applied sciences

MDPI

Article

Data Exfiltration through Electromagnetic Covert Channel of Wired Industrial Control Systems

Shakthi Sachintha [1], Nhien-An Le-Khac [2], Mark Scanlon [2,*] and Asanka P. Sayakkara [1,*]

[1] University of Colombo School of Computing (UCSC), Colombo 7, Sri Lanka
[2] Forensics and Security Research Group, School of Computer Science, University College Dublin, Belfield, 4 Dublin, Ireland
* Correspondence: mark.scanlon@ucd.ie (M.S.); asa@ucsc.cmb.ac.lk (A.P.S.)

Abstract: Industrial control systems (ICS) often contain sensitive information related to the corresponding equipment being controlled and their configurations. Protecting such information is important to both the manufacturers and users of such ICSs. This work demonstrates an attack vector on industrial control systems where information can be exfiltrated through a electromagnetic (EM) radiation covert channel from the wired Ethernet connections commonly used by these devices. The attack leverages compromised firmware for the controller—capable of encoding sensitive/critical information into the wired network as packet transmission patterns. The EM radiation from the wired network's communication is captured without direct physical interaction using a portable software-defined radio, and subsequently demodulated on the attacker's computer. This covert channel facilitates the exfiltration of data from a distance of up to two metres with a data rate of 10 bps without any significant data loss. The nature of this covert channel demonstrates that having strong firewalls and network security mechanisms does not fully protect ICSs, or other critical networked or local, air-gapped infrastrusture.

Keywords: covert channel; EM radiation; exfiltration; air-gap; Ethernet; software-defined-radio

Citation: Sachintha, S.; Le-Khac, N-A.; Scanlon, M.; Sayakkara, A.P. Data Exfiltration through Electromagnetic Covert Channel of Wired Industrial Control Systems. *Appl. Sci.* **2023**, *13*, 2928. https://doi.org/10.3390/app13052928

Academic Editor: Alessandro Gasparetto

Received: 20 January 2023
Revised: 16 February 2023
Accepted: 22 February 2023
Published: 24 February 2023

1. Introduction

Large-scale industrial environments often consist of electrical, electronic, hydraulic, and various other components, which need to function under tight control. Industrial control systems (ICS) are responsible for the monitoring and controlling of these components and are being increasingly employed to control critical infrastructure the world over. The information stored and processed on ICSs can be considered critical to the smooth operation of the industrial environments where they operate [1]. As a result, ensuring the security and preventing leakage of data from ICSs can be considered vital to the successful operation of several public services and industries.

ICSs are designed from the ground up to be as robust as possible, and ideally, to operate for decades without significant maintenance or downtime [2]. Broadly speaking, computing technologies are ever-evolving and advancing, and the relatively frequent (i.e., every few years) replacement and upgrading of consumer and industrial computer equipment is commonplace. However, several factors prevent ICSs from being upgraded or replaced at a similar timescale, including financial, the inability for critical infrastructure downtime, and often the sheer scale of the endeavour. Once deployed, it usually takes in the order of 15 to 20 years for an ICS to be replaced with newer components and products [1]. This upgrade-cycle lag has serious security consequences, as any known vulnerabilities are often left unaddressed for significant periods of time [3]. Vulnerabilities and attack vectors that are easily and frequently patched in modern computing systems can remain present in critical ICS infrastructure for years.

Side channels allow attackers to retrieve information from a system by monitoring inert characteristics of the system, such as execution time, power consumption or memory

usage patterns. The target device does not need a direct output to the attacker's system for the side-channel analysis approach to be viable. Instead, side channel attacks are possible when an attacker has the ability to monitor non-functional characteristics of internal device activity, e.g., its running time, power consumption, memory accesses, or packets communicated over a network. Sensitive data has been successfully discovered and exfiltrated through side-channel attacks in various systems, including web applications [4], electronic and IoT systems [5], and cryptographic systems [6].

A covert channel is a communication channel that leverages a regular, legitimate device communication channel to transfer illegitimate data. System administrators may not be aware such data transmission is occurring as it is transferred through a legitimate channel. Data can even be stolen from highly-secured, air-gapped, non-networked computers through such covert channels. Several side channels exist in a typical computer that can be leveraged as covert channels, such as the sound a computer generates when operating, i.e., an acoustic side-channel [7], the power consumption of a computer, i.e., a power side-channel [8], operational indicator LEDs, i.e., an optical side-channel [9], and electromagnetic radiation (EMR) from different components of a computer, i.e., an EM side-channel [10,11].

Ethernet cables are typically designed to keep electromagnetic (EM) interference at a minimal level to avoid inadvertently impinging with other communication (wired or wireless) devices in the vicinity. Nonetheless, the most commonly used Ethernet cables, i.e., twisted pair, often still emit a small amount of EM radiation due to manufacturing defects. Radiation is difficult to capture when the cable is appropriately shielded [12]. Several recent studies have demonstrated that this small amount of radiation can be detected and leveraged as a side channel to leak information [4,12]. Therefore, there is potential to leverage this unintended EM radiation from wired network cables as a covert channel for sensitive information leakage.

This work demonstrates that the unintended EM radiation from Ethernet cables, commonly used in ICSs and other networked devices, can be leveraged to build a covert channel to leak critical data from the system. Malicious code injected into a critical component of an ICS can cause specific network traffic patterns to intentionally modulate critical data into the EM radiation of the Ethernet cables. The demonstrated attack is performed by sending two carefully-chosen network traffic patterns through the Ethernet cable by a malicious code running on the victim device, i.e., ICS. A software-defined radio hardware along with a decoding software running on the attacker's computer captures the EM radiation of the Ethernet cable and successfully recovers the data modulated into the network by the malicious code. This work makes the following contributions:

- Demonstrates and experimentally evaluates the design of a covert channel protocol that can exfiltrate data through the EM radiation of Ethernet cables.
- Introduces and experimentally evaluates a methodology to automatically detect information-leaking EM frequencies of Ethernet cables.
- Explores the potential of increasing reliability in EM-based covert channels through error correction codes.
- Discusses the impact of the identified attack vector on the large-scale industrial control systems.

The rest of the paper is organised as follows. Section 2 provides an overview of the state of the art literature on the domain of covert channels in general, and EM-based covert channels in particular. Section 3 details the specific covert channel attack vector that is utilised by the presented work. Section 4 provides a comprehensive coverage on the algorithms to detect information-leaking EM radiation frequencies on Ethernet cables. Sections 5 provides the details of the design and the implementation of the introduced covert channel, followed by its experimental evaluation in Section 6. Finally, Section 7 discusses the findings and their implications before Section 8 concludes the paper.

2. Related Work

An air-gapped computer is one that is separated from the outside world to prevent threats coming from outside the local network. A bare-minimum method may be made available for such computers to transfer legitimate data under tight control, i.e., updated configuration files, software updates, log files, etc. Nonetheless, numerous side-channels can be converted into covert channels for data exfiltration from such air-gapped computers.

Table 1 depicts a comparison of different physical covert channels.

Table 1. Comparison of existing physical covert channels.

Covert Channel	Type	Wall-Penetrating	Bandwidth
BitWhisper [13]	Thermal	No	0.002 bps
Fansmitter [7]	Acoustic	No	0.25 bps
Magneto [14]	Magnetic	No	5 bps
Odini [15]	Magnetic	No	40 bps
USBee [16]	Electromagnetic	Yes	640 bps
PowerHammer [8]	Power	No	1000 bps
xLED [9]	Optical	No	1000 bps
Hard Drive LED [17]	Optical	No	4000 bps
Bitjabber [10]	Electromagnetic	Yes	300,000 bps

2.1. Acoustic Side Channels

Acoustic side channels involve a malicious user or a program using computer speakers to transmit sonic and ultrasonic signals to a nearby receiver. For example, *Fansmitter* [7] is malware that can leak data out from audio-gapped computers by utilizing the CPU or casing fan noise to encode information. The malware regulates the device's fan speed to enable a vector for information leakage through the acoustic side-channel. These acoustic patterns can be captured using a remote microphone, a nearby smartphone, or a similar electronic device.

2.2. Visual Based Side Channels

Most electronic devices have built-in LEDs to display the status of the devices. *xLED* [9] is malware that controls the LEDs' status in routers and LAN switches. The flickering of the LEDs can be used to encode and modulate sensitive data. The produced signal can be recorded by distant cameras, surveillance cameras, smartphone cameras, or other optical sensors. The malware can use both amplitude and frequency-based encoding techniques, achieving from 10 bps to 1 kbps of data rate over the LED covert channel.

2.3. Electromagnetic Side Channels

EM side channels are the most common and widely known side channels in the literature. In 1985, a group of researchers demonstrated that EM emissions from VGA cables can be captured and used to reconstruct the visual content [18]. Guri et al. [14] demonstrated the use of covert magnetic signals to leak data from remote, air-gapped computers to neighbouring smartphones. The suggested covert channel still functions even if a smartphone is stored in a Faraday shielding case. Malware regulates workloads on CPU cores to control the magnetic fields emitted by the running machine. This magnetic field can encode and transfer encryption keys, passwords, and keylogging data.

Schulz et al. [12] explored the EM radiation of wired networks as a side-channel. The authors performed an analysis on the 10BaseT Ethernet standard—capturing emissions radiated from Ethernet cables, and subsequently, decoded frames from captured radiation. The authors used a USRP x300 software-defined radio (SDR) for capturing the radiation. A hardware setup similar to their work is used for the research presented as part of this paper. Following a similar approach, Zhan et al. [10] used an emission from a DRAM clock to transmit information from a computer. This approach used a SDR with a log periodic

antenna to capture the covert signals. They achieved a high transmission rate up to 300,000 bps with a shallow error rate of less than 1%.

3. Covert Channel Attack Model

The IEEE 802.3 standard defines the physical and data-link layer behaviour of wired local area networks, which is commonly referred to as *Ethernet*. There exists several variants of the IEEE 802.3 standard with varying hardware configurations and data rates. A standard Ethernet cable has eight wires twisted into four pairs. In the case of Ethernet standards 10BASE-T (IEEE 802.3i) and 100BASE-TX (IEEE 802.3u), only two twisted pairs are used for full-duplex communication—one pair for transmission and another pair for receiving and collision detection. Fast Ethernet standard (100BASE-TX) uses a 4-bit-to-5-bit (4B5B) encoding scheme along with multi-level 3 (MLT-3) line encoding to modulate and transmit data in a physical medium [19]. The 4B5B uses a lookup table to convert every four bits of data into five bits, while the MLT-3 uses three levels of signal to modulate every single bit [20].

The encoding used in the 100BASE-TX standard is designed to minimize the undesirable EM emissions (EMR) from the cable during operation. In addition, the cables are physically designed to minimize the amount of unintentional EM radiation, such as using various types of cable shielding. When an electrical signal is travelling through an individual wire inside a data cable, it generates EM radiation. This EM radiation can induce noise on the other wires, disrupting the data being transferred through them. Instead of using a single wire, the application of a pair of wires twisted with each other and transferring a differential signal helps to cancel out such harmful radiation. Therefore, under ideal conditions, an IEEE 802.3 twisted pair cable will not cause any detectable EM radiation. However, in real-world settings, various manufacturing and handling defects occur; twisted pair cables are not uniformly twisted, and the signal attenuation of the two wires in a twisted pair are not similar. Due to these weaknesses, EM radiation can still emerge from twisted pair cables with a detectable amplitude. The resulting EM radiation is shown to be correlating with the information being transmitted through the Ethernet cable [21].

The proposed covert channel exploits this EM radiation generated by the Ethernet cables used in ICS networks to exfiltrate their internal data. A basic model of the attack configuration is illustrated in Figure 1. There, in order to exfiltrate data from an ICS through an EM-based covert channel, two main components should exist. On the victim ICS device, there should be a specific malicious code running that has access both to the internal data of the device and to the network interface. The role of the malicious code is to read any sensitive information from the host device and modulate it into the EM radiation of the cable. This modulation is to be achieved by the malicious program sending specific network packet patterns over the cable in such a way to cause correlating EM radiation patterns. The second component in the attack model is the attacker's hardware and software setup, which consists of a software-defined radio to capture EM radiation and a computer to process EM data. The attacker's setup will be continuously capturing and demodulating EM radiation of the cable to retrieve data sent by the malicious code running on the victim ICS. In the demonstrated attack in this work, a HackRF One software-defined radio [22] was used with a laptop computer as the attacker's setup, while another laptop computer and a router connected through an Ethernet cable were used to represent the victim network.

In this work, the delivery of the malicious code into a victim ICS device is outside the considered scope. The proposed attack is only viable if such a malicious code is available on the victim device to modulate data. There are various well-explored approaches to inject malicious code into a target computer, such as installing malicious code through the network using weak or default passwords used on victim devices [23].

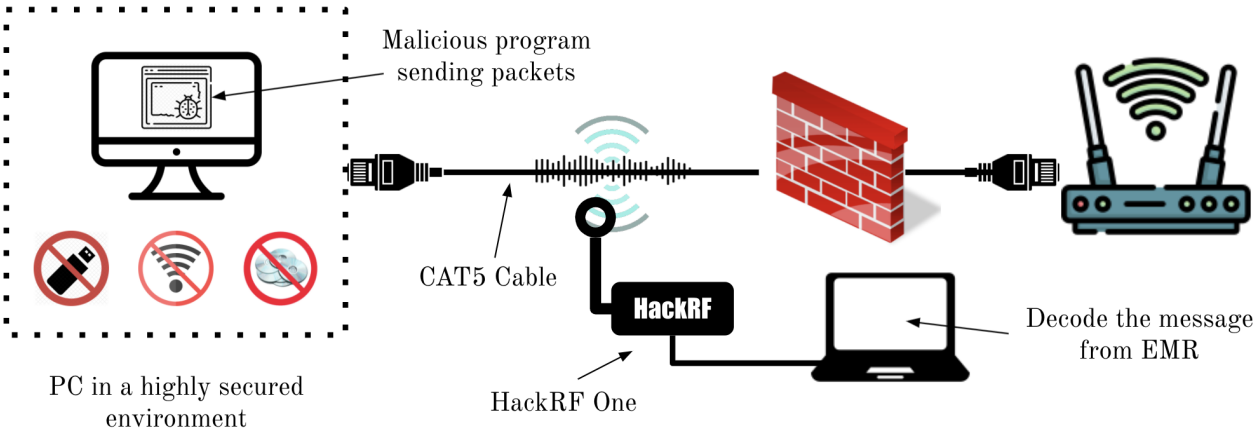

Figure 1. Illustration of the proposed attack model and experimental setup. Malware on an infected target computer manipulates the network communication, resulting in deliberate EMR patterns used to transmit a covert signal to a nearby receiver.

4. Detecting Radiation Frequency

When building a covert channel over the EM radiation of Ethernet cables, it is necessary for the attacker to be aware of the frequency of the EM radiation. The attacker would be able to receive and demodulate data only if this information-leaking frequency is known. Therefore, the first step of developing an EM-based covert channel is to identify the information-leaking EM radiation frequency of the target Ethernet cable using the most reliable approach. Different categories of Ethernet cables have different emission frequencies. The following experiment was conducted to identify the radiation frequency of a cable when two dissimilar network packet patterns are delivered through it. Those two network packet patterns can later be used to modulate a digital `1` and `0` on the physical layer of the covert channel.

4.1. Hardware and Software Setup for Experiments

The experimental hardware setup uses a computer connected to a router through a Cat5 UTP Ethernet cable, representing the target ICS's wired network. The resultant radiation signals are observed through a *HackRF One* SDR device [22], along with a magnetic H-loop antenna, connected to the attacker's computer. The observed EM radiation data are stored on the attacker's laptop for postmortem analysis. Figure 1 illustrates the arrangement of the target network and the attacker's equipment in a realistic scenario. For the experimental evaluation of this phase, i.e., detecting the radiation frequency, the computer to which the Cat5 cable is connected and the attacker's computer to which the SDR hardware are connected are the same machine.

The software running on the attacker's laptop consists of an EM trace collector with three parallel threads collecting data. The first thread sends a set of predetermined network packet patterns through the Ethernet cable to the router, causing the Ethernet cable to produce intentional EM radiation patterns. The second thread reads the EM radiation data captured through the SDR hardware and saves them into EM trace files. The third thread sniffs the Ethernet network interface of the attacker's computer (to which the Ethernet cable is connected for experimental purposes), and saves network traffic packets as PCAP files. These three threads operate in a synchronized fashion so that each captured EM radiation data file corresponds to a specific transmitted network packet pattern through the wired network. The objective is to use these synchronized data to identify two network traffic patterns that cause two uniquely distinguishable EMR patterns.

4.2. Data Collection

Since a HackRF One SDR device was used for capturing EM radiation, the possible EM radiation frequency range to consider can span from 10 Mz to 6 GHz. However,

it was empirically observed that the frequencies closer to the lowest edge of HackRF's sensitive range are often affected by the device's internal noise itself. Therefore, it was decided to consider 30 MHz as the starting frequency in the suspecting frequency range to scan through. Similarly, it is important to decide the upper bound of the suspecting frequency range. Although it could have been set to 6 GHz, i.e., HackRF's maximum sensitive frequency, it was practically impossible to consider such a large frequency range. This is due to the storage and computational overhead posed by the data produced by the HackRF device. Due to this reason, it was decided to consider frequencies only up to 220 MHz in the experimental procedure. As a result, this work considered the frequency range from 30 MHz to 220 MHz in the experiments to identify an information-leaking EM radiation signal.

The EMR data collection thread of the attacker's computer controls the SDR hardware to scan the frequency range from 30 MHz to 220 MHz, with 1 MHz steps. This results in 160 frequency channels to monitor, e.g., 30 MHz, 31 MHz, 32 MHz, . . . , 218 MHz, 219 MHz, and 220 MHz. The procedure in collecting data was to transmit different network traffic patterns through the Ethernet cable while capturing the EM radiation coming from each of the aforementioned frequency channels. For each frequency channel, six EM trace files and six network packet trace files were recorded, respectively, per each network packet transmission pattern. While a broad range of potential network packet patterns can be effective in producing unique and detectable EMR patterns, the following limited set of network packet patterns were used during the experimentation due to simplicity and ease of performing experiments:

- No packets (Medium is idle)—P0;
- 50 UDP packets with non-zero payload—P1;
- 50 UDP packets with all zero payload—P2;
- 50 TCP packets with non-zero payload—P3;
- 50 TCP packets with all zero payload—P4;
- 50 IP packets with all zero payload—P5.

4.3. Dissimilarity Analysis

In order to build a covert channel to infiltrate data, there should be a way to represent bit-level `0` and `1` on the EM radiation patterns. If the two EM radiation patterns are significantly different from each other, it would be possible to demodulate data from the EM radiation with minimum errors. Therefore, after the dataset is collected, a dissimilarity analysis was employed to find two network traffic patterns that create the most dissimilar radiation from the cable. Once it is identified, these patterns can be used to modulate the bit-level `1` and `0` representations. In order to quantify the dissimilarity among different EMR patterns, three metrics were empirically evaluated—namely cosine similarity, Pearson correlation, and cross-correlation. The findings of these metrics are compared with each other below to find the best pairs of network traffic patterns.

In order to perform a comparison between the dissimilarity metrics and identify the two network traffic patterns that are producing a significantly distinguishable EMR signature, it is necessary to have a well-defined process. For this purpose, a leakage frequency analysis algorithm is introduced as illustrated in Algorithm 1. This algorithm takes the Fast Fourier Transformation (FFT) of the EM trace files and uses the three aforementioned similarity metrics on the FFTs. The metrics are recorded for subsequent layer analysis. When calculating the FFTs of the two EM trace files, and slight differences in their size can cause a problem. To overcome this issue, the shortest file length is considered as the window size for the FFT.

After the dissimilarity algorithm was run over the collected EM data set, the two network traffic patterns that produced the most dissimilar EM radiation patterns were selected to represent the bit-level `1` and `0`. As will be described further in Section 6, the two patterns that were selected are P0 and P1. Figure 2 illustrates the waveform of those two EM trace files that correspond to network traffic patterns P0 (in orange colour) and P1 (in

blue colour). A clear difference is observable between the two packet patterns in these two EM radiation waveforms. Due to this difference, the emitting frequency for the Cat5 UTP cable is identified as 220 MHz. Similarly, the best patterns to modulate binary `1` and `0` are chosen as P0, i.e., when the medium is idle, and P1, i.e., when UDP packets with non-zero payloads were sent. Furthermore, the results from the dissimilarity algorithm revealed that cross-correlation is the best metric to analyse the dissimilarity between the two signals.

Algorithm 1 Leakage frequency analysis algorithm

Input: Data = Data set containing pattern traces.
Output: Best patterns pair with minimum similarity.

for $freq \leftarrow start...end$ **do**
  for $pattern_i \leftarrow$Data[$freq$] **do**
    for $pattern_j \leftarrow$Data[$freq$] **do**
      $windowSize \leftarrow minLength(pattern_i, pattern_j)$
      $fft_i \leftarrow getFFT(pattern_i, windowSize)$
      $fft_j \leftarrow getFFT(pattern_j, windowSize)$
      $cosineSimilarity \leftarrow$
      $getCosineSimilarity(fft_i, fft_j)$
      $pearsonCorrelation \leftarrow$
      $getPearsonCor(fft_i, fft_j)$
      $xcor \leftarrow$
      $crossCorrelate(fft_i, fft_j)$
      $results[\,] \leftarrow$
      $(pattern_i, pattern_j,$
      $cosineSimilarity, pearsonCorrelation, xcor)$
    end for
  end for
end for
$Output \leftarrow minimumSimilarity(results[\,])$

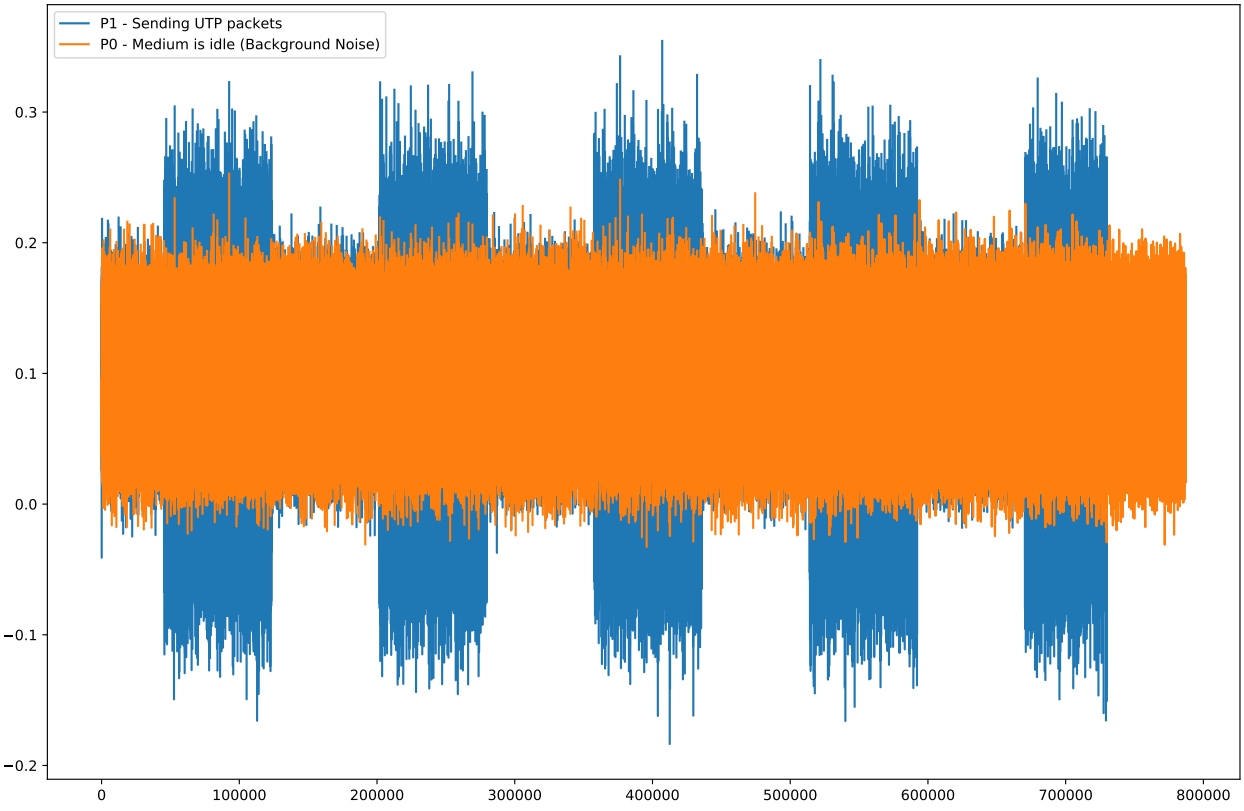

Figure 2. Comparison of waveform of traffic patterns P0 & P1 at 220 Mhz.

5. Covert Channel Design and Implementation

This section presents the overall design and implementation of the covert channel, where data framing, encoding, transmission, and reception are aligned in a coherent manner.

5.1. Structure and Encoding of Data Frame

When designing the covert channel, simply modulating digital `1` and `0` values in the data using the two previously identified network traffic patterns is not reliable. A communication channel of this nature has two important requirements in order to be useful: (1) minimising errors and (2) time synchronisation between the sender and receiver. For the former requirement, a data frame structure was designed incorporating error correcting codes. The frame contains an eight bit preamble, which is an alternating 1 and 0 sequence to detect the start of the frame. The payload is 42 bits long, and an eight bit cyclic redundancy check (CRC) is generated on the payload and appended to the end of the frame. Overall, the data frame is 64 bits long. Both the payload and CRC are encoded with Hamming codes. Hamming codes are a family of binary linear error correcting block codes. They are considered to be a *perfect code* that has a high throughput compared to other error correction codes. Hamming codes can correct one bit errors. Hamming(7,4) code is adopted in this research for error correction. It encodes a four bit word into a seven bit code word by adding three parity bits.

Due to the specific Hamming code in use, seven bits in the payload of the data frame are used to represent four bits of data. Therefore, an ASCII character, i.e., with a size of eight bits, occupies fourteen bits in the payload of a data frame. Since the payload carried by a data frame has a capacity of 42-bits, a single data frame can be used to deliver three ASCII characters at a time. Meanwhile, for the modulation of data to the physical medium, the Manchester encoding scheme is used. With Manchester encoding, binary `1` is represented by a rising edge, while binary `0` is represented by a falling edge. These transitions take place at clock pulses. In this manner, the clock signal and the data stream are encoded together. As a result, any requirement for clock synchronization is eliminated. Since the amplitude of the signals varies over time, simple encoding schemes, such as On–Off Keying (OOK), cannot be used here [24].

5.2. Transmission through Covert Channel

Algorithm 2 illustrates the procedure of preparing frames from the information, and finally modulates these frames into the physical medium by generating Ethernet packet patterns on the cable. Line 1 of the algorithm checks whether there is a valid string to be transferred. It will split the string into chunks of maximum payload size and, in each iteration, only one chunk will be considered. Line 3 conducts the frame preparation from the supplied payload. From Line 5 onward, bits are modulated to the covert channel. Line 6 defines the time limit given to transfer the current bit, and it should transfer the bit in that given time limit. In Manchester encoding, a rising edge denotes binary `1`. Then, the algorithm sleeps for half of the *bitTrxTime*. In another half of the *bitTrxTime*, it sends the packets through the Ethernet cable, which mimics a rising edge. The opposite happens for binary `0`. First, it sends a packet for half a *bitTrxTime* and sleeps for the other half. In an opposite pattern to before, this mimics a falling edge.

Algorithm 2 EM signal modulation algorithm

Input: string, bitTrxTime, payloadSize

```
1: while string.length > 0 do
2:     payload ← getNextChunck(payloadSize, string)
3:     frame ← prepareFrame(payload)
4:     bitTrxEndTime ← getCurrentTime()
5:     for all bit ← frame do
6:         bitTrxEndTime ← bitTrxEndTime + bitTrxTime
7:         halfBitTrxEndTime ← bitTrxEndTime + (bitTrxTime/2)
8:         if bit == 1 then
9:             sleep(bitTrxTime/2)
10:            while getCurrentTime() < bitTrxEndTime do
11:                sendPacketPattern()
12:            end while
13:        end if
14:        if bit == 0 then
15:            while getCurrentTime() < halfBitTrxEndTime do
16:                sendPacketPattern()
17:            end while
18:            sleep(bitTrxTime/2)
19:        end if
20:    end for
21: end while
```

5.3. Reception through Covert Channel

Algorithm 3 illustrates the procedure of demodulating transmitted data from the captured EMR data, which is based on a simple template matching technique. It uses normalized cross correlation to match the patterns to decode the data. In the algorithm, Lines 1 to 5 initialise the required variables and parameters. Inside the main loop, Line 7 picks the next available portion (the size is equal to the window size) from the input array, i.e., *signalFile*. Each window is appended to an array called *samples*. If any preprocessing is required, e.g., taking FFT/Power-Spectral Density, it can be done to the window before appending it to the samples. Lines 9-12 of the algorithm attempts to detect the preamble. If the preamble is not detected and there are enough samples to detect it, then it will proceed to detect a preamble in the *samples*. The *detectPreamble* procedure compares the current samples array with a predefined preamble template to check if the *samples* array looks similar to a preamble. As it was discovered by experimentation in Section 4, cross-correlation performs better in comparing two EM radiation patterns. Therefore, it was used for the comparison between the current sample array and the preamble template. If the result passes a defined threshold value, it is considered that a preamble is detected. Once a preamble is detected, the whole *samples* array is cleared.

Once the preamble is detected and there are enough samples to decode a bit, Line 14 will decode the bit. The *decodeBit* procedure works in a similar manner to *detectPreamble*. It also takes the cross-correlation to compare the samples to check if it looks similar to a 1 or 0. It takes the cross-correlation with both templates 0 and 1 and picks the highest value to classify the decoded bit into a `1` or a `1`. This value should also pass a predefined threshold value. Once a bit is decoded, it is added to *demodulateArray*. When the *demodulatedArray* reaches enough bits for a frame, Lines 18-20 of the algorithm detect and correct errors in the demodulated signal and decode the frame. In the decoding stage, it will validate the CRC, and if it is successful, the bits will be decoded to resolve the original message.

The software code used for the implementation and evaluation of the work presented as part of this paper are available in an open source GitHub repository (https://github.com/shakthisachintha/em-covert-channel (accessed on 20 January 2023)).

Algorithm 3 EM signal demodulation algorithm

Input: signalFile, sampleRate, windowSize, bitTime, preambleSize
Output: Collection of decoded frames

```
enabled ← False
windowsPerBit ← bitTime ∗ sampleRate / windowSize
demodulatedArray[ ] ← [ ]
decodedFrames ← [ ]
samples ← [ ]
while signalFile.hasNext() do
    window ← signalFile.getNextWindow(windowSize)
    samples.append(window)
    if notenabled & enoughSamplesForPreamble(preambleSize) then
        enabled ← detectPreamble(samples)
        samples.pop(0)
    end if
    if enabled & enoughSamplesForBit(windowsPerBit) then
        bit ← decodeBit(samples)
        demodulatedArray.append(bit)
    end if
    if enoughBitsPerFrame(demodulatedArray) then
        frame ← errorCorrect(demodulatedArray)
        frame ← decodeFrame(frame)
        decodedFrames.append(frame)
    end if
end while
Output ← decodedFrames
```

6. Evaluation

The proposed covert channel was evaluated across multiple aspects to evaluate its effectiveness. This section outlines this evaluation.

6.1. Electromagnetic Radiation Dataset

For the purpose of experimentation, an EMR data set was collected on the aforementioned experimental setup. Table 2 illustrates a summary of the dataset collected. This dataset represents EMR trace files captured while each of the network traffic patterns were being transmitted thorough the Ethernet cable.

Table 2. EMR data set details.

Total Size	350 GB
Compressed Size	42 GB
Frequency Range	30–220 Mhz
Data Description	For each frequency from 30 to 220 Mhz, data set contains EMR trace files for each packet pattern and corresponding *pcap* files.
Total Samples	190 × 6 *pcap* files, 190 × 6 IQ files
Total Time	4 h 30 min Approximately
Hardware Setup	UTP cable, HackRF One SDR, H-Loop antenna, SLT Router, MacBook Pro
Software Setup	Python3.7, Scapy Library, GNU Radio Library, h5py Library for data compression

6.2. Leakage Frequency Detection

In order to evaluate the leakage frequency detection algorithm, experiments were conducted against the aforementioned dataset captured using the experimental setup. At the end of each experiment, the dissimilarity metrics for each pair of the considered network traffic patterns were plotted. With the plotted values, it is easier to identify valleys

that provide the similarity between the patterns. The lower the similarity, the better the corresponding pair of network traffic patterns to represent digital `0` and `1` symbols. Out of the three similarity metrics considered, cosine similarity and Pearson correlation demonstrated similar results, while cross-correlation indicated opposite results to the previous two. With these results, further analysis was narrowed down to a few optimal frequencies. Figure 3 illustrates the selected frequencies from the cosine and Pearson correlations: 42–50 MHz, 91–99 MHz, and 200–213 MHz. Similarly, the selected frequencies from cross-correlation are also shown in Figure 4: 30–35 MHz, 120–125 MHz, and 148–152 MHz.

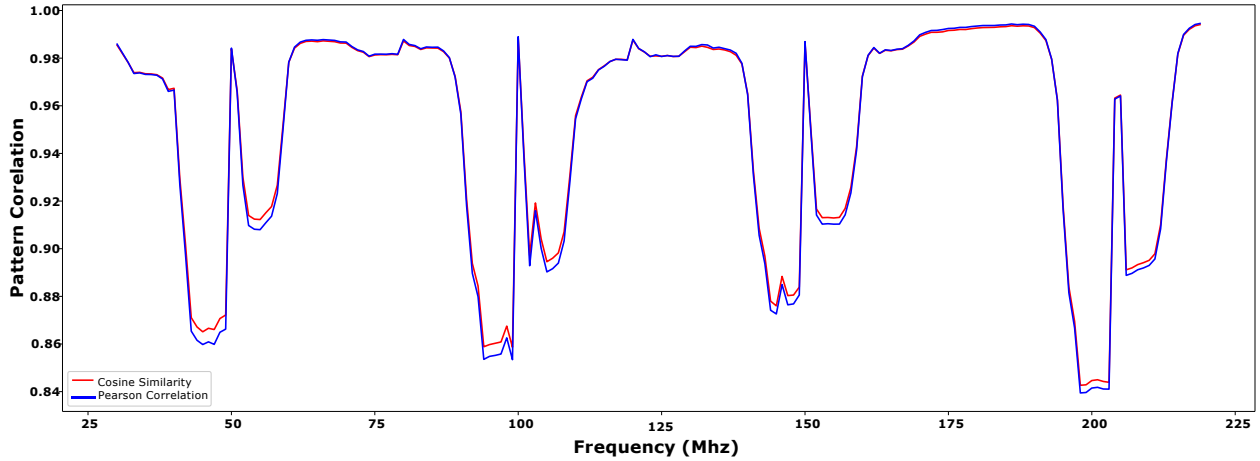

Figure 3. Cosine similarity (red) and Pearson correlation (blue) for the two traffic patterns across the frequencies analysed.

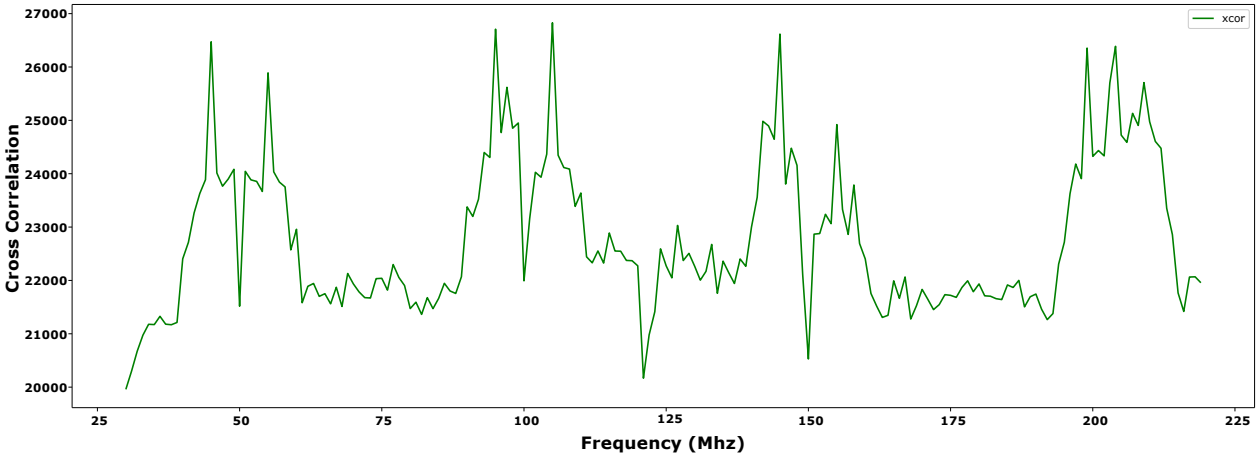

Figure 4. Cross-correlation for the two traffic patterns across the frequencies analysed.

6.3. Quality of Covert Channel

In order to evaluate the quality of the proposed EM covert channel, multiple network quality metrics were considered and experimentally evaluated. Bit error rate (BER) is a measure of the percentage of bits with errors against the total number of bits transferred through the network. One factor that can affect the BER is the physical distance between the transmitter and receiver on the network. In order to evaluate how the BER varies with distance between the target Ethernet cable and the attacker's hardware, the following experiment was conducted. While keeping the data rate of the covert channel to 4 bps, and the Ethernet cable type to Cat5 UTP, the distance between the victim computer and the SDR receiver were varied from 0 m to 2 m. The EMR signal was captured at 220 MHz and demodulated to retrieve the data. Figure 5 illustrates the variation of BER with distance. At 2 m distance, the BER reaches an acceptable 25%.

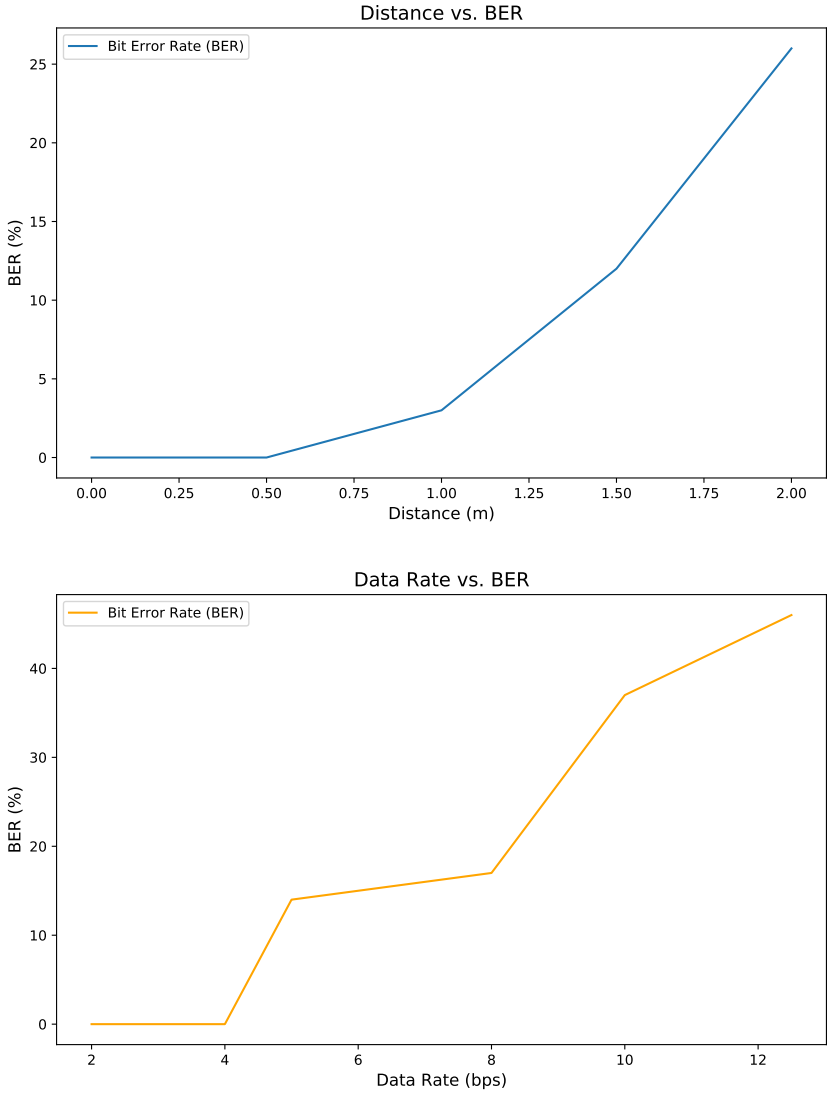

Figure 5. Bit error rate with respect to the distance and data rate.

The signal-to-noise ratio (SNR) is a measure of how clear the transmitted signal is against the background noise in the physical medium. The variation of SNR was calculated for the previous experiments, along with the distance between the victim computer and the attacker's hardware setup. The SNR varied from approximately 19 dB at 0 m to 4 dB at 2 m distance for Cat5 UTP cable.

Another important aspect of the proposed covert channel is the ability to transfer data in an acceptably fast rate. To evaluate the data rates achievable through the proposed covert channel, the following experiment was conducted. From the target computer, frames were transferred at different data rates and captured in the corresponding EM traces. The captured EM traces were demodulated to calculate their BER at different data rates. The bottom of Figure 5 illustrates the variation of BER against the data rate on the network. It is evident that the BER linearly increases from 0% to approximately 45% while the data rate varies from 2 bps to 12.5 bps. According to the figure, with a Cat5 UTP cable, a data rate of 4 bps can be achieved without any bit errors from a distance of 2 m.

7. Discussion

Various countermeasures can be taken to avoid the deployment of these kinds of attacks. One of them is to use shielded Ethernet cables, which can eliminate radiation emissions to a significant extent. Metal shielding around the cable is widely used to reduce

EM interference from other sources, as well as to limit the emission of EM radiation from within the cable to the outside. However, employing fully shielded cables costs over twice the price of using standard UTP cables. This is due to the special material included inside an Ethernet cable, in addition to the twisted pairs of wires. Due to cost constraints, the latter option may be preferred in many networks including industrial environments where ICSs are deployed, even though it reduces the overall security.

Another option would be to use network monitoring tools to monitor the network and detect abnormal activities. These tools can be configured to detect the packet bursts that create EM covert signals. In such a case, it can block access to the network interface or disable the network interface altogether as a countermeasure. It requires further research to explore the potential of such use of network monitoring tools.

When an ICS is operating with a potentially vulnerable wired network, radio signal jammers can be employed as a viable countermeasure. Modern signal jammers can jam the entire bandwidth of a signal's emitting spectrum. When the specific frequency range of a device's EM radiation is blocked by a jamming signal, the noise may increase the error rate of the covert channel's transmission, rendering it unusable. However, it is still possible for higher harmonics of an EM radiation of an Ethernet cable to fall outside the frequency range being jammed. In such circumstances, an attacker may potentially use such a higher harmonic of the information-leaking signal for the covert channel, bypassing the blocked frequency limitations.

When initially designing the experimentation conducted as part of this paper, it was believed that EMR from Ethernet cables would not travel any considerable distance, and therefore, this work used a near-field antenna to capture the EMR. However, it was found that this signal remains capturable without using a near-field antenna from distances as far as 2 m. Therefore, it is possible to employ this covert channel from much further distances with the help of a suitably focused antenna.

ICS networks, as with any other wired network, are under threat from various sources. However, unlike other general purpose wired networks, ICS networks consist of special-purpose devices, which operate autonomously to execute an industrial functionality, such as controlling a production line. Due to the use of specialised hardware in the network, an attacker can fine-tune an attack to specifically match known weaknesses and vulnerabilities on the special-purpose hardware used in an ICS network. This possibility puts ICS networks under a broader threat than general wired networks. Similarly, the demonstrated EM-based covert channel can be fine-tuned to the specific nature of hardware and network cables used on a specific ICS network, such as the specific EM radiation emitting frequencies. Therefore, it is necessary for operators of ICS networks to keep potential EM-based attacks and covert channels under close watch.

8. Conclusions

This work presented a novel method to create a covert channel leveraging wired Ethernet cables, commonly used in ICSs and other networked devices. Malware residing in the target's computer deliberately sends packet bursts to the Ethernet cable, encoding the desired data to exfiltrate by modulating the packet patterns used. The attacker captures the EMR from the Ethernet cable and subsequently demodulates it to recover the transmitted data. The covert channel is evaluated from multiple aspects, revealing that it can operate reliably under low data rates, i.e., 4 bps, from a distance of 2 m. The emitting frequency finding phase generated a massive EM dataset that is feature-rich and can be used in further research. The volume of network generated EM data gathered as part of this approach opens up several avenues for further research leveraging artificial intelligence on the volume of network traffic collected [25].

Author Contributions: Formal analysis, S.S.; Writing—review & editing, N.-A.L.-K. and M.S.; Supervision, A.P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. McLaughlin, S.; Konstantinou, C.; Wang, X.; Davi, L.; Sadeghi, A.R.; Maniatakos, M.; Karri, R. The Cybersecurity Landscape in Industrial Control Systems. *Proc. IEEE* **2016**, *104*, 1039–1057. [CrossRef]
2. Semenkov, K.; Mengazetdinov, N.; Poletykin, A. Extending Operation Lifespan of Instrumentation and Control Systems with Virtualization Technologies. In Proceedings of the 2019 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 8–14 September 2019; pp. 1–5. [CrossRef]
3. Alladi, T.; Chamola, V.; Zeadally, S. Industrial Control Systems: Cyberattack trends and countermeasures. *Comput. Commun.* **2020**, *155*, 1–8. [CrossRef]
4. Hayashi, Y.; Homma, N.; Watanabe, T.; Price, W.; Radasky, W. Introduction to the special section on electromagnetic information security. *IEEE Trans. Electromagn. Compat.* **2013**, *55*, 539–546. [CrossRef]
5. Le, Q.; Miralles-Pechuán, L.; Sayakkara, A.; Le-Khac, N.A.; Scanlon, M. Identifying Internet of Things software activities using deep learning-based electromagnetic side-channel analysis. *Forensic Sci. Int. Digit. Investig.* **2021**, *39*, 301308. [CrossRef]
6. Sayakkara, A.; Le-Khac, N.A.; Scanlon, M. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digital Investig.* **2019**, *29*, 43–54. [CrossRef]
7. Guri, M.; Solewicz, Y.; Daidakulov, A.; Elovici, Y. Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise ('DiskFiltration'). In Proceedings of the 22nd European Symposium on Research in Computer Security, Oslo, Norway, 11–15 September 2017; pp. 98–115. [CrossRef]
8. Guri, M.; Zadov, B.; Bykhovsky, D.; Elovici, Y. PowerHammer: Exfiltrating Data from Air-Gapped Computers through Power Lines. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1879–1890. [CrossRef]
9. Guri, M.; Zadov, B.; Daidakulov, A.; Elovici, Y. XLED: Covert Data Exfiltration from Air-Gapped Networks via Switch and Router LEDs. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust, PST 2018, Belfast, UK, 28–30 August 2018. [CrossRef]
10. Zhan, Z.; Zhang, Z.; Koutsoukos, X. BitJabber: The World's Fastest Electromagnetic Covert Channel. In Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2020, San Jose, CA, USA, 7–11 December 2020; pp. 35–45. [CrossRef]
11. Sayakkara, A.; Le-Khac, N.A.; Scanlon, M. Electromagnetic Side-Channel Attacks: Potential for Progressing Hindered Digital Forensic Analysis. In Proceedings of the Companion Proceedings for the ISSTA/ECOOP 2018 Workshops, Amsterdam, The Netherlands, 16–21 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 138–143. [CrossRef]
12. Schulz, M.; Klapper, P.; Hollick, M.; Tews, E.; Katzenbeisser, S. Trust the wire, they always told me! on practical non-destructive wire-tap attacks against ethernet. In Proceedings of the WiSec 2016—Proceedings of the 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Darmstadt, Germany, 18–22 July 2016; pp. 43–48. [CrossRef]
13. Guri, M.; Monitz, M.; Mirski, Y.; Elovici, Y. BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal Manipulations. In Proceedings of the 2015 IEEE 28th Computer Security Foundations Symposium, Verona, Italy, 13–17 July 2015; pp. 276–289. [CrossRef]
14. Guri, M.; Daidakulov, A.; Elovici, Y. MAGNETO: Covert Channel between Air-Gapped Systems and Nearby Smartphones via CPU-Generated Magnetic Fields. *arXiv* **2018**, arxiv:1802.02317.
15. Guri, M.; Zadov, B.; Daidakulov, A.; Elovici, Y. ODINI : Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields. *arXiv* **2018**, arXiv:1802.02700.
16. Guri, M.; Monitz, M.; Elovici, Y. USBee: Air-gap covert-channel via electromagnetic emission from USB. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust, PST 2016, Auckland, New Zealand, 12–14 December 2016; pp. 264–268. [CrossRef]
17. Guri, M.; Zadov, B.; Atias, E.; Elovici, Y. LED-it-GO: Leaking (a lot of) Data from Air-Gapped Computers via the (small) Hard Drive LED. *arXiv* **2017**, arXiv:1702.06715.
18. van Eck, W. Electromagnetic radiation from video display units: An eavesdropping risk? *Comput. Secur.* **1985**, *4*, 269–286. [CrossRef]
19. *IEEE Std 802.3u-1995*; IEEE Standards for Local and Metropolitan Area Networks: Supplement—Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100Mb/s Operation, Type 100BASE-T (Clauses 21-30). 26 October 1995; pp. 1–415. [CrossRef]
20. Jeffrey, I.; Gilmore, C.; Siemens, G.; LoVetri, J. Hardware invariant protocol disruptive interference for 100BaseTX Ethernet communications. *IEEE Trans. Electromagn. Compat.* **2004**, *46*, 412–422. [CrossRef]

21. Wang, S.; Han, P.; Qiu, Y.; Tian, J. Information leakage of electromagnetic emission from differential transmission lines. In Proceedings of the 2017 IEEE 5th International Symposium on Electromagnetic Compatibility (EMC-Beijing), Beijing, China, 28–31 October 2017; pp. 1–4. [CrossRef]
22. Ossmann, M. HackRF. Available online: greatscottgadgets.com (accessed on 29 August 2020).
23. Wang, A.; Liang, R.; Liu, X.; Zhang, Y.; Chen, K.; Li, J. An inside look at IoT malware. In Proceedings of the Industrial IoT Technologies and Applications: Second EAI International Conference, Industrial IoT 2017, Wuhu, China, 25–26 March 2017; pp. 176–186.
24. Guri, M. Lantenna: Exfiltrating data from air-gapped networks via ethernet cables emission. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 12–16 July 2021; pp. 745–754. [CrossRef]
25. Rizvi, S.; Scanlon, M.; McGibney, J.; Sheppard, J. Application of Artificial Intelligence to Network Forensics: Survey, Challenges and Future Directions. *IEEE Access* **2022**, *10*, 110362–110384. [CrossRef]