

Article

# CRC-Based Reliable WiFi Backscatter Communication for Supply Chain Management

Yun-Hao Liu <sup>1</sup>, Tao Liu <sup>2</sup>, Yimeng Huang <sup>1</sup>, Han Ding <sup>3</sup>, Wei Xi <sup>3</sup> and Wei Gong <sup>1,\*</sup>

<sup>1</sup> School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China

<sup>2</sup> Business School, China University of Political Science and Law, Beijing 102249, China

<sup>3</sup> School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China

\* Correspondence: weigong@ustc.edu.cn

**Abstract:** Supply chain management aims to achieve both efficiency and low cost. Backscatter technology provides a low-energy consumption approach for critical links in the supply chain, such as warehouse management and cargo identification. Traditional backscatter systems achieve tag data transmission through dedicated hardware or controlled transmission sources. An additional access point (AP) can be used to ensure that the original data are always known in tag data decoding. These requirements increase the deployment costs and are not suitable for large-scale applications. To address these challenges, we introduce CRCScatter, a backscatter system based on a cyclic redundancy check (CRC) reverse algorithm, with an uncontrolled source and a single-AP receiver. The CRCScatter decoder at the receiver uses the constraints within 802.11b WiFi packets to recover the original packet and decode tag data from the backscatter packet. Our Matlab simulation results show that CRCScatter is effective in the low signal-to-noise ratio (SNR) regime, and its average decoding time is independent of the length of tag data. By appending redundant bits in tag data, the decoding accuracy of CRCScatter can be improved. In summary, CRCScatter presents a backscatter communication mode based on ambient WiFi signals with fewer hardware requirements and low deployment costs. Furthermore, the decoding idea of calculating unknown data based on the packet constraints has the potential to expand to different types of excitation packages.

**Keywords:** backscatter; CRC reverse; WiFi; internet of things; supply chain



**Citation:** Liu, Y.-H.; Liu, T.; Huang, Y.; Ding, H.; Xi, W.; Gong, W.

CRC-Based Reliable WiFi Backscatter Communication for Supply Chain Management. *Appl. Sci.* **2023**, *13*, 5471. <https://doi.org/10.3390/app13095471>

Academic Editor: Mirco Peron

Received: 26 February 2023

Revised: 25 April 2023

Accepted: 25 April 2023

Published: 27 April 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

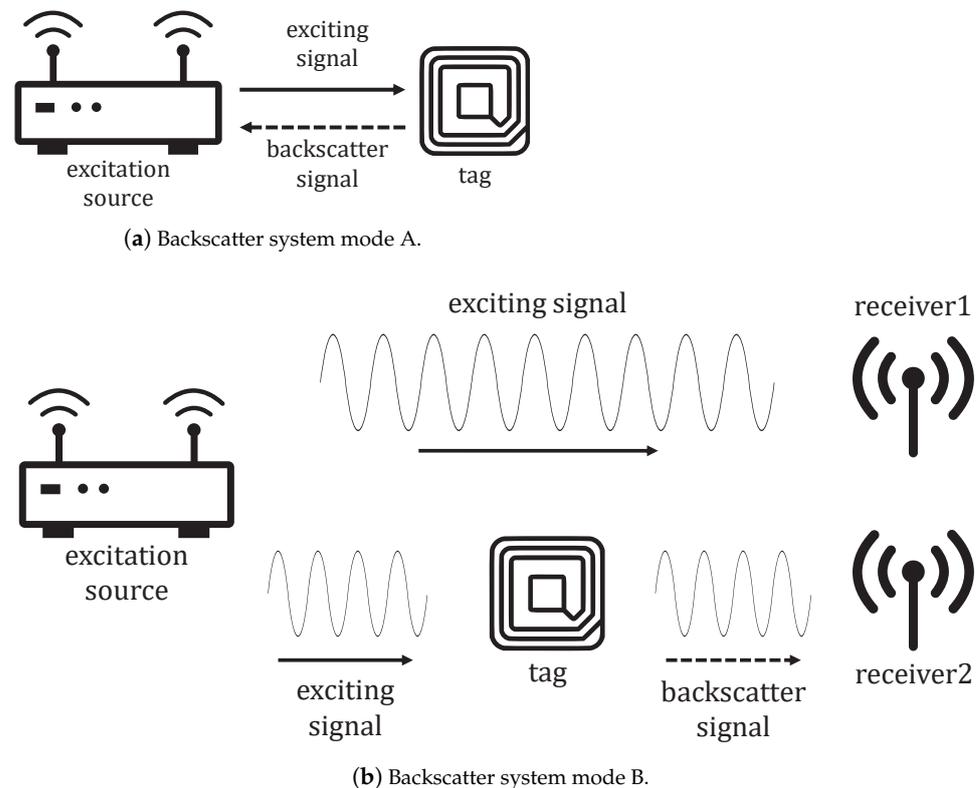
## 1. Introduction

IoT technology has facilitated the use of portable devices by warehouse administrators for connecting goods and various sensors for data communication and remote management. In the service industry, to improve efficiency and control products, there is a high demand for informatization in supply chain management [1]. The supply chain system must be readily available and machinery downtime kept to a minimum [2]. The economic losses caused by replacing aging power sources will be very serious. Moreover, the monitoring and communication devices employed in the supply chain pose a significant power-consumption challenge. According to a survey on wireless communication [3], the bulk of energy consumed by communication systems is attributable to energy-intensive active radio frequency (RF) devices such as power amplifiers. Consequently, enterprises that prioritize the deployment of low-energy-consumption technology in supply chain management will reap higher profits.

To reduce energy consumption and enable long-term communication in passive conditions, backscatter communication systems have been proposed as a novel, low-cost, and energy-efficient solution that can reduce power consumption to a microwatt level. These advantages make backscattering a promising technology for large-scale applications in smart homes [4–8], smart agriculture [9–13], sensors [14–18], and other fields. Backscatter has been implemented in various technologies, including WiFi [19–24], ZigBee [25,26],

LoRa [27], and Bluetooth [28–31]. However, traditional backscatter systems require specialized hardware to achieve backscatter communication. For instance, WiFi backscatter [32] requires a power supply to connect to the network, and BackFi [33] uses customized full-duplex hardware. Passive WiFi [34] needs a dedicated continuous wave signal generator as the excitation signal source. These hardware requirements limit the application scenarios of backscatter systems. HitchHike [35] achieves backscatter communication by commodity devices and introduces the idea of codeword translation, which allows a backscatter tag to embed its information on standard 802.11b packets. Codeword translation has been extended to Bluetooth, ZigBee, and LoRa by FreeRider [36] and LoRa Backscatter [37]. In HitchHike, one more access point is deployed to demodulate excitation signals, enabling the system to decode tag data via a simple and efficient XOR decoder. In addition, ambient backscatter communication and antenna selection methods are also used to improve the technology [38–40].

We can categorize backscatter systems into two modes, as shown in Figure 1. In mode A, the backscatter communication system uses a controlled excitation source for backscattering and decoding. This mode has additional requirements on the excitation source. The pre-defined excitation packet does not carry valid information, and the excitation signal may cause interference with other devices on the same communication channel. In mode B, the system employs an additional AP to receive excitation signals, and it decodes tag data through backscatter data and original data. Compared to active communication infrastructure, both modes of backscatter systems require additional hardware at the transmitter or receiver, which leads to extra costs and may present challenges in large-scale deployment on the current WiFi infrastructure.



**Figure 1.** The structure of backscatter communication systems.

Inspired by these observations, this paper focuses on single-AP decoding with ambient WiFi signals. The main contributions in this work are summarized as follows.

- The current study proposes a CRC reverse-algorithm-based single-AP backscatter system with ambient WiFi signals called CRCScatter. The system uses a CRC re-

verse decoder to solve the problem of decoding original excitation data from a backscatter packet.

- The current research present simulation results to verify the effectiveness of CRCScatter and analyze the performance. The CRCScatter system achieves a decoding tag data bit error rate of  $10^{-2}$  at SNR = -7.5 dB.
- The simulation results verify that the CRC reverse algorithm is better than the brute-force search method in decoding efficiency and the average decoding time of CRCScatter is independent of the tag data length. An improved method for adding redundant bits to the tag data is proposed to improve the decoding accuracy in the presence of noise interference.

## 2. System Model

The system model for the CRCScatter system consists of an excitation source, a tag, and a single-AP receiver, as shown in Figure 2. The ambient excitation source broadcasts exciting signals to the backscatter tag. The tag transmits its data by backscattering and modulating the exciting signal. The CRCScatter receiver uses only one access point to receive the backscatter signal. Considering tag data transmission, tag data are encoded on the 802.11b WiFi signal by codeword translation, and the receiver uses backscatter packets to reverse original packets and decode the tag data. In this section, we present how the tag piggybacks data on a backscatter signal. This paper focuses on the 1 Mbps data rate, and other data rates will be implemented in the future work.

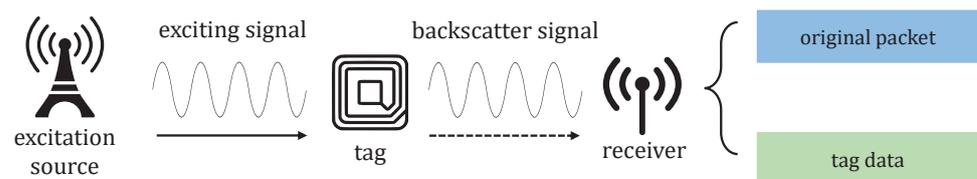


Figure 2. CRCScatter system model.

### 2.1. 802.11b Packet Structure

The 802.11b protocol specifies that the transmitted packet contains PLCP Preamble, PLCP Header, and PSDU. Among them, PSDU has variable length and contains a MAC frame in the CRCScatter system. The MAC frame consists of a MAC header, a frame body, and the frame check sequence (FCS). The frame body field carries the transmitted data, and the CRC32 sequence in the FCS field can detect bit errors in the MAC frame. The part of the packet other than the data segment contains packet control information; if these fields are modified by tag, then the receiver may not be able to demodulate correctly. To receive and demodulate properly, the CRCScatter tag only modulates the frame body field in the packet. The received backscatter packet structure is shown in Figure 3.

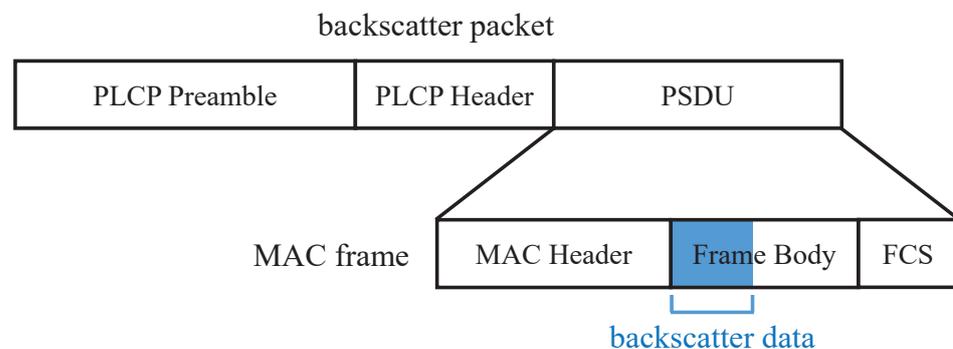


Figure 3. The structure of received backscatter packet.

## 2.2. Codeword Translation

HitchHike [35] proposes a novel technology called codeword translation for tag modulation. This method allows for the piggybacking of tag data information while ensuring that the backscatter signal remains an 802.11b WiFi signal. The backscatter signal after codeword translation can be received and demodulated by the receiver. Specifically, the 802.11b 1Mbps signal utilizes two codewords to encode packets, with a 180° phase difference between them. During tag modulation, the tag uses BPSK modulation to modify the codeword by a phase offset while simultaneously modulating and backscattering the exciting signal. Table 1 shows the relationship between phase offsets and tag bits in tag modulation.

**Table 1.** Encoding at the tag.

Tag Bit	Phase Offset
0	0
1	180°

In tag modulation, one tag data bit corresponds to one data bit. This correspondence indicates that the encoding scheme is efficient and redundancy-free. HitchHike proposes an efficient XOR decoder for extracting tag data from backscatter data and original data. The decoding formula of the XOR decoder can be written as:

$$\text{tag data} = \text{backscatter data} \oplus \text{original data}. \quad (1)$$

In CRCScatter, there is only one access point that receives signals. Thus, while the backscatter data for the XOR decoder can be obtained directly, the original data require calculation by the CRCScatter decoder from the backscatter packet. Therefore, the decoding function for CRCScatter can be expressed as follows:

$$(\text{original packet, tag data}) = \text{CRCScatter}(\text{backscatter packet}). \quad (2)$$

## 3. CRCScatter Decoder Design

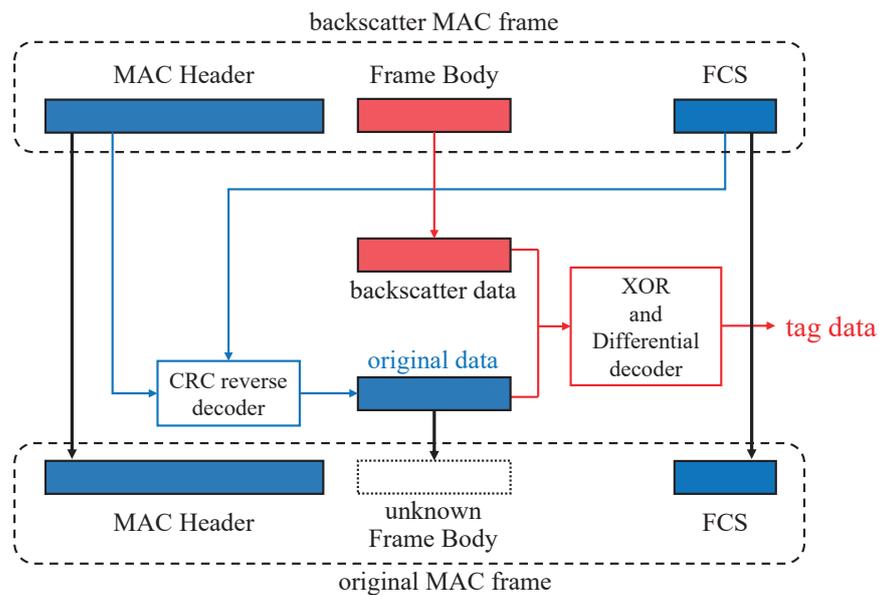
### 3.1. CRCScatter Decoder Overview

The overview of decoding employed by CRCScatter at the receiver is illustrated in Figure 4. The procedure of decoding tag data can be divided into two steps.

The first step involves reversing the backscatter packet into the original packet. The received backscatter MAC frame consists of the MAC header, the frame body, and the FCS, but the tag only modulates the frame body. The MAC header and the FCS in the original packet are the same as those in the backscatter packet. However, the original data cannot be obtained directly because the frame body has been modified. The IEEE 802.11b protocol specifies the correlation between FCS and the transmitted data in WiFi packets. The CRC reverse decoder utilizes this intrinsic correlation as a constraint in calculation and uses two CRC algorithms to reverse original data from the backscatter packets.

The second step is to calculate tag data based on backscatter data and original data obtained through the CRC reverse decoder. In 802.11b data transmission, DBPSK modulation is used, whereas BPSK modulation is adopted by the tag. Due to this difference in modulation, the original XOR decoder in HitchHike cannot obtain the correct tag data. To overcome this, CRCScatter employs a decoder that combines XOR operation and differential decoding to accurately calculate tag data.

In summary, the CRCScatter system utilizes two decoders to compute the original packet and tag data in steps. While the function of the CRC reverse decoder is to retrieve original data from the backscatter packet, the XOR and the differential decoder can obtain correct tag data from original data and backscatter data.



**Figure 4.** CRCScatter decoder overview. The decoding procedure can be divided into two steps. The first step is to reverse the original data marked in blue, and the second step is to calculate the tag data marked in red.

### 3.2. CRC Reverse Decoder

#### 3.2.1. The Algorithms of CRC in the CRC Reverse Decoder

To detect unpredictable bit errors in received packets, the CRC sequence is transmitted with the data. In particular, FCS uses CRC32 to protect the MAC header and frame body. The CRC32 value can be computed by performing bit shifts and XOR operations on a 32-bit CRC register. The bit-oriented calculation algorithm of the CRC value is given by IEEE 802.11b protocol, as shown in Algorithm 1. The CRC algorithm contains three constants, namely, CRCPOLY, INITXOR, and FINALXOR. For CRC32, the value of these constants is given by CRCPOLY = 0x04C11DB7, INITXOR = FINALXOR = 0xFFFFFFFF. In addition, when focusing on the calculation of the CRC register, we can replace INITXOR and FINALXOR with the initial and final states of the CRC register. In each loop of Algorithm 1, bit shift and XOR operations are carried out on the CRC register based on the comparison of the input data bit and the shifted-out bit. The number of bits calculated for a single shift and XOR operation depends on the register length, while the number of cycles corresponds to the length of the input data. When the input data length and CRC type are given, the average run time of Algorithm 1 is stable.

---

#### Algorithm 1 Calculation of the CRC

---

**Input:** data bits  $a$

**Output:** CRC register value  $crcreg$

$crcreg \leftarrow \text{INITXOR}$

$i \leftarrow 0$

**while**  $i < a.length$  **do**

  LEFTSHIFT( $crcreg$ )

**if**  $bit\_just\_shifted\_out \neq a_i$  **then**

$crcreg \leftarrow crcreg \oplus \text{CRCPOLY}$

**end if**

$i \leftarrow i + 1$

**end while**

$crcreg \leftarrow crcreg \oplus \text{FINALXOR}$

---

Assuming that the initial CRC register value  $r'$  and the data bits  $a$  are known, the CRC algorithm can calculate the final value  $r$ . Conversely, if the final CRC register value  $r$  and the calculated data  $a$  are given, we can reverse the procedure of the CRC algorithm to obtain the initial value  $r'$ . The CRC reverse algorithm [41] is presented in Algorithm 2 because Algorithm 2 is the inverse process of Algorithm 1. Similarly, we can also conclude that the average run time is stable when the input bit length is fixed.

---

**Algorithm 2** CRC32 reverse algorithm

---

**Input:** final CRC register value  $r$ , reversed data bits  $a$

**Output:** initial CRC register value  $r'$

```

 $i \leftarrow a.length - 1$ 
 $crcreg \leftarrow r$ 
while  $i \geq 0$  do
  if  $crcreg_{31} = 1$  then
     $crcreg \leftarrow crcreg \oplus CRCPOLY$ 
    RIGHTSHIFT( $crcreg$ )
     $crcreg_0 = a_i \oplus 1$ 
  else
    RIGHTSHIFT( $crcreg$ )
     $crcreg_0 = a_i$ 
  end if
   $i \leftarrow i - 1$ 
end while
 $r' \leftarrow crcreg$ 

```

---

Both the CRC algorithm and the CRC reverse algorithm can be represented as functions, where  $r'$  and  $r$  represent the initial and final value of the CRC register, respectively, while  $a$  represents the calculated data bits. When a set of CRC register values and data bits are given, the two algorithms are opposite computational processes, and the two functions hold simultaneously.

$$r = crc(r', a), \quad r' = crc\_reverse(r, a). \quad (3)$$

### 3.2.2. How to Reverse the Unknown Data from CRC32 Value?

The algorithms of CRC utilize forward or reverse methods to compute the value of the CRC register from the input data bits  $a$ . A highly relevant problem is knowing how to reversing the unknown data bits from the initial and final state values of CRC32. Assuming the unknown data length  $l$  is known, we can employ brute-force search to find the possible original data from all  $2^l$  data sequences. When the data length  $l$  is greater than 32 bits, the number of solutions is  $2^{l-32}$ . If the data length does not exceed 32 bits, the unknown data sequence is unique. The relationship between the number of solutions  $N_l$  and the data length  $l$  can be expressed as follows:

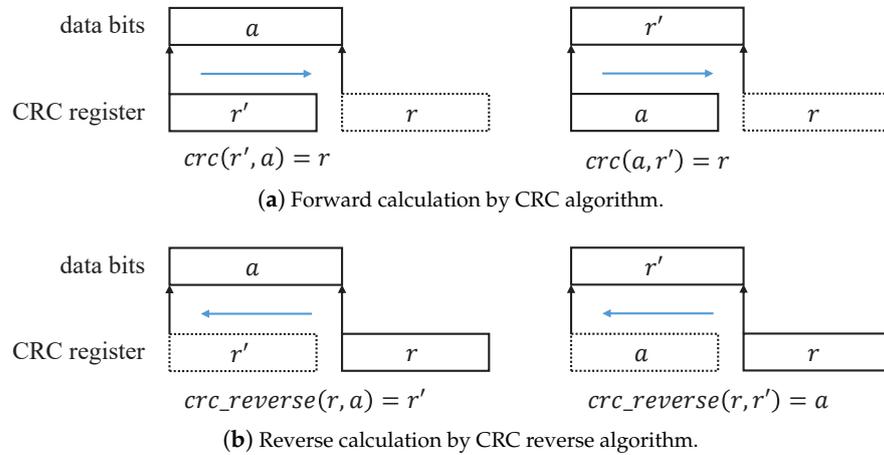
$$N_l = \begin{cases} 2^{l-32}, & l > 32 \\ 1, & l \leq 32 \end{cases} \quad (4)$$

To ensure result uniqueness, we consider the case of unknown data bits with a length of 32 bits. In CRC algorithms, the data sequence  $a$  is an independent variable in functions. To calculate the unknown data bits, we need to establish a new relationship between  $a$  and CRC register values. Assuming the length of data bits is the same as the length of CRC32, the CRC algorithm has certain properties [41], which can be expressed as:

$$crc(r_1, a_1) \oplus crc(r_2, a_2) = crc(r_1 \oplus r_2, a_1 \oplus a_2), \quad (5)$$

$$crc(r', a) = crc(a, r') = r. \quad (6)$$

Through the equivalence relation of (6) and the equation (3), we can derive a new Equation (7) to calculate the data bits  $a$ , as shown in Figure 5.



**Figure 5.** Equation relations in CRC calculation. Because the CRC algorithm and the CRC reverse algorithm are different only in the calculation direction, the functions in (a,b) should hold simultaneously.

$$a = crc\_reverse(r, r'). \tag{7}$$

### 3.2.3. How to Reverse the Original Data?

The MAC frame can be divided into three parts: the MAC header  $K$ , the CRC32 sequence  $R$ , and the original data  $a$ . The transmitter calculates the CRC32 sequence based on the MAC header and the original data. There is a constraint relationship between these parts. The MAC header and CRC32 sequence are known at the receiver because they have not been modified by the tag. The primary challenge is calculating the unknown original data based on known data and constraints. As previously discussed, the tag length should not exceed 32 bits to ensure uniqueness. Therefore, the modified bits by the tag do not exceed 32 bits. We assume that the length of the frame body is 32 bits, indicating the original data length is 32 bits.

Since the unknown original data are limited, we can list all of the possible original data sequences. For each possible sequence, we calculate the CRC value using the MAC header and the possible sequence and compare the value with the CRC32 sequence from the received packet. We continue this process until we find the original data sequence that matches the CRC32 sequence. This brute-force search for the unknown original data is impractical due to the exponential growth of the average number of enumerations with the length of unknown original data. Therefore, a brute-force search cannot satisfy the requirements of immediate communication and the high tag data transmission rate simultaneously.

Fortunately, the original data can be obtained by reversing the unknown data bits based on CRC algorithms. To accomplish this, the initial and final values of the CRC register  $r'$  and  $r$  are computed using CRC algorithms, respectively. The IEEE 802.11b protocol specifies that the initial value  $r'$  for forwarding calculation is INITXOR, while the final value  $r$  can be obtained from the FCS field. Then, the original data can be calculated using the method of reversing unknown data bits. Finally, the original packet can be obtained by replacing backscatter data in the backscatter packet with calculated original data. Algorithm 3 presents the procedure for reversing the original data from the backscatter MAC frame.

The CRC reverse decoder offers an efficient method for decoding transmission tag data up to 32 bits in length. Functionally, the CRC reverse decoder serves as a receiver for a traditional backscatter system that receives the original packet. The CRCScatter system reduces hardware requirements through the use of the CRC reverse decoding method. This design for decoding relies on the presence of a bit sequence within the packet that constrains

the transmitted data. Because the CRC reverse decoder only utilizes the constraints of the FCS field, the maximum length of tag data is limited to 32 bits. If the tag data exceed this limit, the reversed original data at the decoder will not satisfy the uniqueness of the solutions. The length can be extended if more constraints are applied to the packets.

---

**Algorithm 3** Calculation in CRC reverse decoder

---

**Input:** MAC header  $K$ , CRC32 sequence  $R$

**Output:** original data  $a$

$$r \leftarrow R \oplus \text{FINALXOR}$$

$$r' \leftarrow \text{crc}(\text{INITXOR}, K)$$

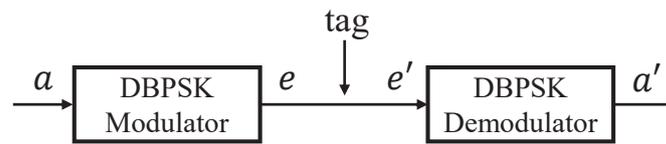
$$a \leftarrow \text{crc\_reverse}(r, r')$$


---

In terms of time complexity, the original data-decoding method based on CRC algorithms requires two CRC algorithms to traverse the entire MAC frame. Even when the tag data is less than 32 bits, the algorithm still needs to calculate the entire original data, while the tag only modifies some of the data bits. Therefore, given the length of frame body, the time for calculating the original data is related to the packet length rather than the tag data length.

### 3.3. XOR and Differential Decoder

The IEEE 802.11 protocol specifies that the DSSS system uses the baseband modulation of DBPSK to provide the 1 Mbit/s data rate and solve the problem of phase ambiguity in BPSK. In DBPSK modulation, the input data  $a$  is calculated by differential encoding to obtain differential data  $e$ ; then, the differential data  $e$  is modulated by the conventional BPSK modulator. In other words, we can assume that the 802.11b signal in our system transmits the differential data  $e$  rather than the original data  $a$ . Since the CRCScatter tag can modify the transmitted data by codeword translation, we set the differential data modified by the tag to  $e'$  and the backscatter data to  $a'$ . According to the decoding Formula (1) and the process shown in Figure 6, the tag data  $t$  can be represented using differential data  $e$  and  $e'$ .



**Figure 6.** The tag modulates the differential data rather than the original data, and tag data  $t$  can be calculated by  $e$  and  $e'$ .

$$t = e \oplus e', \tag{8}$$

In differential encoding, the transmitted data  $a$  and the differential data  $e$  should satisfy the encoding formula written as:

$$e_i = e_{i-1} \oplus a_i. \tag{9}$$

We can use the original data  $a$  and backscatter data  $a'$  to calculate tag data  $t$  by differential encoding formula. The tag data calculation needs to be discussed separately. The first tag data bit  $t_0$  is calculated differently from the other tag data bits.

$$t_0 = e_0 \oplus e'_0 = a_0 \oplus a'_0, \tag{10}$$

$$t_i = e_i \oplus e'_i = (e_{i-1} \oplus a_i) \oplus (e'_{i-1} \oplus a'_i) = (a_i \oplus a'_i) \oplus (e_{i-1} \oplus e'_{i-1}) = (a_i \oplus a'_i) \oplus t_{i-1}. \tag{11}$$

Algorithm 4 presents the procedure for obtaining correct tag data  $t$  from backscatter data  $a'$  and original data  $a$ , as described in this section. From a time complexity standpoint, the number of XOR operations in Algorithm 4 is related to the length of tag data. When the tag data are 32 bits long, we can calculate that the number of bits requiring XOR calculation does not exceed 64. Since a single register operation in CRC algorithms requires a 32-bit operation and the number of cycles is related to the packet length, the time required for differential encoding is significantly shorter than that required for solving the original data. Furthermore, tag data with a length of less than 32 bits can be expanded to 32-bit tag data by adding zeros to the end. When processing backscatter packets from various tags, maintaining a unified tag data length can facilitate the decoder's work and prevent the loss of vital information. Thus, we believe that for the CRCScatter system, the decoding time complexity is related to the packet length, and the tag data length will not affect the average decoding time of the system.

---

**Algorithm 4** Calculation in XOR and differential decoder

---

**Input:** backscatter data  $a'$ , original data  $a$

**Output:** tag data  $t$

```

temp = a' ⊕ a
i ← 0
while i < t.length do
  if i = 0 then
    ti = tempi
  else
    ti = ti-1 ⊕ tempi
  end if
  i ← i + 1
end while

```

---

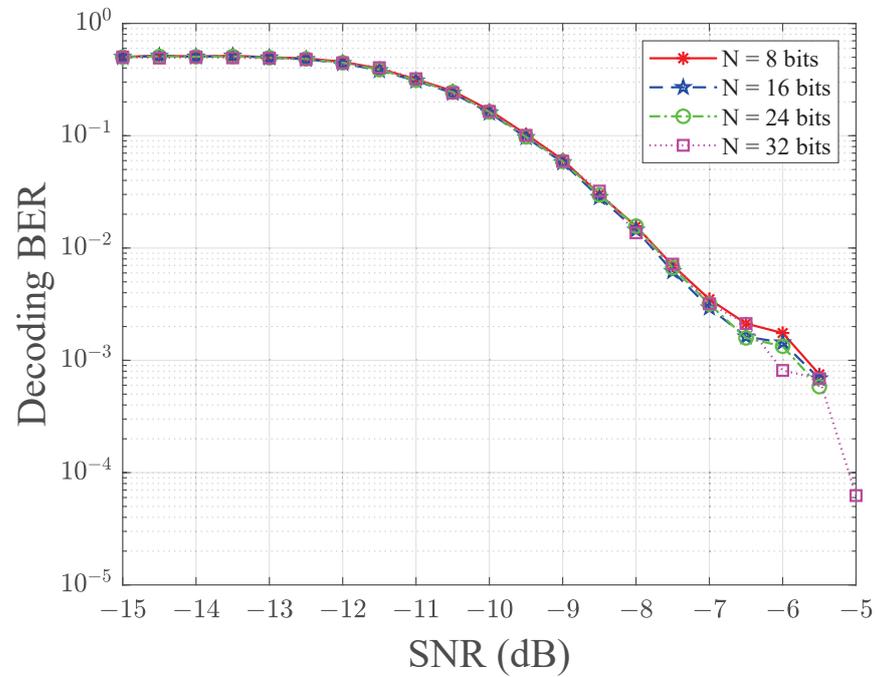
In data communication, noise can cause bit errors in received backscatter packets, leading to errors in decoding tag data. To address this issue, similar to the CRC32 sequence in the MAC frame, the tag can add redundancy check bits and piggyback them along with the real tag data. The tag data containing real tag data and redundant bits can be decoded together by the CRCScatter decoder. Subsequently, the CRCScatter decoder is required to check the decoded tag data. The system can detect incorrect results, and any erroneous tag data that fails to pass verification will be discarded. After error detection steps, the correct decoding data can be preserved, thereby improving decoding accuracy. The verification method and complexity are determined by the type of redundant bits. This work employs a simple parity check code as a redundant error-detecting code, and the decoder needs to count the number of bit "1" in the tag data to verify the results. Due to the limited length of the tag data and the simplicity of the parity check step, the time of parity check will not affect the decoding efficiency of the system. In practical applications, the system can select the suitable redundant bit length and type based on the requirements to balance the tradeoff between the bit error rate (BER) and the verification complexity.

#### 4. Simulation Results

In this section, the simulation results are presented to evaluate the performance of CRCScatter. We use DBPSK modulation in 802.11b transmission and set the length of frame body to 32 bits. The tag data length  $N$  and the SNR are varied to calibrate the results.

First, we verify the effectiveness of the CRCScatter system. In the simulation, the SNR can be set to different values through its relationship with  $E_b/N_0$ . Figure 7 shows the results of the decoding bit error rate of tag data versus the SNR for different tag data lengths  $N$ . The SNR varies from  $-15$  dB to  $-5$  dB, and the tag data length  $N$  is fixed as 8, 16, 24, and 32 bits. In Figure 7, we observe that the decoding the BER of the tag data can be reduced by increasing the SNR. Nevertheless, decoding the BER is similar in terms of the tag data length  $N$ . From the figure, the system can achieve a BER level of  $10^{-2}$  at  $-7.5$  dB,

which proves the effectiveness of the CRC reverse-algorithm-based decoding method in the low SNR regime. Moreover, the SNR does not affect the BER performance when the SNR is less than  $-13$  dB. In this case, the system cannot decode correctly due to excessive noise interference.



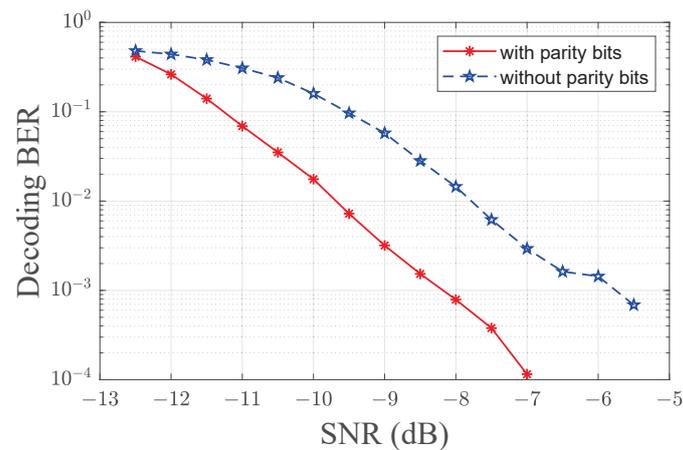
**Figure 7.** The decoding BER versus SNR for different values of  $N$ . The decoding BER is independent of tag data length  $N$ .

Next, we test the decoding time of tag data using the brute-force search and the CRC reverse algorithm. Table 2 exhibits the results of the average decoding time versus different tag data lengths for the algorithms. We observe that the decoding time of brute-force search increases sharply when tag data length  $N$  increases from 4 bits to 18 bits. Due to the excessive decoding time, the brute-force method is not able to meet the requirements of the real-time communication system. Nevertheless, the tag data length  $N$  does not affect the average decoding time of the CRC reverse algorithm, which is close to  $1.0 \times 10^{-2}$  s. Overall, the CRC reverse algorithm is superior to the brute-force search when the tag data length is long. With the average decoding time of the CRC reverse algorithm and the maximum tag data length, we can estimate that the maximum tag data rate of the system is 3.2 kbps, which is sufficient for intelligent meter reading systems, intelligent bracelets, and other micro IoT devices.

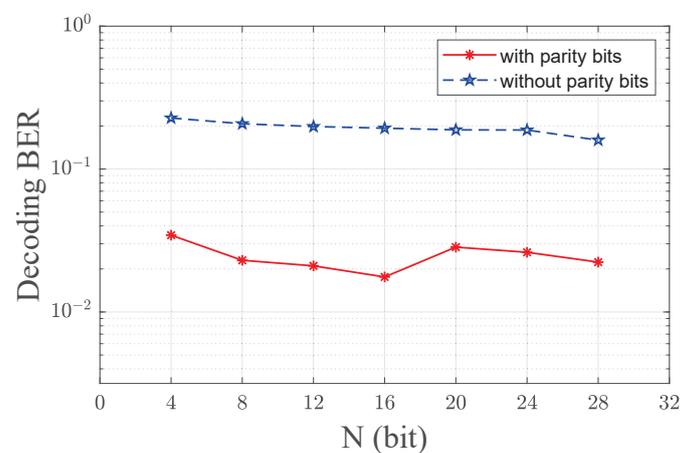
Finally, we test the performance of using redundant bits in tag data to reduce the decoding BER. In the simulations, this study uses four duplicate parity bits as redundant bits for tag data. As shown in Figure 8a, the additional redundant bits can reduce the decoding BER when SNR is greater than  $-12$  dB. In addition, when the SNR is higher than  $-7$  dB, the improved method can achieve accurate decoding with tag data length  $N = 16$  bits. With the same SNR =  $-10$  dB, Figure 8b shows that the decoding BER of tag data with redundant bits is 9% to 15% of the BER without redundant bits. These results suggest that adding redundant bits can significantly reduce the BER in the presence of noise interference. Additionally, because the average decoding time of tag data is stable and the parity check algorithm is simple, adding redundant parity check bits will not affect the efficiency of the decoding.

**Table 2.** Comparison of two decoding methods.

Tag Data Length $N$ (Bit)	Brute-Force Search $T_b$ (s)	CRC Reverse Algorithm $T_c$ (s)
4	0.0026	0.0104
6	0.0049	0.0095
8	0.0194	0.0103
10	0.0586	0.0110
12	0.224	0.0118
14	0.896	0.0099
16	3.529	0.0111
18	38.707	0.0097



(a) The decoding BER versus SNR with  $N = 16$  bits.



(b) The decoding BER versus  $N$  with  $\text{SNR} = -10$  dB.

**Figure 8.** Comparison of the decoding BER between tag data with four duplicate parity bits and tag data without check bits. (a) shows the relationship between BER and SNR, and (b) shows the relationship between BER and tag data length. Adding redundancy check bits can significantly reduce the decoding BER of the CRCScatter system.

In summary, we validate the feasibility of CRCScatter through simulation results. In practical scenarios, such as product identification and quality control in fruit supply chain management, the CRCScatter system has the potential to be deployed on mobile devices. Tags containing key information are affixed to the surface of the fruit. By receiving and decoding the backscatter signal of the tag, managers can simultaneously obtain crucial information such as the location and type of multiple different products. Due to low hardware requirements and low energy consumption, we believe this communication

scheme will facilitate product quality inspection, cargo classification, and other operations in supply chain management. With the improvement of throughput, low-power video communication will also be available.

## 5. Discussions and Conclusions

In this paper, we propose a novel backscatter communication system, named CRCScatter, which enables ambient WiFi backscatter communications with a single-AP receiver. Unlike existing backscatter systems, CRCScatter does not require an additional access point at the receiver, nor does it impose restrictions on the excitation source. The CRCScatter decoder performs a reverse CRC, XOR decoding, and a differential decoding procedure to decode the original packet and tag data from the received backscatter packet. The simulation results demonstrate the effectiveness of the proposed system. To reduce the BER, this work adds redundant parity bits to the tag data for simple error detection. In future work, we aim to implement and test CRCScatter in real-world environments.

We outline some directions for improving upon this work. First, a possible research direction for further exploration is to expand the types and functions of parity bits. More complex error-detecting codes such as CRC can be applied to this work for lower BER. Error-correcting codes could be applied to improve decoding accuracy when the SNR is high and the number of error bits is small. Second, the CRCScatter tag continuously modulates multiple bits based on the length and content of the tag data. Future work could use more modulation methods, such as segmented modulation or sliding window. The decoding algorithm also needs to be redesigned based on the new modulation method. Lastly, while the current design focuses on DBPSK, future work could explore decoding on DQPSK or other excitation signals such as Bluetooth and ZigBee.

**Author Contributions:** Conceptualization, H.D. and W.G.; methodology, W.X. and W.G.; software, Y.-H.L.; validation, Y.-H.L. and Y.H.; formal analysis, T.L.; investigation, Y.-H.L. and T.L.; resources, Y.-H.L.; data curation, T.L. and Y.H.; writing—original draft preparation, Y.-H.L.; writing—review and editing, T.L. and Y.H.; visualization, Y.-H.L.; supervision, H.D., W.X. and W.G.; project administration, W.G.; funding acquisition, W.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by NSFC Grant No. 61971390.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are available from the authors upon request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Menanno, M.; Savino, M.M.; Accorsi, R. Digitalization of Fresh Chestnut Fruit Supply Chain through RFID: Evidence, Benefits and Managerial Implications. *Appl. Sci.* **2023**, *13*, 5086. [[CrossRef](#)]
2. Lolli, F.; Coruzzolo, A.M.; Peron, M.; Sgarbossa, F. Age-based preventive maintenance with multiple printing options. *Int. J. Prod. Econ.* **2022**, *243*, 108339. [[CrossRef](#)]
3. Feng, D.; Jiang, C.; Lim, G.; Cimini, L.J.; Feng, G.; Li, G.Y. A survey of energy-efficient wireless communications. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 167–178. [[CrossRef](#)]
4. Maselli, G.; Piva, M.; Stankovic, J.A. Adaptive Communication for Battery-Free Devices in Smart Homes. *IEEE Internet Things J.* **2019**, *6*, 6977–6988. [[CrossRef](#)]
5. Huang, Y.; Yuan, L.; Gong, W. Research on IEEE 802.11 OFDM Packet Detection Algorithms for Household Wireless Sensor Communication. *Appl. Sci.* **2022**, *12*, 7232. [[CrossRef](#)]
6. Yao, C.; Liu, Y.; Wei, X.; Wang, G.; Gao, F. Backscatter technologies and the future of internet of things: Challenges and opportunities. *Intell. Converg. Netw.* **2020**, *1*, 170–180. [[CrossRef](#)]
7. Maselli, G.; Pietrogiacomi, M.; Piva, M.; Stankovic, J.A. Battery-Free Smart Objects Based on RFID Backscattering. *IEEE Internet Things Mag.* **2019**, *2*, 32–36. [[CrossRef](#)]

8. Wu, W.; Hawbani, A.; Gong, W. Ortho-CodeA: Orthogonal Codes Assisted Backscatter Multiple Access. In Proceedings of the 2023 IEEE International Conference on Pervasive Computing and Communications (PerCom), Atlanta, GA, USA, 13–17 March 2023.
9. Daskalakis, S.N.; Goussetis, G.; Assimonis, S.D.; Tentzeris, M.M.; Georgiadis, A. A uW Backscatter-Morse-Leaf Sensor for Low-Power Agricultural Wireless Sensor Networks. *IEEE Sens. J.* **2018**, *18*, 7889–7898. [[CrossRef](#)]
10. Fieuzal, R.; Baup, F. Estimation of Multi-Frequency, Multi-Incidence and Multi-Polarization Backscattering Coefficients over Bare Agricultural Soil Using Statistical Algorithms. *Appl. Sci.* **2023**, *13*, 4893. [[CrossRef](#)]
11. Daskalakis, S.N.; Kimionis, J.; Collado, A.; Tentzeris, M.M.; Georgiadis, A. Ambient FM backscattering for smart agricultural monitoring. In Proceedings of the 2017 IEEE MTT-S International Microwave Symposium (IMS), Honolulu, HI, USA, 4–9 June 2017.
12. Daskalakis, S.N.; Assimonis, S.D.; Goussetis, G.; Tentzeris, M.M.; Georgiadis, A. The Future of Backscatter in Precision Agriculture. In Proceedings of the 2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, Atlanta, GA, USA, 7–12 July 2019.
13. Lin, K.; López, O.L.A.; Alves, H.; Chapman, D.; Metje, N.; Zhao, G.; Hao, T. Throughput optimization in backscatter-assisted wireless-powered underground sensor networks for smart agriculture. *Internet Things* **2022**, *20*, 100637. [[CrossRef](#)]
14. Gong, W.; Chen, S.; Liu, J.; Wang, Z. MobiRate: Mobility-Aware Rate Adaptation Using PHY Information for Backscatter Networks. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018.
15. Gong, W.; Yuan, L.; Wang, Q.; Zhao, J. Multiprotocol backscatter for personal IoT sensors. In Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies, Barcelona, Spain, 1–4 December 2020.
16. Zhao, J.; Gong, W.; Liu, J. Towards scalable backscatter sensor mesh with decodable relay and distributed excitation. In Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services, Toronto, ON, Canada, 15–19 June 2020.
17. Wang, A.; Iyer, V.; Talla, V.; Smith, J.R.; Gollakota, S. FM Backscatter: Enabling Connected Cities and Smart Fabrics. In Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI '17), Boston, MA, USA, 27–29 March 2017.
18. Zhao, J.; Gong, W.; Liu, J. X-Tandem: Towards Multi-Hop Backscatter Using Commodity WiFi. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018.
19. Zhao, R.; Zhu, F.; Feng, Y.; Peng, S.; Tian, X.; Yu, H.; Wang, X. OFDMA-enabled Wi-Fi backscatter. In Proceedings of the 25th Annual International Conference on Mobile Computing and Networking, Los Cabos, Mexico, 21–25 October 2019.
20. Yang, Y.; Gong, W. Universal Space-Time Stream Backscatter with Ambient WiFi. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), Pisa, Italy, 21–25 March 2022.
21. Wang, Q.; Chen, S.; Zhao, J.; Gong, W. RapidRider: Efficient WiFi Backscatter with Uncontrolled Ambient Signals. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Online, 10–13 May 2021.
22. Yuan, L.; Gong, W. SubScatter: Sub-symbol WiFi Backscatter for High Throughput. In Proceedings of the 2022 IEEE 30th International Conference on Network Protocols (ICNP), Lexington, KY, USA, 30 October–2 November 2022.
23. Dunna, M.; Meng, M.; Wang, P.; Zhang, C.; Bharadia, D. SyncScatter: Enabling WiFi like synchronization and range for WiFi backscatter Communication. In Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation, Boston, MA, USA, 12–14 April 2021.
24. Jia, Z.; Gong, W.; Liu, J. Spatial stream backscatter using commodity wifi. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018.
25. Zhang, P.; Rostami, M.; Hu, P.; Ganesan, D. Enabling Practical Backscatter Communication for On-body Sensors. In Proceedings of the 2016 ACM SIGCOMM Conference, Florianópolis, Brazil, 22–26 August 2016.
26. Li, Y.; Chi, Z.; Liu, X.; Zhu, T. Passive-ZigBee: Enabling ZigBee Communication in IoT Networks with 1000X+ Less Power Consumption. In Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, Shenzhen, China, 4–7 April 2018.
27. Peng, Y.; Shangguan, L.; Hu, Y.; Qian, Y.; Lin, X.; Chen, X.; Fang, D.; Jamieson, K. 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, Budapest, Hungary, 20–25 August 2018.
28. Iyer, V.; Talla, V.; Kellogg, B.; Gollakota, S.; Smith, J. Inter-Technology Backscatter: Towards Internet Connectivity for Implanted Devices. In Proceedings of the 2016 ACM SIGCOMM Conference, Florianópolis, Brazil, 22–26 August 2016.
29. Zhang, M.; Zhao, J.; Chen, S.; Gong, W. Reliable Backscatter with Commodity BLE. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Online, 6–9 July 2020.
30. Kim, T.; Lee, W. AnyScatter: Eliminating Technology Dependency in Ambient Backscatter Systems. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Online, 6–9 July 2020.
31. Huang, Z.; Gong, W. EAScatter: Excitor-Aware Bluetooth Backscatter. In Proceedings of the 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS), Oslo, Norway, 10–12 June 2022.
32. Kellogg, B.; Parks, A.; Gollakota, S.; Smith, J.R.; Wetherall, D. Wi-fi backscatter: Internet connectivity for RF-powered devices. In Proceedings of the 2014 ACM Conference on SIGCOMM, Chicago, IL, USA, 17–22 August 2014.

33. Bharadia, D.; Joshi, K.R.; Kotaru, M.; Katti, S. BackFi: High Throughput WiFi Backscatter. In Proceedings of the ACM SIGCOMM 2015 Conference, London, UK, 17–21 August 2015
34. Kellogg, B.; Talla, V.; Smith, J.R.; Gollakot, S. PASSIVE WI-FI: Bringing Low Power to Wi-Fi Transmissions. In Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16), Santa Clara, CA, USA, 16–18 March 2016.
35. Zhang, P.; Bharadia, D.; Joshi, K.; Katti, S. Hitchhike: Practical backscatter using commodity WiFi. In Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, Stanford, CA, USA, 14–16 November 2016.
36. Zhang, P.; Josephson, C.; Bharadia, D.; Katti, S. Freerider: Backscatter communication using commodity radios. In Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies, Seoul/Incheon, Republic of Korea, 12–15 December 2017.
37. Talla, V.; Hesar, M.; Kellogg, B.; Najafi, A.; Smith, J.R.; Gollakota, S. LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Maui, HI, USA, 11–15 September 2017.
38. Li, D. Two Birds With One Stone: Exploiting Decode-and-Forward Relaying for Opportunistic Ambient Backscattering. *IEEE Trans. Commun.* **2020**, *68*, 1405–1416. [[CrossRef](#)]
39. Li, D. Hybrid Active and Passive Antenna Selection for Backscatter-Assisted MISO Systems. *IEEE Trans. Commun.* **2020**, *68*, 7258–7269. [[CrossRef](#)]
40. Liu, V.; Parks, A.; Talla, V.; Gollakota, S.; Wetherall, D.; Smith, J.R. Ambient backscatter: Wireless communication out of thin air. In Proceedings of the ACM SIGCOMM 2013 Conference, Hong Kong, China, 12–16 August 2013.
41. Stigge, M.; Plötz, H.; Müller, W.; Redlich, J. *Reversing CRC—Theory and Practice*; Humboldt University Berlin: Berlin, Germany, 2006.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.