`FreeEM`: Uncovering Parallel Memory EMR Covert Communication in Volatile Environments

Sihan Yu
Clemson University
sihany@g.clemson.edu

Jingjing Fu
Clemson University
jfu@g.clemson.edu

Chenxu Jiang
Clemson University
chenxuj@g.clemson.edu

ChunChih Lin
Clemson University
chunchi@g.clemson.edu

Zhenkai Zhang
Clemson University
zhenkai@clemson.edu

Long Cheng
Clemson University
lcheng2@clemson.edu

Ming Li
The University of Texas at Arlington
ming.li@uta.edu

Xiaonan Zhang
Florida State University
xzhang@cs.fsu.edu

Linke Guo*
Clemson University
linkeg@clemson.edu

ABSTRACT

Memory Electromagnetic Radiation (EMR) allows attackers to manipulate the DRAM of infiltrated systems to leak sensitive secret information. Although most of the existing works have demonstrated its feasibility, practical concerns, such as the ideal electromagnetic environment and stationary attacking layout, make the covert channel attack less convincing, especially in vulnerable sites such as offices and data centers. This work removes the above impractical assumptions to uncover the potential of memory EMR by proposing the first parallel EMR covert communication protocol. Our design reshapes the current “1-to-1” covert communication mode to “n-to-1” mode via a novel pattern-based 2-dimensional symbol encoding scheme, allowing multiple victim computers to simultaneously perform data exfiltration to one attacker (the receiver) without mutual interference. Meanwhile, this novel scheme design also enables the very first mobile attacker, i.e., a smartphone connected to a software-defined radio (SDR) dongle, to capture parallel memory EMR signals in a volatile environment. Extensive experiments are conducted to verify the performance in a volatile environment with different parameter configurations, distances, motion modes, shielding materials, orientations, hardware configurations, and SDR platforms. Our experimental results demonstrate that `FreeEM` can support up to 4 parallel memory EMR transmissions to achieve an overall throughput of 625Kbps and a decoding accuracy of 96.88%. The maximum communication distance can reach up to 20 meters.

CCS CONCEPTS

• **Security and privacy → Side-channel analysis and countermeasures**; **Mobile and wireless security**.

KEYWORDS

Memory EMR, Covert Communication, Parallelism, DRAM

ACM Reference Format:
Sihan Yu, Jingjing Fu, Chenxu Jiang, ChunChih Lin, Zhenkai Zhang, Long Cheng, Ming Li, Xiaonan Zhang, and Linke Guo. 2024. **`FreeEM`: Uncovering Parallel Memory EMR Covert Communication in Volatile Environments. In *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS ’24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3643832.3661870**

1 INTRODUCTION

Data exfiltration via physical covert channels [18, 19, 21, 26] for compromising the computer system has drawn significant attention recently. To launch the attack, a Trojan horse or similar malware is covertly inserted into the victim’s computer, exploiting the computer’s hardware to generate physical signals. Those signals, usually containing sensitive encoded information, can be transferred by the physical side effect from the victim computer to the receiver (attacker). Various physical side effects can be exploited as covert channels, such as acoustic [4, 12], optical [13, 24], electromagnetic [8, 28], magnetic [7, 14], and thermal [11]. Among them, electromagnetic radiation (EMR) has been widely discussed in the literature [2, 5, 6, 8, 10, 25, 28–30] for its advanced capability of data exfiltration. Different from conventional channels, the EMR covert channel does not need to compromise the victim’s dedicated protocol suite, such that it can bypass a majority of defensive mechanisms, including both cyber-based and physical-based approaches. Although existing research works on the memory EMR covert channel claim that they can improve the data rate to 300Kbps [28] or extend the communication range to 100m [25], most of their designs rely on the following impractical assumptions, making their schemes almost infeasible in real attacking scenarios.

• **Ideal Electromagnetic Environment**: All existing works require the victim computer is the only device within the attacking range. However, many nearby computers will generate clutter signals to significantly impact communication performance in a practical scenario.

• **Stationary Layout**: The change in the layout of both communication parties, including position, orientation, obstacle, motion, and

MOBISYS ’24, June 3–7, 2024, Minato-ku, Tokyo, Japan
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0581-6/24/06…$15.00
https://doi.org/10.1145/3643832.3661870

mobility of the receiver may easily compromise the EMR covert communication performance.

- **Memory EMR Signal Stability**: EMR signals from memory are inherently unstable, depending on the CPU schedule and memory access. When multiple memory EMR signals are transmitted, inevitable and unpredictable collisions will significantly reduce the decoding accuracy at the receiver.

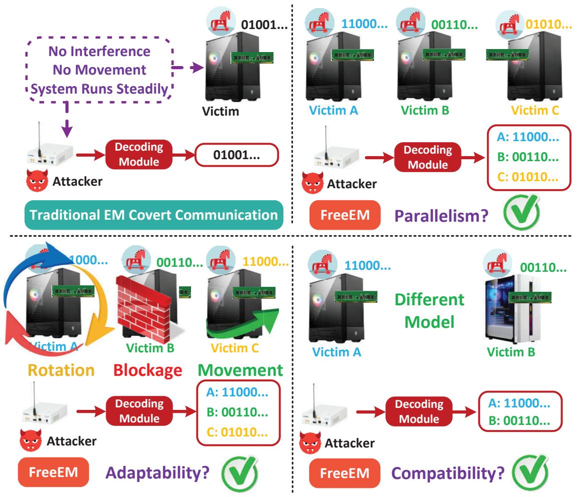


Figure 1: FreeEM System Overview

To uncover the potential of memory EMR in practical volatile environments, we propose FreeEM, the first parallel covert communication without following the above impractical assumptions. As shown in Fig. 1, we expect the proposed paradigm to achieve **parallelism**, **adaptability**, and **compatibility**. Specifically, our idea of enabling parallelism is to leverage the frequency-variable characteristic of memory EMR signals, which can form unique spectral patterns during a period of time. The encoded secret information will be designed as a 2-dimensional (2D) symbol spreading on both frequency and time domains, which not only minimizes collisions on the same frequency band but also enables parallel transmissions. When multiple 2D symbols from various sources overlap during parallel transmission, they are easily recognized. Moreover, the aggregated 2D symbol contains a higher energy to be detected in a volatile environment with position/orientation changes, blockage, or even mobility. Besides the adaptability, our design can also operate on different DRAMs, further extending the compatibility of the memory EMR covert channel. Finally, to overcome the instability of memory EMR signals, we develop a deep neural network (DNN) based decoding scheme to learn the characteristics of received signals from different victims.

As shown in Fig.2, by innovating covert communication from “1-to-1” to “*n*-to-1” paradigm, a mobile receiver can steal data from multiple victim computers concurrently, which is more suitable for real-world scenarios, such as stealing information from offices or data centers with multiple computers. By the novel pattern-based 2D symbol design, EMR from multiple victim computers is no longer considered as mutual interference; instead, they can ”collaboratively" work to significantly improve the data rate. Our proposed FreeEM achieves “4-to-1” parallel covert communication with a data rate up to 625Kbps and a decoding accuracy of 96.88%.

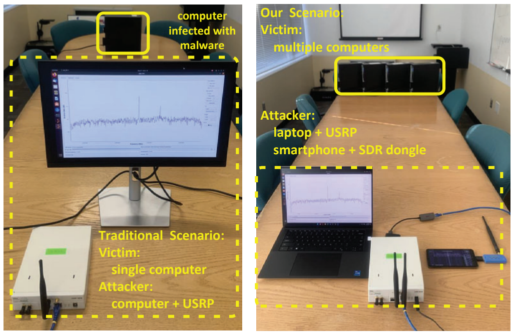


Figure 2: Attacking Scenario Comparison

2 RELATED WORK

- **Physical Side Channel.** Covert channels transfer information using non-standard methods against the system design. This term originated in 1973 by Butler Lampson [16]. Many physical side effects of computers can be exploited to construct physical covert channels, including signals such as acoustic [4, 12], optical [13, 24], electromagnetic [8, 28], magnetic [7, 14], and thermal [11]. We compare some major representatives of the EM covert channel designs in Tab. 1.

Table 1: Comparison of EMR-based Covert Channels

Ref.	Transmitter	Distance	Throughput
[9]	Display cable	1~7m	104~480bps
[10]	USB connectors	≈1m	160~640bps
[8]	DRAM	1~5.5m	100~1Kbps
[28, 29]	DRAM	<3m	100K~300Kbps
[25]	DRAM	40~137m	1.25bps~14bps
[3]	DRAM	0~5m	11.2bps~2.56Kbps
[23]	Power management unit	2.5m	3Kbps
[2]	CPU/memory	<1m	N/A

- **EM Covert Channel:** AirHopper [9] and Soft tempest [15] demonstrate that by manipulating the video display units, an EM covert channel can be established, whose communication range can be several meters, but it is easily noticed. Similarly, the EMR from USB connectors [10], DRAM bus clock [3, 8, 25, 28, 29], power management unit [23], and CPU [2, 5] can also be modulated to carry information. Among the above works, BitJabber [28], EMLoRa [25] and Noise-SDR [3] are three pioneer works. BitJabber is dedicated to optimizing data rates, with a maximum capacity of 300Kbps. EMLoRa focuses on extending communication range, with a maximum achievable distance of 100 meters. Noise-SDR [3] is committed to the customizability of signals, attempting to achieve the functionalities of SDR using EMR. However, they have not taken the parallelism of signals into account and are unable to distinguish signal sources, making them unsuitable for parallel transmission scenarios.

3 PRELIMINARIES ON MEMORY EMR

Previous studies [8, 25, 27, 28] have demonstrated that memory activities can generate EMR on specific spectrum bands. The memory clock acts as a local oscillator and the memory bus acts as an antenna to radiate the generated EMR.

• **Memory Activity.** Fig. 3 shows an example of the memory spectrum pattern, where the signals are generated by a Z97 MPOWER MAX AC motherboard (with DDR3-1600 memories) and received by a Tektronix Real Time Signal Analyzer (RSA507A) with an LP0965 antenna. Since the memory clock usually has a constant frequency, the energy of EMR concentrates on a very narrow frequency band centered on 800MHz. When there are some memory activities, such as read or write operations, the spectrum pattern will significantly change, i.e., more sub-peaks appear on both sides and the density of sub-peaks can vary with the writing frequencies.

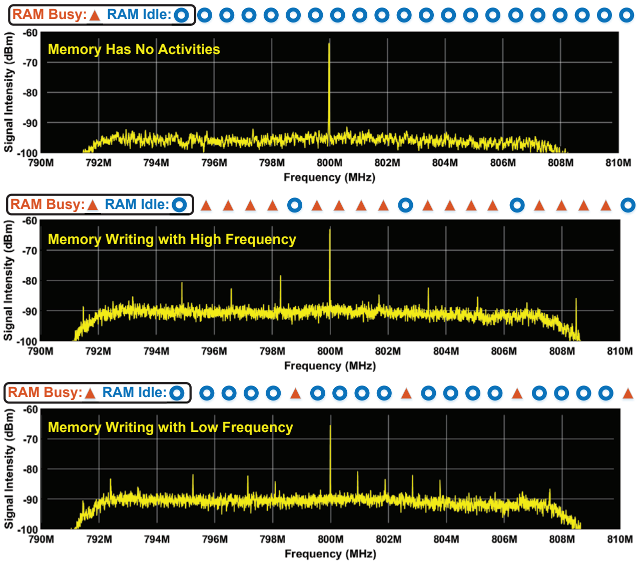


Figure 3: Spectrum of Different Memory Activities

• **Information Encoding.** The presence/absence of sub-peaks can represent bit 1/0. Hence, memory EMR can be used as a carrier to transmit information, enabling the establishment of a covert channel. Besides, a more efficient encoding method is to leverage the position of sub-peaks because high-frequency/low-frequency writing operations can result in the sub-peaks being far from/close to the central frequency. As shown in Fig. 4, we generate 8 kinds of sub-peaks (named as "chip") with different frequencies of writing operations, which can be used to represent bit sequence "000"-"111". Since the frequency of writing operations has hardware limits, the number of definable chips is also limited.

• **Harmonics.** When performing memory operations, there exists harmonics along with the base frequencies as in Fig. 4. For our protocol design, the existence of harmonics is beneficial, because their position features (e.g., equally spaced) can be used to train the deep neural networks for the decoding of combined EMR signals.

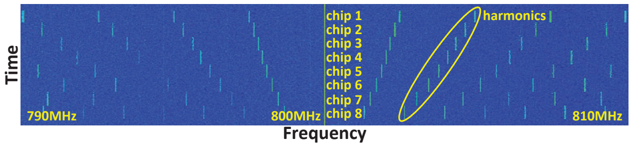


Figure 4: Predefined Chips

4 MOTIVATION

A major objective of this work is to enhance the data rate of memory EMR covert communication by enabling parallel transmissions from multiple victim computers. Since the single-victim data rate is naturally imposed by hardware limits (e.g., CPU frequency, RAM types), a feasible way to improve the data rate is adopting a multi-victim scheme. To this end, we have to exploit available spectrum resources to embed more information in them.

• **Mutual Interference.** Using the widely-used OOK (on-off keying) scheme in memory EMR communication increases the chip error rate from 0.22% to 41.01% when two nearby victim computers are operating simultaneously. Their transmitted EMR signals overlap on the frequency domain as shown in Fig. 5. Memory EMR covert communication schemes usually decode the received signals based on the frequency of high-energy signals, which appear at different positions on the spectrum to represent different information. When two memory EMR signals are received at the same time, e.g., at t_1 in Fig. 5, one is marked as "Data" and the other one marked as "Interference" on the spectrum, the receiver cannot differentiate which one should be used for decoding. Hence, solely relying on frequency of high-energy signals for encoding/decoding will not support parallel EMR covert communication, but only introduce more noise.

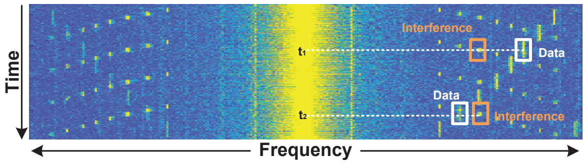


Figure 5: Memory EMR Signals Collisions

• **Intuition of FreeEM Design.** To fully leverage the available spectrum, we design a novel pattern of memory EMR signals. Specifically, the transmitted memory EMR signal is no longer a single high-energy signal on the spectrum, instead, we use a unique pattern that consists of multiple high-energy signals to represent one symbol. Hence, the receiver will detect the pattern for decoding, rather than only using the high-energy signal shown on the spectrum. By carefully crafting the patterns of transmitted symbols, more overlapping high-energy signals can be exploited without causing mutual interference.

5 THREAT MODEL

We focus on achieving parallel covert communications between multiple senders and a receiver to steal data rapidly. This type of attack incident can occur at data centers or any place with multiple computers (e.g., office), as long as the attacker is interested in the data stored on the computers. We assume that the attacker has already obtained some basic information about the victim computers, such as their locations and IP addresses. Therefore, the attacker has clear targets rather than requiring a broad search for victims.

Sender (victim computers):

• The senders have been infected by our malware, whose purpose is to steal data. The malware may be implanted through the Internet or other means, and it does not require root privilege. This is a common assumption in conventional EMR covert channels [19, 25, 28], which can be done by various methods as introduced in [22].

Receiver (the attacker):

- The receiver can be desktop, laptop, and smartphone connected to an SDR device, which is placed in the proximity of senders. The receiver can be placed either within the same room or an adjacent room. The Non-Line-of-Sight (NLoS) can happen between the senders and the receiver.
- The distance between the sender and the receiver is 1-20 meters. Note that FreeEM is not designed for long-range covert communication (e.g., 120m in [25]).
- The receiver can easily determine the frequency of memory EMR because DDR memories typically operate at several fixed and publicly known frequencies (e.g., DDR3-1600/DDR4-3200 operates at 800/1600MHz).

6 FREEEM PROTOCOL DESIGN

First, we elaborate on the pattern-based 2D symbol design. Then, we present the format of the FreeEM packet, followed by the memory EMR signal modulation and generation. Finally, we introduce the optimization process of pattern-based 2D symbol selection to improve the symbol generation efficiency as well as support more parallel transmissions.

6.1 Pattern-based 2D Symbol Design

6.1.1 Overview. Instead of only using the position/existence of high-energy signals on the spectrum for encoding, our idea is to use unique patterns of high-energy signals (similar to the spread spectrum) to encode information.

Different memory activities will generate high-energy EM signals that vary on both the frequency domain and time domain, forming diverse 2-dimensional (2D) patterns on the spectrum. Hence, we exploit all available positions on the spectrum to encode the secret information as unique patterns. Each victim computer will be given a set of available patterns to represent its information. By carefully designing the sets, multiple victim computers can transmit the information in parallel, even experiencing the "overlap" (mutual interference). Since clutter noises cannot have a pre-defined pattern, they can be easily removed upon receiving. Before elaborating on the pattern-based 2D symbol design, we define the following terms:

- **Chip.** A chip is a piece of memory EMR signal having a fixed duration and a specific frequency. We set the chip duration as 12.8μs (=256 samples ÷ 20M samples/s), where 20M is the sampling rate and 256 is the FFT size. Hence, the chip rate is 20M/256=78125 chips/s. Based on the previous discussion, the available lobes near the clock frequency (i.e., 800MHz) is approximately 8. Hence, by changing the memory access frequency, we define 8 chips used for encoding, which can be tuned given different hardware limitations.
- **Symbol.** A symbol consists of multiple consecutive chips, which form specific 2D patterns shown on the spectrum as in Fig. 4. The amount of information that a symbol can represent depends on the size of the symbol set and the number of victim computers. Assuming we design k unique symbols that will be used by d computers, each symbol can represent up to $\log_2 \frac{k}{d}$ bits of information. Therefore, the diversity of chips determines the size of the symbol set (i.e., the number of different symbols that can be defined), which further determines how many bits a symbol can represent (i.e., the data rate).

6.1.2 Parallelism of Pattern-based 2D Symbols. Our design offers opportunities for parallel transmission since the memory EMR signals can be transmitted on different frequencies at the same time slot. We use 2 victim computers as a case study to demonstrate the proposed FreeEM.

- **Encoding.** When symbols are well designed, any two symbols will not overlap with each other on the same frequency. As an example in Fig. 6 (upper part), two senders transmit the same information (symbol 1) with different patterns. From the receiver's perspective, two symbols take up different frequencies at the same time slot without any overlap or interference. Hence, the encoded information from both A and B can be transmitted in parallel. The lower part of Fig. 6 demonstrates another example of parallel communication where the pattern looks more irregular, but it satisfies the requirement that two symbols have no overlap as well. We also demonstrate the actual spectra of combined signals corresponding to the selected pattern-based 2D symbols.

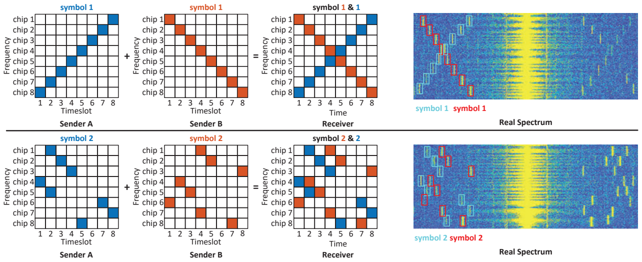


Figure 6: Pattern-based 2D Symbol - Encoding

- **Decoding.**The receiver maintains a table of all valid 2D symbols, it can easily recognize and decode the combined symbols, as shown in the upper part of Fig. 7.

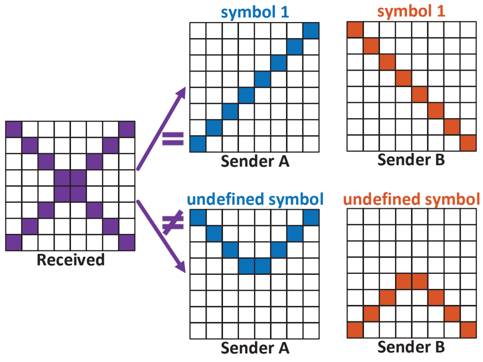


Figure 7: Pattern-based 2D Symbol - Decoding

It is likely that the received symbol can also be decoded in different ways, such as the lower part of Fig. 7. In fact, this case can be easily avoided, because such symbols are undefined in the receiver's table. Besides, our scheme supports the receiver to accumulate more spectral energy from different frequencies for decoding as in LoRa protocol [1] which adopts the chirp spread spectrum (CSS) for modulation. Thus, our proposed scheme will work in an extremely low-SNR environment and provide high decoding accuracy.

6.2 FreeEM Working Process

FreeEM executes the time synchronization before parallel transmission. Then, victims transmit their training sequences to the attacker one by one. Finally, all victims send payload symbols in parallel.

6.2.1 Time Synchronization. Time synchronization is essential for achieving parallel transmission. Without it, the experimental scenario would degrade to a "1-to-1" case. However, time synchronization is not indispensable. Even if multiple computers send symbols independently, the receiver can still decode them correctly, since it employs DNN to recognize symbols one by one. The only disadvantage of non-synchronized symbols is a potentially higher Symbol Error Rate (SER). Since our scenario is a data center, it is highly possible that all computers are connected to a local area network (LAN) or the Internet. Time synchronization can be achieved by a poll-based network time protocol (NTP) [17]. In our experiment, one victim computer acts as the server and waits for connections from clients (i.e., other victims). When all victims are connected, the server will inform clients to begin their transmissions.

6.2.2 Training Sequences. Similar to the WiFi (802.11g) training sequence, which is a predefined sequence transmitted before the payload for channel estimation and better decoding, FreeEM also has training sequences. A training sequence contains all the chips that a victim can produce to construct pattern-based 2D symbols, whose features will be studied by the attacker after they are received. Meanwhile, we also design a guard interval at the end of each training sequence to differentiate different victims at the receiver side.

The reason for designing training sequences lies in the variability of memory EMR signals, which always impedes accurate decoding. According to our empirical study, the spectrum pattern of EMR signals often changes, even when using the same computer to run the same code. Fig. 8 demonstrates the spectrum of running a piece of code (i.e., a training sequence) two times on the same computer. Theoretically, the high-energy signals are supposed to appear at exactly the same position (denoted in the orange dotted circle). However, the positions of high-energy signals vary significantly on the second run of the code, which prevents us from using a common statistical analysis for decoding. Therefore, when initializing the transmission, training sequences of each victim computer should be sent ahead to the attacker for learning. Note that (1) only one-time transmission of training sequences is needed for each victim; (2) training sequence and payload are homogeneous (i.e., 2D symbols). Thus, victim computers can send training sequences under the control of malware without extra requirements or capabilities.

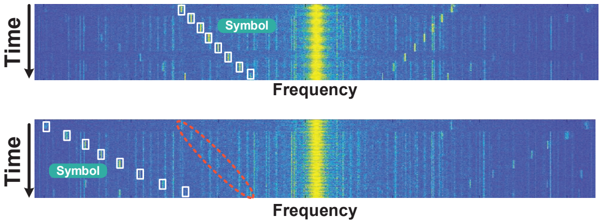


Figure 8: Dynamic Symbol Positions on Spectrum

6.2.3 Parallel Payload Transmission. Ideally, all victim devices can send their pattern-based 2D symbols on synchronized time slots, contributing a perfect combined signal pattern shown on the spectrum, such as the results in Fig. 6. However, our observation shows that sometimes the transmitted symbols still have time shifts, mainly due to 1) inaccurate time synchronization caused by network delays; and 2) different CPU clock rates, which results in different running times even for the same code. Fortunately, these time shifts will not impact the pattern of symbols. The deep-learning-based decoding mechanism can match the received spectrum patterns with pre-defined patterns, which minimizes the decoding error caused by time shifts.

6.3 FreeEM Signal Generation and Modulation

Generating memory EMR signals by repeatedly performing memory reading/writing operations has been proposed in many previous works [25, 28, 31]. To enhance the intensity and stability of EMR signals, we modify the assembly code as illustrated in Listing 1.

6.3.1 Memory EMR Signal Generation. The following factors affect the intensity/stability of memory EMR signals.

- **Memory Operations**: VMOVDQA m r and VMOVDQA r m can perform memory read and write operations, in which m,r represent the memory and register, respectively. The memory write operation usually produces a more intense EMR than the memory read operation. Thus, we adopt memory write operations to generate EMR (as lines 4 and 6 in listing 1).
- **Data Length**: Different assembly instructions, such as MOVD (64 bits), MOVDQA (128 bits), and VMOVDQA (256-512 bits), can perform the memory write operations, each of which can move different lengths of data at a time. Experimental results show that a larger data type can produce more intense EMR signals. Thus, we adopt VMOVDQA to perform the write operation (as lines 4 and 6 in listing 1).
- **Memory Address**: For each time running the program, the operating system will allocate a different memory address, which results in different signal strengths. To address this problem, we request multiple memory addresses to improve the signal stability (as line 10 in listing 1) because using more memory addresses means we have a higher probability of acquiring memory addresses having better effects.

```
1  for (cnt=1; cnt<chipLen; cnt++){
2      asm volatile(
3          " clflush (%0) \n"
4          " vmovdqa %%ymm0, (%0) \n"
5          " clflush (%1) \n"
6          " vmovdqa %%ymm1, (%1) \n"
7          " mfence \n"
8          " mfence \n"
9          :
10         : "r" (addr0), "r" (addr1)
11         : "%ymm0", "%ymm1");
12 }
```

Listing 1: Memory Writing Operations

6.3.2 Memory EMR Signal Modulation. To generate chips with different frequencies, different memory access frequencies should be adopted. As shown in Listing 1, by increasing the quantity of MFENCE instructions, the proportion of writing operation (i.e. VMOVDQA) will decrease, leading to the sub-peaks moving toward the central area of the spectrum. Motivated by this observation, we can achieve precise control of the position of sub-peaks. When conducting signal modulation, the following key points should be considered,

• **Distinguishability**: The chips should be defined with enough distinguishability to each other, i.e., the frequency difference between two chips should be large enough. Since a USRP N210 is used as the receiver with sampling rate=20M samples/s, bandwidth=20MHz, and FFT size=256, any two chips should have a frequency difference of at least 78.1KHz =(20MHz/256). Meanwhile, our experimental results show that the distinguishability is not stable, which will decrease with the decrease of writing frequency. Therefore, the low-frequency area is not suitable for defining too many chips.

• **Unified Signal Intensity**: The intensity of chips also depends on the proportion of VMOVDQA instructions. For example, a symbol having 5 VMOVDQA instructions and 5 MFENCE instructions indicates only about half of the time is used to generate EMR signals (assuming two instructions have the same time duration). Thus, its signal intensity is weaker than a chip having 10 VMOVDQA. To keep all symbols have similar intensity, the proportion of VMOVDQA and MFENCE should be well designed.

• **Unified Duration**: To ensure high decoding accuracy, all chips should have the same duration. However, the fact is that different chips usually have different durations, because 1) the proportions of writing cycles and idle cycles in each chip are different, and 2) the duration of a single writing cycle and idle cycle are also different. Hence, we need to design different numbers of repetitions (i.e., chipLen) for each chip as shown in line 1 of List. 1.

6.4 2D Symbol Selection

According to the previous discussion, each pattern-based 2D symbol should have no overlap with others. By evenly dividing the symbol set, each victim computer can use its allocated subset to transmit data. However, the number of non-overlapping symbols is extremely limited, which will potentially restrict the expected overall data rate. Thus, we propose to relax the constraint of having no overlap to a few overlaps (e.g., 1 or 2 out of 8) in the practical design. The challenging question becomes **how to define pattern-based 2D symbols as many as possible, such that any two symbols have a limited number of frequency overlaps**?

6.4.1 A Naïve Algorithm. Suppose a pattern-based 2D symbol consists of m time slots and n available frequencies. We need to determine the maximum number of symbols that can be used if any two symbols have at most 1 chip of overlap. Apparently, traversing all the possible combinations of symbols is the simplest method. The symbol selection process can be described as follows,

- Step 1: The total number of available symbols is $s = n^m$;
- Step 2: Select s' ($s' = s$ for the initial state) symbols to form a new subset;
- Step 3: Check whether any two symbols of the subset have at most 1 chip of overlap. If yes, the subset is the optimal solution. Otherwise, go to step 4;
- Step 4: If there is no other subset whose size is equal to s', then $s' = s' - 1$;
- Step 5: If $s' \geq 2$, go back to step 2. Otherwise, the selection process ends.

Although the above algorithm can obtain the optimal solution, the efficiency is very low and cannot be used in practice. When the symbol length or available frequencies increase, the extremely high time complexity and space complexity will make this algorithm not applicable. Specifically, the symbol selection will be executed for the following number of rounds if we do not store the previous result,

$$N_1 = \sum_{s'=2}^{n^m} \binom{n^m}{s'} \binom{s'}{2}. \tag{1}$$

When $m = 8, n = 8$, $N_1 = 1.25 \times 10^{5050455}$, which is unacceptable. If we store the previous results to reduce the repeated calculation, we will need a matrix containing n^{2m} elements to store the overlapping status of any two symbols. When $m = 8, n = 8$, the space complexity will be 281TB, which is also unacceptable. Tab. 3 in Appendix A shows more detail about the theoretical time and space requirements to solve this problem, indicating that the traversing algorithm is impracticable.

6.4.2 Advanced Selection Algorithm. In order to reduce the complexity, we propose a fast-traversing algorithm with a pruning function. We prepare a coexistence set and a candidate set. The coexistence set is used to store eligible symbols, while the candidate set contains all symbols in the initial state. For each time, we select a symbol from the candidate set and put it into the coexistence set. Then, we check and delete the overlapping symbols in the candidate set. This pruning process can greatly reduce both time and space complexity because we do not need to repeatedly check the compatibility among symbols or reserve a large space to store the compatibility information. By moving the symbols from the candidate set to the coexistence set, the coexistence set will increase, whereas the candidate set will decrease. Once the candidate set is empty, the selection process is completed.

The pruning algorithm significantly reduces the alternative symbols and the number of times in testing the symbol coexistence. After adopting the pruning algorithm, the size of the candidate set can be reduced to 0 within 10 cycles. The task (e.g., $m = 8, n = 8$) can be completed in seconds. Fig. 25 in Appendix B shows an example of how the size of the candidate set shrinks with the number of times using the pruning function. We can further formulate the decrease of candidate set size as a composite exponential function, $f(x) = ab^x$. Specifically, the number of available symbols will reduce to b^x $(0 < b < 1)$ of the initial value a after the x-th time of pruning. Hence, the coexistence test will conclude in the following rounds.

$$N_2 = \sum_{s'=2}^{n^m} \sum_{x=0}^{s'} \binom{n^m f(x)}{1} n^m f(x) \tag{2}$$

6.4.3 Discussion. With the increase of the problem size, N_1 rapidly grows to an enormous value, beyond the normal computational capabilities. Although N_2 is much less than N_1, it is still a very large number. In practice, we do not need to traverse all cases, because we have an anticipation of the size of the final coexistence set. Experimental results indicate that the size of the final coexistence set is no more than n^2 under the condition that 1) any two symbols have at most 1 chip of overlap; and 2) $n \geq m$. Thus, when traversing, we just need to verify the sets whose size $s' \leq n^2$. The detail of our algorithm is shown in Appendix C. After generating the coexistence set, it is divided into v disjoint subsets and implanted into v victim computers along with malware. Each victim computer possesses a unique subset, allowing the attacker to determine the signal source. Missing signals from some victim computers do not hinder the attacker's ability to receive signals from others. The attacker can accurately determine the identities of victim computers that failed to send EMR signals.

7 FREEEM SIGNAL DECODING

Different from existing EMR covert communication protocols, we leverage the deep learning approach to improve the decoding performance. The decoding process is shown in Fig. 9. After transforming the received signals from time domain to frequency domain, the main component for decoding is the Deep Neural Network (DNN), which classifies the patterns of spectrum into combined chips.

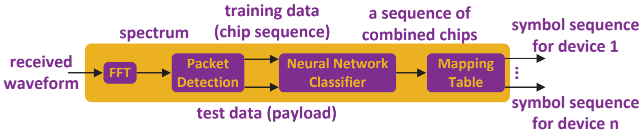


Figure 9: FreeEM Decoding Process

7.1 Packet Detection

7.1.1 Chip Detection. After the FFT, the received signal will be sent to the packet detection module, in which the first step is to perform chip detection to determine whether the received symbol comes from a predefined chip. Based on our pattern-based 2D symbol design, a predefined chip has high-energy features on specific frequencies. We define the following rules for chip detection,

$$p = max \left\{ \frac{\mathbb{E}(max_k\{P_1\})}{\mathbb{E}(P - P_1)}, \frac{\mathbb{E}(max_k\{P_2\})}{\mathbb{E}(P - P_2)}, ..., \frac{\mathbb{E}(max_k\{P_n\})}{\mathbb{E}(P - P_n)} \right\} \quad (3)$$

$$c = \begin{cases} 1 & p \geq \theta \\ 0 & Otherwise \end{cases} \quad (4)$$

where P_n represents the power of high-energy frequency points that correspond to chip n. In particular, P is a power sequence (length=FFT Size) that corresponds to the whole frequency domain. We use max/max_k to denote the finding of maximum or maximum k elements from a sequence. $\mathbb{E}()$ is a function to calculate the average power value, and θ is a threshold set by the receiver. If $p \geq \theta$, this piece of signals will be determined as a "chip"; otherwise, it will be regarded as "no signal" or "noise".

7.1.2 Chip Sequence Delimitation. The received signal can be either the training sequences or the payload symbols. To extract training sequences for DNN training, we use the above method to judge whether the signal is a chip (i.e., c=1) or a guard interval (i.e., c=0). Then, we use a sliding window w_j to detect the boundary of chip sequences,

$$w_j = \sum_{i=j}^{j+L-1} c_i - \sum_{i=j-L}^{j-1} c_i \quad (5)$$

where L is the size of sliding window and c_i is the sequence acquired from Eq. (4). Hence, the peak/valley at the starting/ending point can be calculated using w_j. Finally, we can use those chip sequences to train the DNN classifier.

7.2 Chip Classification

Most existing works [25, 28] demodulate the memory EMR signals by statistical analysis, which can hardly be used in our scheme for the following reasons: 1) the kinds of combined memory EMR signals are too many to be statistically analyzed; 2) the diversity of hardware settings result in the inconsistency between theoretical and real EMR signals. Thus, we cannot demodulate the real EMR signals with a predefined theoretical model.

To tackle this problem, we build a DNN consisting of 3 layers to classify the received chips. The size of the input layer is 256, representing the number of sampling points in a chip. In particular, the size of our proposed DNN output layer depends on the possible number of combined chips. For example, if 8 available chips are used by 4 computers, then, there will be $8^4 = 4,096$ possible combinations. Thus, the size of the output layer should be 4096. Fully connected layers are adopted as hidden layers, whose size gradually increases or decreases to match the dimensions of the input and output layers (e.g., hidden layer 1 = 1,024 and hidden layer 2 = 2,048).

7.3 Chip-to-Symbol Mapping

Given a sequence of combined chips as the output of the classifier, the next step is to perform chip-to-symbol mapping to extract payload symbols from each individual victim.

We use a sliding window with size w (assuming a symbol consists of w chips) to detect whether the current window contains a symbol from a specific victim computer. Since we know the symbol set of each victim computer, we can compare them individually with the symbol in the sliding window to choose the one with the smallest difference as the demodulation result. Then, the sliding window will move forward to demodulate the next symbol. During decoding, the instability of memory EMR signals may degrade the decoding accuracy, mainly due to 1) chips may get lost or cannot be detected (hardware failures), and 2) chips may last longer or shorter than expected. An incorrect chip delimitation will result in the decoding error of not only the current symbol but also the subsequent symbols. To address this issue, we adopt dynamic time warping (DTW) [20] to compare the similarity between received chip sequences and predefined chip sequences. Since DTW does not require that two sequences have the same length, it will be feasible to correct the delimitation error with the process of decoding. By integrating with DTW in chip-to-symbol mapping, the combined chips can be converted to symbols with high accuracy.

8 PERFORMANCE EVALUATION

8.1 Experiment Settings

To better evaluate the parallelism, adaptability, and compatibility of our design, as shown in Fig. 10, we conduct experiments on different hardware settings on the victim computers, including four Z97 MPOWER MAX AC motherboards (with DDR3-1600 memories), four ROG STRIX B-350F Gaming motherboards (with DDR4-2666 memories) and a Z97M-Plus motherboard (with DDR3-1600 memories). For the receiver, a USRP N210 is used together with RFSPACE UWB-3 Antenna, which can capture signals ranging from 675 MHz to 12000 MHz. Besides, 3 SDR dongles are used to further demonstrate mobility and portability of FreeEM.

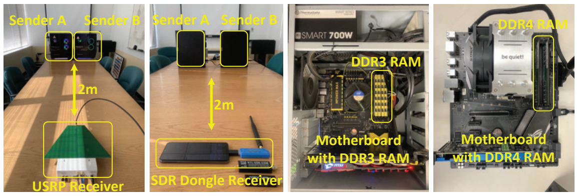


Figure 10: Experiment Scenario

We will focus on diverse metrics, including throughput, error rate, and signal intensity, in various settings such as different relative distances from the victim computers to the receiver, modulation schemes, orientations of victim computers, movement of the receiver, and different obstacles. We also conduct evaluations in a real office scenario where computers are randomly located and the attacker receives FreeEM signals in different places.

8.2 EMR Signal Intensity

In a "1-to-1" communication scenario, a stronger signal generally implies better communication effectiveness. However, in an "*n*-to-1" communication scenario, the presence of mutual interference can lead to different outcomes. Therefore, further research is needed to understand the dynamics in such scenarios. We first use four ROG STRIX B-350F Gaming motherboards (with DDR4-2666 memories) as the senders and evaluate the received signal intensity at different locations. Four senders are placed in the middle of the testing ground, as the pentagrams shown in Fig. 11.

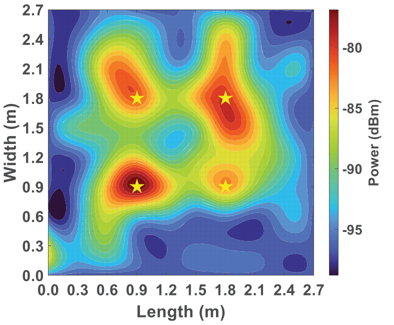


Figure 11: Signal Intensity around Victims

Fig. 11 illustrates the variation in signal intensity across different locations. By comparing the signal intensity and communication performance at various locations, we draw the following conclusions: (1) The central point may not necessarily exhibit optimal communication performance. Intuitively, placing the receiver at the center should yield the best communication performance, but this is not the case. The reason is that the antenna of the receiver will dominate the signal reception. If it faces one of the senders, the signals from other senders will be poor. If it does not point to any sender, the signal intensity will be significantly lower than the expected value. (2) Being overly close to a particular signal source may result in a decrease in data accuracy from other signal sources. When the signal intensity from a particular source is significantly stronger than that of other sources, the DL-based decoder may erroneously assume that the other sources have not transmitted signals and, as a result, produce null decoding results. (3) The appropriate parallel covert communication range is 1-2 meters. Therefore, in the following experiments for evaluating parallelism, we position the communicating parties at a distance of 2 meters apart to evaluate performance metrics.

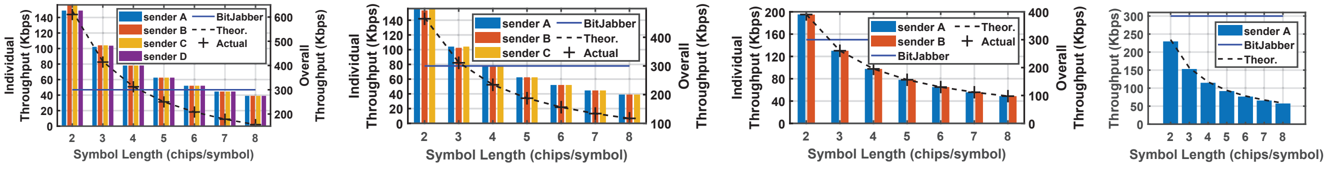
8.3 Parallelism of FreeEM

We carry out experiments to analyze the parallelism of FreeEM when multiple computers work together. As depicted in Fig. 10, victim computers are positioned in front of the attacker, anticipating a Line of Sight (LoS) transmission at equal distances from the victims. We assess the throughput and error rate at a distance of 2 meters between the victims and the attacker. Two memory platforms have been used, i.e., motherboards with DDR3 and DDR4 memory. For the receiver, we use the USRP N210 for received signal processing. Meanwhile, we also compared the performance with other existing works. In our experiments, each transmission lasts for 30 seconds, and to mitigate bias, all experiments were repeated 10 times.

8.3.1 Impact of Encoding Scheme. The encoding scheme refers to the number of chips used to form a symbol, which significantly affects throughput and error rates.

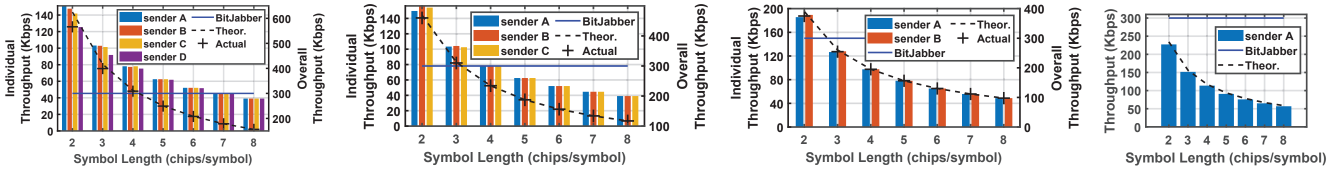
• **Throughput.** The throughput depends on the encoding scheme and the number of victim computers. Both Fig. 12 and Fig. 13 show the throughput when using different numbers of victim computers. The individual throughput (i.e., bar chart) and overall throughput (i.e., line chart) decrease with the increase of symbol length (symbol length=1 is not included in comparison since it is not a parallel communication scheme). When the symbol length gets longer, the number of symbols that can be sent in a unit of time will decrease, and thus the throughput will reduce. Among all cases, the encoding scheme with 2 chips/symbol usually reaches the optimal performance for both individual and overall throughput. In particular, the optimal overall throughput can achieve 625Kbps when 4 victim computers running FreeEM, reaching approximately 2.1X than *BitJabber* in [28] (the fastest memory EMR covert communication scheme). By comparing Fig. 12(a), 12(b) and 12(c), the overall throughput of our FreeEM protocol continuously increases with the increase of the number of victims. Also, victims do not have a significant difference in individual throughput because they coherently transmit EMR signals instead of competing for the channel.

Fig. 13 (DDR4 RAM) shows similar experimental results compared with Fig. 12 (DDR3 RAM) but with slightly lower throughput, mainly due to the weaker EMR signal strength. As given in Fig. 12 and Fig. 13, our throughput performance can achieve more than 90% of the theoretical value. These results not only show the correctness of the coexistence set selection algorithm but also demonstrate excellent anti-interference and parallel performance. Although designed for an "*n*-to-1" scenario, FreeEM can also operate in a "1-to-1" mode, achieving throughputs of 234.37 Kbps for DDR4 memory in Fig. 12(d) and 227.34 Kbps for DDR3 memory in Fig. 13(d), both of which are slightly less than those of *BitJabber*. The performance difference is not due to algorithm design flaws but rather to hardware capabilities, such as the precision with which the sender (memory) and receiver (SDR) can generate and recognize signals.



(a) 4 Victim Computers (b) 3 Victim Computers (c) 2 Victim Computers (d) 1 Victim Computer

Figure 12: Individual and Overall Throughput Analysis - DDR3



(a) 4 Victim Computers (b) 3 Victim Computers (c) 2 Victim Computers (d) 1 Victim Computer

Figure 13: Individual and Overall Throughput Analysis - DDR4

• **Decoding Accuracy.** We also evaluate the error rate of FreeEM, as shown in Fig. 14. The SER measured from DDR3 decreases with the increase in symbol length. As the symbol length increases, more chips are contained in each symbol, allowing for better verification of correctness and reducing the error rate. An interesting observation is Symbol length=1 usually has a lower SER than Symbol length=2. This is because an error in one of the two chips in the Symbol length=2 causes the entire symbol to fail, leading to a higher probability of errors. Our scheme achieves extremely low SERs, nearly reaching zero when symbol lengths are greater than 4. Even for the maximum transmission rate (i.e., the aforementioned 625Kbps, with 4 senders, symbol length=2), the average SER is only 3.12%, which demonstrates good anti-interference performance. Meanwhile, the SER decreases as the number of victims decreases. As for 2 victims, the SER will be lower than 1%. We also observe that different victims may have different SER, which is mainly due to (1) the unbalanced assignment of symbols: some victim computers may have higher quality symbols that are less error-prone; and (2) the randomly allocated memory addresses: different memory addresses may have different effects in emitting EMR. Similar to the throughput analysis, the SER (using DDR4 RAM) in Fig. 15 is generally higher than that of DDR3, because the motherboards with DDR4 usually have weaker EMR than the motherboards with DDR3.

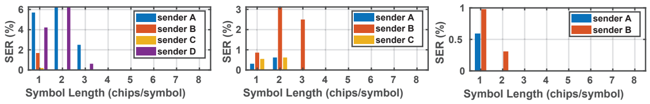


Figure 14: DDR3-Symbol Error Rate w/4, 3, 2 victims

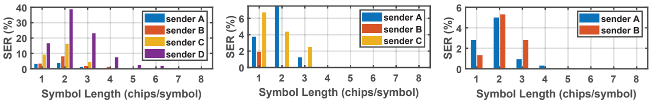


Figure 15: DDR4-Symbol Error Rate w/4, 3, 2 victims

When jointly comparing results in Fig. 12 and 14, we can see a clear tradeoff among symbol length, throughput, and SER. With the increase of symbol length, more chips are used in a symbol, which greatly decreases the error rate, but the overall throughput also degrades. Hence, given different scenarios, our proposed scheme can be fine-tuned to meet diverse communication needs.

8.3.2 Impact of Distance. To assess the impact of distance on communication performance, we arranged the victim computers and the receiver in a row, as illustrated in Fig. 16.

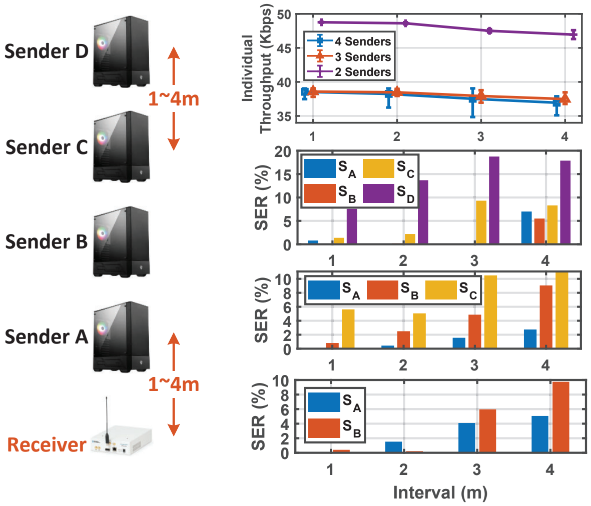


Figure 16: Performance at Different Intervals

The RAMs in each victim computer are oriented towards the direction of the receiver. We uniformly extend the distance between each device, ranging from 1m to 4m. Hence, the maximum distance ranges from 4m to 16m. Fig. 16 shows that the individual throughput gradually decreases from 38.55 / 38.59 / 48.77Kbps to 36.95 / 37.49 / 46.96Kbps for 4 / 3 / 2 victims, respectively. As the interval increases, the SER demonstrates an upward trend. With a determined interval, the more senders, the higher the average SER. The SER of the farthest sender is 17.87%, 11.06%, and 9.75% for the case of 4 / 3 / 2 victims, respectively. In addition, the different positions will make the memory EMR signal reception unbalanced (a.k.a. near-far effect), e.g., sender D usually has a significantly higher SER than sender A.

To sum up, the above results indicate that the proposed FreeEM can still maintain approximately 95% of the theoretical throughput for 4 victim computers.

8.4 Adaptability of FreeEM

We further evaluate the adaptability of FreeEM in detail. In what follows, we will use 8 chips/symbol as the encoding scheme to achieve more reliable covert communication.

8.4.1 Impact of Motion. Memory EMR covert communication is very sensitive to the movement of both communicating parties. We arrange the victims in a row and instruct the receiver to move from near to far at a speed of 0.5 m/s, as shown in Fig. 17. Then, we measure the individual average throughput (i.e., traveling 1 meter over 2 seconds) and SER as the receiver passes through specific locations. It can be seen that the SER increases with the distance. The maximum SER increases from 27.91% to 69.56% when the victim number increases from 2 to 4, because more victims will cause more collisions on the spectrum. However, in most cases, our scheme can ensure a low SER (i.e., <10%) when the distance is less than 10m, which demonstrates its ability to sustain reliable communication during motion. Similar to Fig.14 and Fig. 15, the SER is sometimes unbalanced among different victim computers due to different assigned symbol sets and indeterminately assigned memory addresses.

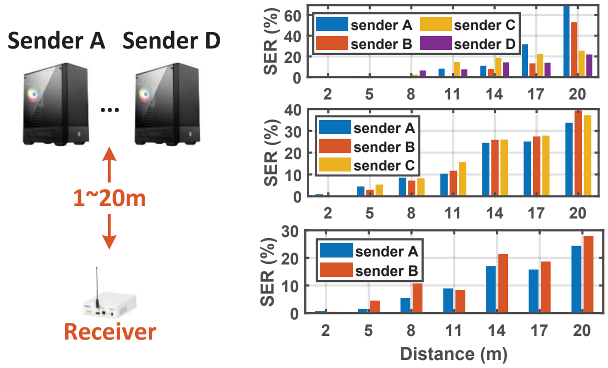
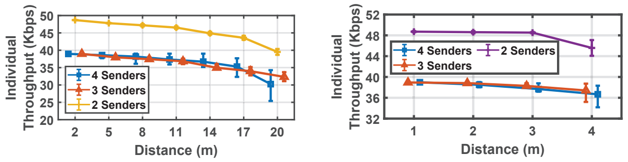


Figure 17: SER in Linear Motion

From Fig. 18(a), our scheme can achieve a long communication range of up to 20m. The individual throughput gradually decreases with distance. 4 senders and 3 senders have similar individual throughput due to their identical symbol set sizes (i.e., 16 symbols/sender). 2 senders has a larger symbol set size (i.e., 32 symbols/sender), resulting in higher individual throughput. Note that the overall throughput still follows the pattern 4 senders > 3 senders > 2 senders, since the number of senders is taken into consideration.

8.4.2 Impact of Rotation. We position the receiver at the center of the victims, rotating at a speed of one revolution every 8 seconds while receiving data, as shown in Fig. 19. All DRAMs are directed towards the center of the circle, with the radius incrementing from 1m to 4m in increments of 1m.

Fig. 18(b) shows the individual throughput decreases from 38.98 / 38.94 / 48.71Kbps to 36.67 / 37.4 / 45.58Kbps for 4 / 3 / 2 victims, respectively. In Fig. 19, the SER shows an upward trend with the increase in distance. When the distance is short, the SERs among different victims do not have significant differences, since the distance between the receiver to all victims is the same. Only when the distance is set to 4m, a higher SER occurs. Different from linear motion, the continuous variation of antenna angles is the main reason for the degradation in decoding accuracy and throughput.



(a) Linear Motion (b) Rotational Motion

Figure 18: Throughput v.s. Types of Motion

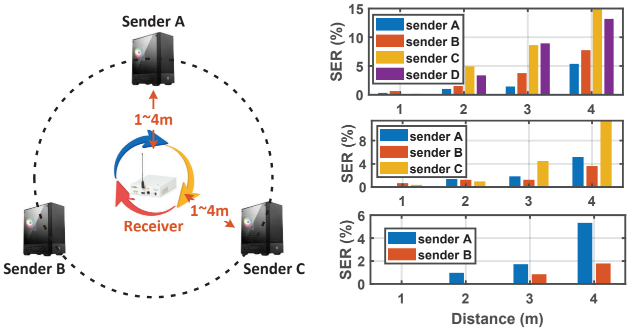


Figure 19: SER in Rotational Motion

8.4.3 Impact of Obstacles. Besides conducting experiments in the LoS scenario, we also evaluate the performance of penetrating obstacles. Three common obstacles are used as obstacles in our experiments, including glass, wood, and concrete wall. As shown in Fig. 20, 4 victims and the receiver are 4m apart from each other with the obstacle in the middle. Tab. 2 shows the symbol decoding accuracy with different materials as the obstacles. It can be seen that the glass and wood have a similar effect in blocking signals, with which the average symbol accuracy is 99.4% and 98.56%, respectively. However, with the concrete wall, the average symbol accuracy decreases to 90.24%. Our scheme can maintain a throughput higher than 148.4Kbps in all the above cases, which demonstrates FreeEM has a good performance in penetrating walls and low-density materials.

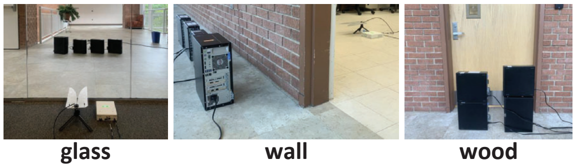


Figure 20: Types of Obstacles

Table 2: Obstacle-penetrating Performance

Material	A	B	C	D	Throughput
Glass	98.69%	100%	99.06%	99.94%	155.7Kbps
Wood	98.75%	99.38%	96.56%	99.56%	155Kbps
Concrete	81.44%	94.5%	86.31%	98.69%	148.4Kbps

8.5 Office Environment Experiments

To assess the compatibility and practicality of FreeEM, we conducted evaluations in a real office setting with concrete walls. As shown in Fig. 21, four computers are placed in different positions, in which sender A uses the Z97M-Plus motherboard, sender B, C and D use Z97 MPOWER MAX AC motherboards. According to the previous discussions, the orientation of a motherboard is critical for communication performance. Hence, these motherboards face different directions (denoted as red arrows). For victim B, the motherboard faces the ceiling. We evaluate the SER and overall throughput at 4 locations (P1 to P4), where P1 is at the center of the room, P2 is at the door, P3 and P4 are outside the room. For P1, there exist LoS to all victims, whereas for P2, P3, and P4, there only exists LoS to 3 victims (i.e., A, B, C), 2 victims (i.e., A, B), and 1 victim (i.e., A), respectively.

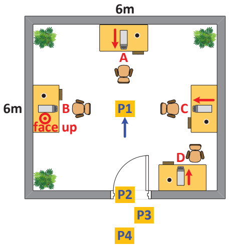


Figure 21: Setting

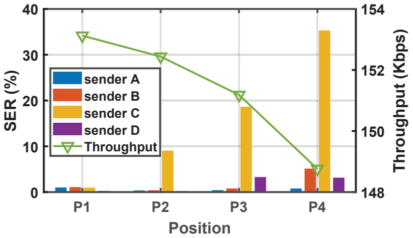


Figure 22: Throughput and SER

In Fig. 22, as the receiver moves towards the outside of the room, the SER (throughput) shows an increasing (decreasing) trend. Victim C has a significantly higher SER than other victims because its radiated signals are orthogonal to the receiver's antenna (denoted as the blue arrow), which has the weakest received signal strength. Other victims can maintain an SER lower than 5%. The overall throughput can be greater than 148.76Kbps even in the worst case. Besides the performance differences caused by orientation, the two different motherboards cooperate well and show no significant variance in communication effectiveness. This experiment indicates that 1) the orientation of a victim computer has a significant impact on the SER, 2) FreeEM demonstrates excellent compatibility, and 3) the transmission effect of one sender does not affect other senders.

8.6 Mobility and Portability Demonstration

Attackers can use mobile/portable devices to receive FreeEM signals (e.g., a laptop/smartphone + SDR dongle), as shown in Fig. 23.

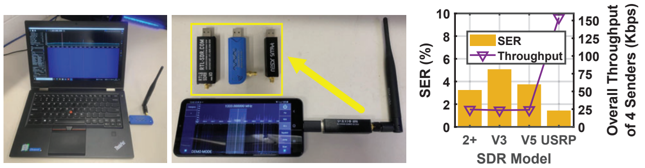


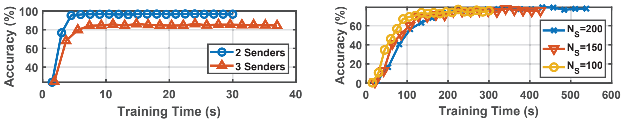
Figure 23: Mobility and Portability

We evaluate three different SDR dongles, including NESDR Mini 2+, RTL-SDR Blog V3, and NESDR SMArt RTL-SDR V5. All three have excellent software compatibility, enabling the direct display of spectra on the screen or real-time exporting of captured data for further analysis. By experiments, we found that three SDR dongles have similar overall throughput (≈ 24Kbps) and SER (≈ 4%), as shown in Fig. 23, because they have similar performance parameters (e.g., frequency capacity, bandwidth, and sampling rate) and tuner chips. Their lower sampling rates (i.e., 3.2MHz) result in lower throughput compared to USRP (i.e., 154Kbps). When using an SDR dongle as the receiver, adjustments to the data rate at the victim end are necessary. This experiment demonstrates that our scheme can be used by mobile attackers.

8.7 DNN Training Cost Analysis

To verify the efficiency of the DNN-based demodulation scheme, we conduct a cost analysis regarding the DNN training on the chip sequences. As illustrated in Sec. 6.2.2, the training time of DNN depends on the number of senders. The more senders, the more diverse combined chips need to be trained. For 2, 3, and 4 senders, there are $8^2 = 64$, $8^3 = 512$, and $8^4 = 4096$ kinds of combined chips in total, respectively. For each combined chip, we use 200 examples (each example contains 256 numbers) to train the DNN. As shown in Fig. 24, for 2, 3, and 4 senders, completing the training process (10 epochs) needs 30s, 37s, and 538s, respectively. The DNN can achieve a fully trained state (i.e., the accuracy reaches 95.19%, 84.26%, and 77.36%) when the training time is 4.5s, 7.4s, and 269s, respectively.

Besides the number of senders, the training speed also depends on the number of examples for each combined chip (e.g., $N_S = 200$ hereinbefore). Fig. 24(b) shows that completing the training process (10 epochs) needs 538s, 426s, and 300s when the $N_S = 200$, $N_S = 150$ and $N_S = 100$, respectively. Note that the N_S is not always better when smaller, because the final training effect may degrade if the N_S is too small. For example, in Fig. 24(b), the final accuracy are 77.36%, 74.88%, and 73.53% for $N_S = 200$, $N_S = 150$ and $N_S = 100$, respectively. The above results demonstrate both the feasibility and efficiency of deploying DNN for combined EMR signal classification.



(a) 2/3 victims

(b) 4 victims

Figure 24: Time Consumption in Training DNN

9 DISCUSSION

- **Supporting more victims.** FreeEM is scalable to support more victim computers. Since we stipulate that any two symbols can have at most 1 chip of overlap, d senders working together will result in at most $d(d-1)/2$ chips of overlap. Note that there is a tradeoff between supported victims and decoding accuracy. If a small number of errors can be tolerated, FreeEM can support more victims working in parallel. For example, 32 victim computers running together (using 64 symbols) can achieve a theoretical throughput of 1250Kbps.
- **Number of overlapping chips.** Allowing more overlapping chips in a combined symbol can increase the number of usable symbols, which will improve the throughput. However, it will also result in higher decoding errors, because the recognizability of symbols will be reduced. With the increase of senders, more overlapping is likely to happen. Thus, when making regulations on the number

of overlapping chips, the number of senders should be taken into consideration.

- **Potential defensive mechanism.** Spread Spectrum Clocking (SSC) technique has been proposed to spread the memory EMR energy across a wider spectrum. However, the de-spreading technique [31] can render it ineffective. One of the potential defensive methodologies could be adding a module on the operating system level to constantly detect malicious memory activities, such as frequently writing/reading to/from a specific address for a long period of time. Besides, we can develop a protection program continuously running in the memory to generate benign EMR jamming signals, in order to deviate the decoding process at the attacker.

10 CONCLUSION

In this paper, we design the `FreeEM`, the first parallel covert communication paradigm using memory EMR, which can support multiple victim computers to transmit their secret information in a volatile environment. `FreeEM` extends the knowledge that memory EMR can only support one-to-one communication. By exploiting vacant frequencies in the same time slot, the newly proposed pattern-based 2D symbols from multiple victim computers can coexist for achieving parallel covert communication. Extensive experimental results have demonstrated that `FreeEM` significantly increases the overall throughput with low error rates in different scenarios.

ACKNOWLEDGEMENT

The work of L. Guo is partially supported by NSF under grants CNS-2008049 and CCF-2312616, and ARO under grant W911NF-24-1-0044. The work of Z. Zhang is partially supported by NSF under grant CNS-2147217. The work of L. Cheng is partially supported by CNS-2239605. The work of X. Zhang is partially supported by CCF-2312617.

APPENDIX

A COMPLEXITY ANALYSIS OF NAIVE ALGORITHM

The time and space complexity of the naive algorithm depends on the scale of the problem (i.e., m and n). Table 3 shows some typical cases. Even if the m and n are very small, the time/space complexity of the problem has already far exceeded the capacity of the computer.

Table 3: Time/Space Complexity

Parameters	Time Complexity	Space Complexity
$m = 3, n = 3$	1.82×10^{9}	729B
$m = 4, n = 4$	4.73×10^{79}	65.54KB
$m = 5, n = 5$	9.12×10^{944}	9.77MB
$m = 6, n = 6$	7.21×10^{1450}	2.18GB
$m = 7, n = 7$	1.04×10^{247919}	678GB
$m = 8, n = 8$	1.25×10^{5050455}	281TB

B EFFICIENCY OF PRUNING ALGORITHM

After adopting the pruning algorithm, the size of the candidate set can be reduced to 0 within 10 cycles, as shown in Fig. 25. The task (e.g., $m = 8, n = 8$) can be completed in seconds.

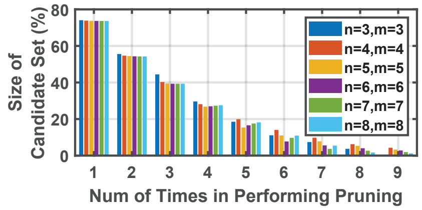


Figure 25: Efficiency of Pruning Algorithm

C ALGORITHM OF FINDING MAX. COEXISTENCE SET

To find the maximum coexistence set, we use a variable-length array I to denote the indexes of symbols we expect to select from the candidate set. The initial array can be $I = [1, 1, ..., 1]$ (n^2 terms in total), i.e., for each time, we intend to select the first symbol of the candidate set (then, delete the overlapping symbols from the candidate set), until the size of the candidate set reduces to 0. If we cannot obtain n^2 terms before the size reduces to 0, it means some items are improperly selected. In this case, we need to restore to the previous state and try other larger I arrays (e.g., $I = [1, 1, ..., 2]$). After determining the coexistence set, we will divide it into S parts evenly and allocate them to S senders. Each sender can use its own symbols to encode and send information with minimum collisions while maintaining a high data rate and decoding accuracy.

Algorithm 1: Finding the maximum coexistence set

```
Input: Candidate sequence $C[n^m]$
       Expected size of the coexistence set $s$
Output: Coexistence set $X$
$I \leftarrow [1, 1, ..., 1]$; // index sequence
$s' \leftarrow 0$; // effective length of the $I$
$r \leftarrow 0$; // whether need to reconstruct $X$
while $s' \leq s$ do
    if $r = 1$ then
        $X \leftarrow$ Reconstruct X$(s', I)$;
        // reconstruct $X$ according to current $s'$ and $I$
        $r \leftarrow 0$;
    else
        if Length$(C) < I[s' + 1]$ then
            // there is no sufficient items in $C$ for selecting
               according to $I$
            $I[s' + 1 : end] \leftarrow 1$;
            $I[s'] \leftarrow I[s'] + 1$;// improve the value of $I$
            $s' \leftarrow s' - 1$;// reduce the effective length of $I$
            $r \leftarrow 1$;
        else
            // it is feasible to pick out item from $C$
               according to $I$
            $X \leftarrow X \cup \{C[I[s' + 1]]\}$;
            $I_c \leftarrow$ ChkCompatibility$(C[I[s' + 1]], C)$;
            // find the indexes of items in $C$ which conflict
               with $C[I[s' + 1]]$
            $C[I_c] \leftarrow \emptyset$;
            if $s' > s_{max}$ then
                $s_{max} \leftarrow s'$;
            $s' \leftarrow s' + 1$;
if $s' > s$ then
    return $X$;
else
    return 0;
```

D ARTIFACT

The research artifacts accompanying this paper are available via https://doi.org/10.5281/zenodo.11089810.

REFERENCES

[1] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Mark Townsley. A study of lora: Long range & low power networks for the internet of things. *Sensors*, 16(9):1466, 2016.

[2] Robert Callan, Alenka Zajic, and Milos Prvulovic. A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 242–254. IEEE, 2014.

[3] Giovanni Camurati and Aurélien Francillon. Noise-sdr: Arbitrary modulation of electromagnetic noise from unprivileged software and its impact on emission security. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1193–1210. IEEE, 2022.

[4] Brent Carrara and Carlisle Adams. On acoustic covert channels between air-gapped systems. In *International Symposium on Foundations and Practice of Security*, pages 3–16. Springer, 2014.

[5] Paizhuo Chen, Lei Li, and Zhice Yang. {Cross-VM} and {Cross-Processor} covert channels exploiting processor idle power management. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 733–750, 2021.

[6] Chaojie Gu, Jiale Chen, Rui Tan, and Linshan Jiang. An electromagnetic covert channel based on neural network architecture. In *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 177–184. IEEE, 2021.

[7] Mordechai Guri. Magneto: Covert channel between air-gapped systems and nearby smartphones via cpu-generated magnetic fields. *Future Generation Computer Systems*, 115:115–125, 2021.

[8] Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval Elovici. {GSMem}: Data exfiltration from {Air-Gapped} computers over {GSM} frequencies. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 849–864, 2015.

[9] Mordechai Guri, Gabi Kedma, Assaf Kachlon, and Yuval Elovici. Airhopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. In *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, pages 58–67. IEEE, 2014.

[10] Mordechai Guri, Matan Monitz, and Yuval Elovici. Usbee: Air-gap covert-channel via electromagnetic emission from usb. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 264–268. IEEE, 2016.

[11] Mordechai Guri, Matan Monitz, Yisroel Mirski, and Yuval Elovici. Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In *2015 IEEE 28th Computer Security Foundations Symposium*, pages 276–289. IEEE, 2015.

[12] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, and Yuval Elovici. Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise ('diskfiltration'). In *European symposium on research in computer security*, pages 98–115. Springer, 2017.

[13] Mordechai Guri, Boris Zadov, and Yuval Elovici. Led-it-go: Leaking (a lot of) data from air-gapped computers via the (small) hard drive led. In *International conference on detection of intrusions and malware, and vulnerability assessment*, pages 161–184. Springer, 2017.

[14] Mordechai Guri, Boris Zadov, and Yuval Elovici. Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields. *IEEE Transactions on Information Forensics and Security*, 15:1190–1203, 2019.

[15] Markus G Kuhn and Ross J Anderson. Soft tempest: Hidden data transmission using electromagnetic emanations. In *International Workshop on Information Hiding*, pages 124–142. Springer, 1998.

[16] Butler W Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973.

[17] Sami M Lasassmeh and James M Conrad. Time synchronization in wireless sensor networks: A survey. In *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, pages 242–245. IEEE, 2010.

[18] Corentin Lavaud, Robin Gerzaguet, Matthieu Gautier, Olivier Berder, Erwan Nogues, and Stephane Molton. Whispering devices: A survey on how side-channels lead to compromised information. *Journal of Hardware and Systems Security*, 5(2):143–168, 2021.

[19] Zhengxiong Li, Baicheng Chen, Xingyu Chen, Huining Li, Chenhan Xu, Feng Lin, Chris Xiaoxuan Lu, Kui Ren, and Wenyao Xu. Spiralspy: Exploring a stealthy and practical covert channel to attack air-gapped computing devices via mmwave sensing. In *The 29th Network and Distributed System Security (NDSS) Symposium 2022*. The Internet Society, 2022.

[20] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[21] Bushra Sabir, Faheem Ullah, M Ali Babar, and Raj Gaire. Machine learning for detecting data exfiltration: A review. *ACM Computing Surveys (CSUR)*, 54(3):1–47, 2021.

[22] Fatima Salahdine and Naima Kaabouch. Social engineering attacks: A survey. *Future Internet*, 11(4):89, 2019.

[23] Nader Sehatbakhsh, Baki Berkay Yilmaz, Alenka Zajic, and Milos Prvulovic. A new side-channel vulnerability on modern computers by exploiting electromagnetic emanations from the power management unit. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 123–138. IEEE, 2020.

[24] Vitali Sepetnitsky, Mordechai Guri, and Yuval Elovici. Exfiltration of information from air-gapped machines using monitor's led indicator. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 264–267. IEEE, 2014.

[25] Cheng Shen, Tian Liu, Jun Huang, and Rui Tan. When lora meets emr: Electromagnetic covert channels can be super resilient. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1304–1317. IEEE, 2021.

[26] Lanqing Yang, Xinqi Chen, Xiangyong Jian, Leping Yang, Yijie Li, Qianfei Ren, Yi-Chao Chen, Guangtao Xue, and Xiaoyu Ji. Remote attacks on speech recognition systems using sound from power supply. In *The 32th USENIX Security Symposium (USENIX Security '23)*, 2023.

[27] Alenka Zajić and Milos Prvulovic. Experimental demonstration of electromagnetic information leakage from modern processor-memory systems. *IEEE Transactions on Electromagnetic Compatibility*, 56(4):885–893, 2014.

[28] Zihao Zhan, Zhenkai Zhang, and Xenofon Koutsoukos. Bitjabber: The world's fastest electromagnetic covert channel. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 35–45. IEEE, 2020.

[29] Zihao Zhan, Zhenkai Zhang, and Xenofon Koutsoukos. A high-speed, long-distance and wall-penetrating covert channel based on em emanations from dram clock. *Journal of Hardware and Systems Security*, 6(1):47–65, 2022.

[30] Zihao Zhan, Zhenkai Zhang, Sisheng Liang, Fan Yao, and Xenofon Koutsoukos. Graphics peeping unit: Exploiting em side-channel information of gpus to eavesdrop on your neighbors. In *2022 IEEE Symposium on Security and Privacy (SP)*, 2022.

[31] Zhenkai Zhang, Zihao Zhan, Daniel Balasubramanian, Bo Li, Peter Volgyesi, and Xenofon Koutsoukos. Leveraging em side-channel information to detect rowhammer attacks. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 729–746. IEEE, 2020.

Received November 30, 2023; revised February 8, 2024; accepted March 6, 2024