive. For $50J$ of energy budget, the LearnJam attack performs 940 successful attacks against 826 for the jointly method and 909 for the HARPAGON framework. In addition, an attacker with $250J$ of energy can successfully jam a network 3200 times with the LearnJam jammer, 3076 times with the Join method Jammer and 3636 times with the HARPAGON jammer. Therefore, for a low energy budget, the LearnJam attack has a slightly better performance than HARPAGON attack, although the gap remains small (i.e 31 successful attacks). For the HARPAGON attack, the number of successful attacks is almost linear with the energy budget, as the probability of accomplishing a jamming attack is basically invariant with the elapse of time. Consequently, the higher the attacker's energy budget, the more effective the HARPAGON attack will be compared to other attacks. Indeed, in the LearnJam model, the higher the energy budget, the longer the learning phase will be. Moreover, our framework compared to the other two models has the possibility of maximising the listening or transmission time depending on the type of attack desired. Our framework is adaptable for different types of attacks and takes into account the idle state which is a mandatory transition state between sleep mode and receive/transmit mode in communication protocols.

A. *Discussion on the victim's lifetime*

VI. Discussion and Conclusion

Jamming attack can create severe and fatal consequences when IoT devices are used as decision aids. In some professional sectors, IoT networks are utilised for operational applications, often associated with maintenance and elementary denial of service attacks can produce immediate consequences. Indeed in some cases like in the chemistry industry, it is necessary to monitor the temperature and vibrations of industrial motors and detect the irregular operation in it. The sensors installed on these machines will ensure industrial maintenance by alerting to the slightest problem. Based on this example, below we show the consequences of HARPAGON combined with a jamming attack.

We suppose that the sensor has the same behaviour that the transmitter of the testbed. It sends its data once a day, which in a usual situation represents an average of 204 packets sent. Moreover, the sensor is equipped with a battery with a capacity of 133,200 J (10,000 mAh) and a Realtek RTL8187L device. In this case study, the energy expended by the sensor to carry out these measurements/calculations and when it is inactive, is not taken into account. Only the energy expended while sending this data is calculated. Based on these data, we are able to estimate the approximate life time of the sensor in a case without attack. An example calculation is listed in Table V.

First of all, we assume an attacker employs the constant jamming strategy and targets the sensor. As we noticed in Section V, from the beginning of the attack, the attacker has an impact on the transmission. Indeed, the packets are no longer transmitted, and the sensor will stay active. However, as we are reported during experimentation, the device will disconnect from the access point after a few seconds. Consequently, the

Process	Value
Number of packets per day	204
Battery Capacity	133,200 J (10,000 mAh)
Time of Tx mode for one transmission	0.45 s
Time of Rx mode for one transmission	0.40 s
Energy Spent in Tx mode for one transmission	0.3015 J
Energy Spent in Rx mode for one transmission	0.136 J
Energy Spent total for one transmission	0.4375 J
Total energy spent in 204 transmission per day	89.25 J
Expected device life time	4 Years

TABLE V: Battery life for an IEEE 802.11 sensor device.

administrator will immediately notice there is a problem with the sensor. This attack is effective in restricting communication, but it is clearly identified without a detection system.

As seen in the previous section, if the attacker uses a reactive jamming attack and no detection system is in place, it will generate an average of 26 re-transmissions. This represents an extra power consumption of 11.375 J per day to transmit 204 packets. Reported in terms of lifetime and taking into account only the energy consumption generated by the transmissions, the sensor will be operational for 3.62 years. However, with the HARPAGON attack, the attacker causes 48 supplementary transmissions, which represents an additional energy expenditure of 21 J. In this case, the existence of the sensor is equivalent to 3.31 years.

It is now assumed that the network has a detection system based on the threshold like explained above. Consequently, when the network detects an attack, the nodes change channel frequency to continue transmitting. Before the system detects a reactive attack, the device will send 6 extra packets which represents an added cost of 2.625 J. The lifetime of the sensor is therefore reduced by 0.05 years. However, if the attacker uses a HARPAGON attack, as we saw in Section V, the PDR decreases less quickly and this type of attack takes longer to be identified. As a result, the network will send 11 additional packets and consumes 4.8125 J more per day. Its lifespan will suffer a 14% reduction, decreasing from 4 to 3.43 years.

In conclusion, a HARPAGON attack increases the energy consumption of its victim by 15% when the network possesses a detection system. The reactive attack decreases the lifespan of its target by 1.25%, which is 13.75% less than HARPAGON attack. Moreover, if we assume our attacker is a node with the same capacity of the battery and a wireless chip as its victim, it is also possible to calculate its lifetime. Based on Table IV, for one reactive attack, the attacker consumes 51,8517 J and for the HARPAGON Attack 39 J. Consequently, a HARPAGON attack can be executed once a day for 9.35 years on this type of node against 7.03 years for the reactive attack. For the same duration of attack, HARPAGON attack allows to save 24.82% of energy compared to reactive attack while having a more significant impact of 15% on the lifespan of its victim. In recent years, new detection methods based on the victim's energy consumption metric have emerged [18], [19]. These methods are based on online time series statistic where the goal is to predict the energy consumption of a node and to detect an anomaly if the difference between the actual observation and the prediction is significant. This new type of method is relevant when attacks generate continuous damage. Reactive and HARPAGON attacks cause retransmissions but not continuously. Indeed, the probability of success of the reactive attack depends essentially on the size of the packets and the HARPAGON attack does not have a constant success rate over time. Consequently, the increase of retransmission and hence of energy consumption will not be constant and the detection will be more difficult. The difference generated by this type of attack and the prediction will be weaker and less apparent and will never be detectable. Therefore, using the energy consumption metric for more elaborate attack strategies is not effective. However, in the future, it could be interesting to combine this type of solution with other metrics so that the latter is also effective against attacks with temporary effects.

B. Discussion on HARPAGON framework

We have seen the effects of using HARPAGON coupled with a jamming attack. This new model is more effective compared to jamming attacks that we can find in the literature. Indeed, the employment of this framework considerably reduces the energy expenditure of the attacker. This factor is not negligible, especially in the IoT network, where the attacker can be also an IoT device. Moreover, the HARPAGON model can anticipate the optimal moment of the attack and produce severe consequences on its victim.

In this paper, we experiment our new model with the IEEE 802.11 protocol. However, the derived framework is general and can be applied to different wireless protocols, provided they include the four operating states. Moreover, this system is also adaptable for other types of attacks. Indeed, instead of optimizing the attacker's transmission period, we could slightly modify the framework to optimize the attacker's listening time during transmission while minimizing its energetic cost. As machine learning begins to be used to create attacks [9], HARPAGON can be utilized to collect data while keeping energy costs low.

From a different perspective, HARPAGON can be also used for effective prevention. Indeed, jamming attacks are employed in several security scenarios to prevent non authorized communications like, e.g., illegal drone surveillance [20]. Therefore, having an efficient jammer that consumes little power can be helpful. In the same way, the collection of data is essential to create a new machine learning based-security system. Hence, using HARPAGON combined with a passive attack to collect data could allow to reduce energy consumption during the design phase of the security system.

Finally, in this article, we experienced an attacker against one receiver and one transmitter. We considered one attacker per area in these first experiments. However it could be interesting in future to have several jammers in the same area in order to have collaborative strategies. Indeed, we think that in terms of attack efficiency, having two jammers in the same area is not going to have a large impact. A jammer transmits within a radius and jams all communications around it. However, studying the consequences of two HARPAGON attacker which could execute their attack in turn to save their energy but also to deceive the detection systems by being located at different places could be interesting. This idea could be realised by adding a state "collaboration" to the Markov chain for example. This state would be dedicated to the time of information exchange between the attacker nodes.

C. Conclusion and future works

With the advent of IoT, the security aspect has become a crucial issue. This is why designing new attacks to discover vulnerabilities is increasingly essential. In this article, we base ourselves on the vulnerabilities generated by the power-saving mode of wireless networks and show that this process can be used by an attacker to improve this efficiency. Indeed, we have developed the HARPAGON framework capable of modeling the interaction between an attacker and his victim. Simultaneously, this framework employs the different operating modes provided by the power saving mechanism to reduce the power consumption of the attacker. In this article, we have combined HARPAGON with a jamming attack and demonstrate the impacts of the latter in a real testbed. We show this type of attack can reduce the attacker's energy consumption by 24.82% and improve this impact by 13.75% compared to a classic jamming attack.

As future work, we aim to integrate this framework in a simulation network tool in order to evaluate the attacker effectiveness on a large network. We would also like to evaluate it against more advanced detection methods. Indeed, in this paper, we considered an attacker able to attack a distinct region in the network. It would be interesting to consider several attackers for one region. By adapting this framework, we can think of having multiple attackers in the same area who collaborate with each other in order to counter the defence system. For example, two jammers at different positions could alternate their attack phase while simultaneously minimizing their energy in order to deceive a defense system trying to locate attackers.

REFERENCES

[1] L. Galluccio, G. Morabito, and S. Palazzo, "Analytical evaluation of a tradeoff between energy efficiency and responsiveness of neighbor discovery in self-organizing ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1167–1182, 2004.

[2] L. Galluccio, A. Leonardi, G. Morabito, and S. Palazzo, "Tradeoff between energy-efficiency and timeliness of neighbor discovery in self-organizing ad hoc and sensor networks," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005, pp. 286a–286a.

[3] T. Salonidis, P. Bhagwat, and L. Tassiulas, "Proximity awareness and fast connection establishment in bluetooth," in *2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC (Cat. No.00EX444)*, 2000, pp. 141–142.

[4] V. Loscri, "An analytical evaluation of a tradeoff between power efficiency and scheduling updating responsiveness in a tdma paradigm," in *The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness Workshops*, ser. QSHINE '07. New York, NY, USA: ACM, 2007. [Online]. Available: https://doi.org/10.1145/1577222.1577269

[5] C. Perera, C. McCormick, A. K. Bandara, B. A. Price, and B. Nuseibeh, "Privacy-by-design framework for assessing internet of things applications and platforms," in *Proceedings of the 6th International Conference on the Internet of Things*, ser. IoT'16. New York, NY, USA: Association for Computing Machinery, 2016, p. 83–92. [Online]. Available: https://doi.org/10.1145/2991561.2991566

[6] M. Mohsin, Z. Anwar, G. Husari, E. Al-Shaer, and M. A. Rahman, "Iotsat: A formal framework for security analysis of the internet of things (iot)," in *2016 IEEE Conference on Communications and Network Security (CNS)*, 2016, pp. 180–188.

[7] W. H. Hassan *et al.*, "Current research on internet of things (iot) security: A survey," *Computer networks*, vol. 148, pp. 283–294, 2019.

[8] B. DeBruhl and P. Tague, "How to jam without getting caught: Analysis and empirical study of stealthy periodic jamming," in *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, 2013, pp. 496–504.

[9] E. Bout, V. Loscri, and A. Gallais, "How machine learning changes the nature of cyberattacks on iot networks: A survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021.

[10] Z. Yang, P. Cheng, and J. Chen, "Learjam: An energy-efficient learning-based jamming attack against low-duty-cycle networks," in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, 2014, pp. 354–362.

[11] B. Ahuja, D. Mishra, and R. Bose, "Optimal green hybrid attacks in secure iot," *IEEE Wireless Communications Letters*, vol. PP, pp. 1–1, 12 2019.

[12] A. G. Emilie Bout, Valeria Loscri. Code of harpagon attack. [Online]. Available: https://github.com/JammingWiFiNs3/HARPAGON

[13] K. Trivedi, "Probability and statistics with reliability queuing and computer science applications," 1992.

[14] M. Vanhoef and F. Piessens, "Advanced wi-fi attacks using commodity hardware," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 256–265. [Online]. Available: https://doi.org/10.1145/2664243.2664260

[15] A. Communications. Single-chip 2x2 mimo mac/bb/radio with pci express interface for 802.11n 2.4 and 5 ghz wlans. [Online]. Available: https://datasheetspdf.com/datasheet/AR9280.html

[16] M. Spuhler, D. Giustiniano, V. Lenders, M. Wilhelm, and J. B. Schmitt, "Detection of reactive jamming in dsss-based wireless communications," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1593–1603, 2014.

[17] A. Benslimane, A. El yakoubi, and M. Bouhorma, "Analysis of jamming effects on ieee 802.11 wireless networks," in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[18] M. Skowron, A. Janicki, and W. Mazurczyk, "Traffic fingerprinting attacks on internet of things using machine learning," *IEEE Access*, vol. 8, pp. 20 386–20 400, 2020.

[19] F. Li, Y. Shi, A. Shinde, J. Ye, and W. Song, "Enhanced cyber-physical security in internet of things through energy auditing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5224–5231, 2019.

[20] A. H. Abunada, A. Y. Osman, A. Khandakar, M. E. H. Chowdhury, T. Khattab, and F. Touati, "Design and implementation of a rf based anti-drone system," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 35–42.

Emilie Bout received her B.Sc degree and her M.Sc degree in computing sciences from the University Polytechnic des Hauts de France in 2017 and 2019. She is currently working toward a Ph.D in Inria Lille researching Networking and Cybersecurity partially granted by the DGA (General Armament Direction). Her main research interests focus on the creation of "smart" denial-of services attacks in wireless networks and their countermeasures.

Valeria Loscrì is a permanent researcher of the FUN Team at Inria Lille–Nord Europe since Oct. 2013. From Dec. 2006 to Sept. 2013, she was Research Fellow in the TITAN Lab of the University of Calabria, Italy. She received her MSc and PhD degrees in Computer Science in 2003 and 2007, respectively, from the University of Calabria and her HDR in 2018 from Université de Lille. Her research interests focus on emerging technologies for new communication paradigms such as VLC and TeraHertz bandwidth and cooperation and coexistence of wireless heterogeneous devices. Since 2019, she is Scientific International Delegate for Inria Lille.

Antoine Gallais is a Full Professor at INSA Hauts-de-France (Université Polytechnique Hauts-de-France), Valenciennes, France. He received M.Sc. (2004) and PhD (2007) degrees in computer science from the University of Lille, France, and was an associate professor at the University of Strasbourg, France, from 2008 to 2019. His main research interests lie in wireless ad hoc and mesh networks, actuator and sensor networks, Industrial Internet of Things, activity scheduling, routing and MAC protocols, mobile networks, fault-tolerance, cybersecurity and performance evaluation.