NoiseHopper: Emission Hopping Air-Gap Covert Side Channel with Lower Probability of Detection

1st Md Faizul Bari
Dept. of ECE, Purdue University, West Lafayette, IN, USA
mbari@purdue.edu

2nd Shreyas Sen
Dept. of ECE, Purdue University, West Lafayette, IN, USA
shreyas@purdue.edu

2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) | 979-8-3503-7394-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/HOST55342.2024.10545402

***Abstract*—To shield against malicious attack vectors and safeguard sensitive data, organizations resort to physical isolation called 'air-gap' where the air-gapped device is isolated from the public internet and can only be connected to an internal, secured, 'air-gap network'. Due to their sensitive nature, air-gap networks have been a coveted target for motivated adversaries, leading to various malware/worms that can infect these devices via insider threats, unauthorized software updates, peripherals, or supply chain attacks and collect data. Due to the absence of a connection to the outside network, collected data cannot be exfiltrated easily. Attackers have developed 'air-gap covert channels' to bridge the gap between the air-gap network and the outside network. These are intentionally generated electromagnetic (EM) emissions produced by varying CPU load or exploiting memory instructions and modulated with data. However, existing covert channels have several limitations. The channels are covert in the sense that the malware is not easy to detect, but the wireless signal itself can be identified as a malicious anomaly by spectrum monitoring tools. Since emission is generated by exploiting CPU/memory which is shared with other parallelly running processes, the channel can be interrupted by their simultaneous activities. Also, most of them have very low data rates ($\leq$1 kbps) that cannot transmit significant data volume in a reasonable time and are not suitable for low-power, air-gapped embedded devices with limited resources. In this work, we propose 'NoiseHopper', an improved covert side channel formed by pulse width modulation (PWM) controlled EM emission with spectrum covertness rendered by frequency hopping. It looks like noise or spurious peaks have been added to the existing RF spectrum, rendering low detection probability. It doesn't depend on any specific shared hardware or peripherals, is suitable for embedded devices, and can transmit data to ~5.5 m range at 100 kbps. We have implemented our proposed method on an ATmega328P microcontroller (part of the AVR family that is found in many embedded systems) and transmitted MNIST dataset images to show its efficacy. The proposed covert channel has been shown to transmit through a 15 cm thick wall to make it more realistic. The bit error rate (BER) has been analyzed. Finally, a few probable countermeasures have been proposed to prevent data leakage.**

***Index Terms*—Covert Communication, Frequency Hopping, Air Gaps, Emanation, Data Exfiltration, Embedded System**

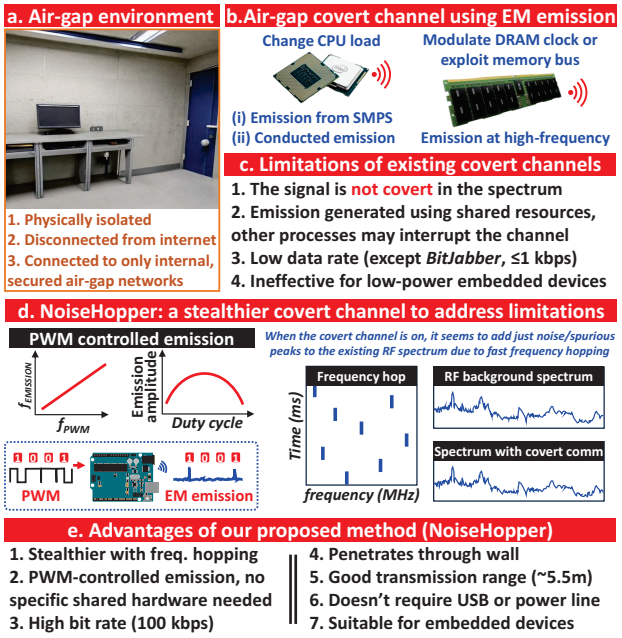

Fig. 1. (a) Air-gap devices in an air-gap environment refers to physically isolated devices with sensitive data. They are not connected to the internet and can only connect to internal air-gap networks. (b) To jump over air-gap, attackers have developed EM emission-based covert channels by varying CPU load or exploiting the DRAM clock/memory bus. (c) However, they have limitations as the malware is stealthy in the code execution domain but not covert in the RF spectrum. They have low data rates and their generation method uses shared resources that other running processes may interrupt. (d) We propose NoiseHopper, a PWM-controlled emission with frequency hopping that looks like adding noise/spurious peaks to the existing RF spectrum, hence more stealthy (shown in the conceptual diagram). (e) Advantages of NoiseHopper include: a stealthier channel with a high bit rate, no shared resources, USB or power line needed, can penetrate through a wall, and has good transmission range. Also, it's suited for embedded devices.

I. Introduction

A. Air-gapped Environment

To protect the confidentiality, integrity, and availability (the CIA triad) of sensitive data from complex cyber-attacks, critical infrastructures often deploy physical isolation called 'air-gap' where air-gapped devices are disconnected from the public internet and can only be connected to a secured 'air-gap network'. Wireless interface (e.g., Bluetooth, WiFi, etc.) of air-gapped computers is removed and physical access to such environment is strictly controlled. Examples of such networks include SIPRNet or Secret Internet Protocol Router Network [1] and JWICS or Joint Worldwide Intelligence Communications System [2] used to exchange 'secret-level' and 'top-secret' level classified information respectively by the Department of Defense (DoD). Along with these internal networks for own officials, the US govt also has other networks in place for multinational intelligence coalitions. For

example, the NSA uses the National Security Agency Network (NSANET) to exchange 'top-secret' level information among the 'Five Eyes' partners [3]. Apart from the defense organizations, other facilities may also use air-gapped environments, such as SCADA (supervisory control and data acquisition) systems, nuclear power plants, aerospace, medical diagnostic systems, etc. Fig. 1(a) shows an air-gapped environment.

B. Air-gap Covert Channels

Despite their isolated nature and robust security measures, air-gapped devices are not completely immune to malicious attacks. Motivated attackers with necessary resources have developed malware such as Stuxnet [4], Agent.BTZ [5], SymonLoader [6], etc. to penetrate air-gapped networks. Once the malware gets into the 'air-gap network' through air-gapped devices, it collects confidential data. However, the collected data needs to be transmitted to the adversary to finish the attack cycle. This is an extremely challenging task due to the lack of connectivity between air-gapped devices and the outside network. This gives rise to a special communication technique where intentionally generated electromagnetic (EM) emissions are modulated with the collected data to establish a covert communication channel known as the 'air-gap covert channel' [7]–[12] as shown in Fig. 1(b).

Mordechai Guri has developed malware that exploits the dynamic power consumption of the CPU cores in a computer [7], [8]. Due to the dynamic voltage and frequency scaling (DVFS) [13], CPU power consumption changes according to its workload. A dynamic power change means a dynamic change in the 'switching frequency' of the switch-mode power supplies (SMPS), leading to EM emission at the switching frequency. Hence, the malware varies CPU load to two extremes (high workload and almost no load) to generate EM emission at two frequencies, leading to binary FSK modulated data. However, the maximum data rate was only 1 kbps. Also, the transmission range for computers was only up to $1\,\mathrm{m}$, which reduces to only $20\,\mathrm{cm}$ for IoT/embedded devices.

In [9], authors proposed a malware called 'USBEE' that generates intentional EM emanations from data buses of a USB connector, essentially turning any unmodified USB device connected to a computer into an RF transmitter. Like before, the intentional emanation is modulated with binary data and transmitted at a maximum rate of 640 bps. Another malware called 'GSMEM' was proposed in [10] that exploits a specific set of memory instructions to generate intentional EM emission at GSM, UMTS, and LTE frequency bands. The signal is amplified using multi-channel memory architecture and sent to a compromised mobile phone from 1 - $5\,\mathrm{m}$ range. But again, the bit rate was only 1 kbps max with advanced receiver hardware. A similar approach to GSMEM, called AIR-FI, was proposed in [11]. Here, data were transferred to a DDR memory bus. This generates an emission at $2.4\,\mathrm{GHz}$ frequency since that is the DDR clock frequency. However, this is also a typical WiFi band. Hence, AIR-FI malware generates a covert channel in the WiFi frequency bands. However, the maximum bit rate was only 16 bps. To circumvent the low bit rate issue, authors proposed BitJabber [12], a covert channel generated by modulating the emission amplitude corresponding to the DRAM clock. They achieved a 100000 baud rate, translating to 100 kbps (OOK modulation) to 300 kbps (3-bit MFSK) data rate. However, the maximum detection range for BitJabber is only 2 m (max) which reduces to 0.5 m when there is a wall between the transmitter and receiver. This is too short to cross the air gap in most cases.

Instead of EM emission, authors in [14] proposed PowerHammer which varied CPU load to modulate the power supply with intended data. It is a 'conducted emission' that goes through the power line for an adversary to pick it up. Since it is conducted via a power line, the range can be long (authors tested up to 112 m). However, the data rate is extremely low (a few bits per second), making it mostly ineffective for a significant volume of data exfiltration in a reasonable time. Also, it's not suitable for battery-powered devices.

C. Limitations of Existing Attacks

The proposed attacks have several limitations as shown in Fig. 1(c). First of all, the covertness is claimed in terms of the covertness of the malware. In most works, authors have shown how low resources the malware uses to avoid detection by API monitoring and resource tracking. No covertness is provided in the wireless channel against spectrum monitoring. Any spectrum monitoring tool can detect the presence of these channels. For example, in 2021 IARPA launched a program called SCISRS [15] that has developed smart radio systems to monitor the RF spectrum and detect RF anomalies. In GSMEM [10] and AIR-FI [11], the generated emission falls in the cellular and WiFi bands (IEEE 802.11b/g/n) respectively. Do the overlapping cellular or WiFi signals provide any cover for hiding? To answer that question, we need to understand 'Mimicked Signal' or 'Mimics' which resemble known communication signals (e.g., GSM, LTE, OFDM) in frequency, bandwidth, and pulse shape but are unrecognizable by the standard receiver of such protocol. EM emissions from GSMEM and AIR-FI do not qualify as mimics because only their carrier frequencies match while bandwidths and pulse shapes do not. So, they can not be disguised as GSM or WiFi signals. Even if they are modified somehow to appear as mimic signals, it is to be noted that 'mimics' can also be detected by spectrum monitoring tools. In summary, none of the existing methods has spectrum covertness.

Secondly, the emanation generation processes use shared resources such as CPU or memory bus. CPU loads may vary significantly due to other software running in the system, which jeopardizes the covert channel proposed in [7], [8], [14]. The same is true for the other methods that depend on memory read/write [10]–[12]. Also, modulating existing emissions due to the DRAM clock requires the same memory instruction at the same memory location to keep the amplitude and frequency of the emission stable. These are very stringent requirements making the attack less effective.

Thirdly, many organizations have restricted access to USB drives to prevent malware infection or data leakage through

them. This makes USB-based malware like USBEE [9] impractical.

Fourthly, except for BitJabber, all other proposed methods have low data rates ($\leq$1 kbps). This limits the data volume that can be exfiltrated within a reasonable time frame.

Finally, these methods form covert channels for air-gapped computers. However, sensitive facilities involve many isolated, low-power, and resource-limited embedded devices. These devices often have sensors that collect sensitive data and can be exploited. They are often battery-powered, making powerline exploitation via conducted emissions like the PowerHammer [14] method futile. Also, they have low-power SoC or microcontroller in them. Generating high-frequency emissions (Wi-Fi and GSM bands) requires significant power, making these methods unsustainable. Not to mention they do not have separate DRAM to exploit. In addition to that, high-frequency emanations have a shorter range [16]. Hence, intentionally generated high-frequency emissions from embedded devices may never cross the air gap at all.

D. Our Proposed Method

Fig. 1(d) shows our proposed method. We propose an improved covert side channel for air-gapped devices (including embedded devices) by generating pulse width modulation (PWM) controlled EM emission with a much higher degree of covertness rendered by frequency hopping. We have designed malware, called NoiseHopper, for the AVR microcontroller family which is used in many embedded systems, and tested its efficacy with the ATmega328p microcontroller. The malware generates a PWM signal, whose duty cycle (D) determines the emission amplitude. A $\sim$50% duty cycle will cause high amplitude. However, a 0% or a $\sim$100% duty cycle means the PWM will essentially be a DC signal. Due to the absence of signal switching, there will be no emission. This leads to On-Off Keying (OOK) modulated data as follows:

- data bit '1' $\rightarrow$ 50% duty cycle PWM $\rightarrow$ 'high' emission amplitude
- data bit '0' $\rightarrow$ 100% duty cycle PWM (basically DC) $\rightarrow$ 'low' emission amplitude (~ 0)

Since the emission frequency matches closely with its source signal frequency, a PWM signal with frequency f_{PWM} causes emission at $\sim f_{PWM}$. The PWM frequency is switched rapidly to implement fast frequency hopping. Inherently low emission power (unlike regular communication systems, there is no power amplifier for this side channel) with frequency hopping makes the emission signal appear like adding noise or spurious peaks to the existing RF spectrum. That's why we call it NoiseHopper (hopping emissions that seem to be added as noise or spurious peaks). Since frequency hopping signals are more resistant to interference, low SNR has a trivial effect on communication performance. Fig. 1(e) shows why our proposed method is better than other proposed methods. Our proposed covert channel: (i) appears as if some noisy or spurious peaks have been added to the existing spectrum (thanks to frequency hopping and low SNR) making it very difficult for spectrum monitoring tools to detect. Also, changing the PWM frequency and duty cycle requires only a few bit flips in a couple of registers. So, the malware is also stealthy, (ii) does not depend on any specific shared hardware (CPU/memory), and in most cases, other processes running on the device can not interfere with the channel, (iii) suitable for low-power embedded devices, (iv) has high bit rate (up to 100 kbps), (v) generates emission in the low-frequency band to provide a good range ($\sim 5.5\,\mathrm{m}$) of communication to cross the air-gap, (vi) can penetrate through $15\,\mathrm{cm}$ thick walls, (vii) doesn't require any peripherals like USB, and (viii) doesn't depend on the power line for conducted emission (hence suitable for battery-powered devices).

E. Our Contribution

Our specific contributions are as follows:

- In this work, we propose an improved covert side channel by generating PWM-controlled EM emission with frequency hopping. During communication, the channel appears as if noise or spurious peaks have been added to the existing RF spectrum, leading to a much higher degree of covertness with a lower probability of detection. The malware (NoiseHopper) is stealthy as well since it only requires setting a few bits in common registers.
- We have designed the covert channel without dependency on any specific shared hardware (e.g., CPU, DRAM, peripherals, dedicated power line, etc.), leading to a sustained channel unperturbed by other parallelly running processes.
- We have implemented our proposed method in an ATMega328P microcontroller and showed its efficacy by transferring MNIST dataset images. This also proves its suitability for embedded devices.
- We have explored the properties of the covert channel (BER, data rate, transmission range, obstacle penetrability, etc.) in detail. NoiseHopper can transmit data up to $\sim$5.5 m range at a maximum of 100 kbps rate and can transmit through a 15 cm thick wall.

F. Organization of the Paper

The rest of the paper is structured as follows: Section II discusses relevant works. In Section III, our attack model is described. In Section IV, covert channel formation using a PWM signal is analyzed. Next, Section V discusses data transmission using our covert channel in detail. The covertness of the channel is increased using frequency hopping which is discussed in Section VI. In Section VII, various properties of the proposed covert channel including bit error rate (BER), data transmission range, data transmission through an obstacle, etc. are described. Section VIII provides a comparison between NoiseHopper and the existing electromagnetic emission-based air-gap covert channels. A few probable countermeasures are discussed in Section IX. Finally, the work is summarized and concluded in Section X.

II. Related Works

During the switching of data and clock signals between the high and low states, a portion of switching energy (i.e., dynamic energy = $\frac{1}{2}CV^2$) converts to electromagnetic radiation called emanation [17]. This is an unintentional emission that often has a strong correlation with the source signal. It is possible to recover the source data partially or totally by processing the emanation signal. Hence, emanation forms a 'side channel' for information leakage bypassing any physical and/or cryptographic access-to-data control methods at hardware, software, and network levels. Although compromising emanations can be of different types, e.g. electromagnetic [18], acoustic [19], optic [20], ultrasonic [21], thermal [22], etc., electromagnetic (EM) emanations are most commonly utilized and widely studied.

EM emanation has been exploited for numerous eavesdropping purposes. Authors in [23] have used emanation from data storage devices to monitor and classify its activity (reading, writing, or silence). In [24], authors have exploited GPU emanation to detect DNN architecture. In [25], emanations from smartphones are used to detect the camera status (both front and rear). Out of the many sources, EM emanations from display devices and display cables are relatively stronger [26] and have been an attractive target for attackers for decades. They have been exploited mostly to recover the screen content [27]–[30]. Due to its leakage property, researchers sought to generate 'intentional emanation' (basically switching emissions). Erik Thiele [31] developed and distributed an open-source program back in 2001 called 'TEMPEST for Eliza', which uses a computer monitor to send out AM-modulated signals.

Intentional generation of EM emissions paved the way for the development of electromagnetic covert channels [7]–[12]. These channels have been discussed earlier in section I-B. An electrical covert channel has also been proposed using 'conducted emission' in the PowerHammer work [14]. Authors in [32] proposed a similar conducted covert channel called NoDE that exploits high-frequency voltage ripples generated by power factor correction circuits built into today's computers. Along with these electromagnetic and electrical channels, other types of covert channels have also been proposed, e.g., magnetic, acoustic, thermal, etc.

EM emissions can be blocked by metal shielding, i.e., Faraday Cage. However, the Faraday cage cannot block magnetic fields. Guri et al. exploited this property to propose ODINI [33] and MAGNETO [34], two types of malware that exploit low-frequency magnetic fields generated by CPUs to exfiltrate data from air-gapped computers, even through Faraday cage. Authors in [35] proposed a magnetic covert channel using smartphone magnetic sensors. They have successfully communicated between a laptop and mobile phone using the proposed channel. While these magnetic covert channels can penetrate the Faraday Cage, they have a very low data rate and a very short data transmission range. Hence, they are not attractive choices for data exfiltration from air-gapped devices.

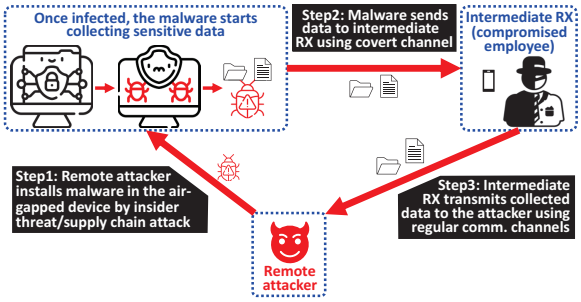

Fig. 2. Attack model of our proposed method. In step 1, the attacker infects air-gap devices with malware through insider threats, supply chain attacks, unauthorized software updates, or peripherals. Once infected, the malware starts collecting sensitive data. Since air-gap devices have no connection to the internet, the malware generates a PWM-controlled, EM-emission-based covert channel and transmits data to an intermediate receiver (a compromised employee or visitor with a phone/radio, or a bug receiver implanted by him/her). This is the second step. Finally, in step 3, the intermediate receiver transmits the collected data to the attacker using a regular digital communication method (WiFi, LTE, etc.) or storage devices (e.g., Pendrive).

Modern computing equipment (keyboard, hard disk drive, desktop, etc.) often includes one/more LED lights for some sort of status indication. However, these optical emissions have been exploited as well to propose an optical covert channel. Authors in [36] exploited keyboard LEDs (Caps-Lock, Num-Lock, and Scroll-Lock) to encode information and exfiltrate data optically. Similar approaches were presented for HDD or camera LEDs [37]. Authors in [38] showed that attackers can send commands to preinstalled malware for data exfiltration by using LASER (attached to a tripod or carried by drones) in combination with multifunction printers. It was shown in [39] that the efficiency of these methods can be improved further by using BFSK modulation. Computer screens have also been exploited through the screen brightness [40] or hidden images [41]. The obvious issue with optical channels is that they can not penetrate a nontransparent obstacle (a wall). Also, optical receivers (e.g., cameras) are harder to sneak into a critical facility.

There are also acoustic covert channels as all devices produce some acoustic footprint while running. Inaudible ultrasound has been used to form acoustic covert channels in [42], [43]. Hard disk noise has also been used to create acoustic covert channels [44] which the author terms as 'DiskFiltration'. Similar acoustic channels have been proposed in [45], [46]. Temperature variation in electronic equipment has also been exploited to form thermal covert channels such as BitWhisper [47] and HOTSPOT [48].

III. Attack Model

Fig. 2 shows our attack model. The attack is performed in 3 steps: (1) infecting the target device and collecting data, (2) crossing the air gap using the covert channel and transmitting collected data to an intermediate receiver, and (3) relaying the collected data from the intermediate receiver to the remote attacker. These steps are discussed below in detail.

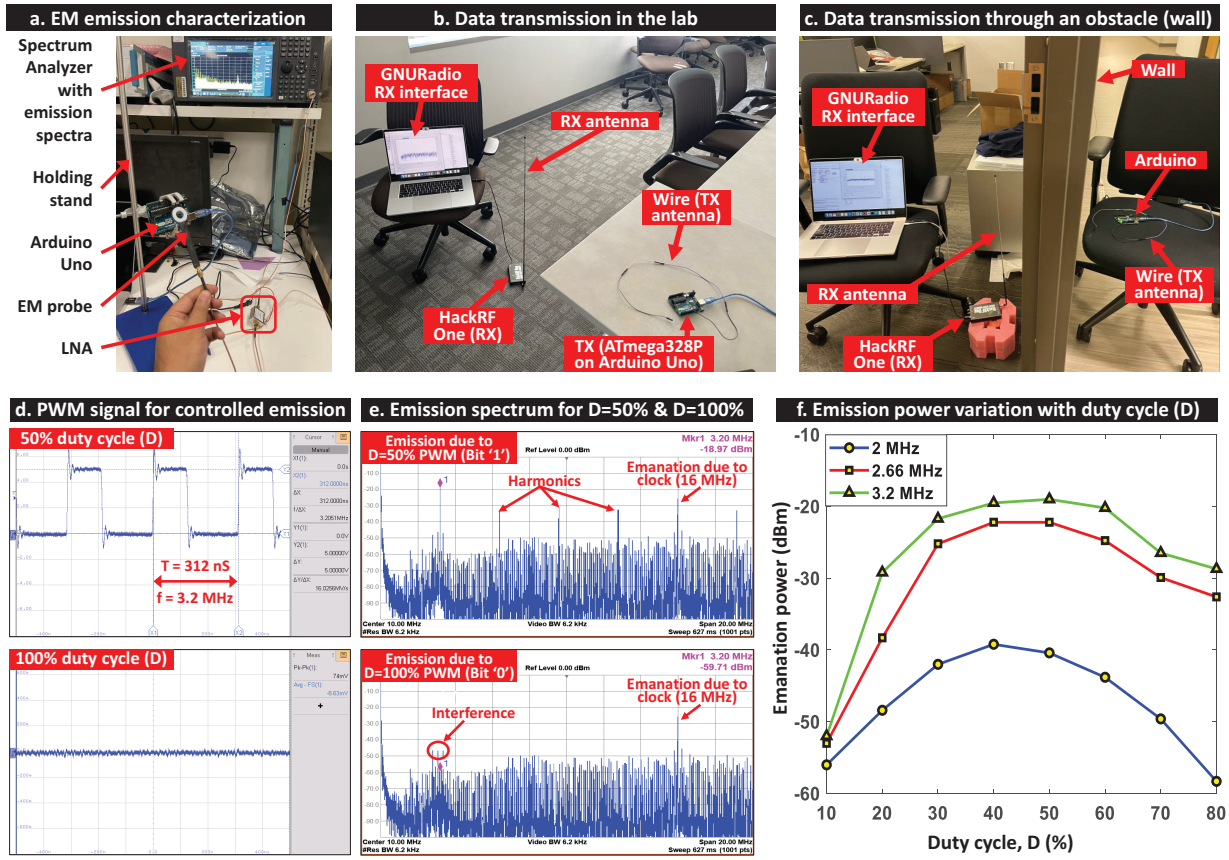

Fig. 3. (a) Experimental setup to characterize the electromagnetic emission pattern. Emission is captured using an EM probe which sends the signal to a spectrum analyzer through an LNA. (b) Experimental setup for data transmission in an office room. PWM-based emission is generated in ATmega328p microcontroller onboard an Arduino Uno, which is used as a transmitter. A HackRF One SDR with a telescopic antenna works as a receiver. The receiver interface is designed using GNURadio. (c) Experimental setup for data transmission through a 15 cm thick wall. The transmitter is kept outside and the receiver is kept inside a room, keeping a wall in between. (d) Pulse Width Modulation (PWM) signal generated in the Arduino for both 50% and 100% duty cycle at 3.2 MHz. It is to be noted that this is an inverted, fast PWM. Hence, 100% duty cycle renders as ∼0 V instead of 5 V. (e) Emission spectra for both 50% and 100% duty cycle PWM when the probe is placed right on top of the microcontroller. 3.2 MHz 50% duty cycle PWM generates EM emission of 3.2 MHz (and its harmonics). This emission is gone for 100% duty cycle PWM. For both cases, unintentional emission (emanation) due to 16 MHz Arduino clock and a few interference signals are present. (f) Emission power variation with duty cycle. Power (& amplitude) is highest near 40-50% duty cycle, which drops on both sides. For <10% and >80% emission goes away. While not completely symmetric, the curve follows the expected trend.

A. *Infecting the Target Device*

At the first stage of the attack, a target is selected and malicious code/malware is installed. This can be achieved via various infection vectors: using malicious insiders (spies), deceived employees, social engineering techniques, peripheral devices, unauthorized software updates, supply chain attacks, etc. This has been done in many attacks on air-gapped networks. A well-known example is the Stuxnet worm which primarily infected the Iranian nuclear facilities [4]. Other similar worms include Agent.BTZ [5], SymonLoader [6], Copperfield [49], etc. Such incidents show the practical effectiveness of such attack vectors.

B. *Intermediate Receiver*

After the initial infection of the air-gapped devices with malware, a covert channel is established and the malware starts collecting data. Unlike traditional communication system that utilizes a power amplifier (PA) at the transmitter to boost the signal being transmitted, covert emissions have no dedicated PA as the hardware is not designed for this channel. Hence, air-gap covert channels are inherently weak, and depending on the generation method some are weaker than others. These EM emission-based covert channels are good for taking the data outside of an air-gapped room within a facility, but not out of the facility to the attacker who might be far away. This is why an intermediate receiver, carried by a man-in-the-middle, is necessary for this type of attack. This middleman can be a compromised employee, service provider, contractor, or even a visitor. Edward Snowden is a well-known example of a compromised insider who leaked highly classified information from the National Security Agency (NSA) in 2013 [50]. It was also reported that he stole the private keys of other employees by convincing them that he needed the key for his work, an example of deceived employees [51].

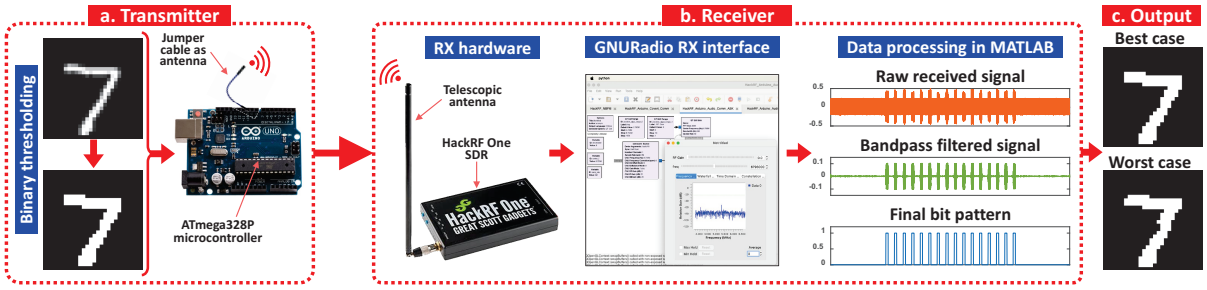

Fig. 4. (a) Images from the MNIST dataset are processed by applying binary thresholding where each pixel value >10 is replaced by 1, otherwise 0. Then they are loaded into the Arduino Uno and transmitted using PWM-controlled emission. (b) A HackRF One is used as a receiver, located 1 m apart. A receiver interface has been designed using GNURadio which captures and stores the received data. These data are then processed in MATLAB by filtering, demodulating, and recovering the bit pattern. (c) The final received image for the best and the worst case.

This kind of man-in-the-middle can receive the data transmitted through the covert channel using his receiver (cellphone or specialized RX device). Alternatively, he can implant a rogue receiver near the air-gapped chamber to collect data.

C. Data Exfiltration

At this stage, the intermediate receiver (man-in-the-middle) transmits the collected data to the actual attacker(s). The attack cycle is now complete and data are stolen. The process runs in a loop where the malware continuously transmits collected data to the intermediate receiver who forwards it to the attacker(s).

IV. Covert Channel Formation

A. Experimental Setup

Fig. 3(a) shows our initial experimental setup for EM emission characterization. A pulse width modulation (PWM) signal has been generated in an ATmega328P microcontroller (onboard Arduino Uno). The generated emission has been captured using an EM probe. The probe is connected to a low noise amplifier (LNA) which sends the amplified signal to a spectrum analyzer (SA) where it is observed and analyzed. The spectrum pattern, dependence of emission frequency on PWM frequency, harmonic pattern, dependence of emission power on the duty cycle of the PWM signal, etc. have been analyzed at this phase. This initial analysis paves the way for designing a data transmission system using the EM emission.

Fig. 3(b) shows our setup for actual data transmission. An Arduino Uno is used as a transmitter (TX). Four jumper cables are connected in series to form $\sim$30 cm long wire which is attached to pin 10 (our PWM pin). This wire works as a transmitting antenna. On the receiver side, a HackRF one, connected with a telescopic antenna, is used as a receiver (RX). A receiver interface has been designed in GNURadio. It samples received data at 8 MSps and saves. Fig. 3(c) shows a similar setup, except there is a 15 cm thick wall between the transmitter and the receiver. The total linear separation (including the wall) between the transmitter and the receiver is $\sim$50 cm. This setup is used to show the efficacy of our proposed channel through an obstacle.

B. PWM Signal Generation

Generating PWM requires setting appropriate waveform generation mode (WGM) bits, compare output mode (COM) bits, and clock select (CS) bits in register TCCR0A and TCCR0B. Here, we have used the 'FastPwmPin' library [52]. It provides the enablePwmPin method which takes a pin number, frequency, and duty cycle (D) to generate the target PWM signal. Fig. 3(d) shows the generated PWM signal at pin 10 for both 50% and 100% duty cycles, measured on an Oscilloscope. By default the PWM signal is inverted, meaning 100% duty cycle provides $\sim$0 V DC instead of 5 V. The sample PWM signal is generated at 3.2 MHz.

C. Channel Formation

Switching in the PWM signal at frequency f_{PWM} causes an EM emission at $\sim f_{PWM}$. Fig. 3(e) shows the emission spectra (0 - 20 MHz span) collected right on top of the ATmega328p for D=50% and D=100% PWM signals respectively, Both the top and bottom figures have unintentional emission (emanation) at 16 MHz due to the Arduino clock, along with a few interferences. However, the switching activity of 3.2 MHz 50% duty cycle PWM generates additional EM emissions with 3.2 MHz fundamental frequency and its harmonics. Due to an absence of switching in 100% duty cycle PWM, the bottom figure shows no intentional emissions.

Fig. 3(f) shows the effect of the duty cycle (D) of the PWM signal on EM emission power at 3 different frequencies (2, 2.66, and 3.2 MHz). At $\sim$40-50% duty cycle, the emission power (hence amplitude) is maximum for all cases. It gradually decreases on both sides of D. However, at $D < 10\%$ or $D > 80\%$, the emission simply goes away. The curves are not perfectly symmetric but follow the expected trend closely. This paves the way for amplitude-modulated emission controlled by the duty cycle. In this work, we have used it for 'On-Off Keying' (OOK) modulation, where D=50% is used to represent bit '1' and D=100% is used to represent bit '0'. An interesting phenomenon that has been noticed is that if D=50% PWM is set even for 1 µs, there is high amplitude emission at the receiver end for several microseconds. To address this spreading, we have used the return-to-zero (RZ) type encoding

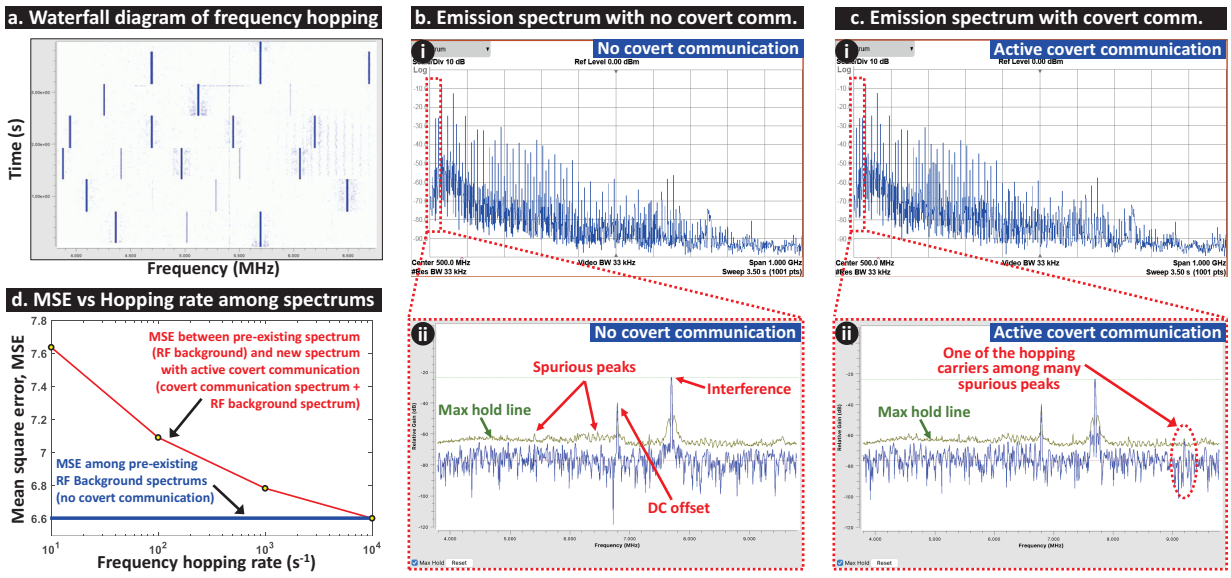

Fig. 5. (a) Received electromagnetic emission with frequency hopping, exaggerated (emission stays at each frequency for $\sim$0.5 s) for better viewing. (b)(i) & (c)(i) shows the emission spectra of 1 GHz span both without and with covert communication. At this wide span, there is a trivial visual difference. (b)(ii) & (c)(ii) shows a zoomed-in version of the spectrum captured with only a 6 MHz span using HackRF. 'Max Hold' is turned on to keep track of hopping peaks. When there is no covert communication, there are just DC offset, some interferences, and spurious/noisy peaks. When covert communication starts, the NoiseHopper peaks sneakily blend in with the other noisy peaks. (d) Mean square error (MSE) between the existing spectrum (RF background) and new spectrum (RF background+covert communication spectrum) decreases as the hopping rate increases and approaches to the reference value (MSE among RF background spectra). The values are average of 35 different spectra measurements to make it statistically significant.

scheme. To transmit bit '1', 50% duty cycle PWM is on for only a small fraction of the bit period. For the rest of the bit period, it is D=100%. For bit '0', it's always 100% duty cycle PWM for the whole bit period.

Once the covert channel is formed, we are ready for data transmission. A demo video of random data transmission has been uploaded for the interested readers (Link: https://youtu.be/__jrUnPGXsg).

V. Covert Communication

A. Data Transmission

Fig. 4(a) shows the transmitter. For sample data, we have chosen images from the MNIST dataset [53]. These are 28×28 pixel, grayscale images of handwritten digits. Each pixel has values between 0 to 255. To simplify data loading, binary thresholding has been applied to each pixel as follows:

- New pixel value = 1, if original pixel value >10
- New pixel value = 0, if original pixel value ≤10

Each row of the images is now represented by 28 bits. In Arduino, an array of 28 unsigned integers has been used to represent each image where each integer value corresponds to an image row converted in decimal numbers. Finally, the data are transmitted at a 10 kbps rate in multiple frames. Each frame contains a preamble and 28 bits of data.

B. Data Processing

Fig. 4(b) shows our receiver: a HackRF One SDR module with a telescopic antenna. A receiver interface has been designed using GNURadio which captures the transmitted data, samples at 8 MSps, and stores them. The stored data are then processed in MATLAB. Raw data are filtered, demodulated and the bit pattern recovered. Fig. 4(b) shows the processing of 9^{th} row of the handwritten digit '7'. It has the following bit pattern '0000001-1111111-1000000'. Since we have used a return-to-zero (RZ) coding scheme while transmitting, there are 'off' periods between each bit '1'. Finally, received bits are arranged as a 28×28 matrix to reconstruct the transmitted image. Fig. 4(c) shows the final image for the best and worst cases. The comparison between transmitted and received images shows that they are almost identical. Since our BER at 10 kbps is in the order of 10^{-4}, a few erroneous bits can be seen in the worst-case scenario. Detailed analysis of BER is provided in Section VII-A.

VI. Increasing Covertness - Frequency Hopping

Most of the current methods proposed in the literature claim their covertness in terms of the designed malware/malicious code that uses very low resources and often does not require the execution of any privileged instructions. However, they are clearly visible in the wireless spectrum and can be detected by the 'spectrum monitoring tools'.

To provide spectrum covertness, we propose frequency hopping EM emission. PWM frequency (f_{PWM}) controls emission frequency ($f_{emission}$). By switching f_{PWM}, OOK modulated EM emission can be made to hop around (we used 16 hopping frequencies). Fig. 5(a) shows an exaggerated version of our frequency hopping emission where the transmitter transmits at each frequency for $\sim$0.5 seconds so

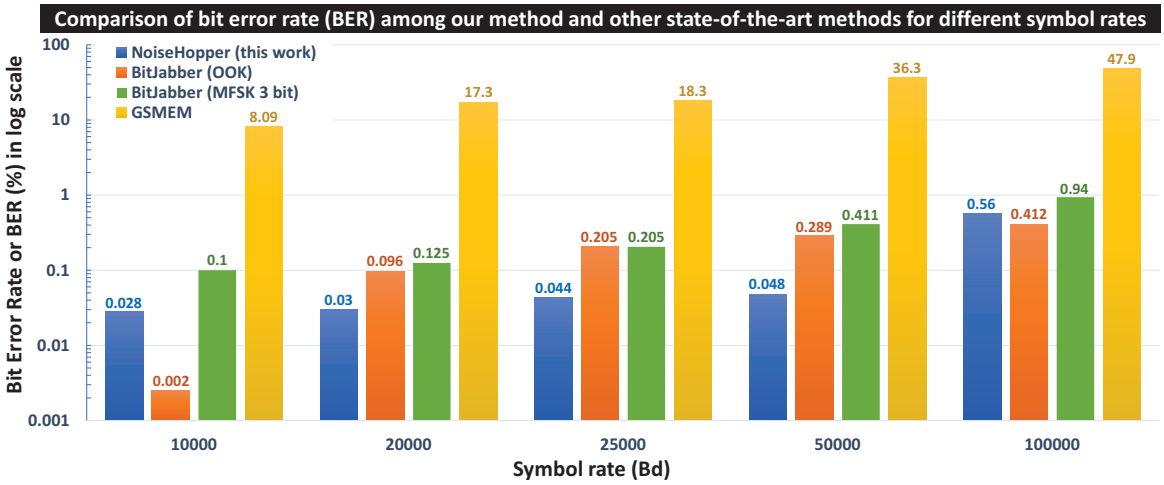

Fig. 6. Bit error rate (BER) of NoiseHopper at 1 m distance for different symbol rates are plotted in log scale and compared with GSMEM and BitJabber. The comparison shows that NoiseHopper performs similarly (and sometimes better) to BitJabber, both of which outperform GSMEM.

that they are visible to us. Fig. 5(b)(i) and (c)(i) show the RF spectra captured by a spectrum analyzer (1 GHz span) when the covert channel is inactive and active respectively. Visually there isn't any significant difference. However, can a smaller span reveal the emission peaks? To answer that, we inspect a much narrower span, 6 MHz, as shown in Fig. 5(b)(ii) and (c)(ii) using HackRF at 1.5 m TX-RX separation. 'Max Hold' was turned on to keep track of the peaks as they were hopping. Fig. 5(b)(ii) shows that, even without the covert channel, the spectrum has spurious peaks and interferences from different sources. Fig. 5(c)(ii) shows that when the emission hopping covert communication is turned on, the hopping peak appears as just another spurious peak.

Along with visual inspection, rigorous mathematical analysis is also needed. To address that, the mean square error (MSE) between the existing spectrum (RF background) and the new spectrum (RF background + NoiseHopper channel) has been calculated at different hopping rates. Fig. 5(d) shows the MSE vs hopping rate plot. For reference, MSE among 35 different RF background spectra has also been calculated and their average is plotted as a constant line. MSE between RF background and new spectrum (with covert channel) decreases with increasing hopping rate, which becomes almost equal to the reference at $10^4 s^{-1}$ hopping rate. This means at a high hopping rate, the spectrum with a covert channel becomes almost identical to the spectrum without it. Frequency hopping and low SNR (controlled by the PWM duty cycle) help NoiseHopper hide better. That said, while NoiseWhopper has a lower detection probability, it is not '0' as there are frequency-hopping signal detectors. However, by varying the hopping frequency set (by f_{PWM}) with low SNR (inherently weak, further controlled by D_{PWM}), the carrier peaks can be moved around the spectrum, making them even more difficult to detect by spectrum monitoring tools which scan the whole spectrum in multiple chunks (usually 100 MHz chunks to scan 6 GHz spectrum). In brief, NoiseHopper takes the state-of-the-art one step ahead by adding a new layer of covertness in the spectrum which is absent in the currently existing methods.

VII. Communication Properties

A. Bit Error Rate (BER)

To calculate the bit error rate, a stream of 50000 bits has been transmitted to the receiver, and the number of erroneous bits is calculated. Fig. 6 shows the BER of our OOK modulated data for different symbol rates and compares it with GSMEM and BitJabber (plotted in log scale to accommodate large variation). Since NoiseHopper (our proposed method) uses OOK modulation, a fair comparison with it will be OOK-modulated BitJabber data. From the plot, both NoiseHopper and BitJabber (blue and orange) have BER in similar order for most cases and outperform GSMEM which has high BER even for low data rates. Though the BER of NoiseHopper and BitJabber is similar, NoiseHopper is still better since it is spectrum stealthy and has almost $3\times$ data transmission range compared to BitJabber.

B. Communication through Obstacle

One limitation of earlier covert channels was that they could not penetrate through an obstacle (a wall). This, along with a short transmission range, posed a challenge for the attackers. They would have to place an intermediate receiver somehow inside the secured room with air-gapped devices. This is not only difficult but also easy to be caught as these rooms are heavily monitored. BitJabber was the first work that showed that their channel works through a wall. A similar test has been performed for our channel as well. Fig. 3(c) has already shown the experimental setup for this. Data have been transmitted successfully keeping the Arduino (TX) outside and the RX inside the lab room, with a $15\,\mathrm{cm}$ wall in between. The wall adds path loss to the channel, but the received signal is still good enough for data transmission.

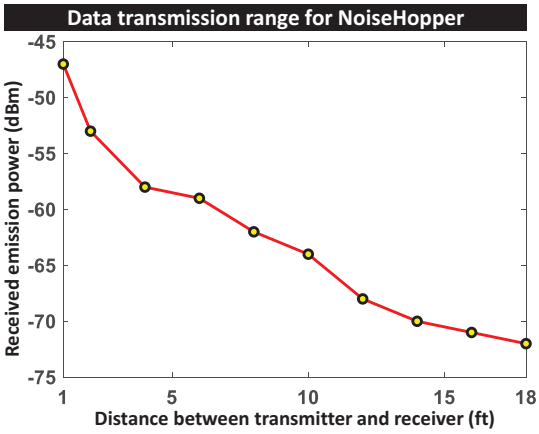

Fig. 7. Emission power plotted against the separation between the transmitter and the receiver. PWM-controlled emission can be received at 18 ft ($\sim$5.5 m), which can be extended by a receiver with higher sensitivity.

C. Data Transmission Range

As mentioned in the attack model (Section III), EM emission-based air-gap covert channels are inherently weak. Hence, it is important to characterize the data transmission range for this type of channel. Fig. 7 shows the emission power plotted against the distance between the transmitter and the receiver in feet. Transmitted signals can be received up to 18 ft ($\sim$5.5 m) range before noise becomes significant. A point to note here is that HackRF has relatively lower sensitivity compared to other better-quality receivers (such as Ettus N210). The transmission range should be higher with those components.

VIII. Comparison with Previous Works

In this section, we will compare our work, NoiseHopper, with the existing electromagnetic emission-based air-gap covert channels: USBEE [9], GSMEM [10], AIR-FI [11], and BitJabber [12]. Fig. 8 summarizes the comparison.

- *Emission generation method:* USBEE manipulates data on USB data buses (D^+ and D^-) to generate intentional EM emission. GSMEM, AIR-FI, and BitJabber exploit the memory bus. Emissions are at the DRAM clock frequency (800 MHz or 2.4 GHz). Coincidentally, 800 MHz falls in GSM band and 2.4 GHz falls in the WiFi band. Our proposed NoiseHopper method generates EM emission by generating PWM signal.
- *Spectrum covertness:* All these covert channels are stealthy in terms of the malware in the code execution domain which uses low resources and doesn't require privileged instructions in most cases. However, the signal is visible in the spectrum with no obfuscation. GSMEM and BitJabber fall in the GSM (or UMTS/LTE) and WiFi bands. However, they are not completely 'mimicked signals' as only the frequency matches, while bandwidth and pulse shapes do not. So, their presence can be detected by spectrum monitoring tools by checking the preamble and signal pattern. On the other hand, NoiseHopper makes hopping emissions with low SNR, as if noise or spurious peaks have been added to the spectrum. Hence, they are very stealthy in the spectrum.
- *Shared resources:* USBEE uses USB data buses on a USB port. Since many facilities prevent USB drive attachments, such resources are often unavailable. GSMEM, AIR-FI, and BitJabber use DRAM bus. However, it requires specific memory instructions to access specific memory blocks to get a stable emission with stable frequency [12]. Another program may use the same instruction for another memory block (more likely) or access the same memory portion. This hurts the channel's stability and interrupts data transmission. Contrary to these approaches, NoiseHopper doesn't depend on any specific shared hardware block.
- *Bit rate:* The maximum data rates for USBEE, GSMEM, and AIR-FI are low (0.64, 1, and 0.016 kbps respectively). Using these, exfiltrating a significant amount of data will require a very long time (e.g., $\sim$2.9 hours for a 10 MB file!!). BitJabber was proposed to address that issue with a maximum bit rate of 300 kbps (though it has a very short transmission range). Our proposed NoiseHopper can reach up to 100 kbps which is fast enough for a reasonable time (e.g., $\sim$1.7 min for a 10 MB file) and can be improved with a more sensitive receiver.
- *Data transmission range:* USBEE doesn't specify its exact range other than saying "USBee can be used for transmitting binary data to a nearby receiver." From this statement and their experimental setup, the range appears to be $<$1 m. BitJabber can be detected up to 2 m (0.5 m with a wall between TX and RX). These are not long enough to transmit data from an air-gapped device in a monitored room to an outside receiver. On the other hand, GSMEM, AIR-FI, and NoiseHopper can go up to 5.5 m, 8 m, and 5.5 m respectively. These ranges are just good enough to cross the air gap.
- *Embedded device suitability:* Since embedded devices barely have USB 2.0 ports, USBEE is not suited for those devices. Also, the USB port is disabled for most air-gapped devices anyway. GSMEM, AIR-FI, and BitJabber generate high-frequency (800 MHz and 2.4 GHz) emissions which require sustained high power consumption. This is a challenge for low-power embedded devices. Also, embedded devices have microcontrollers or single SoC and no separate memory bus or DRAM clock to exploit. Contrary to these methods, NoiseHopper generates low-frequency emissions and has no DRAM clock dependency. We have implemented NoiseHopper in the ATmega328p microcontroller (part of the AVR family, widely used in embedded systems) to show its efficacy for low-power embedded devices.

IX. Countermeasures

The root of the EM emission generation is stealthy malware. Since these malicious codes don't require privileged permis-

Name	Emission generation method	Spectrum covertness	Shared resources	Bit rate (kbps)	Range (m)	Embedded device suitability
USBEE	Manipulating data on USB data buses	Absent	USB data bus	0.64	<1	No
GSMEM	Exploiting memory instructions and using multi-channel memory architecture to amplify	Absent	RAM bus	1	5.5	Mostly no
AIR-FI	Transferring data on DRAM bus to generate emission at DRAM clock frequency (2.4 GHz)	Absent	DRAM bus	0.016	8	Mostly no
BitJabber	Modulating EM emission due to DRAM clock	Absent	DRAM	300	2	Mostly no
NoiseHopper	Generating a PWM signal whose switching causes EM emissions at PWM frequency	Present	None	100	5.5	Yes

Very bad Bad Good Very good

Fig. 8. Comparison of existing EM emission-based air-gap covert channels with our proposed method in terms of various channel parameters and implementation metrics. The table is color-coded for easier comparison.

sion and don't behave suspiciously, they are undetectable in traditional software which only considers logical impacts, not 'electromagnetic impacts'. However, the antivirus/anti-malware tools can incorporate a new feature to consider the possible 'electromagnetic impact' of an instruction execution. A probable emission pattern can be derived for programs that use certain instructions repeatedly. If the emission pattern seems suspicious, the program should be flagged. This method has the potential to identify not only malware for covert channels but also other programs that unintentionally leak significant data (emanation).

A commonly proposed mitigation for EM emissions is to use jammer signals. However, due to frequency hopping, this method is ineffective against NoiseHopper. A classic approach to block EM emissions is metal shielding, just like the famous 'Faraday Cage'. However, Power supply and other port access require some gap in the shielding which reduces its efficacy significantly. Also, this type of shielding can weaken the EM emission, but can not eliminate it totally [54], [55]. Also, manufacturing custom shielding for different devices of different sizes and shapes is difficult and costly.

NATO follows a zoning system of safety perimeters classified as zone 0-4. They require specific screening and background checks before granting access to the area of radiated systems. However, insider threats have breached that several times as mentioned earlier. To intercept intermediate receivers, a zero-trust approach can be applied for each personnel where everyone, regardless of clearance level, is properly authenticated and scanned before allowing entry to these zones. Also, the sensitive zones need to be searched regularly to find any potential bugs implanted by compromised insiders.

X. Conclusion

In this work, we propose 'NoiseHopper', an improved air-gap covert side channel with much better covertness provided by frequency hopping. The electromagnetic emission is generated through a PWM signal whose frequency and duty cycle control the emission frequency and amplitude. Unlike previous methods, this generation process doesn't require shared hardware resources like CPU, specific memory buses, or DRAM clock and hence, cannot be interrupted by other parallel programs. The frequency hopping emission with low SNR (inherently weak, further controlled by the PWM duty cycle) appears in the existing RF spectrum as if some spurious peaks or noise have been added to it. Hence, it has a much lower probability of detection by the spectrum monitoring tools and is immune to jamming. Our proposed method has been implemented on the ATmega328p microcontroller (onboard an Arduino Uno) to show its efficacy for low-power, embedded devices. The correlation between PWM frequency and duty cycle with emission frequency and amplitude (power) has been analyzed to design the covert channel. Using the covert channel, images from the MNIST dataset have been successfully transmitted at 10 kbps, which can reach up to a maximum of 100 kbps. It has been shown that the generated emission can penetrate through a 15 cm thick wall and has a data transmission range of ∼5.5 m. Properties of the covert communication channel (e.g., BER, data transmission range, penetrability through a wall, etc.) have been analyzed. The performance of our proposed method has been compared against existing methods in terms of various communication parameters and implementation considerations. Finally, a few possible countermeasures have been proposed at software, hardware, and network levels. This work takes the state-of-the-art EM emission-based covert channels to a new height by adding an extra layer of covertness in the spectrum and also expands their attack surface by exposing air-gapped embedded devices as new victims.

Acknowledgment

This research was supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract: 2021-21062400006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation thereon.

REFERENCES

[1] S. Shaneman and C. Murphy, "Enhancing the Deployment and Security of SIPRNET and JWICS Networks using Intrinsic Fiber Monitoring," MILCOM 2007 - IEEE Military Communications Conference, Orlando, FL, USA, 2007, pp. 1-6, doi: 10.1109/MILCOM.2007.4455073.

[2] "Intelligence Communications System gets Tech Refresh," U.S. Department of Defense, https://www.defense.gov/News/News-Stories/Article/Article/1954347/intelligence-communications-system-gets-tech-refresh/ (accessed Dec. 7, 2023).

[3] P/K, "US military and intelligence computer networks." https://www.electrospaces.net/2015/03/us-military-and-intelligence-computer.html (accessed Dec. 7, 2023).

[4] D. Kushner, "The real story of stuxnet," IEEE Spectrum, 26-Feb-2013. [Online]. Available: https://spectrum.ieee.org/the-real-story-of-stuxnet. [Accessed: 08-Dec-2023].

[5] "Agent.btz," Mitre.org. [Online]. Available: https://attack.mitre.org/software/S0092/. [Accessed: 08-Dec-2023].

[6] K. Hayashi and M. Harbison, "SymonLoader archives," Palo Alto Networks Blog. [Online]. Available: https://www.paloaltonetworks.com/blog/tag/symonloader/. [Accessed: 08-Dec-2023].

[7] M. Guri, "Air-Gap Electromagnetic Covert Channel," in IEEE Transactions on Dependable and Secure Computing, 2023, doi: 10.1109/TDSC.2023.3300035.

[8] M. Guri, "Near Field Air-Gap Covert Channel Attack," 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Wuhan, China, 2022, pp. 490-497, doi: 10.1109/TrustCom56396.2022.00074.

[9] M. Guri, M. Monitz and Y. Elovici, "USBee: Air-gap covert-channel via electromagnetic emission from USB," 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 2016, pp. 264-268, doi: 10.1109/PST.2016.7906972.

[10] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, & Y. Elovici, "GSMem: Data exfiltration from Air-Gapped computers over GSM frequencies," In 24th USENIX Security Symposium (USENIX Security 15) (pp. 849-864).

[11] M. Guri, "AIR-FI: Leaking Data From Air-Gapped Computers Using Wi-Fi Frequencies," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 2547-2564, 1 May-June 2023, doi: 10.1109/TDSC.2022.3186627.

[12] Z. Zhan, Z. Zhang and X. Koutsoukos, "BitJabber: The World's Fastest Electromagnetic Covert Channel," 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), San Jose, CA, USA, 2020, pp. 35-45, doi: 10.1109/HOST45689.2020.9300268.

[13] C. Zhuo, S. Luo, H. Gan, J. Hu and Z. Shi, "Noise-Aware DVFS for Efficient Transitions on Battery-Powered IoT Devices," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 7, pp. 1498-1510, July 2020, doi: 10.1109/TCAD.2019.2917844.

[14] M. Guri, B. Zadov, D. Bykhovsky and Y. Elovici, "PowerHammer: Exfiltrating Data From Air-Gapped Computers Through Power Lines," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1879-1890, 2020, doi: 10.1109/TIFS.2019.2952257.

[15] "SCISRS - Securing Compartmented Information with Smart Radio Systems," IARPA. [Online]. Available: https://www.iarpa.gov/index.php/research-programs/scisrs. [Accessed: 07-Dec-2023].

[16] M. F. Bari, M. R. Chowdhury, B. Chatterjee and S. Sen, "Detection of Rogue Devices using Unintended Near and Far-field Emanations with Spectral and Temporal Signatures," 2022 IEEE/MTT-S International Microwave Symposium - IMS 2022, Denver, CO, USA, 2022, pp. 591-594, doi: 10.1109/IMS37962.2022.9865347.

[17] M. G. Kuhn, "Compromising emanations: eavesdropping risks of computer displays," (Doctoral dissertation, University of Cambridge), 2002.

[18] M. Martin, F. Sunmola, and D. Lauder, "Unintentional Compromising Electromagnetic Emanations from IT Equipment: A Concept Map of Domain Knowledge," Procedia Computer Science, vol. 200, pp. 1432–1441, Jan. 2022, doi: https://doi.org/10.1016/j.procs.2022.01.344.

[19] K. R. Teo, B. T. Balamurali, C. J. Ming and J. Zhou, "Retrieving Input from Touch Interfaces via Acoustic Emanations," 2021 IEEE Conference on Dependable and Secure Computing (DSC), Aizuwakamatsu, Fukushima, Japan, 2021, pp. 1-8, doi: 10.1109/DSC49826.2021.9346271.

[20] J. Loughry, "Optical TEMPEST," 2018 International Symposium on Electromagnetic Compatibility (EMC EUROPE), Amsterdam, Netherlands, 2018, pp. 172-177, doi: 10.1109/EMCEurope.2018.8485128.

[21] P. Naveen, D. Saranesh, V. K. Vishnukanth, B. Sabarish, D. Vishnu and S. R. Ashokkumar, "Detection and Recognition of Species using Deep Convolution Neural Network," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1830-1835, doi: 10.1109/ICACCS54159.2022.9785170.

[22] T. Kaczmarek, E. Ozturk and G. Tsudik, "Thermanator: thermal residue-based post factum attacks on keyboard data entry," in Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (pp. 586-593).

[23] B. Liu, Y. Xu, W. Huang and S. Guo, "Detecting USB Storage Device Behaviors by Exploiting Electromagnetic Emanations," ICC 2022 - IEEE International Conference on Communications, 2022, pp. 4980-4985, doi: 10.1109/ICC45855.2022.9839155.

[24] S. Liang, Z. Zhan, F. Yao, L. Cheng and Z. Zhang, "Clairvoyance: Exploiting Far-field EM Emanations of GPU to "See" Your DNN Models through Obstacles at a Distance," 2022 IEEE Security and Privacy Workshops (SPW), 2022, pp. 312-322, doi: 10.1109/SPW54247.2022.9833894.

[25] B. B. Yilmaz, E. Mert Ugurlu, A. Zajić and M. Prvulovic, "Cell-Phone Classification: A Convolutional Neural Network Approach Exploiting Electromagnetic Emanations," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2862-2866, doi: 10.1109/ICASSP40776.2020.9054006.

[26] M. F. Bari, M. R. Chowdhury and S. Sen, "Long Range Detection of Emanation from HDMI Cables Using CNN and Transfer Learning," 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2023, pp. 1-6, doi: 10.23919/DATE56975.2023.10137263.

[27] M. G. Kuhn, "Electromagnetic eavesdropping risks of flat-panel displays," in International Workshop on Privacy Enhancing Technologies,2004, pp. 88-107. doi: https://doi.org/10.1007/11545262_20

[28] H. S. Lee, D. H. Choi, K. Sim and J. -G. Yook, "Information Recovery Using Electromagnetic Emanations From Display Devices Under Realistic Environment," in IEEE Transactions on Electromagnetic Compatibility, vol. 61, no. 4, pp. 1098-1106, 2019, doi: 10.1109/TEMC.2018.2855448.

[29] D. -H. Choi, E. Lee and J. -G. Yook, "Reconstruction of Video Information Through Leakaged Electromagnetic Waves From Two VDUs Using a Narrow Band-Pass Filter," in IEEE Access, vol. 10, pp. 40307-40315, 2022, doi: 10.1109/ACCESS.2022.3162686.

[30] D. -H. Choi, E. Lee, T. Nam and J. -G. Yook, "Recent Trends in Image Information Recovery Using Leaked Electromagnetic Wave from Electronic Equipment," in IEEE Electromagnetic Compatibility Magazine, vol. 11, no. 3, pp. 77-83, 3rd Quarter 2022, doi: 10.1109/MEMC.2022.9982567.

[31] E. Thiele, "Tempest for Eliza," 2001. [Online]. Available: http://www.erikyyy.de/tempest/. [Accessed Dec. 7, 2023].

[32] Z. Shao, M. A. Islam, and S. Ren, "Your Noise, My Signal," Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 4, no. 1, pp. 1–39, May 2020, doi: https://doi.org/10.1145/3379473.

[33] M. Guri, B. Zadov and Y. Elovici, "ODINI: Escaping Sensitive Data From Faraday-Caged, Air-Gapped Computers via Magnetic Fields," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1190-1203, 2020, doi: 10.1109/TIFS.2019.2938404.

[34] M. Guri, "MAGNETO: Covert channel between air-gapped systems and nearby smartphones via CPU-generated magnetic fields," Future Generation Computer Systems, vol. 115, pp. 115–125, Feb. 2021, doi: https://doi.org/10.1016/j.future.2020.08.045.

[35] N. Matyunin, J. Szefer, S. Biedermann and S. Katzenbeisser, "Covert channels using mobile device's magnetic field sensors," 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, 2016, pp. 525-532, doi: 10.1109/ASPDAC.2016.7428065.

[36] M. Guri, B. Zadov, D. Bykhovsky and Y. Elovici, "CTRL-ALT-LED: Leaking Data from Air-Gapped Computers Via Keyboard LEDs," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 2019, pp. 801-810, doi: 10.1109/COMPSAC.2019.00118.

[37] M. Guri, B. Zadov, and Y. Elovici, "LED-it-GO: Leaking (A Lot of) Data from Air-Gapped Computers via the (Small) Hard Drive LED,"

Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 161–184, 2017, doi: https://doi.org/10.1007/978-3-319-60876-1_8.
[38] B. Nassi, A. Shamir and Y. Elovici, "Xerox Day Vulnerability," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 2, pp. 415-430, Feb. 2019, doi: 10.1109/TIFS.2018.2854708.
[39] Z. Zhou, W. Zhang, Z. Yang, and N. Yu, "Optical Exfiltration of Data via Keyboard LED Status Indicators to IP Cameras," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1541–1550, Apr. 2019, doi: https://doi.org/10.1109/jiot.2018.2842116.
[40] M. Guri, D. Bykhovsky and Y. Elovici, "Brightness: Leaking Sensitive Data from Air-Gapped Workstations via Screen Brightness," 2019 12th CMI Conference on Cybersecurity and Privacy (CMI), Copenhagen, Denmark, 2019, pp. 1-6, doi: 10.1109/CMI48017.2019.8962137.
[41] M. Guri, O. Hasson, G. Kedma and Y. Elovici, "An optical covert-channel to leak data through an air-gap," 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 2016, pp. 642-649, doi: 10.1109/PST.2016.7906933.
[42] M. Hanspach and M. Goetz, "On Covert Acoustical Mesh Networks in Air," arXiv.org, Jun. 04, 2014. https://arxiv.org/abs/1406.1213
[43] B. Carrara and C. Adams, "On Acoustic Covert Channels Between Air-Gapped Systems," Lecture Notes in Computer Science, pp. 3–16, Jan. 2015, doi: https://doi.org/10.1007/978-3-319-17040-4_1.
[44] M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici, "Acoustic Data Exfiltration from Speakerless Air-Gapped Computers via Covert Hard-Drive Noise ('DiskFiltration')," Computer Security – ESORICS 2017, pp. 98–115, 2017, doi: https://doi.org/10.1007/978-3-319-66399-9_6.
[45] M. Guri, Y. Solewicz and Y. Elovici, "MOSQUITO: Covert Ultrasonic Transmissions Between Two Air-Gapped Computers Using Speaker-to-Speaker Communication," 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 2018, pp. 1-8, doi: 10.1109/DESEC.2018.8625124.
[46] M. Guri, "CD-LEAK: Leaking Secrets from Audioless Air-Gapped Computers Using Covert Acoustic Signals from CD/DVD Drives," 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 2020, pp. 808-816, doi: 10.1109/COMPSAC48688.2020.0-163.
[47] M. Guri, M. Monitz, Y. Mirski and Y. Elovici, "BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal Manipulations," 2015 IEEE 28th Computer Security Foundations Symposium, Verona, Italy, 2015, pp. 276-289, doi: 10.1109/CSF.2015.26.
[48] M. Guri, "HOTSPOT: Crossing the Air-Gap Between Isolated PCs and Nearby Smartphones Using Temperature," 2019 European Intelligence and Security Informatics Conference (EISIC), Oulu, Finland, 2019, pp. 94-100, doi: 10.1109/EISIC49498.2019.9108874.
[49] N. Sehatbakhsh, B. B. Yilmaz, A. Zajic and M. Prvulovic, "A New Side-Channel Vulnerability on Modern Computers by Exploiting Electromagnetic Emanations from the Power Management Unit," 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), San Diego, CA, USA, 2020, pp. 123-138, doi: 10.1109/HPCA47549.2020.00020.
[50] G. Greenwald, E. MacAskill, and L. Poitras, "Edward Snowden: the whistleblower behind the NSA surveillance revelations," The Guardian, Jun. 11, 2013. https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance (accessed Dec. 7, 2023).
[51] M. Hosenball and W. Strobel, "Exclusive: Snowden persuaded other NSA workers to give up passwords - sources," Reuters, Nov. 08, 2013. Available: https://www.reuters.com/article/net-us-usa-security-snowden/exclusive-snowden-persuaded-other-nsa-workers-to-give-up-passwords-sources-idUSBRE9A703020131108/ (accessed Dec. 7, 2023).
[52] M. R&D, "FastPwmPin," GitHub, Sep. 03, 2023. https://github.com/maxint-rd/FastPwmPin (accessed Dec. 7, 2023).
[53] "MNIST in CSV," www.kaggle.com. https://www.kaggle.com/datasets/oddrationale/mnist-in-csv/data (accessed Dec. 07, 2023).
[54] A. Zajić and M. Prvulovic, "Experimental Demonstration of Electromagnetic Information Leakage From Modern Processor-Memory Systems," in IEEE Transactions on Electromagnetic Compatibility, vol. 56, no. 4, pp. 885-893, Aug. 2014, doi: 10.1109/TEMC.2014.2300139.
[55] M. F. Bari, M. R. Chowdhury and S. Sen, "Is Broken Cable Breaking Your Security?," 2023 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, USA, 2023, pp. 1-5, doi: 10.1109/ISCAS46773.2023.10181751.