Journal of Hardware and Systems Security (2022) 6:47–65
https://doi.org/10.1007/s41635-022-00128-6

A High-Speed, Long-Distance and Wall-Penetrating Covert Channel Based on EM Emanations from DRAM Clock

Zihao Zhan[1] · Zhenkai Zhang[2] · Xenofon Koutsoukos[3]

Accepted: 30 June 2022 / Published online: 10 August 2022

Abstract

An air-gapped computer is physically isolated from unsecured networks to guarantee effective protection against data exfiltration. Due to air gaps, unauthorized data transfer seems impossible over legitimate communication channels, but in reality many so-called physical covert channels can be constructed to allow data exfiltration across the air gaps. Most of such covert channels are very slow and often require certain strict conditions to work (e.g., no physical obstacles between the sender and the receiver). In this paper, we introduce a new through-wall physical covert channel named *BitJabber* that is extremely fast and has a long attacking distance. We show that this covert channel can be easily created by an unprivileged sender running on a victim's computer. Specifically, the sender constructs the channel by using only memory accesses to modulate the electromagnetic (EM) signals generated by the DRAM clock. While possessing a very high bandwidth (up to 300,000 bps), this new covert channel is also very reliable (less than 1% error rate). More importantly, this covert channel can enable data exfiltration from an air-gapped computer enclosed in a room with thick walls up to 15 cm and the maximum attacking distance is more than 6 m.

Keywords Covert channel · Electromagnetic · Through-wall · Air-gapped

1 Introduction

In organizations where information security and privacy are top priorities, physical isolation is often used to prevent data exfiltration. Air-gapping is considered as one of the strongest physical isolation methods that have been widely used by, e.g., militaries and governments. An air-gapped computer has no connections with the outside unsecured networks, so that it is believed that protection against unauthorized data transfer can be effectively guaranteed.

However, recent research has discovered that many physical side effects of computation on air-gapped computers can be exploited to construct so-called physical covert channels to re-enable data exfiltration. The physical side effects that can be exploited are various, including thermal [2], optical [3–6], magnetic [7–9], acoustic [10–13], or electromagnetic (EM) [14–17]. The communication distance of such covert channels is usually very short, ranging from several centimeters to several meters, due to the high attenuation of the exploited physical effects in the distance. Information is encoded within the physical effects and transferred over the air gaps between a sender and a receiver. Normally, a sender is a piece of malware, like a Trojan horse, that has been stealthily inserted into a victim's computer, and a receiver is some device in the proximity of the sender that can capture the exploited physical effects.

Nevertheless, the security risks of such covert channels are often neglected, as they are considered hardly posing any real hazards for two reasons. First, the bandwidth of such physical covert channels is usually very low. For example, the transmission rate of the covert channel proposed

This is an extension to our previous work [1]

✉ Zihao Zhan
zihao.zhan@vanderbilt.edu

Zhenkai Zhang
zhenkai@clemson.edu

Xenofon Koutsoukos
xenofon.koutsoukos@vanderbilt.edu

[1] Department of Electrical & Computer Engineering, Vanderbilt University, Nashville 37235, TN, USA

[2] School of Computing, Clemson University, Clemson 29634, SC, USA

[3] Department of Computer Science, Vanderbilt University, Nashville 37235, TN, USA

in [2] is only 8 bits/hour (i.e., 0.002 bps). Even the fastest one reported in [3] can only reach 4000 bps. Therefore, if a large amount of data needs to be exfiltrated, an attacker has to maintain the covertly communicating status for a long period of time. In a situation where the attacker can briefly have her foothold in the proximity to the targeted computer, any lingering action may cause suspicion. Second, most of these covert channels require no physical obstacles between the sender and receiver. Thus, an attacker may encounter great difficulties in managing the placement of the receiving device. In particular, locking an air-gapped computer in an enclosed room has been regarded as a sufficiently secure protection against such physical covert channels.

In this paper, we demonstrate that there in effect exist powerful covert channels that are extremely fast and strong enough to penetrate even thick walls. Specifically, we construct such a covert channel named *BitJabber* from the EM signals generated by the DRAM clock. As discovered in [18], there are strong EM signals generated by different clocks in a computer that can propagate far, and these EM signals can be amplitude-modulated (AM) by activities driven by the corresponding clocks. Therefore, the EM signals generated by the DRAM clock can be AM-modulated by normal memory accesses to carry and transfer information over the air gaps between a pair of sender and receiver, namely forming an electromagnetic covert channel. Our experimental results show that the transmission rate of this new covert channel can reach 100,000 bps using binary frequency-shift keying (B-FSK) modulation with error rate around 0.05%, and 300,000 bps using multiple frequency-shift keying (M-FSK) modulation with error rate less than 0.1%. Moreover, this covert channel is resilient to a reasonable level of background noise and works well even in the presence of 15 cm thick walls between the sender and the receiver.

The main contributions of this paper are three-fold:

- We present a new physical covert channel named *BitJabber* that can allow expedited data exfiltration between air-gapped sender and receiver. We show that this covert channel can be easily constructed and effectively operate on modern computer platforms.
- We analyze the possible error sources of *BitJabber* and we experimentally verify that our covert channel is much more resilient to background noise compared with the state-of-the-art ones.
- We demonstrate that this new covert channel can achieve reliable communication within a few meters, even under the scenario where the sender and the receiver are in separate rooms with walls in-between.

Notice that we have previously described *BitJabber* and some preliminary results in [1]. This paper substantially extends our prior work as follows: (1) We re-implement the communication protocol with an improved synchronization technique (see Sect. 4.4). Compared to our original one in [1], this improved method can correctly align a signal with significantly lower signal-to-noise ratio (SNR). (2) We carry out more evaluations against some up-to-date computer systems equipped with DDR4 memory modules (see Sect. 5.1). The new evaluation results further demonstrate the good performance of our *BitJabber* covert channel with respect to its bandwidth. (3) We perform evaluations on the data exfiltration distance of *BitJabber* (see Sect. 5.5), which helps us understand the performance of *BitJabber* from another important perspective. The evaluation results show that *BitJabber* can be used for high-speed data exfiltration from more than 6 m away. (4) We thoroughly analyze how bit errors are caused in *BitJabber* (see Sect. 5.6). Such analyses help us better understand the evaluation results, highlight the advantages as well as limitations of this EM covert channel, and provide us with new directions on how to further improve the *BitJabber*'s performance. Additionally, we include some very recently published works on physical covert channels and add more metrics for comparison (see Sect. 2).

The rest of this paper is organized as follows: Sect. 2 briefs existing work on physical covert channels and makes a comparison across different approaches. Section 3 states the threat model considered in this paper. Section 4 presents our *BitJabber* covert channel in detail, including the techniques for modulation, demodulation and synchronization. Section 5 evaluates the performance of *BitJabber*. Section 6 lists some possible countermeasures against this new covert channel and Sect. 7 concludes this paper.

2 Related Work

The confinement problem was brought forth by Lampson in 1973 [19], which made the first mention of possible data exfiltration via covert channels. Since then, extensive research has been conducted on this topic. Basically, a covert channel is an unintended communication channel that can be used to transfer information between a sender and a receiver. Depending on the construction, covert channels can be classified into logical and physical ones. Logical covert channels usually manipulate the microarchitectural states in a processor to encode and transfer information [20], and the receiver normally runs on the same processor/platform/cloud as the sender [21–27]. On the other hand, physical covert channels are usually used to enable illegitimate communication between air-gapped computers, and are constructed from certain physical side effects of computation. In this section, we will mainly focus on physical covert channels.

2.1 Physical Convert Channels

A running computer can affect its physical environment in many ways, such as issuing heat, producing sound, emitting light, and generating EM signals. These affections are often called physical side effects of computation. To construct covert channels using such physical side effects, the sender needs to manipulate them in a controlled way such that information can be encoded within the physical side effects. As these physical side effects can propagate to a certain distance in the air, the carried information can be transferred over the air gaps. On the receiver side, the attacker measures the environmental changes introduced by the sender and interprets the measurement correctly to recover the exfiltrated information. Many physical side effects of computation have been reported as exploitable for constructing physical covert channels.

Since many components (e.g., clocks and voltage regulators) in a computer have switching behavior and thus emit strong EM signals, several EM covert channels have been created. For example, EM emanations from graphics card's clock were exploited to implement covert channel [28]. Guri *et al.* implemented multiple EM covert channels by exploiting the EM emanations from either video display unit [14], USB connectors [16], or DRAM bus clock [15]. Similar to our *BitJabber* cover channel, their *GSMem* covert channel described in [15] also relies on the EM signals related to the DRAM clock. They discovered that memory accesses could increase the strength of the EM signals in a wide frequency range around the DRAM clock frequency. By controlling the presence/absence of intense memory access behavior, the EM signals around the DRAM clock frequency can carry information through on-off keying (OOK) modulation. In our work, *BitJabber* is implemented using a different carrier with a much higher SNR and new modulation techniques. Section 5 will present the results showing that *BitJabber* outperforms *GSMem* significantly in terms of both speed and reliability. Note that EM signals can penetrate walls and be easily measured by some cheap devices, e.g., software-defined radios or mobile phones, but they can be blocked by metal shields like Faraday cage.

As the magnetic field around a computer can be affected by manipulating components like hard disk drives [9] and CPUs [7, 8], magnetic covert channels have also been constructed. The magnetic field can be measured by either specialized equipment like a digital magnetometer or any hardware equipped with magnetic sensors like mobile phones. Normally, magnetic covert channels have very low transmission speed and extremely short transmission distance. Unlike EM signals, the low-frequency magnetic emanations cannot be blocked by metal shields. However, due to their limited transmission distance, magnetic emanations are unlikely to be exploited for data exfiltration through a thick obstacle like a concrete wall.

A running computer frequently produces sound, and any device with a microphone can receive these signals. The first acoustic covert channel was implemented by Carrara *et al.*, who used speakers and microphones on computers to communicate through ultrasound [10]. Furthermore, Hanspach *et al.* leveraged ultrasound to establish covert acoustical mesh network [13]. Because ultrasound frequencies are higher than the upper audible limit of human hearing, communication cannot be easily noticed. Later, Guri *et al.* designed speakerless acoustic covert channels, where cooling fans [11], hard disk drives [12] and power supply unit [29] were used to generate acoustic emissions. However, the abnormal noise generated by fans and/or hard disk drives may be easily noticed by perceptive people, which makes them less stealthy. To some extent, acoustic signals can travel through obstacles, but their strength may be significantly attenuated depending on the material of the obstacles. Besides, most acoustic covert channels have very low bandwidth.

Optical emissions can also be exploited to create covert channels. The exploitable optical emissions may be generated by light-emitting diode (LED) in components like keyboards [5], monitors [6], and even hard disk drives [3]. Most LED-based optical covert channels use OOK modulation, and Zhou *et al.* showed that the efficiency could be improved by replacing OOK modulation with B-FSK modulation [30]. Another kind of optical covert channel manipulates the monitor screen [31, 32]. By modifying a small amount of content displayed on the screen, information may be transmitted without being noticed by humans. Theoretically, optical covert channels can reach a very high bandwidth with the help of optical instruments as long as the sender is in the sight of the attacker. However, exploiting optical emissions is harder than expected in practice, because it is rare that a highly secured target machine can be monitored by a malicious camera. In addition, it is very difficult, if not impossible, to create optical covert channels when the target machine is enclosed in a room with non-transparent walls. Similar to acoustic covert channels, some optical emissions like abnormal blinking of LED can also raise administrator's suspicion.

A thermal covert channel was constructed in [2] to transmit information between two physically adjacent but air-gapped computers. The advantage of this covert channel is that it can realize two-way communication. However, the performance of this covert channel is extremely poor. The maximum bandwidth reported is 8 bits/hour, and the sender and the receiver must be very close to each other. A similar thermal covert channel can be implemented by using a smartphone as a receiver to exfiltrate data from air-gapped computers [33]. This covert channel also has the problems of low bandwidth and short transmission distance.

Power consumption is also exploitable for establishing covert channel [34]. In this work, CPU was manipulated to affect the power consumption of a computer to transmit information through power lines. The receiver can be mounted either on the in-home power lines that are directly attached to the electrical outlet or on the main electrical service panel. The bandwidth of this covert channel can reach 1000 bps, but it requires the installation of malicious hardware devices on the power lines connected to victim machines.

Some recent works also showed the possibility of recovering the sound by detecting the air vibrations using components like light sensors [35] and hard drives [36]. However, they are out of the scope of this paper because we mainly focus on the physical covert channels that exfiltrate data from computers using physical side-channel effects.

A recent study presented an EM covert channel [37] exploiting EM emanations from power management unit capable of exfiltrating data at the bandwidth of 4000 bps. This is the fastest EM covert channel prior to our work. *EMloRA* is a study parallel to our work [38]. Both *BitJabber* and *EMLoRa* are featured by explicitly addressing spread spectrum clocking to improved communication performance. However, *EMLoRa* focuses on ultra-long-distance data transmission with lower bandwidth while *BitJabber* aims at achieving extremely high-speed data exfiltration.

2.2 Comparisons

To highlight the advantages of *BitJabber*, we compare the existing physical covert channels in Table 1. The comparisons are made in terms of their wall-penetrating ability, maximum achievable bandwidth, error rate of this bandwidth and maximum attacking distance. From the table we can see, before our work, the fastest physical covert channel was the one proposed in [3], which can achieve 4000 bps. Compared to that covert channel, our *BitJabber* improves the performance by 75x.

Moreover, most of the existing physical covert channels have difficulties in penetrating physical obstacles like a wall. (We mark “maybe” on acoustic covert channels in terms of wall-penetrating ability, although we think it is very unlikely that they can actually penetrate a wall.) From the table, we can observe that the EM covert channels have considerable advantages over others in terms of penetrating walls. However, as illustrated in Sect. 5, when penetrating concrete walls, approaches like *GSMem* actually have a too large error rate (from 38% to 50%) to be actually used in reality, while our *BitJabber* has an error rate even less than 0.5%. Therefore, compared to other physical covert channels, it can

Table 1 Comparison of existing physical covert channels

Covert Channel	Type	Wall-Penetrating	Bandwidth	Error Rate	Distance
GPU Clock [28]	Electromagnetic	Yes	N/A	N/A [a]	< 1 m
BitWhisper [2]	Thermal	No	0.002 bps	0 [b]	0.4 m
HOTSPOT [33]	Thermal	No	0.03 bps	N/A [a]	0.5 m
Fansmitter [11]	Acoustic	Maybe	0.25 bps	0 [b]	8 m
Hard Drive [9]	Magnetic	No	2 bps	3%	0.3 m
DiskFiltration [12]	Acoustic	Maybe	3 bps	0 [b]	2 m
MAGNETO [7]	Magnetic	No	5 bps	0 [b]	0.12 m
Screen Brightness [32]	Optical	No	10 bps	0	9 m
EMLoRA [38]	Electromagnetic	Yes	14 bps	0	250 m
Monitor LED [6]	Optical	No	20 bps	N/A [a]	N/A
ODINI [8]	Magnetic	No	40 bps	30%	1.5 m
POWER-SUPPLaY [29]	Acoustic	Maybe	50 bps	0 [b]	5 m
UltraSonic [10]	Acoustic	Maybe	230 bps	2%	11 m
Keyboard LED [5]	Optical	No	450 bps	N/A [a]	20 m
AirHopper [14]	Electromagnetic	Yes	480 bps	0.12%	22.2 m
USBee [16]	Electromagnetic	Yes	640 bps	N/A [a]	1 m
GSMem [15]	Electromagnetic	Yes	1000 bps	0.087%	5.5 m
PowerHammer [34]	Power	N/A	1000 bps	0	N/A
Hard Drive LED [3]	Optical	No	4000 bps	0 [b]	20 m
PMU [37]	Electromagnetic	Yes	4000 bps	3%	2.5 m
BitJabber	Electromagnetic	Yes	300,000 bps	0.1%	6 m

[a] No error rate was reported

[b] Error rate obtained after using the error correction code

be found that our *BitJabber* imposes more realistic security risks on air-gapped isolation protection.

Note that some existing works did not explicitly provide error rates and distances. Some numbers are estimated according to their descriptions of the experimental setups and evaluation results for these works.

3 Attack Model

Similar to the previous work [2–16, 34], in this paper, we explore how to construct a covert communication channel between a pair of air-gapped sender and receiver. We assume that the sender has been placed on the victim computer that stores or processes the secret data of interest, and the sender can acquire the secret through techniques like microarchitectural side-channels [39]. (How to place the sender there is out of scope, but, as presumed in the previous work, the attacker is capable of achieving this by methods like social engineering, USB interface, or physical access.) Note that we do not assume the sender has any privilege higher than the regular user level.

We assume that the attacker can use a radio frequency (RF) receiver (like a cheap software-defined radio) to collect the EM signals emanated from the victim machine somewhere nearby. Note that we do not require the receiving device to share the same room with the sender or to be physically adjacent to the sender. The sender and the receiver may be in different rooms with concrete walls, and the straight-line distance between them can be several meters.

All computers using standard DDR DRAMs can be susceptible to this attack. Although we mainly focus on desktop computers in this paper, other computer systems like servers, gaming laptops, and some all-in-one computers generate equally strong EM emanations that can be exploited to launch similar attacks. However, devices including most embedded systems, mobile phones, tablets, and ultrabooks like the MacBook are less vulnerable. These devices typically use low-power DRAMs and have compact designs with better shielding techniques, thus leading to weaker EM emanations.

4 The Design of *BitJabber* Covert Channel

As mentioned above, our *BitJabber* is an EM-based covert channel. The carrier EM signal is generated by the DRAM clock, and memory accesses are used to modulate the carrier signal to encode information. When modulated carrier signal is captured, demodulation is used to decode information from that signal. The overview of our *BitJabber* covert channel is illustrated in Fig. 1. In the following, we will describe the main components and techniques used in *BitJabber*.

4.1 Spread Spectrum Clocking

Before going forward to describe the details of our *BitJabber* covert channel, we need to present a challenging problem caused by a feature named *spread spectrum clocking* (SSC). SSC has been widely used in electronic products like computers for meeting electromagnetic compatibility (EMC) regulations [40]. Due to SSC, the energy of the EM signals generated by the DRAM clock will be spread over a wide range of frequencies. Such an energy dispersion makes the exploitation of these EM signals much harder, because the power of the exploitable signals becomes weaker but the power of the background noise stays the same. As a result, the SNR is much decreased, and thus our covert channel capacity will be considerably affected. To increase the SNR, we need to use a de-spreading technique to gather the scattered signal energy back.

Fortunately, this problem has been solved recently [41]. For self-containedness, we will summarize the solution here. The detailed presentation can be found in [41].

Given a clock signal whose frequency is f_c, SSC uses FM-modulation to vary the clock frequency in accordance with a signal $f_m(t)$ that is generated in the SSC hardware chip but undocumented. Normally, $f_m(t)$ is a periodic function, namely we have $f_m(t) = f_m(t + T_m)$ where T_m is the fundamental period of $f_m(t)$. At time t, the instantaneous frequency $f_i(t)$ of the clock signal becomes:

$$f_i(t) = f_c + Af_m(t), \tag{1}$$

where A is some proportionality constant. In an analytic form, the effect of SSC is equivalent to multiplying the clock signal by a complex exponential function $\theta(t)$, which is defined as:

$$\theta(t) = e^{j2\pi \int_0^t Af_m(t)dt}, \tag{2}$$

where j denotes $\sqrt{-1}$. Hence, if $\theta(t)$ can be estimated, for the purpose of de-spreading, we just need to multiply the measured signal by $\theta^{-1}(t)$. However, $f_m(t)$ is an undocumented non-linear function and thus it is difficult to derive $\theta(t)$. In [41], we have developed a technique that can effectively counter the scattering effect of SSC. Our de-spreading technique can be summarized in the following steps:

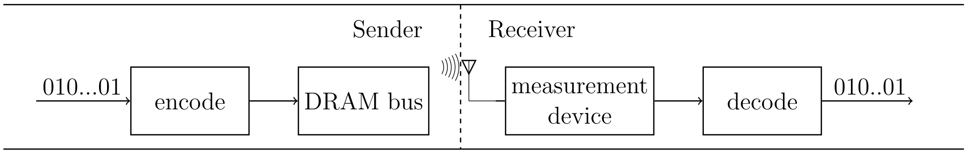


Fig. 1 Overview of *BitJabber* cover channel

1. Collect a sequence of samples at the DRAM clock frequency f_c when no user processes are running.
2. Compute a phase difference sequence, i.e., $\delta_l = \phi_{l+1} - \phi_l$, where ϕ_l is the phase angle of the lth sample.
3. Find the fundamental period T_m of $f_m(t)$ by observing the repeated patterns in the phase difference sequence.
4. Derive a smaller sequence $\Delta = \{\delta_1, \delta_2, ...\}$ over one T_m. (To reduce noise, average multiple ones to obtain Δ.)
5. Align Δ with the targeted EM signals by cross-correlation.
6. Multiply each sample by $e^{-j\Phi}$, where Φ is the running sum of the aligned elements in Δ, to perform de-spreading.

De-spreading can significantly improve the capacity of our covert channel in several ways. First, de-spreading gathers the scattered energy of the exploitable EM signals (i.e., it helps strengthen the signal), while de-spreading also inadvertently acts like SSC on background noise (i.e., it helps weaken the noise). Thus, the SNR will be greatly increased. Second, the EM signals of interest will be located in a narrow frequency range after de-spreading, which allows us to use more advanced modulation techniques to utilize the spectra.

4.2 Modulation

To encode information into the EM signals generated by the DRAM clock, modulation is required to vary the EM wave with respect to the message contents. As it is known that the EM radiation of the DRAM clock is AM-modulated by memory accesses, the modulation for *BitJabber* covert channel is accomplished through manipulating the memory access behavior.

To understand how memory access behaviors affect the EM signals generated by the DRAM clock, we perform different memory activities on a computer equipped with DDR3-1600 memory modules (i.e., the DRAM clock frequency is 800 MHz) and investigate the corresponding spectra, which are shown in Fig. 2. At first, no intense memory accesses are performed. As illustrated in Fig. 2, the EM radiation after de-spreading has most of its energy concentrated near the clock frequency (i.e., 800 MHz). When memory accesses with execution time around 350 ns are repeatedly performed, raised energy can be observed at certain frequencies in the lower and upper sidebands. The offsets of these lobes from 800 MHz are multiples of the memory access frequencies (i.e., 2.86 MHz), which indicates that the EM radiation is AM-modulated by a non-sinusoidal wave with the same frequency as the memory accesses. If some delay is added to make the memory accesses slower, the positions where the lobes locate indicate that the frequency of the modulating non-sinusoidal wave also decreases. (Note that we use non-temporal load/store instructions like `MOVNTI` to avoid memory accesses being served directly from the CPU caches.)

The above observation shows that not only do intense memory accesses introduce obvious lobes in the sidebands, but also the memory access frequency has influence on where these lobes locate. Accordingly, two modulation techniques can be applied to encode information into the EM signals generated by the DRAM clock:

- The first and also the simplest modulation method is OOK. As shown in Fig. 3(a), OOK uses the presence and absence of repeated memory accesses to encode bit “1” and bit “0”. Consequently, the AM-modulated EM

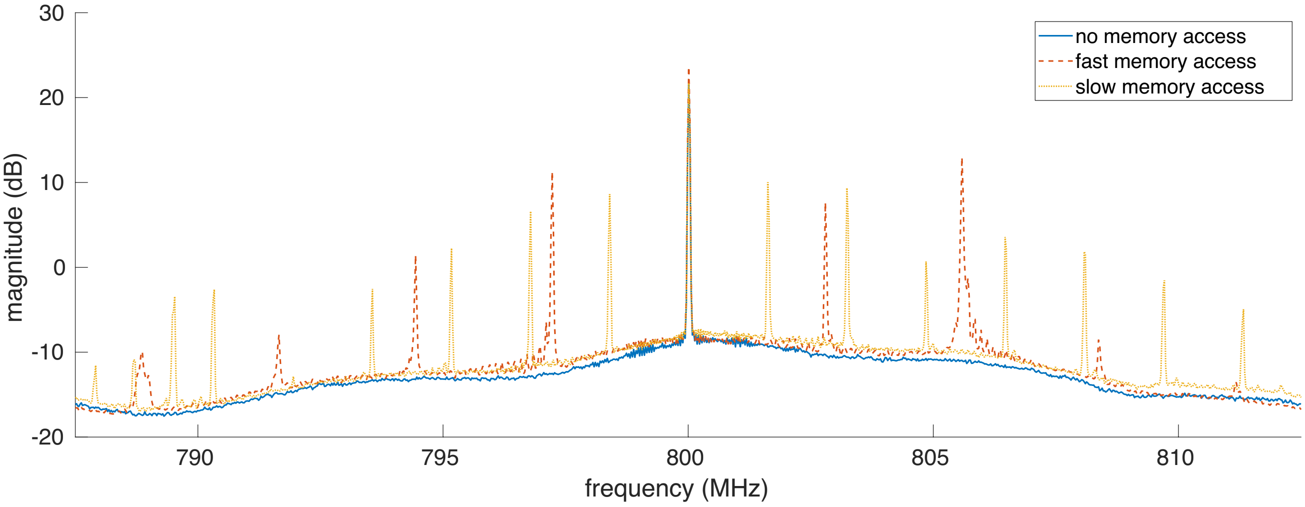


Fig. 2 Spectra of different memory access behaviors

signal will have side lobes in its spectrum only when “1” is transmitted; otherwise, “0” is sent.

- The other modulation method is FSK, indicated by Fig. 3(b), where different symbols are represented by different memory access frequencies. For example, to send bit “1”, fast memory accesses are repeated, and to send bit “0”, slow accesses are repeatedly made. Thus, different distances between the side lobes and the clock frequency in the spectra can distinguish these two cases. To realize different memory access frequencies, we can use a normal memory access as the fast one and introduce some delay to derive the slow one.

Note that the above-mentioned FSK modulation is not limited to B-FSK, in which case either bit “0” or “1” is transmitted. Because any two different memory access frequencies can result in distinguishable side lobe positions in the spectra, M-FSK modulation is also achievable by adding distinct delays to a base memory access activity `BaseMemAcc` as depicted in Algorithm 1. (The details of the `BaseMemAcc` activity will be described later.)

Algorithm 1 Memory activities for M-FSK modulation

```
Input: $T_i$ = delay time for transmitting symbol $i$
if Transmitting symbol $i$ then
    BaseMemAcc
    DELAY($T_i$)
end if
```

4.3 Base Memory Access Design

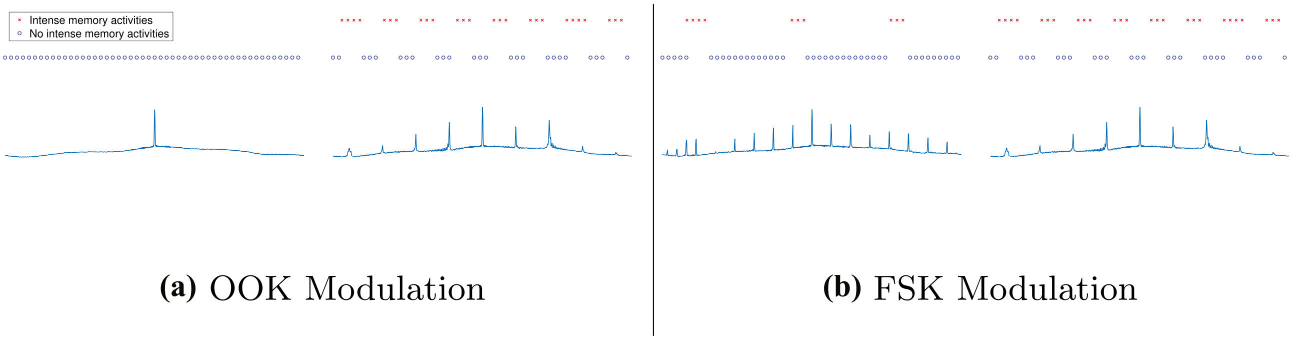
We have observed that randomly accessing some memory addresses may not AM-modulate the EM signals generated by the DRAM clock well. Thus, we need to have a systematic way to construct a memory access activity such that the probability of AM-modulating the EM signals of interest well is very high. We term this systematically constructed memory access activity as base memory access `BaseMemAcc`.

We need `BaseMemAcc` to have the following three properties:

1. It should have a very short execution time (e.g., a few hundreds of nanoseconds).
2. It should have a relatively stable execution time.
3. It should induce obvious change in the amplitude of the EM signals generated by the DRAM clock.

To design such a base memory access activity, we need to understand how memory accesses affect the DRAM clock. Although it has been investigated in some prior work [15, 18], factors that influence the AM-modulation effect were not fully identified.

To satisfy the first two properties, we decide to use non-temporal memory access instructions, such as `MOVNTI`, `MOVNTDQ` and `VMOVNTDQ`. Since they will bypass the CPU caches, we can use them to directly access the main memory in a rapid manner. Otherwise, `CLFLUSH` instruction needs to be used to flush the cache after each memory access, which brings in more overhead and execution time variation. These non-temporal memory access instructions can support operands of different sizes, e.g., either 32-bit or 64-bit operands can be used in `MOVNTI`. The operand size can affect the execution time slightly, but can result in observable differences in side lobe positions in the spectrum.



(a) OOK Modulation

(b) FSK Modulation

Fig. 3 Encoding of 0 and 1 using two modulation methods — OOK and FSK

We notice that memory locations may have a significant influence on the AM-modulation. In order to find out how the AM-modulation effect is related to the memory access instructions and memory locations, we conduct experiments and empirically conclude the following:

1. When the same memory access instruction is used to access the same memory location, the AM-modulation effect (e.g., side lobe positions and their energy) is fixed.
2. When different types of non-temporal instructions are used to access the same memory location, the AM-modulation effect is slightly different.
3. When the same instruction is used to access different memory locations, the amount of amplitude change of the EM signals of interest may be significantly different. The relationship between accessed address and the amount of amplitude change is still not clear, but in our tested platforms we notice that accessing memory addresses in the same DRAM bank tends to change the amplitude similarly.

Algorithm 2 Grouping virtual addresses *w.r.t.* banks

```
Input: $AP$ = address pool
Output: $G$ = addresses mapped to the same bank
RefAddr $\leftarrow$ $AP$.DEQUEUE()
$G$.ENQUEUE(RefAddr)
$n$ $\leftarrow$SIZEOF($AP$)
for $i \leftarrow 1...n$ do
    RemAddr $\leftarrow$ $AP$.DEQUEUE()
    if LATENCY(RemAddr,RefAddr) then
        $G$.ENQUEUE(RemAddr)
    else
        $AP$.ENQUEUE(RemAddr)
    end if
end for
```

Because an unprivileged user does not have enough knowledge of the physical memory address information, the AM-modulation effect of accessing a random memory location is unpredictable. Therefore, the more memory locations are accessed, the higher the possibility that obvious amplitude change will arise is. Based on the above observations, to satisfy the third property, `BaseMemAcc` needs to access several fixed memory locations using the same non-temporal memory access instruction. Apparently, there is a trade-off, because the more memory locations are accessed, the slower `BaseMemAcc` will become. We empirically find that accessing 4 memory locations is sufficient to have obvious AM-modulation effect while keeping the execution time short.

Note that if these fixed memory locations are randomly selected, it may incur unpredictable variations in the execution time due to row buffer conflicts in the same DRAM banks [42–44]. Such variations may make the second required property of `BaseMemAcc` violated. Therefore, it is preferable to have these memory locations in different DRAM banks. Moreover, considering that in some platforms the amplitude change is bank-dependent, this memory location selection strategy can even help `BaseMemAcc` hold the third property. Thus, we design `BaseMemAcc` to be a memory access activity that uses a fixed non-temporal memory access instruction to access 4 fixed memory locations in different DRAM banks.

However, finding memory locations belonging to different DRAM banks can be a problem, because the address mapping information is unavailable to unprivileged attackers. To obtain such memory locations, we use a method exploiting a timing side-channel introduced by the row buffer conflicts in the same DRAM banks [42–44]. Given two virtual addresses a_1, a_2, a function `LATENCY`(a_1, a_2) is used to check whether they are in the same bank. If they are in the same bank, accessing them consecutively is relatively slow due to the delay induced by the row buffer conflict, and `LATENCY`(a_1, a_2) returns `True`; otherwise, accessing them is faster and `LATENCY`(a_1, a_2) returns `False`. The memory location selection method is described in Algorithm 2. By repeating this method, we can derive several groups, in each of which the addresses are located in the same DRAM bank within a few seconds.

4.4 Communication Protocol and Demodulation

As indicated in Sect. 4.2, modulated signals with varied energy distribution on frequency domain are transmitted to send symbols of different values. On the receiver's side, after the EM signals are captured, demodulation is a necessary step to recover the encoded information from the modulated signals. In order to demodulate the received signals correctly, three problems need to be tackled:

1. How can we extract the features to distinguish different transmitted signals?
2. How is the receiver synchronized with the sender?
3. How can the receiver map the extracted features to correct symbol values?

In this section, we will describe the feature extraction method and communication protocol implemented to handle these problems.

4.4.1 Feature Extraction

For our *BitJabber* covert channel, the key problem of demodulation is to classify different symbol values according to the signal's energy distribution in the frequency domain. As shown in Fig. 2, when memory accesses are performed at a fixed frequency to transmit a symbol value corresponding to that frequency, side lobes appear at the first few harmonics of that frequency. Instinctively, features corresponding to these side lobes should be extracted.

To better describe the feature extraction process, we will use an example in which B-bit FSK modulation is employed. In this case, S possible symbol values may be sent, where $S = 2^B$. We assume the clock frequency is f_c, memory access frequencies $f_0, f_1, ..., f_{S-1}$, are used for encoding S different symbol values. M symbols are transmitted with a known symbol rate R_{symbol} and the EM signal of interest is sampled with a known sampling rate R_{sample}. The steps of feature extraction are as follows:

1. Find all the frequencies where side lobes locate in the spectrum of the captured EM signal (which is a sequence of sample size of sampled values). In our example, for each symbol value $s \in \{0, ..., S-1\}$, let us assume there are $2K_s$ lobes at $f_c \pm k_s f_s$ where $1 \leq k_s \leq K_s$.
2. For each frequency where a side lobe locates, apply a bandpass filter on the original signals and extract the envelope of filtered signals to preserve only the energy of that frequency. For our example, we can obtain K filtered signals, where $K = \sum_{s=0}^{S-1} 2K_s$. Hence, the captured signals are converted to a sequence of K-dimension vectors $\mathbf{v}$.
3. Segment the vector series $\mathbf{v}$ using the boundary finding technique described in Sect. 4.4.3. The length of each segment L is:

$$L = \frac{R_{\text{sample}}}{R_{\text{symbol}}} \tag{3}$$

(Note that we choose R_{sample} divisible by R_{symbol} by design, so L is an integer.)

4. Average all the values within each segment. Assume the segment head found for symbol m is sample n, the correspoinding feature vector $\mathbf{V}_m$ is computed using:

$$\mathbf{V}_m = \frac{1}{L}\sum_{l=0}^{L-1} \mathbf{v}_{n+l} \tag{4}$$

After this step, M K-dimension feature vectors are derived for all symbols.

4.4.2 Message Structure

To implement our *BitJabber*, we structure the message Q as shown in Fig. 4. It consists of a header $\{Q_0^H, ..., Q_{M_h-1}^H\}$ and its payload$\{Q_0^P, ..., Q_{M_p-1}^P\}$. The header is a pseudo random number sequence whose seed is shared by the sender and receiver, which is used for signal synchronization and deriving symbol mapping.

4.4.3 Finding Segment Boundaries

Successful synchronization is the prerequisite of demodulation, which guarantees that the feature vectors in Eq. 4 is computed at the right position, i.e., the correct pair of (m, n) are found. Assume in a symbol sequence, we know segment for symbol m_0 starts at sample n_0, the segment for the next symbol $m_0 + 1$ will start at sample $n_0 + L$. Because the sender and receiver are driven by different clocks, there exists inevitable clock drift δ. Although in reality δ is very small (e.g., around 0.0001%), the accumulated error can reach a level such that a compensation in the symbol length is needed. Therefore, the symbol m will actually start at sample n expressed as:

$$n = n_0 + (m - m_0) \times L + \lfloor (m - m_0) \times \delta \times L \rfloor \tag{5}$$

If we make $m_0 = 0$, symbol m starts at:

$$n = \lfloor m \times L \times (1 + \delta) \rfloor + n_0 \tag{6}$$

Thus, finding segment boundaries means finding the values of δ and n_0, which can be accomplished through performing linear fit on some known pairs of (m, n). Such pairs

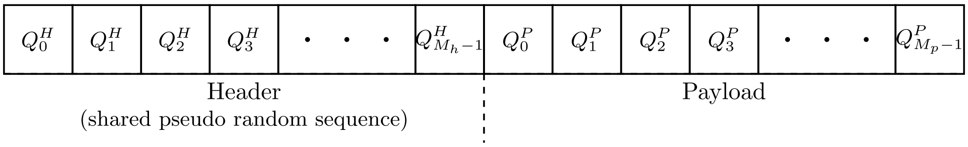


Fig. 4 Message structure

can be found within the header with it being shared knowledge between the sender and receiver.

To find the correct segment head n_r for a symbol m_r in header, we use the following steps:

1. With a guessed segment head $n_r = n'$, we can obtain a feature vector $\mathbf{V}^H_{m_r}(n')$ using Eq. 4.
2. Find an integer Δm such that $2\Delta m + 1$ is a large enough sample size for performing statistical analysis while satisfying $\Delta m\delta \ll 1$. In reality, Δm is usually a number ranging from 100 to 1000.
3. With a guessed segment head n' and a properly chosen Δm, segment heads for symbols $\{m_r - \Delta m, ..., m_r + \Delta m\}$ can be estimated as $\{n' - L\Delta m, ..., n' + L\Delta m\}$. Subsequently, a sequence of feature vectors $\{\mathbf{V}^H_{m_r-\Delta m}(n'), ..., \mathbf{V}^H_{m_r+\Delta m}(n')\}$ can be obtained.
4. For each possible symbol value s, these feature vectors can be split into two groups according to the value of Q^H_m, which gives us $\mathbf{V}^H_s(n') = \{\mathbf{V}^H_m(n') \mid Q^H_m = s\}$ and $\mathbf{V}^H_{\sim s}(n') = \{\mathbf{V}^H_m(n') \mid Q^H_m \neq s\}$.
5. With this splitting, we take advantage of the fact that *with the correct segmentation, feature values in each dimension of the feature vectors will have the minimum standard deviations within the same group and the maximum differences between different groups*. Intuitively, to measure the segmentation quality for each symbol value s w.r.t each feature dimension k, we can define a score T as follows:

$$T(n', s, k) = \ln\left(\frac{\sum_{m_i,m_j} \left(V^H_{s,k,m_i}(n') - V^H_{\sim s,k,m_j}(n') \right)^2}{\mid V^H_{s,k}(n') \mid \times \mid V^H_{\sim s,k}(n') \mid \times \sigma\left[V^H_{s,k}(n') \right] \times \sigma\left[V^H_{\sim s,k}(n') \right]} \right) \tag{7}$$

   Then the actual n_r can be found through searching for the maximum score.

$$n_r = \arg\max_{n'} \sum_{s,k} T(n', s, k) \tag{8}$$

After enough pairs of symbol and segment heads are identified in the header, we can fit Eq. 6 to obtain δ and n_0 for the header. With this knowledge, segment heads n of a symbol m in the payload can be computed using:

$$n = \lfloor (m + M_h) \times L \times (1 + \delta) \rfloor + n_0 \tag{9}$$

4.4.4 Payload Decoding

After successful synchronization, we can correctly compute the feature vectors for all symbol transmitted in the message. The last step of demodulation is mapping these feature vectors to the correct symbol values. As the header is a symbol sequence shared by the sender and receiver, the feature vectors obtained from the header can be used to train a simple classifier. This classifier is then used to translate the feature vectors in payload part to symbol values. Because the feature vectors to be recognized are simple, any classification technique can be used. In our case, we find the performance of SVM (support vector machine) to be satisfactory as demonstrated in Sect. 5.

5 Experimental Results

In this section, we will evaluate the performance of our *BitJabber* covert channel in terms of its bandwidth, error rate, and capability of wall-penetrating. In the evaluations, we also compare our *BitJabber* with the existing *GSMem* covert channel [15] for the following two reasons:

1. The performance of covert channels depends on many factors like background noise and the physical architecture of the sender machine.
2. Both *BitJabber* and *GSMem* covert channels use the EM emanations generated from the DRAM clock.

5.1 Experimental Setup

The performance of *BitJabber* and *GSMem* covert channels are evaluated on three different platforms listed in Table 2. These platforms use different motherboards and DRAMs of multiple frequencies. On all platforms, two DIMMs are installed on two DRAM channels.

The receiver uses a log-periodic (LP) antenna, a telescope antenna and a software-defined radio (SDR) platform LimeSDR-USB development board to collect the EM signals around the DRAM clock frequency as shown in the left part of Fig. 5. The EM signals are preprocessed using the GNU Radio.

The experiments are performed in a typical office environment. In such an environment, much background noise exists, including EM waves radiated from wireless communication systems (e.g., radio stations and cell towers), nearby electronic devices, and other components in the victim computers.

Table 2 Platforms on which our covert channel is evaluated

Platform	Motherboard	Memory	Case Material
A	Dell Optiplex 990	2 × 4GB DDR3-1333	Metal
B	Dell Optiplex 3020	2 × 4GB DDR3-1600	Metal
C	Asus PRIME Z270-P	2 × 8GB DDR4-2400	Metal & Plastic

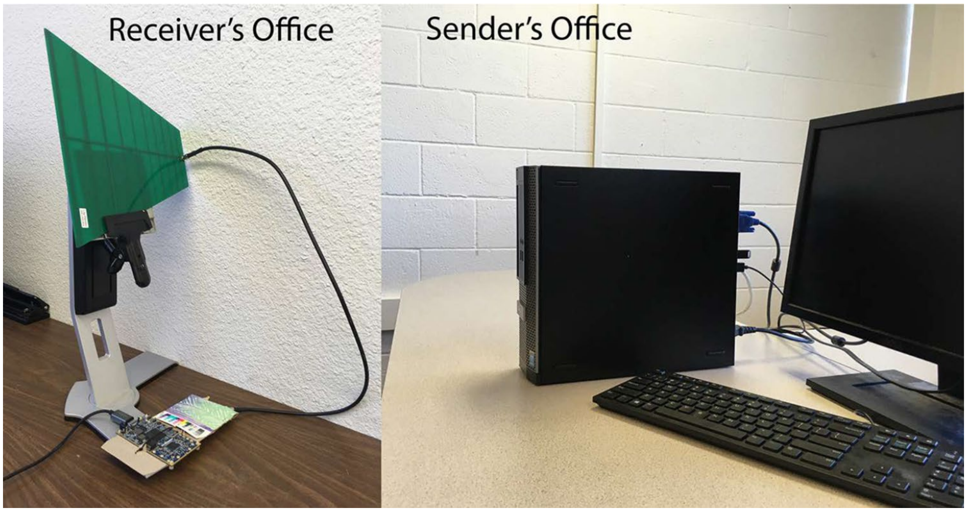


Fig. 5 Experimental setup for wall-penetrating performance evaluation

The experiments are performed in three different scenarios. First, the antenna is put close to the victim machine to receive the strongest EM emanations from the DRAM clock. This experiment will show the performance upper bound of different approaches. The second scenario is to experiment with a more practical setting, as shown in Fig. 5, where the sender and receiver are located in two different offices sharing a 15-cm-thick wall. This experiment compares the wall-penetrating data exfiltration capability of the covert channels. Additionally, we evaluate the performance of *BitJabber* with the antenna and victim machine separated by different distances, with no obstacles in between. This experiment evaluates *BitJabber*'s long-distance data exfiltration ability.

Our paper focuses on high-speed data exfiltration only, so the lowest evaluated symbol rate is 1000 Bd. Nevertheless, in all evaluations, we use a 20 Bd symbol rate sequence to speed up the process of locating the beginning of signals (which is optional for the implementation). These 20 Bd patterns are very visually perceptible in all measurements, i.e., all evaluations in this section have zero error rates at 20 Bd symbol rate.

5.2 Symbol Distinguishability

For all covert channels exploiting physical side-channel effects, the receiver measures certain physical changes introduced by senders and transforms the measurements into different symbols. A good covert channel should have good symbol distinguishabilities. In Fig. 6, we compare the symbol distinguishabilities of two covert channels *GSMem* and *BitJabber* using the B-FSK modulation.

For transmitting binary symbols, we can use a single feature value to represent how likely a measurement is identified to a certain symbol (either "0" or "1"). In *GSMem*, only the magnitude of the EM signal is used for distinguishing symbols with binary values, and thus we can use this as the feature value. For *BitJabber* using B-FSK modulation, an SVM model is trained to distinguish the feature vectors, and thus we use the difference of two prediction scores as the feature value. The feature values of *GSMem* at 1000 Bd symbol rate and *BitJabber* using the B-FSK modulation at 100,000 Bd symbol rate are illustrated in Fig. 6. Compared to *GSMem*, it is

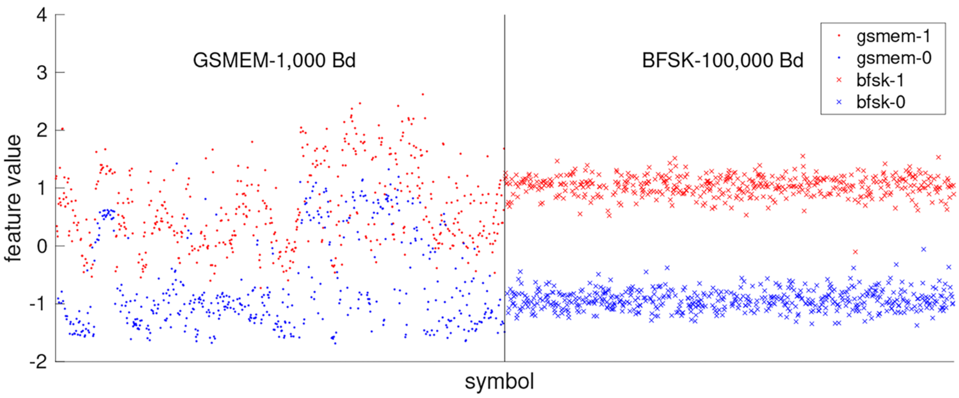


Fig. 6 Symbol distinguishability of *GSMem* and *BitJabber* using the B-FSK modulation

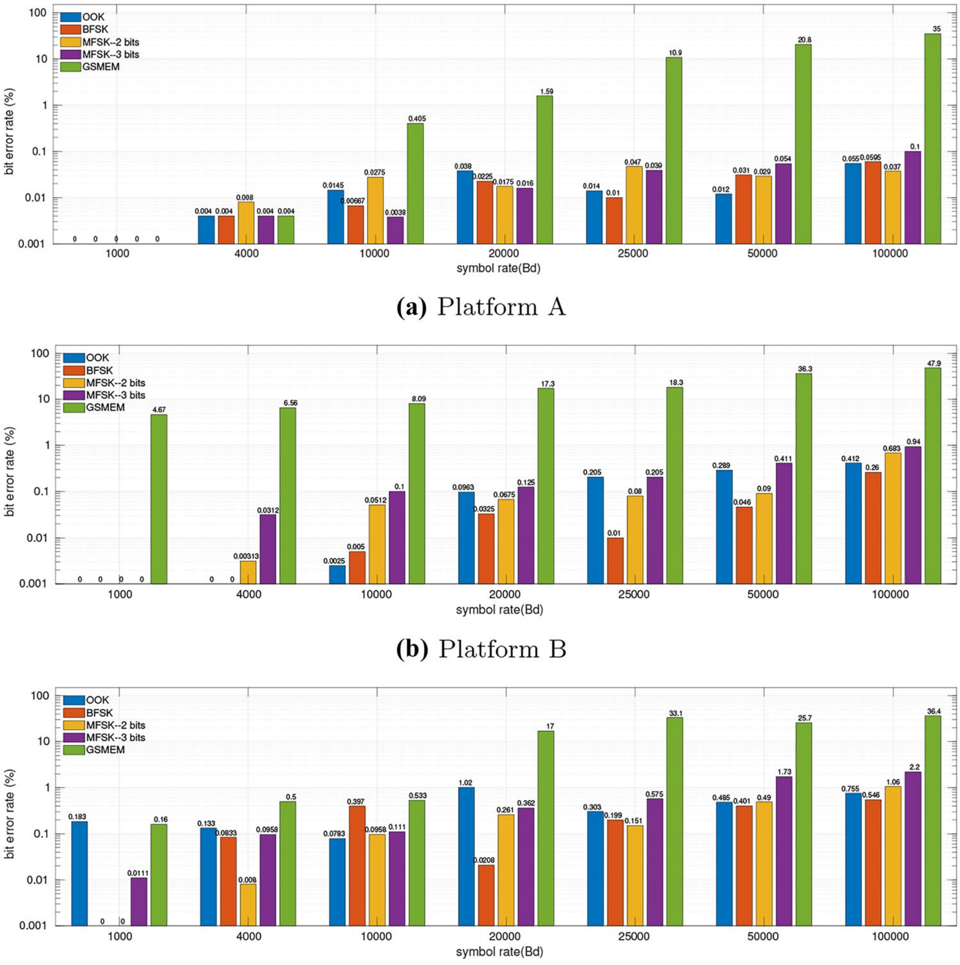
apparent that the measurements of *BitJabber* have much larger difference between different symbol values and smaller variances between same symbol values even if the symbol rate is 100 times higher. This comparison indicates that our *BitJabber* can greatly outperform the *GSMem*, which is demonstrated by the following experimental results.

5.3 Bandwidth Evaluation

The first group of experiments measure the maximum bandwidth of *GSMem* and our *BitJabber*. To measure the performance upper bound, all measurements are performed with the antenna set at a fixed position, at which the strongest EM emanations from the DRAM clock can be collected. The EM signals are modulated by the OOK, B-FSK, and M-FSK modulation methods. Examined symbol rates range from 1000 Bd to 100,000 Bd and the evaluation results are shown in Fig. 7. Because of the huge performance difference between *GSMem* and our *BitJabber*, **logarithmic scale** is used in this plot. Note that the original *GSMem* uses the EM signals at only 800 MHz. To make a fair comparison, here we evaluate *GSMem* using the EM signals at the frequencies of DRAM clocks, where memory behaviors cause the maximum amplitude changes.

In our evaluations, we limit the maximum symbol rate to 100,000 Bd and the maximum symbol length for M-FSK to 3 bits. Theoretically, larger symbol rate and symbol length can be used for this covert channel. Nevertheless, selection of these two parameters highly depends on the hardware device used to implement this covert channel. The `Base-MemAcc` used in victim computers typically takes several hundred nanoseconds to execute. If symbol rate higher than 100,000 Baud is used, the actual symbol duration tends to be more unstable, which will greatly increase the error rate. As for the symbol length, when 3 bits are represented by a single symbol, 8 different memory access frequencies are used and the resulting EM emanations almost affect the entire 25 MHz frequency range. If more bits are transmitted, frequency ranges affected during transmission of different symbol values may overlap too much and variance between different symbol values' feature vectors tends to be smaller,

Fig. 7 Bit error rate at different symbol rate for *GSMem* and *BitJabber* using different modulation methods



(a) Platform A

(b) Platform B

(c) Platform C

which will also increase the error rate. With more advanced SDR devices used to exfiltrate data from more powerful computers, *BitJabber* may be implemented at larger symbol rates and symbol lengths.

The evaluations on different platforms have slight differences but in general for different covert channels and modulation methods we can summarize that:

- For all evaluated approaches, the error rates increase as symbol rates get higher.
- The error rate reported in [15] at the symbol rate 1000 Bd is 0.087%. Our evaluations indicate that the performance of *GSMem* is highly dependent on the platforms where it is implemented and the error rates range from 0 to 4.67% at 1000 Bd symbol rate. Even though *GSMem* can be implemented with relatively low error rate in some platforms when the symbol rate is low, as the symbol rate increases, the error rates of *GSMem* implemented in all platforms become extremely high. Therefore, *GSMem* covert channel cannot be used to exfiltrate data in high bandwidth.
- When the OOK modulation is used in *BitJabber*, it has a low error rate which is close to 0 at low bandwidth. On most platforms, the error rates are also low when the bandwidth is 100,000 bps. Using the same OOK modulation, *BitJabber* outperforms *GSMem*.
- Most of the time, *BitJabber* implementing B-FSK modulation exfiltrate data with the lowest error rate among all evaluated approaches.
- Using the M-FSK modulation, *BitJabber* can transmit multiple bits with each symbol effectively, the error rate is very low at a low symbol rate. Under the same conditions, 2-bit M-FSK always results in lower error rate than 3-bit M-FSK.
- Considering that 3-bit M-FSK modulation can transmit 3 bits with each symbol, the fastest transmission can reach **300,000 bps**. Compared to *GSMem* at its fastest transmission rate (i.e., 1000 bit/sec), **BitJabber increases the bandwidth by 300 times even with significantly lower error rate**.

According to Fig. 7, evaluated covert channels' performances highly depend on the victim platforms. By comparing the evaluation results and features of EM signals, the intensities of emanated EM signals and background noise have significant impacts on evaluated covert channels, especially for *GSMem*. When the antenna is put close to the victim platforms, the received EM signals from computers all have high intensities, but background noise at different frequency ranges varies a lot. The strongest noise is observed around 800 MHz when platform B is evaluated and it is not emanated from the victim computer. Therefore, we can observe that on platform B, *GSMem* has the worst performance. As mentioned before, the despread technique used for implementing *BitJabber* enhances EM signal emanated from computers and suppresses the other irrelevant signals so the performance of *BitJabber* on platform B is not seriously affected. More detailed analysis of the factors influencing error rates will be given in Sect. 5.6.

5.4 Through-Wall Evaluations

Compared to the other covert channels, one advantage of EM covert channels is that EM signals can travel through many non-metal obstacles with little energy loss. In this experiment, *GSMem* and *BitJabber* are evaluated in a more practical scenario. The sender machine is put in an isolated room with a 15-cm-thick wall. The distance between the sender and the wall is 50 cm. The receiver is set in the next door sharing the same wall with the sender's room.

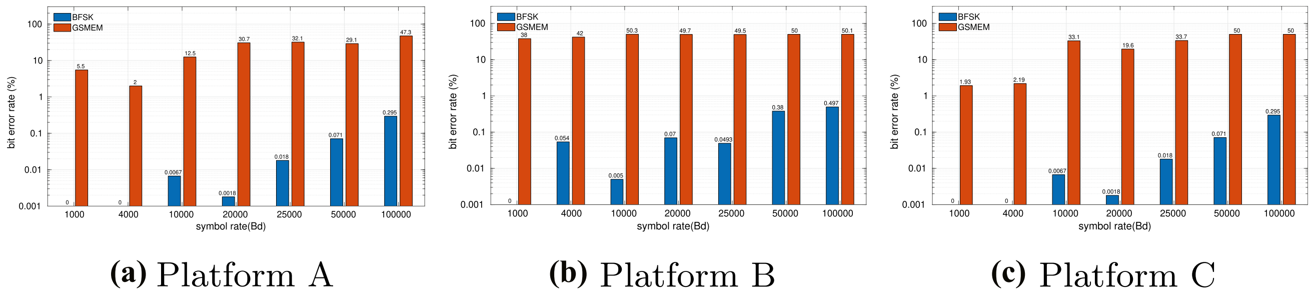
Similar to the previous evaluation, background noise exists in both rooms and there are even some wire cables with unknown layout in the wall. In this scenario, the received EM emanation generated by the DRAM clock is weaker and more noise is in the transmission process. Wall-penetrating performance of *GSMem* and our *BitJabber* using the B-FSK modulation are evaluated and the results are shown in Fig. 8. From the figures, we can conclude that:

- Compared to results in Fig. 7, performances of both covert channels get worse to some extent.
- *GSMem*'s performance is seriously affected and the error rates exceed 25% with symbol rate of 25,000 Bd on all platforms.
- Performance of our *BitJabber* using the B-FSK modulation is only slightly affected compared to *GSMem*.

During the evaluations, we found that the office wall has little influence of EM signal intensities but the distance between senders and receivers matters. Similar results to Fig. 8 can be obtained when the receivers and senders are separated in same distances but without being wall-gapped.

5.5 Attacking Distance Evaluations

Benefit from the stronger carrier signal, *BitJabber* can be used to perform long-distance data exfiltration. In our experimental environment, implementing *GSMem* at distances longer than 1 meter is very hard because the SNR gets too low to be exploitable. Therefore, in this section, we only measure the performance of *BitJabber* when the receiver is located at different distances away from the sender. In the experiments, *BitJabber* implementing B-FSK are evaluated at bandwidth range between 1000 bps and



(a) Platform A

(b) Platform B

(c) Platform C

Fig. 8 Bit error rate of *GSMem* and *BitJabber* using the B-FSK modulation measured with a wall between the receiver and sender

100,000 bps. For platforms A and B, the measured distances range from 0 m to the longest distance where the carrier signals are visible. For platform C, the longest measured distance is 6 m due to the space limitations of our experiment environment.

The experimental results are shown in Fig. 9

From the figures, we can observe that:

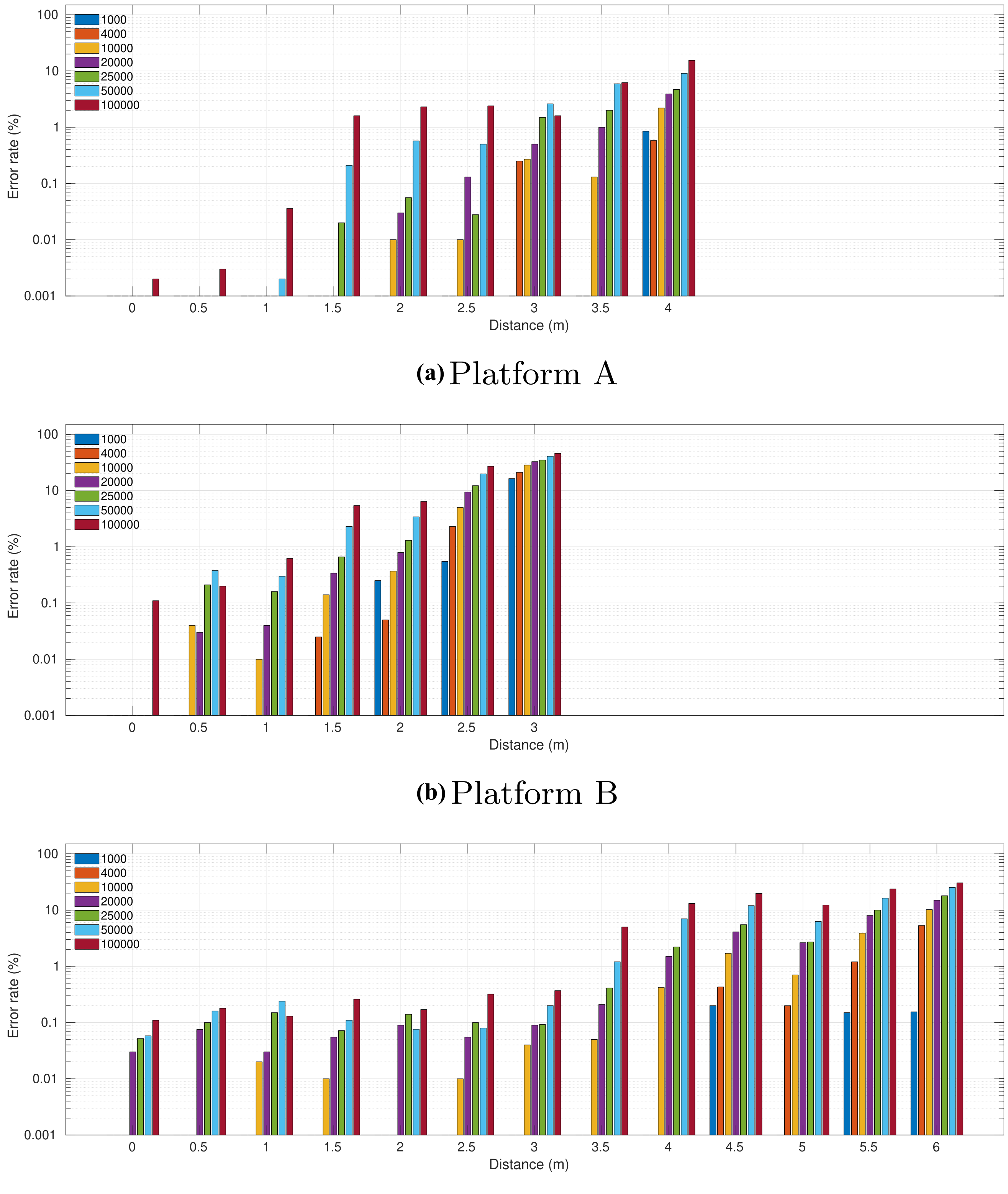
- The error rates are low for all methods with short attacking distances, and they increase as the attacking distances become longer.
- The correlation between error rates and distances gets strong when the error rate is high. While this correlation is weak when the error rate is low.
- Platform C is most vulnerable to *BitJabber* at long distances, and the error rate is around 0.1%, with the bandwidth being 1000 bps at 6 m away.
- *BitJabber* implemented on Platform C has the highest error rates when the attacking distance is short.

Besides the result presented in Fig. 9, one thing worth mentioning is that longer distance between senders and receivers not only reduces the EM signal intensities, but also increases the difficulty of setting antennas. The EM signals emanated from victim computers have different intensities in different directions so the location and orientation of antenna have large influence on the collected signals. When the receiver is moved away from the victim computers, the antenna locations receiving the strongest signals are harder to be determined, i.e., an attacker needs much more effort to receive the exfiltrated data. The difficulty of finding the best antenna locations is related to the computer case materials. When the receivers and senders are put more than 4 m away, we cannot find any antenna locations to collect exploitable EM signals from platforms A and B, but the EM emanations from platform C can still be easily observed. If we replace a metal plate of platform A's case with tempered glass, we can observe strong EM emanations even if the receiver is put more than 6 m away.

5.6 Error Analysis

We have observed that the error rate is influenced by factors including symbol rates and attacking distances from the above evaluations. Although we can conclude that the error rate generally increases with higher symbol rates and longer attacking distances, which also agrees with our intuition, we notice that some measurements do not strictly follow this relation. In order to better understand the threat posed by *BitJabber*, we try to find out all factors influencing the error rate using data collected at Sect. 5.5. Finally, we identify three types of errors. For each type of error, we select a symbol value sequence where that error occurs. We plot the symbol value sequences and spectrograms around the corresponding frequencies in Fig. 10

The first type of error is caused by low SNR, as shown in Fig. 10a. This sequence is collected when *BitJabber* implements 100,000 bps B-FSK modulation on platform C with the sender and receive located 3.5 m away from each other. As we can see from the spectrogram, the signal collected at this distance is very noisy. Two symbol values cannot be clearly distinguished by looking at the feature values due to the low SNR. This type of error can explain how the error rate is related to the symbol rate and attacking distance. Because the SNR decreases with increased symbol rate and attacking distance, we observe the inverse correlations between error rate and symbol rate and between error rate and attacking distance. Under the same conditions, signals emitted from platform C have the highest SNR, which makes it most vulnerable to long-range attack. However, when the SNR is high enough for distinguishing different symbols, higher SNR does not help further lower the error rate, which is why the above correlations get weaker with low error rates. As evaluations in previous sections indicated, error rates do not always reach zero with very short distances between the sender and the receiver when the SNRs are high enough for distinguishing two symbol values. The other two types of errors dominate in these cases.

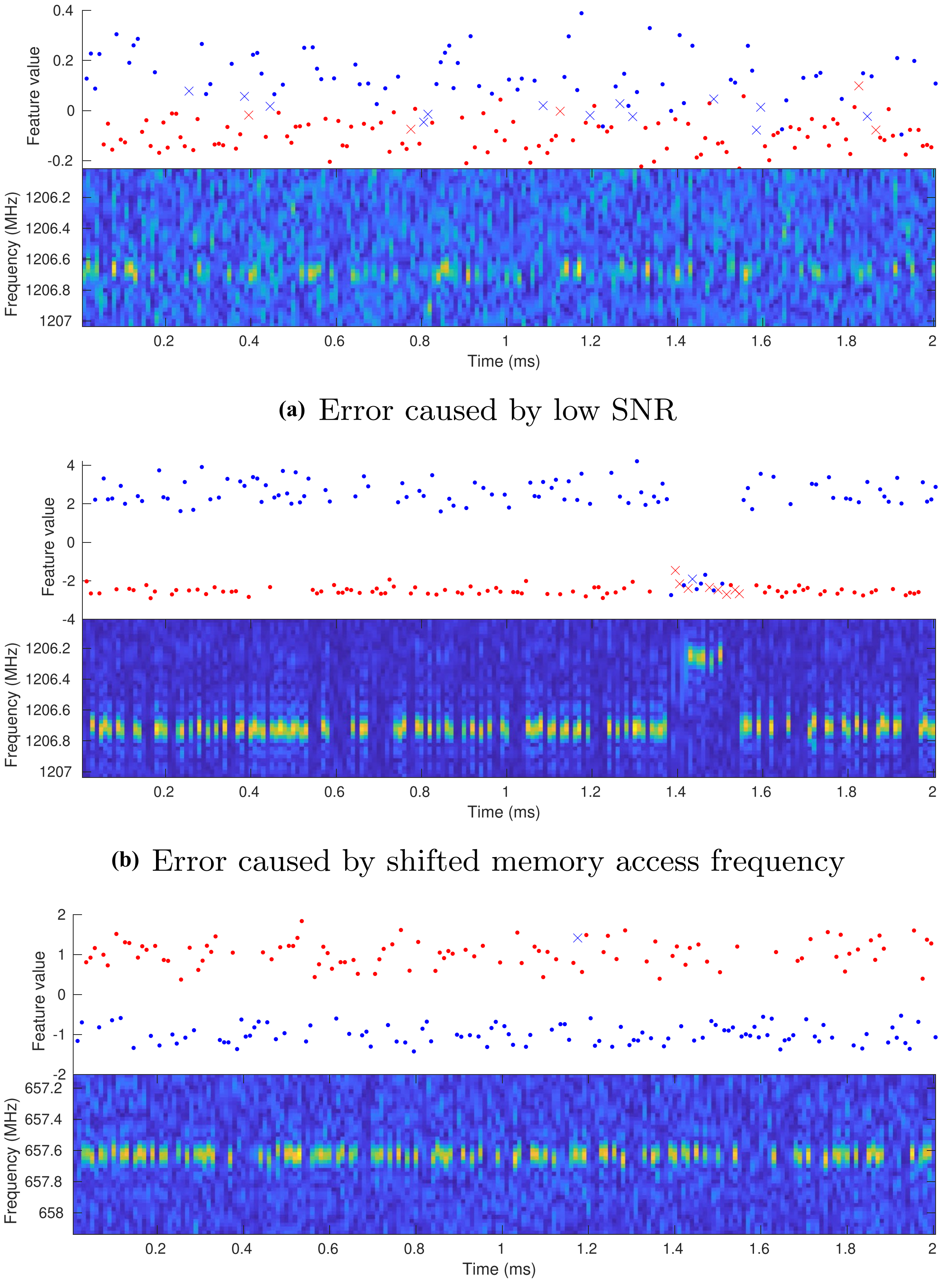


(a) Platform A

(b) Platform B

(c) Platform C

Fig. 9 Error rate measured at varied distances for BitJabber implementing B-FSK at different bandwidths



(a) Error caused by low SNR

(b) Error caused by shifted memory access frequency

(c) Error caused by sending non-updated symbols

◂**Fig. 10** Feature value sequences and the spectrograms around the corresponding frequencies when three different types of error occur. Color blue and red represent transmitted symbols with values '0' and '1' respectively. Dots (•) and crosses (×) denote correctly and wrongly classified symbols respectively

The second type of error is caused by shifted memory access frequency, as shown in Fig. 10b. This sequence is collected when *BitJabber* implements 100,000 bps B-FSK modulation on platform C with the sender and receive located very close to each other. As we can see from the spectrogram, the captured signal has a very high SNR. Nevertheless, we observe that the side lobe frequency is shifted by 0.5 MHz at 1.5 ms. Accordingly, the computed feature vectors are erroneous at this part and many errors occur. A reasonable guess is that such errors occur more frequently when more processes use the memory. In our experiments, this kind of unexpected frequency shift is most often observed on platform C but rarely on platforms A and B. This type of error explains why *BitJabber* implemented on platform C fails to reach a very low error rate despite the highest SNR.

The third type of error is introduced when the wrong symbol is transmitted by the sender, as shown in Fig. 10c. This sequence is collected when *BitJabber* implements 100,000 bps B-FSK modulation on platform A with the sender and receive located very close to each other. As we can see from the spectrogram, the received signal has a very high SNR with no unexpected frequency shift. The only misclassified symbol has a feature value entirely in accordance with the feature values of the other symbol value. When the sender transmits data at a very high speed, it may fail to update the data to transmit in time (e.g., due to task scheduling). Subsequently, the sender modulates the carrier based on the incorrect data and transmits the wrong information. All our evaluations are performed with sender running on relatively idle computers, in which case this type of error appears occasionally on platforms A and B but rarely on platform C.

In conclusion, the first type of error is related to the signal quality received by the receiver. Both the second and third types of errors depend on the working states of senders on victim machines. These error analysis results offer us deeper insight into countermeasures of *BitJabber*.

6 Countermeasures

Based on the error analysis results in Sect. 5.6, we propose countermeasures against *BitJabber* in several different directions.

One direction of countermeasures aims at lowering the SNR of received signals to increase the first type of error. This can be achieved either by reducing the EM emanation intensity or increasing the background noise. EM shielding is a commonly used technique to reduce the EM emanation intensities. Since EM signals can travel through normal walls, metal shields like the Faraday cage are needed to block EM wave propagation. As reported in [17], EM emanations from metal-shielded computers are weakened. Our evaluations in Sect. 5.5 also show that long-range data exfiltration from computers in metal cases using *BitJabber* is harder to implement. However, we need to keep in mind that metal-shielding does not completely eliminate this covert channel, as we've seen in our evaluations that data exfiltration is still possible from 4 m away for platform A that has a metal case. SNR can also be lowered by increasing the noise level in the surrounding environment, which can be achieved using signal interference devices to jam the frequency range around the carrier signal. However, our approach will disperse the power of random noise after de-spreading the EM signal generated by the DRAM clock. As our evaluations indicate, the random noise irrelevant to victim computers only has a slight influence on *BitJabber*'s performance. However, the noise with SSC patterns can effectively increase the error rate. Therefore, to better mitigate this covert channel, the noise generator can produce noise with an SSC pattern to disturb the de-spreading process.

The other direction of countermeasures targets the sender running on the guarded computer. Because performing memory activities in stable frequencies is essential to implement *BitJabber* with a low error rate, we can execute some memory-intensive applications to disturb the sender's memory access behavior. In this way, the second type of error may be significantly increased. Besides, we can also increase the third type of error by introducing more computation load in the protected computers. However, all these methods require the computer to stay busy to some extent, which may hurt the computer's performance sometimes. Even with this mitigation, it is still possible for the attacker to circumvent it. The sender program can determine how busy the system is by timing several memory accesses. Using this information, an attacker can dynamically adjust the symbol rate or wait until the system is idle to start data exfiltration.

The implementation of *BitJabber* may be detected by looking for repeated memory accesses to the same memory locations. The methods relying on last-level cache miss detection [45] do not work because the non-temporal memory access instructions do not cause cache effects. However, such access patterns can be identified by memory controllers [46] or using side-channel information [41].

Furthermore, since *BitJabber*'s performance is highly dependent on the de-spreading of SSC signal, a good idea of mitigation is preventing the de-spreading process. In most modern computers, SSC is implemented by FM modulating the clock signal with a simple periodical signal. This despreading process can be easily reversed to recover the modulating signal. If we use a more complicated SSC technique (e.g., using a secret random number sequence to FM modulate the clock signal), the attacker cannot restore the high-SNR carrier, and the implementation of *BitJabber* is much harder.

7 Conclusion

In this paper, the EM radiation of the DRAM clock is exploited to implement a covert channel. We restore a high-SNR carrier by de-spreading the DRAM clock's EM emanations and applying multiple modulation techniques to exploit EM signals to efficiently exfiltrate data from air-gapped computers. The performance of our covert channel *BitJabber* is evaluated and compared with an existing covert channel *GSMem*, which exploited the same EM emanations from the DRAM clock. *BitJabber* can reach a bandwidth of 300,000 bps with an error rate under 1%. It can also perform wall-penetrating data exfiltration and long-range data exfiltration. According to [28], people used to consider the SSC technique a countermeasure for EM side-channel attacks, but our work shows that this countermeasure can be easily invalidated. Although only the DRAM clock's EM emanations are investigated in this work, we also notice that many other components in computers also generate strong EM emanations after de-spreading, which may also be exploitable for performing threatening covert channels. This covert channel greatly increases the maximum data exfiltration speed for air-gapped computers by exploiting EM side-channels, making people pay more attention to the protection against EM attacks.

Author Contribution N/A.

Funding This work is supported in part by the National Science Foundation (CNS-1739328, CNS-2147217).

Availability of Data and Material The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code Availability N/A.

Declarations

Ethics Approval N/A.

Consent to Participate All the authors have approved.

Consent for Publication All the authors have approved.

Competing Interests Yier Jin.

References

1. Zhan Z, Zhang Z, Koutsoukos X (2020) Bitjabber: The world's fastest electromagnetic covert channel. In: 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE
2. Guri M, Monitz M, Mirski Y, Elovici Y (2015b) Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In: 2015 IEEE 28th Computer Security Foundations Symposium (CSF '15), pp 276–289
3. Guri M, Zadov B, Elovici Y (2017b) 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 (DIMVA '17), pp 161–184
4. Lopes AC, Aranha DF (2017) Platform-agnostic low-intrusion optical data exfiltration. In: Proceedings of the 3rd International Conference on Information Systems Security and Privacy, pp 474–480
5. Loughry J, Umphress DA (2002) Information leakage from optical emanations. ACM Transactions on Information and System Security (TISSEC) 5(3):262–289
6. Sepetnitsky V, Guri M, Elovici Y (2014) Exfiltration of information from air-gapped machines using monitor's led indicator. In: 2014 IEEE Joint Intelligence and Security Informatics Conference, IEEE, pp 264–267
7. Guri M, Daidakulov A, Elovici Y (2018a) Magneto: Covert channel between air-gapped systems and nearby smartphones via cpu-generated magnetic fields. arXiv preprint arXiv:180202317
8. Guri M, Zadov B, Daidakulov A, Elovici Y (2018c) Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields. arXiv preprint arXiv:180202700
9. Matyunin N, Szefer J, Biedermann S, Katzenbeisser S (2016) Covert channels using mobile device's magnetic field sensors. In: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC '16), pp 525–532
10. Carrara B, Adams C (2014) On acoustic covert channels between air-gapped systems. In: International Symposium on Foundations and Practice of Security, Springer, pp 3–16
11. Guri M, Solewicz Y, Daidakulov A, Elovici Y (2016c) Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers. arXiv preprint arXiv:160605915
12. Guri M, Solewicz Y, Daidakulov A, Elovici Y (2017a) Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise ('diskfiltration'). In: European Symposium on Research in Computer Security (ESORICS '17), pp 98–115
13. Hanspach M, Goetz M (2013) On covert acoustical mesh networks in air. J Commun 8(11)
14. Guri M, Kedma G, Kachlon A, Elovici Y (2014) 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 '14), IEEE, pp 58–67
15. Guri M, Kachlon A, Hasson O, Kedma G, Mirsky Y, Elovici Y (2015a) Gsmem: Data exfiltration from air-gapped computers over gsm frequencies. In: 24th USENIX Security Symposium (USENIX Security 15), pp 849–864
16. Guri M, Monitz M, Elovici Y (2016b) Usbee: air-gap covert-channel via electromagnetic emission from usb. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST '16), pp 264–268
17. Zajić A, Prvulovic M (2014) Experimental demonstration of electromagnetic information leakage from modern processor-memory systems. IEEE Trans Electromagn Compat 56(4):885–893
18. Callan R, Zajić A, Prvulovic M (2015) Fase: finding amplitude-modulated side-channel emanations. In: 2015 ACM/IEEE 42nd

Annual International Symposium on Computer Architecture (ISCA '15), pp 592–603

19. Lampson BW (1973) A note on the confinement problem. Commun ACM 16(10):613–615
20. Szefer J (2019) Survey of microarchitectural side and covert channels, attacks, and defenses. Journal of Hardware and Systems Security 3(3):219–234
21. Masti RJ, Rai D, Ranganathan A, Müller C, Thiele L, Capkun S (2015) Thermal covert channels on multi-core platforms. In: 24th USENIX Security Symposium (USENIX Security 15), pp 865–880
22. Maurice C, Neumann C, Heen O, Francillon A (2015) C5: cross-cores cache covert channel. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '15), pp 46–64
23. Ristenpart T, Tromer E, Shacham H, Savage S (2009) Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM conference on Computer and communications security (CCS '09), ACM, pp 199–212
24. Sullivan D, Arias O, Meade T, Jin Y (2018) Microarchitectural minefields: 4k-aliasing covert channel and multi-tenant detection in iaas clouds. In: NDSS '18
25. Wang Z, Lee RB (2006) Covert and side channels due to processor architecture. In: 2006 22nd Annual Computer Security Applications Conference (ACSAC '06), IEEE, pp 473–482
26. Wu Z, Xu Z, Wang H (2012) Whispers in the hyper-space: High-speed covert channel attacks in the cloud. In: Presented as part of the 21st USENIX Security Symposium (USENIX Security 12), pp 159–173
27. Xu Y, Bailey M, Jahanian F, Joshi K, Hiltunen M, Schlichting R (2011) An exploration of l2 cache covert channels in virtualized environments. In: Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW '11), pp 29–40
28. Davidov M, Oldenburg B (2020) Tempesthome - finding radio frequency side channels. Tech. rep., Duo, URL https://duo.com/labs/research/finding-radio-sidechannels
29. Guri M (2020) Power-supplay: Leaking data from air-gapped systems by turning the power-supplies into speakers. arXiv preprint arXiv:200500395
30. Zhou Z, Zhang W, Yang Z, Yu N (2017) Exfiltration of data from air-gapped networks via unmodulated led status indicators. arXiv preprint arXiv:171103235
31. Guri M, Hasson O, Kedma G, Elovici Y (2016a) An optical covert-channel to leak data through an air-gap. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST '16), pp 642–649
32. Guri M, Bykhovsky D, Elovici Y (2019) Brightness: Leaking sensitive data from air-gapped workstations via screen brightness. In: 2019 12th CMI Conference on Cybersecurity and Privacy (CMI), IEEE, pp 1–6
33. Guri M (2019) Hotspot: Crossing the air-gap between isolated pcs and nearby smartphones using temperature. In: 2019 European Intelligence and Security Informatics Conference (EISIC), IEEE, pp 94–100
34. Guri M, Zadov B, Bykhovsky D, Elovici Y (2018b) Powerhammer: Exfiltrating data from air-gapped computers through power lines. arXiv preprint arXiv:180404014
35. Nassi B, Pirutin Y, Shamir A, Elovici Y, Zadov B (2020) Lamphone: Real-time passive sound recovery from light bulb vibrations. Cryptology ePrint Archive
36. Kwong A, Xu W, Fu K (2019) Hard drive of hearing: Disks that eavesdrop with a synthesized microphone. In: 2019 IEEE symposium on security and privacy (SP), IEEE, pp 905–919
37. Sehatbakhsh N, Yilmaz BB, Zajic A, Prvulovic M (2020) 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), IEEE, pp 123–138
38. Shen C, Liu T, Huang J, Tan R (2021) When lora meets emr: Electromagnetic covert channels can be super resilient. In: 2021 2021 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, Los Alamitos, CA, USA, pp 1304–1317. 10.1109/SP40001.2021.00031, URL https://doi.ieeecomputersociety.org/10.1109/SP40001.2021.00031
39. Ge Q, Yarom Y, Cock D, Heiser G (2018) A survey of microarchitectural timing attacks and countermeasures on contemporary hardware. J Cryptogr Eng 8(1):1–27
40. Departments, agencies of the Federal Government (2019) Code of federal regulations. URL https://www.ecfr.gov/cgi-bin/text-idx?SID=8c3c331bc40fd1a017dbf9917665f6c6&mc=true&node=pt47.1.15&rgn=div5
41. Zhang Z, Zhan Z, Balasubramanian D, Li B, Volgyesi P, Kousoukos X (2020) Leveraging em side-channel information to detect rowhammer attacks. In: 2020 IEEE Symposium on Security and Privacy (S&P '20), pp 729–746
42. Hassan M, Kaushik AM, Patel H (2015) Reverse-engineering embedded memory controllers through latency-based analysis. In: 21st IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '15), pp 297–306
43. Pessl P, Gruss D, Maurice C, Schwarz M, Mangard S (2016) Drama: Exploiting dram addressing for cross-cpu attacks. In: 25th USENIX Security Symposium (USENIX Security 16), pp 565–581
44. Xiao Y, Zhang X, Zhang Y, Teodorescu R (2016) One bit flips, one cloud flops: Cross-vm row hammer attacks and privilege escalation. In: 25th USENIX Security Symposium (USENIX Security 16), pp 19–35
45. Aweke ZB, Yitbarek SF, Qiao R, Das R, Hicks M, Oren Y, Austin T (2016) Anvil: Software-based protection against next-generation rowhammer attacks. ACM SIGPLAN Not 51(4):743–755
46. Yağlıkçı AG, Patel M, Kim JS, Azizi R, Olgun A, Orosa L, Hassan H, Park J, Kanellopoulos K, Shahroodi T, et al. (2021) Blockhammer: Preventing rowhammer at low cost by blacklisting rapidly-accessed dram rows. In: 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA), IEEE, pp 345–358