Demo Abstract: On preventing GTS-based Denial of Service in IEEE 802.15.4

Roberta Daidone
University of Pisa
Pisa, Italy
Email: roberta.daidone@iet.unipi.it

Gianluca Dini
University of Pisa
Pisa, Italy
Email: gianluca.dini@iet.unipi.it

Marco Tiloca
University of Pisa
Pisa, Italy
Email: marco.tiloca@iet.unipi.it

Abstract—The IEEE 802.15.4 standard features some optional services, including the Guaranteed Time Slot (GTS) mechanism. It provides network devices with collision-free access to the medium to assure Quality of Service. GTS suffers from a severe security vulnerability: an adversary can easily perform a Denial of Service attack by selectively jamming collision-free communications. We present Secure GTS, our solution to the GTS-based Denial of Service attack, and our implementation for the TinyOS platform on Tmote Sky motes. Our test application shows that Secure GTS manages to prevent Denial of Service attacks.

I. INTRODUCTION

The IEEE 802.15.4 communication standard [1] is designed for low-cost, low-power devices organized in a Personal Area Network (PAN), and is widely adopted for applications based on Wireless Sensor Networks (WSNs). Typically, these applications rely on devices with scarce hardware resources, and require to boost communication performance, or even achieve some Quality of Service (QoS) guarantees.

IEEE 802.15.4 provides the Guaranteed Time Slot (GTS) mechanism. GTS allows network devices to ask the PAN Coordinator for a dedicated time slot. If the request is accepted, they can communicate during an exclusively pre-assigned slot, so that no collisions occur while accessing the medium.

However, GTS has been proved to suffer from a severe security vulnerability. As described in [2], an adversary can easily perform a selective jamming attack, and disrupt communication even during dedicated GTS slots. This can jeopardize even the entire network activity, thus resulting in an actual Denial of Service attack.

Secure GTS is our solution to prevent the GTS-based Denial of Service attack, and consists in the following steps:

1) Hide the GTS slots assignment process from the adversary, by encrypting related information.
2) Shuffle the allocation of dedicated slots over time. This forces the adversary to perform the attack randomly.

We implemented Secure GTS for the TinyOS platform [3] and the Tmote Sky motes [4]. Our implementation is compliant with the IEEE 802.15.4 standard.

In order to test Secure GTS, we consider a simple application scenario with real sensor devices. We show that we succeed in preventing the adversary from disrupting collision-free communications, and that Secure GTS reduces the attack success rate up to 1/7.

In the following, we provide a brief summary of IEEE 802.15.4 and its GTS mechanism. Then, we describe the GTS-based Denial of Service attack, and present our solution. Finally, we show how our implementation of Secure GTS effectively prevents Denial of Service attacks.

II. IEEE 802.15.4 AND GTS

An IEEE 802.15.4 Personal Area Network (PAN) includes a PAN Coordinator, which is responsible for managing network activity. Communication can rely on the beacon-enabled mode, in which the PAN Coordinator periodically broadcasts beacon frames. Thus, the medium access is bounded as a sequence of superframes delimited by two consecutive beacon frames, and network devices are synchronized with each other.

A superframe is composed by an active portion and an inactive portion (see Figure 1). The active portion is composed by 16 equally sized superframe slots, and is divided into a Contention Access Period (CAP) and a Contention Free Period (CFP). During the CAP, devices access the medium on a contention basis, according to a slotted CSMA-CA algorithm. Instead, during the CFP, devices can ask the PAN Coordinator for a dedicated portion of the superframe, namely a GTS Slot. Thus, they are able to access the medium without colliding with each other. This mechanism is known as Guaranteed Time Slot (GTS), and is managed by the PAN Coordinator.

GTS Slots (Slots for short) can be composed by one or more superframe slots. Each device can send a GTS Allocation Request to the PAN Coordinator and ask for one Slot, specifying the amount of needed superframe slots and the traffic direction. Also, nodes can ask the PAN Coordinator to deallocate previously assigned Slots. Both allocation and
deallocation requests take place by means of *GTS Request Command* frames.

The PAN Coordinator assigns up to seven Slots in a *First Come First Served (FCFS)* fashion. Then, it includes in each beacon frame i) the list of devices whose request has been accepted; and ii) the time they are supposed to access the medium, i.e. when their Slot starts.

### III. ATTACK

GTS allocation requests are managed by the PAN Coordinator, which notifies the accepted ones using beacon frames. Broadcasting a beacon results in a GTS vulnerability [2].

We define the *sniper attacker*. He selects a target (i.e. a user), and exploits the knowledge of which users have been granted a collision-free Slot, in order to interfere with communications of his victim. Thus, the sniper attacker realizes a *Denial of Service (DoS)* attack against his target.

Figure 2 shows a GTS-based Denial of Service attack. The sniper attacker eavesdrops the medium to extract the GTS-related information from the beacon frame. Thus, the adversary becomes aware of during which specific Slot a specific user will access the medium. This means it is very easy for the sniper attacker to cause collisions between legitimate GTS clients and the PAN Coordinator, corrupt data, and *selectively* interfere with transmissions.

### IV. SOLUTION

GTS is vulnerable because the adversary can access the GTS-related information within beacon frames. A simple solution to avoid this would be *encrypting and authenticating* GTS-related information. However, the standard does not allow for encrypting the beacon payload portion containing GTS-related information. Thus, another solution is needed.

Secure GTS is our standard-compliant solution to the GTS-based DoS attack. Note that Secure GTS is not effective in case the adversary interferes with all transmissions by continuously jamming all available channels. However, we believe that, in a WSN, it is very likely that the adversary performs attacks by means of a sensor node. Therefore, it is very likely for him to behave as described, making Secure GTS effective. Secure GTS consists in two steps:

1) **MAC frames smart authentication and encryption.** Secure GTS prevents the attack by moving the GTS-related information to a different portion of the beacon frame. Thus, GTS-related information can be authenticated and encrypted. MAC command frames should be encrypted to avoid the adversary recognizing GTS Request Commands even by analyzing network traffic. Authentication prevents the adversary from spreading fake GTS Request Commands all around the network.

2) **Random Slots allocation.** According to the standard, if no deallocations occur, assigned Slots do not change their position in the CFP. As a consequence, the attacker is still able to infer the GTS-related information by observing the sequence of transmissions during the CFP. Secure GTS changes the position of Slots randomly on a superframe basis to make this analysis pointless.

We implemented Secure GTS by extending an open-source TinyOS implementation of the IEEE 802.15.4 standard [5]. We tested Secure GTS on a realistic scenario which includes four Tmote Sky motes: one PAN Coordinator, two sender nodes and one sniper attacker. The effectiveness of our countermeasure has been evaluated considering the probability of success of the attacker. Secure GTS reduces the probability of success of the sniper attacker to 1/7. Since the GTS information carried within the beacon is encrypted, the sniper attacker can only pick a Slot at random.

### V. CONCLUSION

We have presented Secure GTS, our solution to the GTS-based Denial of Service attack in IEEE 802.15.4. We implemented our solution for TinyOS on Tmote Sky motes, and tested it on a real application scenario. Our tests show that Secure GTS reduces the attack success rate up to 1/7.

### ACKNOWLEDGMENT

This work has been supported by EU FP7 Network of Excellence CONET (Grant Agreement no. FP7-224053) and EU FP7 Integrated Project PLANET (Grant agreement no. FP7-257649). Thanks also to Mário Alves and Ricardo Severino from ISEP for the useful feedback during the early stages of this work and the standard GTS implementation for TinyOS.

### REFERENCES


