BitDance: Manipulating UART Serial Communication with IEMI

Zhixin Xie
Zhejiang University
Hangzhou, Zhejiang, China
zhixinxie1999@zju.edu.cn

Chen Yan*
Zhejiang University
Hangzhou, Zhejiang, China
yanchen@zju.edu.cn

Xiaoyu Ji
Zhejiang University
Hangzhou, Zhejiang, China
xji@zju.edu.cn

Wenyuan Xu
Zhejiang University
Hangzhou, Zhejiang, China
wyxu@zju.edu.cn

ABSTRACT

Wired serial communication protocols such as UART are widely used in today's IoT systems for their simple connection and good industry ecology. However, due to the simplicity of these protocols, they are vulnerable to attacks that falsify the communication. In this work, we propose the `BitDance` attack that can arbitrarily flip the bits of serial communication without any physical contact utilizing intentional electromagnetic interference (IEMI). We describe the physical process of how electromagnetic interference influences the voltage, build up a model to demonstrate the bit-level control principle of our work, and implement the attack on 6 different sensors with UART, a widely used serial communication protocol. The result shows we can inject bit-level information and disable legitimate communication from the system with a maximum success rate of 45.4% and 100%. Finally, we propose countermeasures to mitigate the impact of this attack.

CCS CONCEPTS

• **Security and privacy → Hardware attacks and countermeasures; Embedded systems security**;

KEYWORDS

Serial communication; IEMI attack; Embedded system

ACM Reference Format:
Zhixin Xie, Chen Yan, Xiaoyu Ji, and Wenyuan Xu. 2023. BitDance: Manipulating UART Serial Communication with IEMI. In *The 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23), October 16–18, 2023, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3607199.3607249

*Chen Yan is the corresponding author.

RAID '23, October 16–18, 2023, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-0765-0/23/10…$15.00
https://doi.org/10.1145/3607199.3607249

1 INTRODUCTION

Serial communication is the process of sending data sequentially, i.e., one bit at a time. Due to its simple structure, easy implementation, and low cost, it has been widely used to connect devices in the Internet of Things (IoT) and cyber-physical systems (CPS). For example, most sensors, e.g., laser range sensors, carbon dioxide gas sensors, infrared temperature sensors, and weight pressure sensors, adopt serial communication to transmit data to controllers. As one of the most fundamental communication methods, the trustworthiness of the transmitted data on serial communication is critical to the normal functioning of IoT and CPS. However, the security of serial communication has not received due attention. Many existing studies focused on the security of data before the communication, e.g., whether the data from sensors reflects the environment situation correctly or whether sensors transmit the correct data to other units [31][33][23][22][29][25][28].

Our motivation in this work is to investigate the research question of to what extent can an attacker falsify and manipulate the information transmitted in serial communication without any physical contact. In specific, we wonder whether it is feasible to accurately control every bit of the data transmission, i.e., flipping "1" to "0" and "0" to "1" arbitrarily at the attacker's will. Like any other communication methods, serial communication can also be influenced by external environment factors like electromagnetic interference (EMI in short) [3][17][16]. In addition to unintentional interference, we believe EMI can also become one of the attacker's means as it can affect the operation of the communication system without contact, which is beneficial to the concealment of the attack. However, at the first glance, the impact of EMI on serial communication seems to be largely unpredictable as the signal dissipates and changes rapidly with distance, hence it may be difficult to be exploited for the bit-level targeted attack against the wired protocol. Though the feasibility of EMI attack on serial communication has been verified before [3][6][20], to the best of our knowledge and as we elaborate in Section 2, there has been no published EMI attack that can manipulate arbitrary bits on the serial communication lines at a distance without any physical contact.

In this paper, we propose the BitDance, an EMI-based attack that can flip arbitrary bits and manipulate the content of the serial communication without any contact with the victim wires. The core idea is to use electromagnetic radiation to change the voltage of the signal line and then exert influence on every bit of the transmission. In specific, we design the attack against UART (Universal asynchronous receiver-transmitter), which is one of the most used serial

communication protocols for its simple connection, straightforward interface, and good industry ecology.

To achieve such an attack, we have to overcome two challenges. (1) It is difficult to predict and control the voltage of the signal line precisely by EMI. To address this challenge, we analyze the detailed process from IEMI generation to voltage induction of the circuit loop, and finally, we find that the induced voltage is the direct ratio to the derivative of the signal generated by the attacker. Based on this observation, we carefully design the signal frequency and signal phase to influence every sample result. (2) It is difficult to attack the serial communication system at a long distance. To increase the attack distance, we use high-frequency EMI to obtain a higher amplitude of the coupling voltage. However, it will result in an essential problem: the frequency difference between the receiver sample rate and the attack waveform frequency. To deal with it, we build up a model to describe what serial signal the receiver will read under this non-ideal situation and exploit the aliasing effect of Analog-to-digital Converter (ADC) to derive controllable sampling results on the receiver, which will be thoroughly discussed in Section 5.

Overcoming these challenges, our attack can achieve two types of attack effects. (1) *Shield attack*. When the serial communication system is working, our attack can disrupt all legitimate signals transmitted on the signal line and can prevent the receiver from getting any valid information. (2) *Creation attack*. When the communication system is working, our attack can inject a false signal into the wires that can flip arbitrary bits and make the receiver get erroneous data regardless of what legitimate data is being transmitted. We evaluate our attack method on six types of sensors and validate the feasibility of falsifying sensor data and textual messages. For example, when the MCU is receiving data from the MPU6050 acceleration sensor, the shield attack can disable the MCU from receiving the acceleration data from the sensor, and the creation attack can make the MCU receive false acceleration data injected by the attacker. In addition, our attack can also inject meaningful text such as "helloworld" into the communication between two MCUs. According to our experimental result, the shield attack has a maximum success rate of 100% and the creation attack has a maximum success rate of 45.4%.

Our contributions are as follows:

- We propose the BitDance attack, a bit-level control attack against a serial communication system at a distance that can disrupt legitimate communication or inject erroneous data.
- We evaluate bit-level control attacks successfully on six different sensors with two microcontrollers. Our attack can influence every bit of the victim communication system.
- We suggest countermeasures to protect serial communication from such attacks.

2 RELATED WORK

There have been several studies about EMI injection attacks over the past few years. EMI is utilized to falsify behaviors of many kinds of sensors, computation units, and actuators. In the following, we provide a summary of the existing EMI attacks on analog circuits and digital circuits.

EMI attack on analog circuits. **(a) Attacks on sensors**. Works in [31][12] exploited the nonlinearity of microphones to inject EMI signals, making microphones receive inaudible commands designed by the attacker. Works in [4] injected commands using IEMI into a microphone when it's wirelessly charging. Works in [10][28][15] induced current and injected fake operations like touches and slides on the capacitor touch screen. Shoukry et al. [21] influenced the magnetic field around speed sensors to spoof the results of it. There is other existing work on GPIO and ADC [30], temperature sensors [26], photo-diode sensors [20] and CCD sensors [11]. **(b) Attacks on actuators.** Selvaraj et al. [20] spoofed the PWM signal to make an originally motionless servo motor rotary in a certain direction. Dayanikli et al. [5] attacked a converter of a vehicle battery and led to some further problems like battery overheating. Our bit-level control attack focuses on the communication phase and tries to induce a controllable signal into serial communication.

EMI attack on digital circuits. Digital circuits use bits to store, calculate, and transmit information. Bit-flip is a general term for a class of physical phenomenon that means flipping bits in the digital system, and it can be man-made or naturally occurring. This topic is discussed in many fields like programs' error resilience [19], neural network [18], and communication decoder [32]. As the attacker, we focus more on deliberately flipping bits to mislead the system. There have been many works on attacking computer hardware like DRAM [27][9], NAND [13], and flash [14]. Our work focuses more on wired digital communication and aims to flip every bit of communication. Works in [3][17][16] discussed that serial communication can be influenced by EMI. Selvaraj et al. [20] qualitatively explored the impact of high-frequency electromagnetic waves (EM waves in short) on serial communication.

As far as we know, Dayanikli's work [6] is most similar to ours. Dayanikli et al. tried to utilize EMI to exert influence on the communication system. To get a stable attack effect, the attacker needed to utilize the low-frequency EMI and loop the victim signal line around the toroid closely to obtain a strong enough EMI to complete the attack which is difficult in a real-world attack scenario. When increasing the attack range, the attack effect became unstable. The successful rate of injecting a single bit was no more than 63%, and the injection of UART format signal was also hard to achieve. In our work, we utilize the high-frequency EMI to achieve a longer attack range and successfully inject UART format information into the victim system at a distance.

3 BACKGROUND

Our attack utilizes EMI to attack the communication system which adapts UART protocols, and the UART protocol is implemented by the general purpose input output (GPIO in short). Therefore, we introduce the target protocol, the hardware structure, and the physical principle respectively.

3.1 Universal Asynchronous Receiver-Transmitter (UART)

UART is a duplex asynchronous serial communication protocol for two-microcontroller (MCU in short) communication. In this subsection, we introduce the UART protocol from three aspects: Hardware properties and communication process.

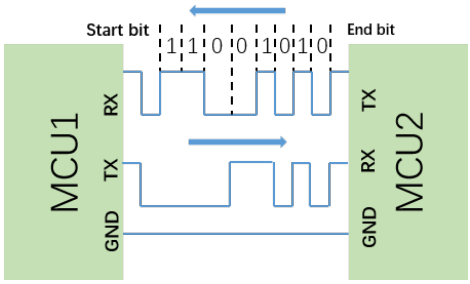

Figure 1: The circuit and the protocol of UART

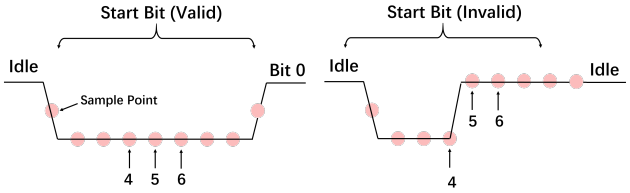

Figure 2: Recognition of start bit

Hardware properties. As shown in Fig. 1, the MCU needs two GPIOs, TX and RX, for UART communication. TX is used for sending signals and RX is used for receiving signals. TX is configured in output mode while RX is configured in input mode. TX of MCU1 is connected with RX of MCU2 by a signal line, and vice versa. Because the communication parties use two GPIOs for communication, they can send and receive signals simultaneously. The ground electrodes of the two MCUs also need to be connected.

As UART does not require a clock line, it requires that the communication parties agree on a baud rate for synchronization before the communication starts. Common baud rates are 9600,19200,38400, 115200 bit/s. The TX outputs digital signals to represent logic 0 and logic 1, and RX will sample the digital signals with frequency 8 times (in some MCUs, 16 times) and the baud rate to recognize whether the bit is logic 0 or logic 1.

Communication process. A frame of UART usually has four parts, start bit, data bits, parity bit (optional), and stop bit(s). It supports the different lengths of data bits (5,6,7,8,9 data bits) and stop bits (1,1.5,2 data bits). However, no matter what the frame structure is, our attack works according to the same principle and steps. For the convenience of description in the following, if not otherwise specified, we assume a frame consists of one start bit, eight data bits (which means one frame of UART communication transmits one byte of data), and one stop bit. We assume a situation in which MCU1 is sending data and MCU2 is receiving data. When there is no communication between two MCUs, the system is in an idle state, and the voltage of the signal line keeps high. If MCU1 wants to send a frame of information, it pulls down the voltage of the signal line to send the start bit. As Fig. 2 shows, the MCU2 will detect the voltage fall, record the next eight sampling results, and judge whether the start bit is valid according to the 4th, 5th, and 6th sampling results. If the three samples' results are all low voltage, the MCU2 will recognize this start bit. Otherwise, the MCU2 will regard the voltage fall on the signal line as a glitch and discard it, and wait for the next time of voltage fall. After a valid start bit,

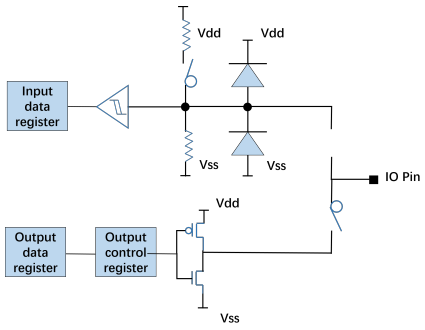

Figure 3: The structure of GPIO.

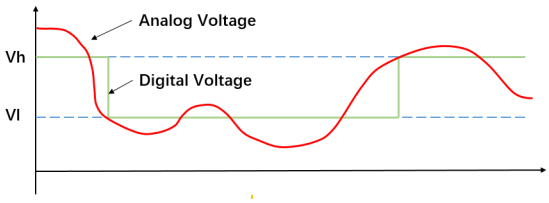

Figure 4: The corresponding relationship between the analog signal and the digital signal sampled by GPIO.

the MCU1 begins sending 8-bit data. The high (or low) voltage represents logic 1 (or 0). The MCU2 also samples the signal line eight times to confirm the value of every bit. The MCU2 determines the value of this bit according to the majority of the 4th, 5th, and 6th sampling results. After sending 8-bit data, the stop bit will be sent. The MCU1 pulls up the voltage to stop a frame of information.

3.2 General Purpose Input Output (GPIO)

GPIO is a universal peripheral of MCUs that can send or receive digital signals, and the structure of the GPIO is shown in Fig. 3. Generally speaking, the GPIO has two switches to control its mode. When it is in output mode, the user can write the value into the output data register of the MCU. Then the output method, like push-pull and open-drain will be configured by writing certain values (for different MCUs, the values are different) into the output control register. When it is in input mode, there are two diodes connected to the pin to limit the input voltage. Then two switches are connected to V_{dd} and V_{ss} respectively. They can control the input modes as pull-up or pull-down. A Schmitt trigger is used at the next stage. It sets two voltage thresholds, V_h and V_l, as shown in Fig. 4. If the voltage is higher (or lower) than V_h (or V_l) then the output of the Schmitt trigger will get a logic 1 (or logic 0). If the voltage is between V_h and V_l, then it will not change the sample result. The Schmitt trigger is used to prevent the influence of noise or unexpected voltage glitches.

3.3 Electromagnetic Induction Theory

Electromagnetic induction theory describes a series of natural phenomena [7]. Assume that there is a wire with the current in the space, shown as Fig. 5. Firstly, the current can generate the magnetic

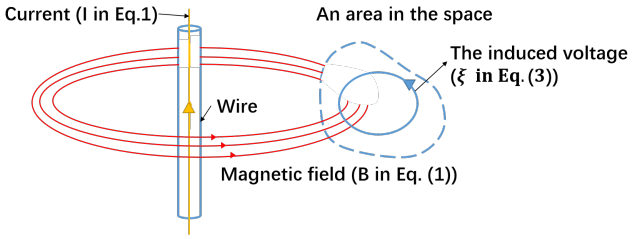

Figure 5: The AC current can generate AC electrical field nearby.

field in the surrounding space, and Eq. (1) describes this phenomenon

$$\oint_{loop} B \cdot \mathrm{d}l = \mu I \tag{1}$$

where I is the current of a wire, B is the magnetic field generated by the current, the loop can be one of any path that loops around the wire, and μ is the permeability of free space. As Eq. (1) shows, the magnitude of the magnetic field is in direct proportion to the magnitude of the current. In the surrounding space, there will fill up with the magnetic field, and the integral of the magnetic field with respect to space is called the magnetic flux, which can be represented by

$$\Phi_B = \iint_{area} B \cdot \mathrm{d}s \tag{2}$$

where *area* is a piece of space, and Φ_B is the magnetic flux that passes through the *area*. Φ_B is in direct proportion to the magnetic field, thus it is also in direct proportion to the current in the wire. If the current is alternating current, then the Φ_B will also change with respect to time. Therefore, according to the electromagnetic induction theory, the time-variant Φ_B generates a time-variant electrical field. And the electrical field is integrated to voltage along the edge of the *area*, as shown in Eq. (3)

$$\xi = \frac{\mathrm{d}\Phi_B}{\mathrm{d}t} \tag{3}$$

where ξ is the induced voltage. As ξ is proportional to the differential of Φ_B, the ξ is also proportional to the differential of the current. Therefore, the ξ can also be described as the product of a constant and the current differential, as shown in Eq. (4)

$$\xi = constant \cdot \frac{\mathrm{d}I}{\mathrm{d}t} \tag{4}$$

4 THREAT MODEL

Attacker's goal. The attacker's goal is to control the bit-level data transmitted by serial communication operating at the TTL logic level in a contactless manner. We further induce two attack effects, shield attack which shields the normal data transmitting on the signal line, and creation attack which injects false data that doesn't exist originally on the signal line.

Attacker's knowledge. The attacker needs to know the details of the serial communication protocol such as frame structure, length of stop bit, etc. The attacker also needs to know the model of the victim's device. The attacker does not need to know the baud rate of the communication because there are only several fixed and commonly used baud rates, as mentioned above. Even if the attacker does not know the exact baud rate, the attacker can test all the possible baud rates.

Table 1: The parameters list.

Parameters	Meaning
V_a	The amplitude of the attack waveform
$f_a(T_a)$	The frequency (period) of the attack waveform
V_h	The upper threshold of the receiver
V_l	The lower threshold of the receiver
$f_s(T_s)$	The sample rate (period) of the receiver

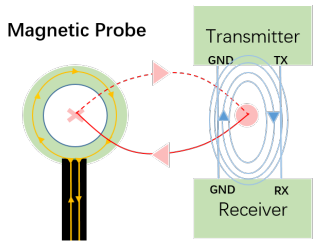

Figure 6: A simple diagram of the attack setup.

Attacker's capability. The attacker can carry his attacking equipment in the vicinity of the victim system, but there is no physical contact between the attacker and the victim system. The attacker can only utilize IEMI to interfere with the specific content of the communication. In other words, the attacker cannot read or write the memory of the MCU or sensor, change the connection of the digital communication system, or affect the signal received by the MCU or sensor through software attack methods (such as malware, etc.).

5 ATTACK MECHANISM

In this section, we use a prototype serial communication system to illustrate the attack mechanism of the BitDance attack. As shown in Fig. 6, there is a transmitter and a receiver connected by a signal line, and the receiver samples the voltage of the signal line with the sample rate f_s. We assume that the receiver recognizes voltage higher than V_h as logic 1 and less than V_l as logic 0. The original waveform on the signal line is generated by the transmitter, and it's a valid frame of the UART signal. Now the attacker injects the attack waveform and changes the voltage of the signal line. The frequency of the attack waveform is fa and its amplitude is Va. The receiver samples the signal line and records the received results. The signal waveform in the whole process is shown as Fig. 7. In the process above, the induced attack waveform is the only handle for the attacker to influence the communication system. Therefore, the attacker needs to well design the attack waveform. We assume that the attacker induces the sine wave as the attack waveform and the attacker can adjust the parameters of the sine wave like phase, amplitude, and frequency. We list the parameters frequently used in this paper in Table 1. The basic mechanism of our attack is using one cycle of the sine wave to influence one sample result. As shown in Fig. 8, both low-frequency and high-frequency EMI can do it, but

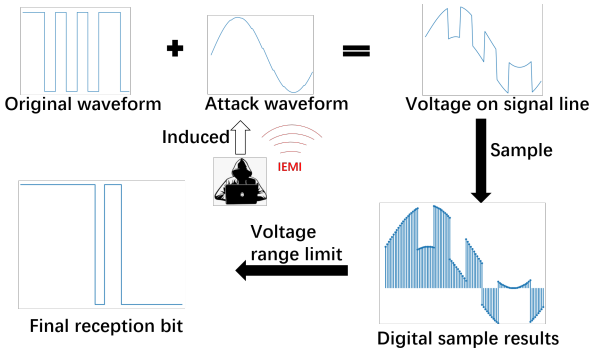

Figure 7: Some important waveforms in the whole process

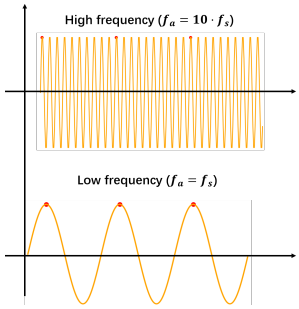

Figure 8: Both high and low-frequency attack waveform satisfies the attack mechanism, but the high-frequency attack waveform helps increase the attack distance.

in order to increase our attack distance, we choose high-frequency EMI. Therefore, it's important for the attacker to keep f_a equal to f_s or multiple times f_s as much as possible. The different f_a only leads to different attack distances, and have no other differences in other attack analyses. Therefore, for simplicity of description, we take $f_a = f_s$ as an example to illustrate our attack mechanism. However, the device of the attacker may not be ideal, and it's hard to make f_a exactly equal to f_s especially when the device is generating a high-frequency signal. Therefore, we differentiate the attacker's ability by whether he can absolutely control the frequency of the attack waveform. In this section, we thoroughly discuss how the attacker can induce the voltage on the signal line and then control the received result with different levels of attack ability.

5.1 Voltage Induction

In this subsection, we discuss how to induce voltages on the signal line. The simple diagram of the attack setup is shown in Fig. 6. The yellow curve represents the current in the probe and it emits the magnetic field, shown as a red curve with arrows in the figure. Part of the magnetic field falls in the circuit loop, which forms the magnetic flux. If the amount of magnetic flux changes, then it will generate induced voltage around the circuit loop, shown as the blue curve in the figure. According to Eq. (4), the amplitude of the induced voltage is determined by the current derivative in the probe with respect to time. It means the amplitude of the induced voltage will be influenced by the frequency of the current waveform.

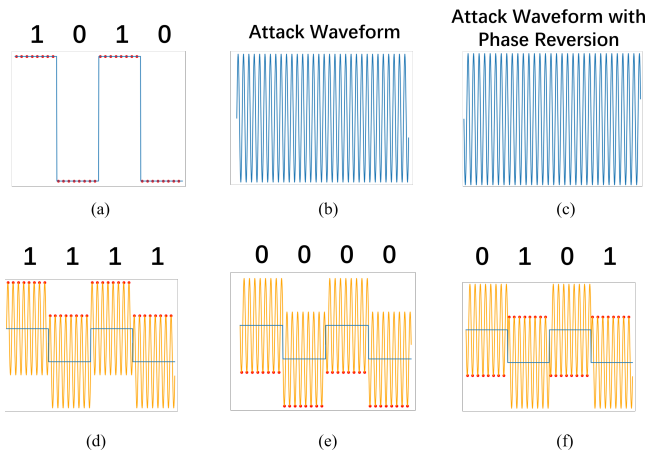

Figure 9: (a) The blue waveform is the original waveform of the signal line and red points represent the results of sampling. The original digital signal is "0101" (b) The blue waveform is the attack waveform induced by the attacker. (c) The blue waveform is the attack waveform whose phase is shifted by 180° to exert a reversed effect on the victim. (d) The orange waveform is the voltage on the signal line which is the sum of the waveform of (a) and (b), and the red points represent sample results. After inducing the attack waveform, the received signal becomes "1111". (e) The orange waveform is the sum of the waveform of (a) and (c), the reversed attack waveform pulls down the voltage on the signal line, then the sampling results become "0000". (f) Mix the attack waveform of (b) and (c) to induce "0" and "1".

If we use the sine wave current to generate EM waves, then the induced voltage must also be a same frequency sine wave with a $\frac{\pi}{2}$ phase shift compared with the current waveform. Therefore, we can control the frequency and amplitude of the induced voltage by adjusting the current in the probe antenna.

5.2 Attackers with Accurate Frequency Control

In this subsection, we discuss how to attack the victim system with absolute control of f_a. As mentioned above, the receiver receives the information by sampling the voltage on the signal line. We assume that the transmitter sends the bit stream of "0101". So the original waveform on the signal line is a square wave as shown in Fig. 9(a). The receiver samples eight times to confirm every bit.

Our basic idea of signal manipulation is to induce a sine wave attack waveform that has the same frequency as the receiver's sample rate. Then the sine wave will influence every sample result. Fig. 9(b) is the attack waveform with sufficiently large amplitude, and Fig. 9(c) is the reversed waveform of Fig. 9(b). The attacker induces the attack waveform to the signal line, then the voltage on the signal line is the sum of the attack waveform and the original waveform. As the frequency of the attack waveform equals the sample rate, when the receiver samples, the phase of the attack waveform doesn't change. The attacker can carefully adjust the phase of the attack waveform so that the induced sine wave is

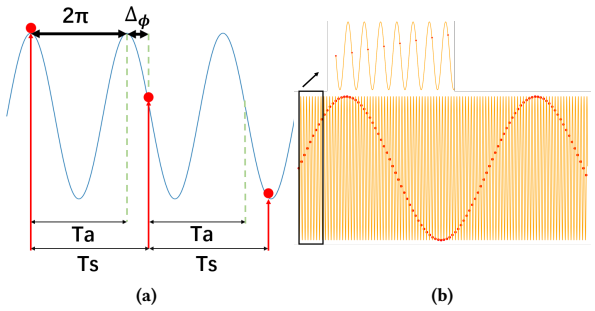

Figure 10: (a) The inherent deviation of the instruments makes T_a not equal T_s. (b) The difference of T_a and T_s makes sample results a sine wave in a long time scale.

sampled at its maximum values, i.e. at the peaks or troughs. After inducing the attack waveform, the voltage on the signal line increases and is shown as the orange waveform in Fig. 9(d). The sample result changes from the original "0101" to "1111" because the sampled values all exceed the upper threshold V_h. If the attacker wants to alter the sample result in the opposite direction, he only needs to reverse the phase of the attack waveform and pull down the voltage every time the receiver samples. Therefore, every sample result will be logic 0, as shown in Fig. 9(e), and the original "0101" becomes "0000". If the attacker wants to mix "1" and "0" into the injected information, he only needs to reverse the phase when he wants to reverse the injected bit, as shown in Fig. 9(f). This can be implemented by phase shift keying (PSK) modulation. PSK modulation is a widely used modulation method and it is integrated into many signal generators on the market nowadays. Using PSK, it's easy for the attacker to reverse the phase of the attack waveform. By phase reversion, the waveform will pull up or pull down the voltage on the signal line, and the received bit will also be alternated.

The above paragraph discusses how to set frequency and phase to manipulate the sample results. Then we discuss the range of the attack waveform's amplitude that can sufficiently alter the sample results. We assume V_a is the amplitude of the attack waveform. V_{ol} and V_{oh} are the amplitude of the original waveform when it represents logic 0 and logic 1 respectively. V_h and V_l are the two thresholds of the receiver. When the attacker wants to alter the voltage from low to high, then he needs to hold the inequality $V_a + V_{ol} > V_h$, that is $V_a > V_h - V_{ol}$. When the attacker wants to alter the voltage from high to low, then he needs to hold the inequality $V_{oh} - V_a < V_l$, that is $V_a > V_{oh} - V_l$. To summarize it,

$$V_a > max(V_{oh} - V_l, V_h - V_{ol}) \tag{5}$$

If V_a satisfies Eq. (5), then it's large enough for the attacker waveform to alternate the sample results.

5.3 Attackers with Inaccurate Frequency Control

In the last subsection, we discuss the attack mechanism when the attacker can absolutely control the frequency of the attack waveform. However, in a real-world attack, the output of the instrument always deviates slightly from its set value. For example, when the attacker plans to generate a sine wave at 10 MHz, he may generate a sine wave at 10.00001 MHz or 9.99999 MHz because of the inherent error of the signal generator. Therefore, even if the attacker knows the sample rate of the victim system, he cannot generate the attack waveform whose frequency exactly equals the sample rate which also varies slightly with time. In this situation, the phase of the induced waveform changes as the receiver samples, as shown in Fig. 10(a).

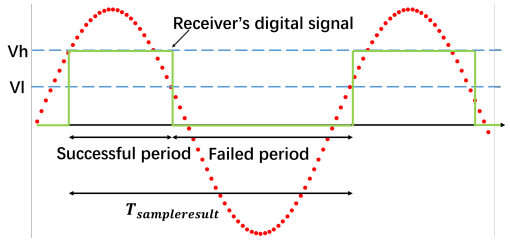

Figure 11: Sample results and received digital signals. The red points represent sample results, and the green waveform is the digital signal received.

In Fig. 10(a), we assume that the period of the attack waveform (T_a) is a little shorter than the period of sampling (T_s). Every time the receiver sample the voltage on the signal line, the corresponding phase of the attack waveform will increase by Δ_ϕ. The time corresponding to the radian angle 2π is T_a, so the radian angle corresponding to $T_s - T_a$ is shown as

$$\Delta_\phi = \frac{abs(T_s - T_a)}{T_a} \cdot 2\pi \tag{6}$$

According to Eq. (6), after $\frac{2\pi}{\Delta_\phi}$ times of samples, the phase of the corresponding attack waveform will change by 2π, and that is also 0. Therefore, the influence of instruments' deviation is periodic. In Fig. 10(b), we assume the original waveform is constantly low voltage and the voltage on the signal line is only determined by the attack waveform. The red points are sample results, and the corresponding attack waveform phase changes a little every time the receiver samples. The sample results form a sine wave with a period

$$T_{sampleresults} = \frac{2\pi}{\Delta_\phi} \cdot T_s \tag{7}$$

Therefore, due to the inherent deviation of the device, the sample results consist of a sine wave whose period is $T_{sampleresults}$. As shown in Fig. 11, the receiver will get a bit stream with "011100001110...". If the attacker's target is to alternate the reception from logic 0 to logic 1, then part of the time he fails. We can calculate the percentage of time that attacks can be successful

$$percentage = \frac{(arcsin(\frac{V_l}{V_a}) - arcsin(\frac{V_h}{V_a}))}{2\pi} \tag{8}$$

Because of the mathematical essence of Eq. (8), when using the non-ideal device, the attack can be successful no more than 50% of the time. It's the theoretical upper limit if the attacker cannot absolutely control the frequency of the waveform. The length of the successful period is

$$T_{successful} = percentage \cdot T_{sampleresults} \tag{9}$$

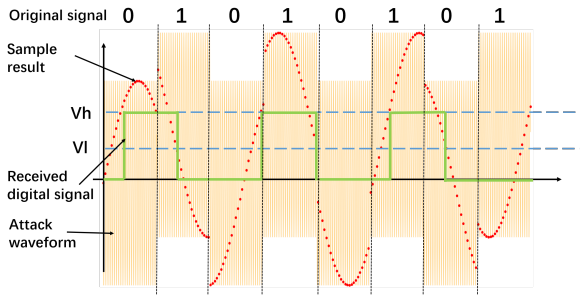

Figure 12: The attack waveform shields the original signal.

The above analysis is true when the original signal is constantly 0. The results are nearly the same when the original signal is a normal digital signal. As shown in the Fig. 12, the original signal is "01010101", and after inducing voltage, the original signal has already been influenced. The attack may not flip every bit of the original signal, but it's enough to shield the victim from receiving the normal signal. On the base of the above analysis, if the attacker wants to inject more meaningful information, he must keep the length of the successful period longer than the injecting time. For example, it takes about 1.041 ms to transfer one frame (one frame consists of ten bits, as mentioned above) of information in UART protocol with a 9600 bit/s baud rate, and the attacker must keep the successful period longer than 1.041 ms to succeed injecting one byte of data. In the real world, UART is usually used to transfer application layer instructions that are made of operators and operands. It often needs several bytes to transmit one meaningful instruction. Therefore, to inject more content, the attacker needs to lengthen the successful period as long as possible.

According to Eq. (7)(8)(9), there are two methods to increase the successful period. Firstly, the attacker can enhance the amplitude of the attack waveform. It's simple because it's configurable for most EM-generating devices like the signal generator and the power amplifier. Secondly, the attacker can reduce the difference between the f_s and f_a. According to the attacker's ability mentioned in the threat model, the attacker can purchase the same model device in advance. Then, the attacker needs to run the device to watch the real-time received content of it. The attacker can analyze the content and then subtly adjust the f_a to make its frequency close to the f_s. The detailed analysis method will be illustrated in Section 6.

6 ATTACK DESIGN

In this section, we present the attack design of the BitDance attack. We will first overview our attack workflow. Then, we will describe the basic method of EM generation and voltage induction. Based on the ability that we can inject voltage in the victim system, we will illustrate how to inject more falsified content by sample rate approaching that lengthens the successful period, and how to inject meaningful bits by PSK control that reverses the phase of the attack waveform.

6.1 Attack Workflow

The workflow of the attack is shown in Fig. 13. Firstly, the attacker needs to build up an attack device, that can stably generate EM waves in any frequency and induce the voltage on the signal line. After that, as the attacker knows the device model of the victim, he could buy the same model device as the victim and monitor the received content in real time. The attacker can analyze the received content and find the proper f_a. After that, the attacker uses a controller, usually an MCU, to control the PSK trigger to adjust the phase of the attack waveform, which helps inject meaningful information.

6.2 EM Generation and Voltage Induction

In this attack, we use a signal generator to generate the voltage signal. The signal generator can produce a sine wave signal with a maximum power of 80 mW, which is not enough to induce a 2 V voltage on the signal line. Therefore, we use an amplifier to convert this small voltage signal to a high-amplitude current signal. We connect a magnetic probe to the output of the amplifier and utilize the magnetic probe to emit the EM wave into space. The probe contains several loops of copper wire which makes it emit massive magnetic fields. There are two key points that can increase the induced voltage, and thus enhance the attack effect.

Frequency of the attack waveform. As mentioned in Section 5, the f_a will be coarsely tuned to multiple times of the f_s. As the amplitude of the induced voltage is proportional to the f_a, the attacker can increase the frequency as much as possible, only if it's multiple times the f_s.

Probe position. If we put the probe at different angles, the amount of the magnetic flux that falls in the circuit loop will also be different. To maximize the induced voltage, we need to find the right angle to place the magnetic probe. According to [34], we can use an n-gonal conductor to approximate the circular conductor when calculating the magnetic field. Therefore, we simplify the circular probe to many finite-length straight current-carrying conductors. To get the most magnetic flux, a fixed shape closed curve should be in the same plane as the conductor. Therefore, the attacker should try to place the probe in the same plane as the victim circuit loop to get a maximum induced voltage.

6.3 Sample Rate Approaching

As discussed in the attack mechanism, the most ideal situation is that the attacker can generate the attack waveform whose frequency is exactly the same as the sample rate, then the attacker can keep the sample results at a stable value. But the real-world devices always have inherent deviation, which leads to the sample results being like a square wave. A cycle of a square wave, that is, $T_{sampleresult}$, is divided into successful period and failed period, and the length of the square wave is determined by the difference of the f_a and f_s. The smaller the difference is, the longer the $T_{sampleresult}$ is. To inject more information, the attacker needs to keep $T_{sampleresult}$ as long as possible, which means the attacker needs to find a proper f_a close to f_s (or multiple times the f_s). In this subsection, we discuss the method that how to carefully adjust the f_a to lengthen the $T_{sampleresult}$. The challenge of this step is that because of the hardware properties of the MCU, the sample

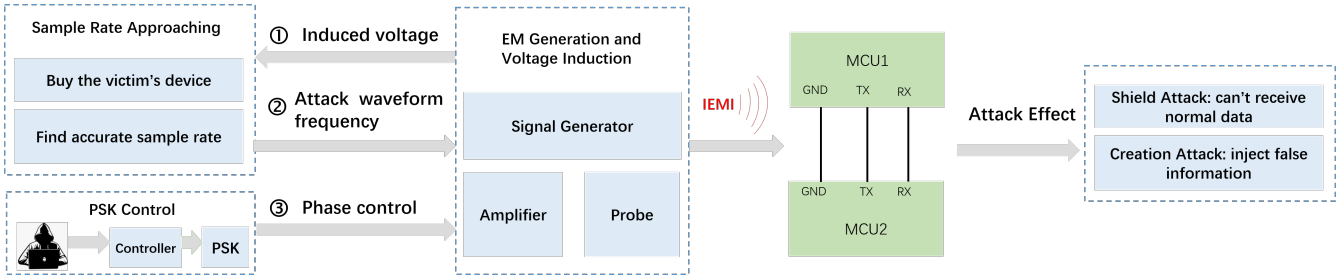

Figure 13: The steps of our attack.

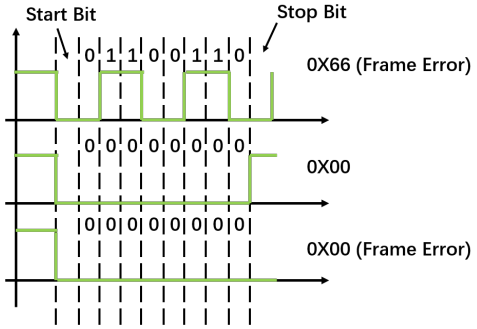

Figure 14: The value of different UART frames under the different period square waves.

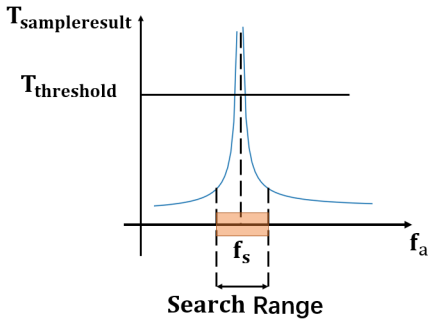

Figure 15: The relationship between $T_{sampleresults}$ and f_a.

result cannot be directly observed, so the attacker cannot directly get the value of $T_{sampleresult}$. The attacker must infer it from other information.

How to infer the value of $T_{sampleresults}$. As mentioned above, the sample results form a square wave. The square wave has one rise edge and one fall edge in a cycle. According to the UART protocols, when the receiver detects a falling edge on the signal line, then it regards that the transmitter sends a start bit, then it continuously samples the signal line to receive eight data bits which consist of a UART frame. As Fig. 14 shows, different received UART frame implicates different $T_{sampleresults}$. When the $T_{sampleresults}$ is shorter than a UART frame, the value will be random and not stable. When the $T_{sampleresults}$ is exactly long enough to inject a complete UART frame, the value of the frame will be 0x00. If the $T_{sampleresults}$ is longer, then the stop bit will be logic 0, which will trigger a frame error. Most MCUs can be configured to ignore this error and record the data bits as normal. Then the value of the UART frame will be 0x00. Until the next cycle of the square wave comes and the receiver detects a falling edge as the start bit, the receiver will receive nothing. Therefore, every time there comes a cycle of a square wave, there comes a falling edge of the square wave, and there will be a 0x00 received by the MCU. Therefore, $T_{sampleresults}$ is the same as the period of the 0x00 received by the MCU. The attacker can utilize it to infer the value of $T_{sampleresults}$ by measuring the frequency of received 0x00. Thus, if the attacker can get the received content, the attacker can infer the value of $T_{sampleresults}$. He can connect MCU's RX pin to a high voltage, and connect the TX pin to the computer by some UART-USB convert chips, like CH340. The attacker can program the MCU to make it send what it receives so that the computer can also obtain the received content of the MCU. Then the attacker can get the value of $T_{sampleresult}$.

How to search the proper f_a. After getting the value of $T_{sampleresult}$, the attacker can start the attack and search the proper f_a. He can set a frequency range centered on f_s (or multiple times f_s) in advance, and use the brute force method to search every possible frequency in the frequency range to get the optimal frequency. However, the brute force method costs much of time. Modern signal generators' frequency resolution can be as small as 0.1 Hz. It's impossible to test every frequency and observe its corresponding $T_{sampleresults}$.

To address this challenge, we utilize the three-divided algorithm to accelerate the search process. According to Eq. (6)(7), relationship between T_a and $T_{sampleresult}$ is

$$T_{sampleresult} = \frac{T_s \cdot T_a}{abs(T_s - T_a)} = \frac{1}{abs(f_s - f_a)} \tag{10}$$

As shown in Fig. 15, the function of $T_{sampleresult}$ is a convex function, then the attacker can utilize a three-divided algorithm to get its extremum. This algorithm is specifically designed to find the extremum of the convex function. The algorithm divides the search range into three parts with four points. The left end, the left trisection point, the right trisection point, and the right end. Then it compares the function value of the left trisection point and the right trisection point. If the function value of the left trisection point is larger than that of the right trisection point, then the extremum must fall in the range [left end, right trisection point], and vice versa. Then the search range is reduced, and we can regard the right trisection point as the new right end, calculate the new left and right trisection points, and repeat the above comparing process. As the search range decreases as the iteration rounds increase, there

will be one iteration and the difference between the two trisection points will be 1. Then we find the extremum. The process can be summarized as Algorithm 1. This algorithm greatly reduces the time spent on searching. In the above illustration, the attacker completes the step of "set f_a, observe $T_{sampleresult}$" manually. If the signal generator supports software control, this step can be completed automatically.

Algorithm 1 Three divided algorithm for searching proper f_a

Input: frequency range $[f_l, f_h]$, time threshold t_{th}
Output: f_a that makes $T_{sampleresult}$ larger than t_{th}
Used sub-function: getPeriod(f): this sub-function returns the $T_{sampleresult}$ when the frequency of attack waveform is f.

```
Low = f_l
Up = f_h
left = Low+1/3·(Up-Low)
right = Low+2/3·(Up-Low)
while getPeriod(left)<t_th do
    if getPeriod(left)<getPeriod(right) then
        Low = left
        left=Low+1/3·(Up-Low)
        right=Low+2/3·(Up-Low)
    else
        Up = right
        left=Low+1/3·(Up-Low)
        right=Low+2/3·(Up-Low)
    end if
end while

return left
```

6.4 PSK Control

After ensuring the length of the $T_{sampleresult}$, the attacker needs to control the phase of the attack waveform to inject different information. PSK is a modulation method that can alter the phase of the baseband signal according to the PSK trigger. Some signal generators integrate this function in the device and set a pair of external trigger pins for users to modulate the baseband signal whenever they want. The user can control the PSK module with an MCU. He can connect a GPIO pin (we call this pin control pin) of the MCU to the trigger of the PSK module. The control pin will output a digital signal. When output logic 1, the PSK module will reverse the phase to 180°, when output logic 0, the phase will be turned back to 0°. Therefore, the attacker can use an MCU and choose one of its GPIO as the control pin. Then the attacker can control the phase of the attack waveform by configuring the MCU.

As the voltage of the control pin can determine the phase of the attack waveform, then it can determine the injected voltage. In the successful period, the relationship between the control signal of PSK, phase of attack waveform, original waveform, and received content is shown in Fig. 16. It shows that the injected content is the same as the control signal of PSK. Therefore, the attacker can directly connect the TX pin of his MCU to the PSK trigger. The output signal of the TX pin will be the same as the injected signal to the victim system.

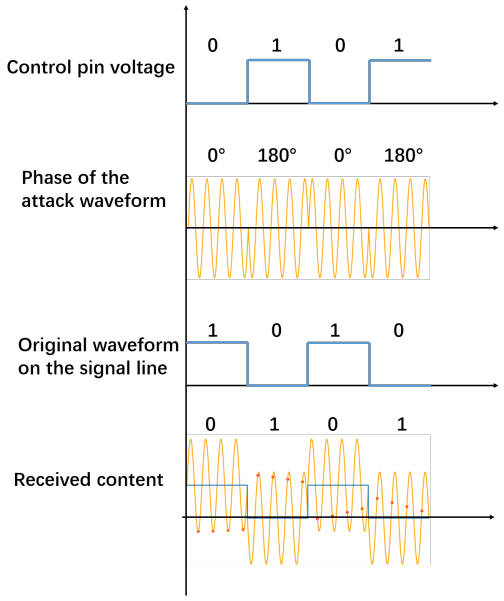

Figure 16: The attacker can use the control signal of PSK to control the injected content. In order to make the waveform concise and clear, four sampling points (instead of eight or sixteen sample points as in actual situation) are used here to determine the value of a bit

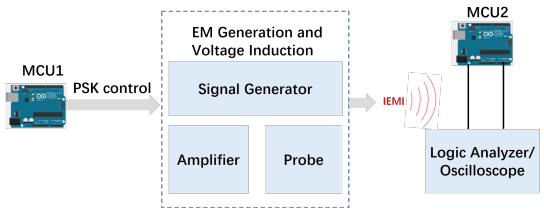

Figure 17: Setup to attack the logic analyzer.

7 MECHANISM VALIDATION

In this section, we validate the correctness of the attack mechanism using the method described in our attack design. We first use an oscilloscope to verify that we can inject large enough voltage, and then we use a logic analyzer to test that we can successfully inject meaningful information. The logic analyzer is an ideal device for this experiment because it can show every sample result and its supporting software can recognize the binary waveform according to the UART protocol. We connect the EM generation devices including the signal generator, amplifier, and probe as mentioned in the attack design, and connect the oscilloscope or the logic analyzer to an MCU called MCU1. We also use another MCU called MCU2 to control the PSK modulation. The diagram of the validation setup is shown in Fig. 17. To thoroughly validate the attack mechanism, we will prove our ability to inject large enough voltages first. Besides, we will verify that in the real-world attack, the victim's received bit is not constant but a stream of alternating "1" and "0". Lastly, we will verify that by phase reversion, the attacker can inject a meaningful string into the received results.

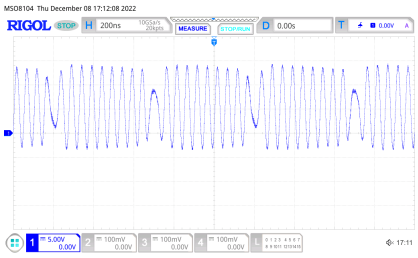

Figure 18: The induced voltage under PSK mode.

7.1 Ability of Injection

In this part, we launch some experiments to show we can inject large enough voltages into the signal line. We connect the pin of the MCU2 with the probe of an oscilloscope, then we can observe the waveform of the induced voltage directly. We generate and emit EM waves using the magnetic probe.

We test the amplitude of the injected voltage and the attacker's ability of phase reversion. For a 3.3 V CMOS device, its V_h, V_l, V_{oh} and V_{ol} are respectively 0.8 V, 2 V, 3.3 V and 0V. According to the Eq. (5), to attack this 3.3 V CMOS device, the amplitude of the voltage induced should be at least 2.5 V. As our experiment results show in Fig. 18, the amplitude of the induced voltage is 7 V. It proves that when the magnetic probe is 5 cm away from the victim device, the frequency of the current higher than 20 MHz can induce a sufficient voltage. Although the sample rate may be less than 20 MHz, we can generate a signal with a frequency that is close to 20 MHz and an integer multiple of the sampling rate. The phase of the attack waveform is determined by the voltage of the trigger, when the trigger gets a high voltage, the phase will be reversed. The results are shown in Fig. 18. The phase will change near 180° after the control pin output a high voltage to the trigger.

7.2 Meaningful information injection

In this experiment, we set the f_s of the logic analyzer and the f_a as 10 MHz. Before the attack, we set the TX pin of MCU2 as high voltage, then the results of the logic analyzer are all from the attacker's influence. After we begin our attack, the waveform of the logic analyzer is a square wave shown as Fig. 19 (a). The $T_{sampleresult}$ is short. Although both the f_s and f_a are set to be 10 MHz, the inherent deviation of the instruments makes differences between the two frequencies. Therefore, we use the method described in Section 6.3 to adjust the f_a until the period of the square wave is long enough. For example, when we set the frequency to 9.999939 MHz, the period of the square wave is about 5 s, which is long enough to inject complex UART information, as shown in Fig. 19(b). Then we test the effect of the phase reversion. We program the MCU1 to make its TX pin output high voltage for 10 ms, then output low voltage for 100 ms to control the PSK module. As explained in Section 6.4, the injected waveform will be the same as the PSK control signal in the successful period. The received waveform of the logic analyzer is shown as the black waveform, and the PSK control waveform is shown as the green waveform in Fig. 19(c). We can see that in the successful period, the green waveform is the same as the black one. In the failed period, the green waveform is exactly opposite to the black one.

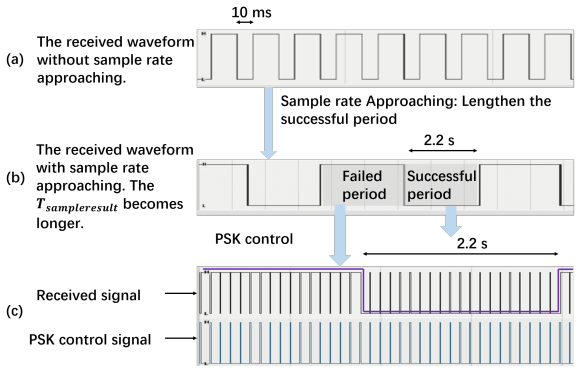

Figure 19: The experiment result of meaningful information injection.

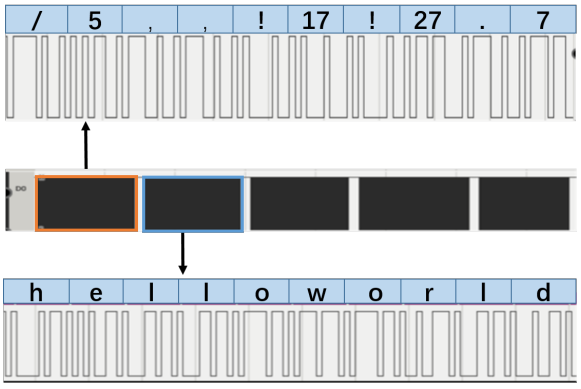

Figure 20: The waveform of the injected UART format string on the logic analyzer.

Then we try to make the MCU1 TX pin output digital signal in UART format. According to the results shown in Fig. 19, in the success period, the injected signal will be the same as the output of the TX pin. We make MCU1 output the string "helloworld". The received result of the logic analyzer is shown in Fig. 20. The blue box contains waveform in successful period, which is the same as the digital signal of the TX pin of MCU1. The waveform can be interpreted as the string "helloworld". The orange box contains waveform in the failed period. Therefore, the waveform is opposite to the waveform in the blue box, and it can be interpreted as a meaningless string.

8 EVALUATION

In this section, we evaluate the attack effect of the BitDance attack from different aspects. We describe our experimental setup in detail, test our attack ability with different parameters on six different sensors, and propose a case study to enhance our attack success rate under certain conditions.

8.1 Experiment Setup

The experiment setup is shown in Fig. 21. The attack devices include a RIGOL DG832 signal generator for EMI signal generation, a Mini-Circuits ZHL-100W-GAN+ power amplifier for EMI signal

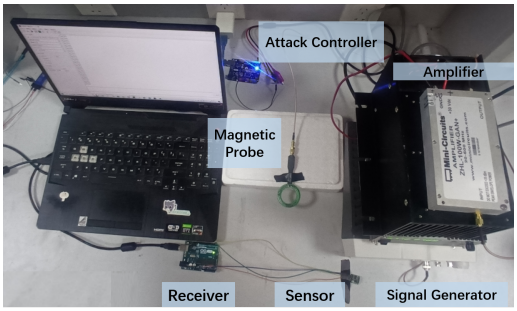

Figure 21: Setup of the evaluation.

amplifying, a convenient magnetic field probe (with a radius of 45mm and operating frequencies ranging from 9 kHz to 3 GHz) for EMI signal generation transmission, an Arduino UNO board for attack waveform phase control. And the victim MCU is connected to a sensor, and it's also connected to a computer to count the success rate. The circuit loop area is about 100 cm^2. If not otherwise specified, we maintain the experiment setup like this.

8.2 Attack Ability

In this subsection, we test the relationship between attack ability and other parameters. The attack ability is mainly represented by the amplitude of the induced voltage and attack success rate. According to Eq. (8), the amplitude of the induced voltage influences the successful period and then influences the success rate. In our experiment, we measure two kinds of attack success rates, creation success rate (CSR in short) and shield success rate (SSR in short). The method we use to measure the two kinds of success rates is as followed. Before the attack, the victim is receiving normal data from the sensor. Then we try to inject string "attack" into the victim every 10 ms and observe the received data for 5 s. If the attack is completely successful, the received content should consist of 500 numbers of the string "attack" without any normal data. Therefore, we measure CSR by counting the number of string "attack" that the attacker receives and dividing it by 500. We also count the number of normal data the victim receives and calculate SSR, which represents how many proportions of normal data are shielded. In this part, we use the acceleration sensor, MPU6050 as the sensor and we set the UART baud rate of the communication system as 9.6 kHz, so the sample rate is 153.6 kHz.

8.2.1 Distance. We evaluate the relationship between attack distance and attack ability. In our evaluation, since the attack device and the victim communication system are on the same plane, the attack distance is defined as the horizontal distance between the magnetic probe and the communication circuit loop. We set the frequency of the attack waveform to 15.36 MHz, which is 100 times of the sample rate, and the circuit loop area of the victim as about 100 cm^2. Fig. 22(a) shows the results of our experiments. The results show that as distance decreases, once we inject the signal successfully, the SSR rapidly increases from a low level to near its theoretical max limit (100%). That may be because there exists an application layer interface between the sensor and the MCU, and only several bits influenced leads to communication interference

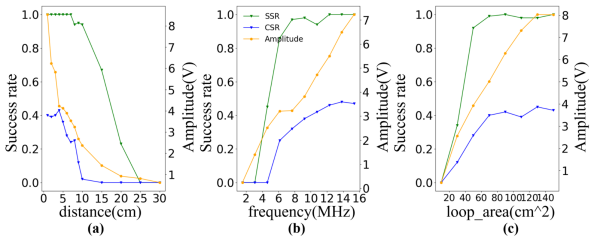

Figure 22: The relationship between attack capability and distance, frequency and loop area.

Figure 23: 6 sensors used in experiment

and shield the normal data. The upward trend of CSR is much more gradual and reaches its top at 42%. The amplitude of induced voltage increases as an inversely proportional function to distance, which is as we expect.

8.2.2 Frequency. We evaluate the relationship between the frequency of attack waveform and attack ability. In our attack, the frequency of the attack waveform will be an integral multiple of the sample rate. We set the distance between the attack device and the victim as 5cm, and the circuit loop area of the victim as about 100 cm^2. The result is shown in Fig. 22(b). Theoretically speaking, the amplitude of the induced voltage will be proportional to the frequency of the attack waveform. The induced voltage grows as the frequency grows, which agrees with our analysis. The two kinds of success rates also grow as the frequency grows. The effect of frequency on success rate is similar to that of distance, SSR reaches its theoretical limit quickly and CSR grows gradually.

8.2.3 Loop Area. We evaluate the relationship between the circuit loop area of the victim and the attack ability. The circuit loop area is the area enclosed by the signal line and the ground line on the plane where the magnetic field probe is located. According to (2), theoretically speaking, the larger the loop area, the more significant the magnetic flux will be injected, and the induced voltage will also be more significant. We set the distance between the attack device and the victim as 5cm and the attack frequency as 15.36 MHz. The result is shown in Fig. 22(c). When the loop area is small, the induced voltage increases as the loop area, and when the loop area gets larger, the circuit loop has already contained most of the magnetic flux. On this condition, increasing the loop area has a limited influence on the amplitude of the induced voltage.

8.3 Real Sensor Attack

We launch our attack on 6 different sensors, shown as Fig. 23, with 2 kinds of MCUs (Arduino UNO and STM32F103) as data receivers. The Arduino UNO only has one hardware serial port, therefore we use a software serial port to display serial communication reception

on the computer. The attack device is 5 cm away from the victim. In this experiment, we construct different UART frames according to the different application layer protocols of the sensors. We take the MHZ19 carbon dioxide gas sensor as an example, it uses 8 bytes to send the gas concentration to the MCU. The first bytes of the data must be 0xFF, according to its datasheet. Therefore, before we inject false data into this sensor, we inject a 0xFF into it. Also, the baud rate of communication changes as the victim sensors change. The result is shown in Table 2. The sensors with UART protocol have different default baud rates, we set the attack frequency near 20 MHz and integral multiples of the sample rate, which ensure the amplitude of EMI is large enough to successfully attack the victim. The result shows that almost all the sensors can be attacked successfully. The most SSR is 100%, which means the MCU does not receive any meaningful information. The most CSR is 45.4%, which means we can inject falsified data in about half of the time. And in the other half of the time, the attack makes the MCU receive meaningless data.

Table 2: Success rate of attack against those sensors. The left side of the success rate is CSR, and the right side is SSR.

Sensors	CSR	SSR	MCU
SFM-V1.7	36.4%	100%	Arduino
fingerprint sensor	38.8%	99.7%	STM32
MHZ19 carbon	41.2%	97.6%	Arduino
dioxide gas sensor	39.6%	97.4%	STM32
HX711 weight	45.4%	92.3%	Arduino
sensor	44.6%	86.8%	STM32
MPU6050	41.6%	98.5%	Arduino
acceleration sensor	37.8%	97.2%	STM32
MLX90614 THERMOMETER	43.2%	98.6%	Arduino
infrared sensor	42.8%	96.8%	STM32
TOF050F	39.4%	97.1%	Arduino
laser distance sensor	38.6%	98.2%	STM32

We find that if the SSR of the attack against a certain sensor is high, then the CSR of the attack against the same sensor will be relatively low. We explore the reason for this phenomenon. Beyond bare UART protocol, the sensors have application layer protocols to implement more complex functions. In the application layer, several bytes consist of meaningful instructions. The lengths of the sensors' instructions are different from each other. The more complex the application protocol is, the easier for the attacker to make it meaningless by injecting EMI noise, which means the SCR will be higher. However, it's also more difficult for the attacker to inject complex instructions, which means the CCR is lower. For example, the SFM-V1.7 fingerprint sensor has a complex application layer frame structure, it has 8 bytes of data as the data head. Therefore, it has the lowest CSR and the highest SSR. The HX711 Weight sensor simply transmits the data with 3 bytes, therefore it has the highest CSR and the lowest SSR.

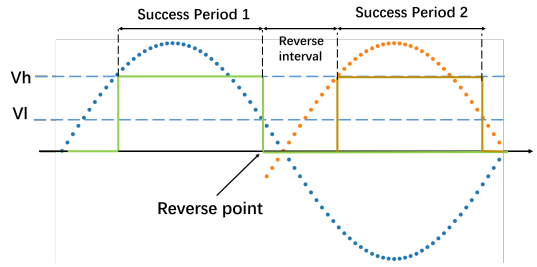

Figure 24: Phase reversion can increase the successful period.

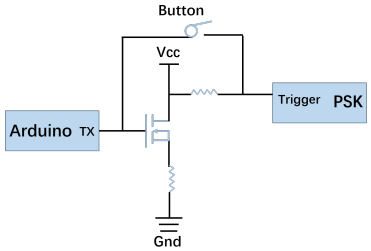

Figure 25: A button and an inverter are used to control the phase reversion.

8.4 Case Study: Increase CSR with Received Content

As the analysis and experiment above show, the theoretical upper limit of CSR is 50%, and the experiment results show that the CSR reaches up to 45.4%. We want to discuss that if the ability of the attacker is more than what we describe in the threat model, can the CSR be higher than 45.4% or even break the theoretical limit 50%? In this case study, we assume that the attacker can watch the received content of the victim. We discuss how the attacker can utilize this ability and enhance the attack effect.

Fig. 24 shows the fundamental of how to increase CSR. The original sample results are shown as blue dots in the figure. If the attacker can see the sample results, he will know that at the reverse point in the figure, the sample result will be logic 0. The attacker can reverse the phase of the waveform, and the sample results of the reversed waveform are shown as the orange dots in the figure. Then there will be another successful period (successful period 2 in the figure) in the same cycle. If the phase reversion happens in the reverse interval, it will not influence the sum of the successful period. Therefore, if the attacker can watch the sample results of the victim, the success rate will be doubled, and the theoretical upper limit of CSR will be 100%. In the real-world attack, although we cannot watch every sample result, we can infer when the creation attack fails by the received UART format content. If the MCU receives the correct injected data, it shows the attack is successful. When the MCU receives meaningless data or even receives frame error information, we know that here comes the reverse interval and we should manually reverse the phase of the attack waveform.

Then we launch the experiment to verify the analysis above. The setup of this case study is as follows. The RX pin of the Arduino is connected to the VCC of an STM32, and the ground pins of the

two MCUs are connected. We also connect the Arduino to our computer and watch the received content in real-time. We also use another Arduino to control the PSK module, but different from other experiments, we add a button and an inverter implemented by a MOSFET between the Arduino and the PSK module. The circuit is shown as Fig. 25. When the attacker pushes the button, the phase will be alternated by 180°, and the digital signal from TX will control the PSK on this basis. When he releases the button, the phase will go back to 0°, and the Arduino controls the PSK module as in other experiments. We watch the received content and push (or release) the button every time the received content is wrong. The CSR reaches up to 85.4%.

8.5 Discussion

Real-world examples that our attack potentially works on. Since our attack needs to inject magnetic flux into the loop area, we envision our attack could affect product that uses wire for UART communication instead of integrating the sender and the receiver into one PCB. Besides, we expect the victim system to stay static, because a moving target may get out of the effective attack range, and the target's movement will change the attack angle and distance, making it difficult to keep the magnetic field probe and the loop area in the same plane and control the strength of injected signals. We find several real-world examples where our attacks potentially work on. In the case of factory production, some of the equipments are both static and communicating in UART protocol by wires. For example, robotic arms are widely used in industrial automation. In robotic arms, the serial servo is one kind of commonly used servos that utilizes the UART protocol to control the specific rotation angle and torque of the robotic arm. Because of its size, it is hard to integrate a servo on a PCB, and many serial servos, such as ST3215 and LX-15D, use wire to communicate with the MCU, as shown in Fig. 26(a)(b). Besides, digital pressure sensors are also crucial to industrial production. Possibly to prevent the MCU from being damaged by external pressure, many digital pressure sensors, such as the ACD-302, also separate the sensing module from the MCU and connect them with wires, as shown in Fig. 26(c)(d). These products can be potentially influenced by our attack method.

Attack ability limitation. As illustrated above, our attack has limitations in several aspects when considering real-world scenarios. Firstly, the induced voltage depends a lot on the circuit loop area. Therefore, our attack method has limited influence on sensors integrated into the PCB because the area between PCB traces is small. Secondly, when the target is moving quickly, our attack method cannot work very well because of the reasons discussed above: the attack device is hard to move, the attack range is limited and it is difficult to keep the magnetic probe the same plane as the loop area.

9 COUNTERMEASURE

Several methods can be used to mitigate the impact of our attack. **Differential lines**. The basic idea of differential lines is to transfer information not by the absolute value of the voltage of a single line, but by the voltage difference between two lines. In most cases, the two lines are wrapped together, thus the EMI exerts the same influence on the two lines and the difference will stay the same.

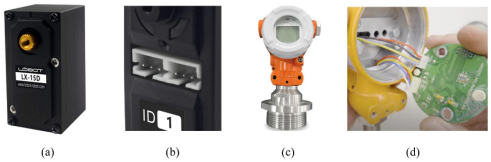

Figure 26: (a) The appearance of the LX-15D[1]. (b) The UART port of the servo[8]. (c) The appearance of the digital pressure sensors[2]. (d) After tearing down the device, it is found that the weight sensor is connected with MCU and display module by wires[24].

Low pass filter. As we launch a high frequency of EMI to pursue a larger distance, the receiver can connect a low pass filter to get rid of the high-frequency component of the outside signal before sampling. **Shielded cables**. Shielded cables are designed with a layer of shielding material that reduces their ability to emit and receive external electromagnetic waves. This shielding layer is typically made of conductive materials, such as copper or aluminum, that act as a barrier to EMI, which helps ensure that the voltage of the signal line remains stable and reliable.

10 CONCLUSION AND FUTURE WORK

In this work, we propose the BitDance attack, a new attack method to inject bit-level information with UART format into the serial communication system at a distance for the first time. We build a model to describe the process of an IEMI attack against the UART protocol, design and implement the attack on the ideal device like the logic analyzer, and examine the attack effect on 6 different kinds of sensors in the real world. In the future, we will explore how to control bit-level information for more than one lines.

ACKNOWLEDGMENTS

We appreciate the anonymous reviewers'valuable comments. This work is supported by China NSFC Grant 62222114, 62201503, 61925109, and 62071428.

REFERENCES

[1] Amazon. 2020. LX-15D High torque durable all metal gear Serial bus server. https://www.amazon.com/LewanSoul-Real-Time-Feedback-Temperature-Indicator/dp/B07G11VJWR
[2] ANCN. 2018. Pressure transmitter ACD-302. http://www.ancn.com/a/chanpinxilie/xilieyi/weishengxingyaliyibiao/2016/0321/425.html
[3] MJ Basford, C Smartt, DWP Thomas, and S Greedy. 2018. On the disruption of wired serial communication links by time domain interference. In *2018 IEEE International Symposium on Electromagnetic Compatibility and 2018 IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (EMC/APEMC)*. IEEE, 183–186.
[4] Donghui Dai, Zhenlin An, and Lei Yang. 2022. Inducing wireless chargers to voice out for inaudible command attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 503–520.
[5] Gökçen Yilmaz Dayanikli, Rees R Hatch, Ryan M Gerdes, Hongjie Wang, and Regan Zane. 2020. Electromagnetic sensor and actuator attacks on power converters for electric vehicles. In *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 98–103.
[6] Gökçen Yılmaz Dayanıklı, Abdullah Zubair Mohammed, Ryan Gerdes, and Mani Mina. 2022. Wireless Manipulation of Serial Communication. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 222–236.
[7] Bhag Singh Guru and Hüseyin R Hiziroglu. 2009. *Electromagnetic field theory fundamentals*. Cambridge university press.

[8] hiwonder. 2020. Hiwonder LX-15D Intelligent Serial Bus Servo with RGB Indicator for Displaying Robot Status. https://www.hiwonder.com/products/lx-15d

[9] Patrick Jattke, Victor van der Veen, Pietro Frigo, Stijn Gunter, and Kaveh Razavi. 2022. Blacksmith: Scalable rowhammering in the frequency domain. In *2022 IEEE Symposium on Security and Privacy (SP)*, Vol. 1.

[10] Yan Jiang, Xiaoyu Ji, Kai Wang, Chen Yan, Richard Mitev, Ahmad-Reza Sadeghi, and Wenyuan Xu. 2022. WIGHT: Wired Ghost Touch Attack on Capacitive Touchscreens. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1537–1537.

[11] Sebastian Köhler, Richard Baker, and Ivan Martinovic. 2022. Signal injection attacks against ccd image sensors. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 294–308.

[12] Denis Foo Kune, John Backes, Shane S. Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost Talk: Mitigating EMI Signal Injection Attacks against Analog Sensors. In *2013 IEEE Symposium on Security and Privacy*. 145–159. https://doi.org/10.1109/SP.2013.20

[13] Anil Kurmus, Nikolas Ioannou, Matthias Neugschwandtner, Nikolaos Papandreou, and Thomas Parnell. 2017. From random block corruption to privilege escalation: A filesystem attack vector for rowhammer-like attacks. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*.

[14] Anil Kurmus, Nikolas Ioannou, Matthias Neugschwandtner, Nikolaos Papandreou, and Thomas Parnell. 2017. Is there a "rowhammer" for MLC NAND Flash SSDs? An analysis of filesystem attack vectors.

[15] Seita Maruyama, Satohiro Wakabayashi, and Tatsuya Mori. 2019. Tap'n ghost: A compilation of novel attack techniques against smartphone touchscreens. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 620–637.

[16] Karol Niewiadomski, Robert Smolenski, Piotr Lezynski, Jacek Bojarski, David WP Thomas, and Frede Blaabjerg. 2022. Comparative Analysis of Deterministic and Random Modulations Based on Mathematical Models of Transmission Errors in Series Communication. *IEEE Transactions on Power Electronics* 37, 10 (2022), 11985–11995.

[17] AE Pena-Quintal, MJ Basford, K Niewiadomski, S Greedy, M Sumner, and DWP Thomas. 2020. Data Links Modelling under Radiated EMI and its Impact on Sampling Errors in the Physical Layer. In *2020 International Symposium on Electromagnetic Compatibility-EMC EUROPE*. IEEE, 1–5.

[18] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. 2019. Bit-flip attack: Crushing neural network with progressive bit search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1211–1220.

[19] Behrooz Sangchoolie, Karthik Pattabiraman, and Johan Karlsson. 2017. One bit is (not) enough: An empirical study of the impact of single and multiple bit-flip errors. In *2017 47th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE, 97–108.

[20] Jayaprakash Selvaraj, Gökçen Yılmaz Dayanıklı, Neelam Prabhu Gaunkar, David Ware, Ryan M Gerdes, and Mani Mina. 2018. Electromagnetic induction attacks against embedded systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 499–510.

[21] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. 2013. Non-invasive spoofing attacks for anti-lock braking systems. In *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 55–72.

[22] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *24th USENIX Security Symposium (USENIX Security 15)*. 881–896.

[23] Liwei Song and Prateek Mittal. 2017. Poster: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2583–2585.

[24] Hardcore teardown. "2023". Disassemble an industrial device called a digital pressure sensor. https://www.youtube.com/watch?v=E8wfSdKtP7s

[25] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In *2017 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 3–18.

[26] Yazhou Tu, Sara Rampazzi, Bin Hao, Angel Rodriguez, Kevin Fu, and Xiali Hei. 2019. Trick or heat? Manipulating critical temperature-based control systems using rectification attacks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2301–2315.

[27] Victor Van Der Veen, Yanick Fratantonio, Martina Lindorfer, Daniel Gruss, Clémentine Maurice, Giovanni Vigna, Herbert Bos, Kaveh Razavi, and Cristiano Giuffrida. 2016. Drammer: Deterministic rowhammer attacks on mobile platforms. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 1675–1689.

[28] Kai Wang, Richard Mitev, Chen Yan, Xiaoyu Ji, Ahmad-Reza Sadeghi, and Wenyuan Xu. 2022. GhostTouch: Targeted Attacks on Touchscreens without Physical Touch. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 1543–1559. https://www.usenix.org/conference/usenixsecurity22/presentation/wang-kai

[29] Zhengbo Wang, Kang Wang, Bo Yang, Shangyuan Li, and Aimin Pan. 2017. Sonic gun to smart devices: Your devices lose control under ultrasound/sound. *Black Hat USA* (2017), 1–50.

[30] David A Ware. 2017. *Effects of intentional electromagnetic interference on analog to digital converter measurements of sensor outputs and general purpose input output pins*. Ph. D. Dissertation. Utah State University.

[31] Zhifei Xu, Runbing Hua, Jack Juang, Shengxuan Xia, Jun Fan, and Chulsoon Hwang. 2021. Inaudible Attack on Smart Speakers With Intentional Electromagnetic Interference. *IEEE Transactions on Microwave Theory and Techniques* 69, 5 (2021), 2642–2650. https://doi.org/10.1109/TMTT.2021.3058585

[32] Yongrun Yu, Zhiwen Pan, Nan Liu, and Xiaohu You. 2019. Belief propagation bit-flip decoder for polar codes. *IEEE Access* 7 (2019), 10937–10946.

[33] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 103–117.

[34] Dejun Liu Zhiyong Guo. 2013. A Numerical Calculation Method for Spatial Magnetic Field of Circular Current. (2013).