Extended Abstract: Towards Physical-Layer Authentication for Backscatter Devices

Thiemo Voigt[1,2][0000−0002−2586−8573], Carlos Pérez-Penichet,[1][0000−0002−1903−4679], and Christian Rohner[1][0000−0002−1527−734X]

[1] Uppsala University Sweden
[2] RISE Computer Science, Sweden
firtname.lastname@it.uu.se

Abstract. Backscatter communications relieves sensor tags from the energy-intensive task of generating their own radio waves. This enables sensor tags to transmit their sensor readings at an energy consumption that is several orders of magnitude lower than that of conventional low-power radios. The resource-constraints of typical backscatter tags, however, make it challenging to provide security for them. In this extended abstract, we take a first step towards authentication of backscatter transmissions. We propose to add authentication information in the chip sequences of the physical layer. We discuss design issues and in particular the trade-off between security and reliability and propose mechanisms to enable low-power authentication suitable for backscatter tags.

Keywords: Backscatter · Authentication · Security

1 Introduction

Backscatter communications enable data transmissions while avoiding the need to generate a radio wave at the backscatter device, which is one of the most energy-consuming tasks for low-power Internet of Things (IoT) devices. Instead, an external device generates the carrier wave on which the backscatter tags modulate their sensed data values. Recent progress in backscatter communications enables IoT sensors and actuators to transmit physical-layer protocols such as Bluetooth [2], WiFi [8, 23], IEEE 802.15.4 (often called ZigBee) [15, 16] and LoRa [14, 18] with a power consumption below one milliwatt, several orders of magnitude lower than with conventional low-power radios and, in some cases, at distances in the range of kilometers [18]. This dramatic reduction of power consumption makes it increasingly feasible to power devices by energy harvested from the environment. At the same time, using commodity physical-layer protocols removes the need for an expensive dedicated device (RFID reader) to generate the required carrier wave on which the devices modulate their data. Backscatter promises large-scale data collection from battery-free IoT devices that run on harvested energy. We are, however, not aware of any security-related efforts in backscatter, other than for RFID [20].

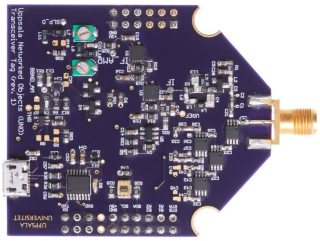

Fig. 1. Backscatter Transceiver Prototype

While security based on Public Key Infrastructure (PKI) systems are becoming possible for resource-constrained, battery-powered devices [5], the resources on backscatter tags are usually too limited for dealing with, for example, certificates required for device authentication. In this paper, we propose an alternative physical-layer authentication mechanism for backscatter devices that employs watermarking to authenticate packets. Watermarking here refers to embedding secret information in the packets [7]. Several IoT physical layers such as IEEE 802.15.4 and IEEE 802.11b (WiFi) use Direct-Sequence Spread Spectrum (DSSS), a spread-spectrum modulation technique used to better handle interference. In the case of 802.15.4, the transmitter maps one 4-bit symbol to one out of 16 32-chip pseudonoise codes (PN-code). The receiver maps the received code to the best matching 4-bit symbol. The 16 codes are chosen so that this matching process is robust against chip flips. Our key idea is that backscatter tags, upon transmission, could intentionally flip one or more chips to enable the receiver to identify the transmitter, without altering the original data. This is similar to recent efforts in using the 802.15.4 chip sequence as steganographic channel [9, 12, 24] or watermarking [11] but with a focus on authentication for ultra-low-power backscatter devices. Similar mechanisms could be implemented in other parts of the packets as we discuss in Section 6.

Contributions. Our main contribution is the design of a physical-layer authentication scheme for backscatter-based IoT devices. We discuss several design trade-offs such as the one between security and reliability. To the best of our knowledge, this is the first paper that addresses security for backscatter with IoT physical layer protocols.

2 Background

In this section we present a brief background on backscatter transmissions and DSSS, both with a focus on IEEE 802.15.4.

Backscatter. Backscatter transmitters selectively reflect an external Radio Frequency (RF) signal to convey information such as their sensor readings [8, 23]. By offloading the carrier generation to an external device, backscatter tags avoid the energy-consuming task of generating their own radio wave, which reduces

their power consumption by up to three orders of magnitude compared to traditional low-power radios. The backscatter transmitter controls how the carrier wave is reflected by changing the load attached to its antenna to create a specific impedance mismatch. This enables the tags to control the amplitude, phase and frequency of the reflected signal. Therefore, the tags can backscatter almost any standard physical layer protocol including packets conforming to the IEEE 802.15.4 standard [15, 16]. A backscatter prototype is shown in Figure 1.

DSSS. Physical layer protocols such as those employed by IEEE 802.15.4 and IEEE 802.11b use DSSS, a spread-spectrum modulation technique used to better handle interference. Using DSSS, this is achieved by widening (spreading) the bandwidth of the transmitted signal. When despreading at the receiver, unintentional and intentional interference is reduced. In IEEE 802.15.4 each symbol (consisting of 4 bits) is mapped to a 32-chip long pseudonoise code (PN-code). There are 16 of these PN-codes as defined in the standard. These 16 PN-codes have been selected to maximize the number of chip positions in which the two PN-codes are different (Hamming distance). In 802.15.4, the minimum distance between two PN-codes is 12 and the maximum is 20. A receiver takes the received PN-code and matches it to the symbol whose PN-code has the minimum Hamming distance to the received PN-code.

3 Related Work

There are a number of studies that discuss steganography and watermarking for IEEE 802.15.4. Ko proposes a first system using a steganographic channel that is based on the 802.15.4 chip sequence [9]. The main contribution of the paper is to show that this channel enables the transmission of additional data to save energy. Along the same lines, Metha et al. also propose to use this channel to communicate with a specialized receiver while sending data to a conventional one [12]. Towards this end, they expand the original 802.15.4 chip sequences with additional chip sequences that still resolve to the original ones for a conventional 802.15.4 receiver. Zielinska and Szczypiorski add additional scrambling to the modified chip sequences to complicate detection of the steganographic channel [24]. They demonstrate the possibility of creating a covert channel with the same data rate as 802.15.4, with a low impact on the bit error rate and only a slight decrease in receiver sensitivity. Li et al. study the same issues for watermarking in 802.15.4 and also implement a prototype system to gain experimental results [11]. Nain et al. extend the channel with acknowledgements to make it reliable [13]. In contrast to these approaches, we aim at exploiting the 802.15.4 PN-codes for authentication of extremely resource-constrained backscatter devices.

Ureten and Serinken are among the first to propose to use RF fingerprints for identifying individual nodes in wireless networks by means of their RF fingerprints [19]. Xu et al. differentiate between the conventional passive approaches for fingerprinting and active approaches [21]. The latter approaches do not only observe ongoing communication but also try to trigger responses from devices to make them transmit useful features. Oracle [17] goes beyond the previous

approaches by using transmitter-side modifications to increase the chances of correct identification at the receiver. In contrast to our approach, Oracle requires machine learning methods to differentiate between different nodes whereas our approach relies on much simpler methods at the receiver as we discuss in the next section. Some studies have shown that RF fingerprinting is feasible also for RFID tags [1, 22].

4 Design Issues

4.1 Overview

On a high level, we devise a backscatter communications authentication system that works as follows: A carrier generator that could be a software-defined radio, a WiFi or an IEEE 802.15.4 device, generates an unmodulated carrier. One or more backscatter tags transmit their collected sensor data by modulating their sensed data in 802.15.4 frames on top of the unmodulated carrier. In order to enable authentication, the tags embed authentication information by flipping selected chips in the PN-code. We call these chips authentication chips. The chips are flipped according to the tag's individual random sequence that is known by both the receiver and the backscatter tags. In our case, the receiver needs to be able to detect the flipped chips. Current 802.15.4 radios do not offer access to such low-level information and hence we would need a specialized receiver that we implement with a software-defined radio, similar to related work. Note that low-power software-defined radios exist [4].

Using the 802.15.4 PN-code for authentication by watermarking is interesting for backscatter devices as it allows us to add information without increasing the packet size. On the other hand, it needs to be done with care since it reduces the robustness of the backscattered signals which are particularly weak and vulnerable. We discuss these and other issues in the sequel of this section.

4.2 Message Authentication Code

We use a Message Authentication Code (MAC) to provide authenticity. To that end, we split the MAC into 5-bit chunks to identify one of the 32 PN-code positions to be flipped. Given the maximum IEEE 802.15.4 packet size of 127 bytes and the fact that there are 4 bits per DSSS symbol we can encode up to 254 bits in one packet by flipping one chip per DSSS symbol. Knowing the message and the tag's random sequence (key) the receiver can decode and verify the MAC.

4.3 Preliminary Reliability Analysis

The goal of this section is to provide a preliminary analysis about the impact of flipping chips of the PN-code on the reliability of packet transmissions and outline ways of improving reliability. We implement a Monte Carlo simulation

Number Auth. chips	Number RX chip errors	Symbol correct	Auth. chips correct	Symbol and Auth. correct
1	0	1.0	1.0	1.0
1	1	1.0	0.97	0.97
1	2	1.0	0.94	0.94
1	4	1.0	0.88	0.88
1	6	0.99	0.81	0.8
1	8	0.88	0.75	0.64
2	0	1.0	1.0	1.0
2	1	1.0	1.0	1.0
2	2	1.0	0.997	0.998
2	4	0.998	0.99	0.99
2	6	0.96	0.97	0.93
2	8	0.78	0.94	0.73

Table 1. The results show that under worse channel conditions there is an increase in symbols and authentication chips that cannot be detected. Using two instead of one authentication chips per chip PN-code improves the situation.

in Python. In the simulation, we take one of the 16 PN-codes and first flip one or more chips of the code before transmission. The first chip is flipped as part of the authentication, the other chips are flipped to increase security by adding additional chip flips. We flip an additional number of chips to simulate errors at the receiver where interference actually occurs. Note that such a chip flip during reception could undo the transmitter's chip flip. After the reception of a packet, the receiver decodes the PN-code and retrieves the authentication chips.

Basic Reliability. The goal of our first simulation is to evaluate if the correct PN-code and hence symbol is detected. Further, we evaluate if the authentication chip is correctly detected, i.e., if it has not been flipped again which could happen because of interference at the receiver.

Table 1 depicts the simulation results. The table shows how the number of authentication chips and the number of chip errors at the receiver impact the correctness of the received symbol (i.e., the receiver selects the correct symbol out of the 16 possible ones), if the authentication chip is still valid (i.e., it has not been flipped again during reception) and finally in the right-most column if both the PN-code and the authentication chip are still correct. The table shows that with only one authentication chip, there is a high chance that interference (additional chip flips) prevents that both the correct symbol and the correct authentication chip can be detected. Therefore, we opt for using two authentication chips in each symbol. With two authentication chips, we consider the authentication chips also correctly received if only one of them is correct.

The results in the table show that under worse channel conditions, i.e., when we flip more chips at the receiver, there is an increase in symbols and authentication chips that cannot be detected. The table also shows that only less than in total six chip flips per symbol can be corrected. As Zielinska and Szczypiorski

Number Auth. chips	Number RX chip errors	Symbol correct	Auth. chips correct	Symbol and Auth. correct
2+1	0	1.0	1.0	1.0
2+1	1	1.0	1.0	1.0
2+1	2	1.0	0.998	0.998
2+1	4	0.99	0.99	0.98
2+1	6	0.91	0.97	0.87
2+1	8	0.66	0.94	0.61
2+2	0	1.0	1.0	1.0
2+2	1	1.0	1.0	1.0
2+2	2	0.998	0.998	0.996
2+2	4	0.97	0.99	0.96
2+2	6	0.83	0.97	0.8
2+2	8	0.56	0.94	0.51

Table 2. While flipping additional chips before transmission increases security, it leads to a further decrease in the correct detection of symbols and authentication chips

denote [24], the maximum error correcting capability in 802.15.4 is $\lfloor \frac{d_{min}-1}{2} \rfloor$ where d_{min} is the minimum Hamming distance between two PN-codes (12 in 802.15.4).

Since we require that only one out of the two authentication chips needs to be correct, we increase the risk that an attacker that sends faked packets and tries to guess the authentication chips succeeds. With only one authentication chip per symbol the probability that the attacker flips the correct chip is $\frac{1}{32}$ since there are 32 chips per symbol. Using two authentication chips and requiring that only one of them is correct increases this probability to $\frac{1}{32} + 2 \cdot \frac{1}{32} \cdot \frac{30}{32}$, i.e., from 0.03125 to 0.0596. If we assume that the attacker needs to guess 10 symbols correctly, the probability of a correct guess decreases beyond 5.63e-13. We make two assumptions here: First, we assume that we have two distinct authentication chips in the (manipulated) PN-code of a symbol. Second, since chips may be flipped at the receiver due to interference, the receiver does not require that all symbols have the expected authentication chips but a subset is sufficient for successful authentication.

Additional chip flips. One possibility to make it more difficult for the attacker to identify the random sequence based on observed chip flips is to create additional chip flips. While this increases security the chances increase that the receiver cannot decode the transmitted symbol correctly. We quantify this risk with additional simulations that we depict in Table 2.

The table shows that as expected the risk of symbol mismatch increases. Assuming that during reception eight chips are flipped because of interference, the ability to decode the right symbol decreases from 78% (Table 1) to 68% with one additional chip flip before transmission to 56% with two additional chip flips.

These results demonstrate that adding additional chip flips in order to increase the security has as expected a negative impact on the symbol detection

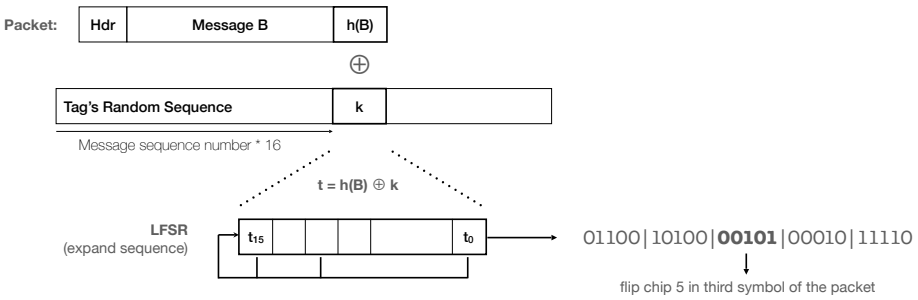

Fig. 2. MAC Implementation

ability of the receiver when there is radio interference. Therefore, the decision if and how many additional chip flips should be performed depends on the state of the radio channel. If the channel conditions are good and there is a low risk for interference and hence chip flips during reception, additional chip flips seem an attractive idea. If, however, the radio channel conditions are bad adding additional chip flips may cause symbol errors at the receiver. As in such scenarios we may expect additional chip flips caused by interference, it is not necessary to add artificial chip flips for security reasons.

Conventional low-power radios have mechanisms such as CCA (Clear Channel Assessment) checks to get an understanding of the state of the radio channel. There are, however, currently no similar mechanisms for backscatter tags. Therefore, we expect that we need to measure the channel conditions at the receiver (where interference actually takes place) and inform the backscatter tags about the channel conditions.

4.4 MAC Implementation

As illustrated in Figure 2, we base our MAC on a CRC-based MAC $t = h(B) \oplus k$ [10], where B is the b-bit message to be sent, $h(B) = \text{coef}(B(x) \cdot x^m \bmod p(x))$ a function of the (b, m) hash-family with polynomial $p(x)$ of degree m, and k is a one-time key. We use the 16-bit CRC (polynomial) of IEEE 802.15.4 because it has to be calculated and concatenated to the packets for error detection purposes anyway. The one-time key is derived from the tag's random sequence. For each packet we use another 16-bit sub-sequence with an offset depending on the message sequence number times 16 (modulo the sequence length).

We finally use a linear-feedback shift register to extend the $m = 16$ bit MAC t (used as initial value) into a longer pseudo-random sequence. For this purpose we use the feedback polynomial $x^{16} + x^{15} + x^{13} + x^4 + 1$ to achieve a maximum-length period of $2^{16} - 1$, which is plenty for practical packet lengths. An LFSR is easy to implement in both soft- and hardware.

The success probability of an adversary that tries to modify a single message that results in the same MAC is at most ϵ for ϵ-opt-secure hash families [10]. For

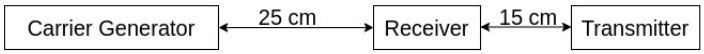

Fig. 3. Experimental Setup

the (b, m) hash-family: $\epsilon = (b+m)/(2^{m-1})$. This shows that the message length m has a significant impact on ϵ, and hence on the security of the system. Short messages reduce the search space for brute force attacks, long messages increase ϵ. The hash function $h()$ (in our case the CRC) can be reused if for each new message a different one-time key k is used. It is therefore important to chose a sufficiently long random sequence for each tag to derive unique keys.

5 First Prototype

In order to demonstrate the basic feasibility of our approach, i.e., that we can flip chips even on our resource-constrained prototype, we implement chip flipping in our backscatter tag [15]. The modifications are implemented in the FPGA that features the baseband logic to generate 802.15.4 frames. As a carrier generator, we use a USRP B200 software-defined radio and as a receiver a Zolertia Firefly IoT development platform. We place the devices at close distances as shown in Figure 3 to avoid undesired chip flips due to weak communication links. We flip a specific number of chips and evaluate whether the Zolertia node is still able to receive the packets. In order to consider a worst case, we flip the chips so that the minimum Hamming distance is minimized. For example, when flipping two chips, the minimum Hamming distance that is 12 without any chip flips, becomes 10, 12 or 14 dependent on which chips are flipped. For the purpose of this experiment, we flip the chips that lead to a minimum Hamming distance of 10. We send 160 packets with the same number of flipped chips, 10 with each of the different PN-codes.

Our results show that the Zolertia node receives almost all packets correctly when we flip up to four chips in each PN-code. In fact, we see one error (out of 640 packets) in these scenarios which we attribute to external interference since the experiments are conducted in an office environment. If we flip five or more chips in one chip sequence, however, we see an increasing rate of symbols that are wrongly detected. When we flip 8 chips in each PN-code, roughly half of the packets can still be correctly received.

6 Discussion

While our first prototype has been implemented in the physical layer, we could also implement the same algorithms in other places in the packet, for example, in the application layer. The main disadvantage of implementations at higher layers is that the packets increase in size due to the additional bytes required for authentication. Note that while the smallest unit we can manipulate in the

physical layer is a chip, in higher layers it is a bit which corresponds to eight chips. Hence, implementing the same approach in a higher layer may lead to a non-negligible increase of the size of the packets and larger packets have a higher risk of being corrupted [6]. On the other hand, our approach of flipping chips in the 802.15.4 frames has a negative impact on the packet reliability. Furthermore, the implementation of our approach requires a gateway that is capable of dealing with physical layer information. Note, however, that there is a current trend of more capable gateways for the Internet of Things [3]. In the long run, we intend to combine our approach with RF fingerprinting which also requires information from the physical layer and hence gateways that are able and need to handle physical layer information.

7 Conclusions

In this extended abstract, we have taken the first steps towards authentication for extremely resource-constrained sensor devices using backscatter communication. In particular, we target backscatter devices that use IEEE 802.15.4 as their physical layer protocol where we embed the authentication information as chip flips. We have discussed several design options that highlight the trade-off between security and reliability and have shown that chip flipping is feasible also in those resource-constrained backscatter devices. Towards the best of our knowledge, this is the first paper that targets security for backscatter with standard IoT protocols.

Acknowledgements

This project is financially supported by the Swedish Foundation for Strategic Research and the Swedish Research Council (grants 2017-045989 and 2018-05480).

References

1. Boris Danev, Thomas S Heydt-Benjamin, and Srdjan Capkun. Physical-layer identification of rfid devices. In *USENIX security symposium*, pages 199–214, 2009.
2. Joshua F Ensworth and Matthew S Reynolds. Every smart phone is a backscatter reader: Modulated backscatter compatibility with bluetooth 4.0 low energy (ble) devices. In *IEEE international conference on RFID*, pages 78–85. IEEE, 2015.
3. Saptarshi Hazra, Simon Duquennoy, Peng Wang, Thiemo Voigt, Chenguang Lu, and Daniel Cederholm. Handling inherent delays in virtual iot gateways. In *International Conference on Distributed Computing in Sensor Systems*, 2019.
4. Mehrdad Hessar, Ali Najafi, Vikram Iyer, and Shyamnath Gollakota. TinySDR, A Software-Defined Radio Platform for Internet of Things. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–3, 2019.
5. Joel Höglund, Samuel Lindemer, Martin Furuhed, and Shahid Raza. Pki4iot: Towards public key infrastructure for the internet of things. *Computers & Security*, 89:101658, 2020.

6. Martin Jacobsson and Christian Rohner. Estimating packet delivery ratio for arbitrary packet sizes over wireless links. *IEEE Communications Letters*, 19(4):609–612, 2015.
7. Taeho Kang, Xiang Li, Chansu Yu, and Jong Kim. A survey of security mechanisms with direct sequence spread spectrum signals. *Journal of Computing Science and Engineering*, 7(3):187–197, 2013.
8. Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R Smith. Passive wi-fi: Bringing low power to wi-fi transmissions. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016.
9. Thomas Kho. Steganography in the 802.15. 4 physical layer. *UC Berkeley*, 2007.
10. H Krawczyk. Lfsr-based hashing and authentication. In *14th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO)*, 1994.
11. Xiang Li, Chansu Yu, Murad Hizlan, Won-Tae Kim, and Seungmin Park. Physical layer watermarking of direct sequence spread spectrum signals. In *IEEE Military Communications Conference*. IEEE, 2013.
12. Ankur M Mehta, Steven Lanzisera, and Kristofer SJ Pister. Steganography in 802.15. 4 wireless communication. In *2008 2nd International Symposium on Advanced Networks and Telecommunication Systems*, pages 1–3. IEEE, 2008.
13. Ajay Kumar Nain and Pachamuthu Rajalakshmi. A reliable covert channel over ieee 802.15.4 using steganography. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 711–716. IEEE, 2016.
14. Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. Plora: A passive long-range data network from ambient lora transmissions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 147–160, 2018.
15. Carlos Pérez-Penichet, Frederik Hermans, Ambuj Varshney, and Thiemo Voigt. Augmenting IoT networks with backscatter-enabled passive sensor tags. In *Proceedings of the 3rd Workshop on Hot Topics in Wireless*, pages 23–27, 2016.
16. Carlos Pérez-Penichet, Dilushi Piumwardane, Christian Rohner, and Thiemo Voigt. Tagalong: Efficient integration of battery-free sensor tags in standard wireless networks. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 169–180. IEEE, 2020.
17. Kunal Sankhe, Mauro Belgiovine, Fan Zhou, Shamnaz Riyaz, Stratis Ioannidis, and Kaushik Chowdhury. Oracle: Optimized radio classification through convolutional neural networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 370–378. IEEE, 2019.
18. Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R Smith, and Shyamnath Gollakota. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–24, 2017.
19. Oktay Ureten and Nur Serinken. Wireless security through rf fingerprinting. *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, 2007.
20. Nguyen Van Huynh, Dinh Thai Hoang, Xiao Lu, Dusit Niyato, Ping Wang, and Dong In Kim. Ambient backscatter communications: A contemporary survey. *IEEE Communications Surveys & Tutorials*, 20(4):2889–2922, 2018.
21. Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1):94–104, 2015.
22. Davide Zanetti, Boris Danev, and Srdjan Capkun. Physical-layer identification of uhf rfid tags. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 353–364, 2010.

23. Pengyu Zhang, Dinesh Bharadia, Kiran Joshi, and Sachin Katti. Hitchhike: Practical backscatter using commodity wifi. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 259–271, 2016.
24. Elzbieta Zielinska and Krzysztof Szczypiorski. Direct sequence spread spectrum steganographic scheme for ieee 802.15. 4. In *2011 Third International Conference on Multimedia Information Networking and Security*, pages 586–590. IEEE, 2011.