PowerBridge: Covert Air-Gap Exfiltration/Infiltration via Smart Plug

Yongyu Liang¹,², Hong Shan¹,², Zelin Luo¹, Lanlan Qi¹,² and Yi Xie¹,²,*

Abstract: Power lines are commonly utilized for energy transmission, and they serve as a conduit for data exfiltration or infiltration in some specific scenarios. This paper explores the feasibility of establishing bidirectional communication between a modified plug and the equipment power line within an air-gapped network organization and with external entities. Bidirectional air-gap communication includes two scenarios, the data leak from air-gapped networks and the transmission of external data to air-gapped networks, namely, exfiltration and infiltration. In the exfiltration scenario, software in the air-gapped networks modulates and encodes data by manipulating the power consumption of the equipment during transmission, which is then sent outside through the power line. The device utilizes a smart plug power meter to record current fluctuations and subsequently decode any leaked data. In the infiltration scenario, a smart plug is used to control the power supply status of a device’s power cord, enabling data encoding and decoding by turning the power supply on and off. The software in the air-gapped equipment captures and decodes the power supply status to infiltrate. We discuss relevant literature and provide scientific background on smart plugs and power line communication. We simulate the communication scenario, propose a communication scheme, and present data modulation techniques as well as a communication transmission protocol for air-gap channels. Our evaluation of the PowerBridge air-gap channels demonstrates that data can leak from the air-gapped computer into the power line at an approximate rate of 30 bps, which can be captured by the smart plug. Additionally, it is possible for data to penetrate from the smart plug into air-gapped networks at a speed exceeding 1 bps.

Keywords: air-gapped channel; smart plug; exfiltration; infiltration; laptop

1. Introduction

The firewall is an access control device that separates internal and external networks, which are typically installed at their intersection points. The most common technology used in firewalls is packet filtering, which involves examining network packets, discarding those that do not comply with the settings, and allowing those that do to pass through. Intruders must first breach the firewall’s security measures in order to gain access to the target computer. By implementing such measures, strict network security policies can be enforced to prevent unauthorized data access. However, despite advancements in new breakthrough technologies and emerging firewalls, they cannot completely eliminate an unauthorized user’s ability to access computers and data [1].

Sensitive and important data should be protected by more secure tools to ensure that internal network data are not leaked or compromised. Many organizations adopt air-gap isolation to improve security, which involves establishing no physical connection between the internal network containing crucial user data and the external Internet, as shown in Figure 1. These works effectively isolate important information from potential threats and
significantly enhance security measures. Physical isolation is commonly implemented in government departments, security agencies, business operations planning units, vital scientific research divisions, and similar scenarios.

![Diagram of Air-Gapped Network](image)

**Figure 1.** Air-gapped isolation network.

In the past ten years, it has been proved that hackers’ destructive behavior cannot be completely blocked by firewalls and air-gap isolation. They can still complete the destruction through sophisticated technology, and malware can still infect the organization’s internal network. Computer users’ intentional or unintentional network crossing behavior, as well as the pollution in the software and hardware supply chain, may introduce malicious programs, which may lead to the infection of the entire organization’s intranet [2].

1.1. Network Covert Channel

When malicious programs invade air-gapped networks, they often aim to establish communication channels for command control and data transmission. To avoid triggering security device alarms such as firewalls, malicious programs usually utilize covert channels to establish communication. A network covert channel refers to a channel where the communicating parties encode and transmit information by modifying the value, characteristics, or status of shared resources. Over the years, academia has proposed various types of network covert channels, such as hybrid covert channels and behavioral covert channels [3]. These covert channels can typically evade detection by devices like firewalls in order to achieve long-term control over the target. However, traditional covert channels are based on network connections; without a network environment, an air-gap channel cannot be established.

1.2. Air-Gap Covert Channels

Air-gap isolation has always been considered as a simple and effective means for blocking intrusion, which places important computers in isolated environments without any physical connection to public networks, in order to prevent hackers from damaging or infiltrating them. However, many researchers have posed challenges to such methods. Security researchers have proposed different types of air-gap channels that can breach the air gap and establish communication with computers within an organization. These methods create invisible bridges to bypass the air-gapped system. These publicly available methods can be broadly categorized as follows: electromagnetic channel [4], thermal channel [5], acoustic channel [6], power line channel (PowerHammer) [7], and optical channel [8].

1.3. Smart Plug Security Risks

In recent years, security researchers have discovered that IoT devices commonly possess vulnerabilities that can be exploited remotely [9]. Due to their frequent connection to the Internet, smart plugs have also become a focal point for security researchers [10]. Ling et al. conducted an analysis of a typical smart plug system and, through reverse engineering, revealed its complete communication protocol, while identifying vulnerabilities that could potentially enable remote control of smart plugs [11]. Compromised and
controlled smart plugs pose security risks and privacy breaches for home users, as well as potential catastrophic consequences for commercial or industrial users [12]. The safety concerns surrounding smart plugs have garnered significant attention from both industry and academia in recent years [9]. More alarmingly, Microsoft has recently reported that 80% of businesses encountered firmware threats at least once within the past two years [13].

In 2019, the McAfee team found that the Wemo Insight smart plug might be remotely turned off or overloaded; such a vulnerability potentially causes overheating or a power outage in the plug and even serves as a gateway to damage more devices [14]. In 2020, Asif Iqbal et al. conducted a comprehensive analysis of evidence collection for criminal activities using smart plugs [15]. The vulnerability of smart plugs makes them susceptible to remote control. However, there is currently no research on utilizing smart plugs to breach air gaps and establish bidirectional air-gap channels.

1.4. Our Contribution

We propose a new communication model that can bridge the air gap by utilizing smart plugs. Technically, we can achieve air-gapped host data acquisition by monitoring the energy consumption of computers through smart plugs. Additionally, the switch behavior of smart plugs enables us to send commands and data to infected laptops within the air-gap network. We discuss the required software and hardware environment and operations for connecting the air-gapped networks. Furthermore, we propose detection methods and preventive measures for such threatening behaviors.

- To the best of our knowledge, this is the first air-gap channel that utilizes smart plugs to achieve bidirectional communication through an air-gapped network. It employs smart plugs for power consumption detection and incorporates manufacturing power switches for bidirectional communication. In comparison to the existing PowerHammer method, which enables only one-way communication, our approach achieves bidirectional communication on the laptop end.
- We conducted extensive experiments on smart plugs and various laptop models to determine the effectiveness and maximum communication speed of PowerBridge.
- The code does not require special permissions (such as root or administrator). Users can operate from ordinary user-level processes or applications with a wide range of security risks.
- PowerBridge utilizes a standard CPU to enable air-gap communication, enabling the establishment of communication channels on nearly any computer or device equipped with a CPU.

The rest of this article will have the following arrangement: The second part will elaborate on threat scenarios in detail. Next, in the third section, we will introduce the relevant research results of previous scholars. Subsequently, in the fourth section, we provide the technical background for implementing PowerBridge based on smart plugs. Then, in the fifth section, we discuss issues related to data encoding and decoding. The content of signal generation is described in Section 6. Furthermore, Section 7 provides a detailed introduction to evaluation and analysis methods. Finally, an in-depth discussion is conducted in Section 8, and countermeasures are proposed.

2. Bridge Scenario

We propose an air-gap channel based on smart plugs, which can be applied in two main scenarios; Figure 2 illustrates the exfiltration and infiltration of data. In the data exfiltration scenario, sensitive files or device passwords are leaked from the air-gapped state to the outside. In the infiltration scenario, external data are infiltrated into computer equipment with air-gap isolation.
2.4. Dangerous Places

Figure 4. Scenes of data infiltrating and exfiltrating through smart plugs.

2.1. Preparation Work

The academic community generally believes that the implementation process of APT can be divided into five steps: information collection, the breakthrough of defense lines, the establishment of channels, horizontal destruction, and external dissemination [16]. In this study, our main focus is on information collection, channel establishment, and information dissemination. Before formal implementation, testers will extensively collect information on specific organizational network systems and related employees and then carry out targeted technical preparation work [17].

After completing technical preparations, malicious attackers will use illegal means such as CD drives, USB devices, supply chain pollution, and social engineering to infiltrate the target’s internal environment with pre-prepared programs, infecting both the target computer and smart plugs. It should be noted that numerous cybersecurity incidents have demonstrated that even networks with robust protection measures can still possess vulnerabilities that may eventually be exploited, leading to network intrusion [18].

After completing these tasks, a foothold can be established within the organization’s internal network for the purpose of information gathering and dissemination.

2.2. Exfiltration

The exfiltration scenario is illustrated in Figure 3. Malicious software actively scans the targeted network for pertinent content and subsequently utilizes a designated mechanism to transmit it to a predetermined destination.

Figure 3. Schematic diagram of the air-gap channel for data exfiltration, where malware on the air-gap workstation modulates binary information onto power line transfer device power consumption and is intercepted by a nearby smart plug.

- Data transmission: Malware utilizes the CPU and other hardware to manipulate power consumption fluctuations in devices, encoding and transmitting data through the power line. The attacker can schedule a specific time for malware transmission
or send it after receiving instructions from the attacker, which involves bidirectional communication.

- Data reception: The signal transmitted by the air-gapped device can be received by the smart plug that provides power to it. Subsequently, the received signal will be recorded and forwarded to the receiver for decoding. The hardware responsible for receiving signals may either be deployed by internal personnel or consist of IoT devices that have already been remotely controlled.

2.3. Infiltration

The infiltration scenario is illustrated in Figure 4. This article assumes the contamination of the smart plug. It is well-known that smart plugs can be remotely controlled by users in real time or programmed to regulate device power supply activation and deactivation. When the power supply of a device is turned off, it ceases to function and becomes detectable by the user. However, laptops powered by batteries do not encounter this issue as they possess both hardware interfaces [19] and software interfaces (ACPI) [20] for monitoring power supply status. By remotely controlling the initiation and termination of the smart plug power supply, laptops can receive information regarding their power supply status, which can then be transmitted to internal software for decoding.

![Figure 3. Schematic diagram of the air-gap channel for data exfiltration, where malware on the air-gap workstation modulates binary information onto power line transfer device power consumption and is intercepted by a nearby smart plug.](image)

![Figure 4. Schematic diagram of the air-gap channel for data infiltration. Attackers use a smart plug to transmit data to a laptop, which is then received by software on the laptop.](image)

2.4. Dangerous Places

The most critical scenario arises when a computer equipped with air-gap isolation becomes compromised, along with the smart plug that supplies it. This compromise not only ensures the existence of a reliable air-gap channel. Although environments with stringent safety requirements may not incorporate smart plugs within their air-gapped setup, establishing an air-gap channel remains challenging. However, the water pit attack poses a perilous trap [21]. Consider a situation where an unsuspecting victim brings an important computer to an unfamiliar setting like a conference room or hotel and relies on an external power source for energy, as depicted in Figure 5. If this power supply device transforms into a malicious trap, it inadvertently provides avenues for infiltration and data exfiltration.
hence, researchers focus on exploring the possibilities of establishing the second type. 

3. Related Work

Covert channels can generally be classified into two types: the first type involves network-based methods, such as modifying TCP and UDP packet structures or exploiting packet time delays and other characteristics. The second type encompasses air-gap channels that leverage the physical properties of devices, including sound, light, magnetism, and heat. In scenarios where sensitive and critical computers are prohibited from connecting to external networks, establishing the first type of air-gap channel becomes infeasible; hence, researchers focus on exploring the possibilities of establishing the second type.
Air-gap channels can be established by transmitting signals such as electromagnetic, acoustic, optical, and power line signals.

The utilization of electromagnetic signals to establish air-gap channels has a long-standing history, yet researchers continue to explore novel methodologies in this field. In 2020, Zhang, Zhao et al. demonstrated BitJabber [23], the utilization of memory access for modulating electromagnetic (EM) signals generated by DRAM clocks to construct channels with a bandwidth reaching up to 300,000 bps and the capability to penetrate concrete walls up to 15 cm thick.

In 2022, Guri et al. developed AIR-FI [24], which enables signal transmission exclusively through a memory bus and the interception of these signals via devices equipped with WiFi capabilities, such as smartphones, laptops, and IoT devices. In 2024, Guri et al. discovered RAMBO [25], which generates radio signals from the memory bus (RAM) and employs software-defined radio (SDR) hardware along with readily available antennas for signal reception at communication speeds of up to 1000 bit/s.

Vibration and sound waves offer alternative approaches. In 2021, Guri successfully addressed the air-gap channel by utilizing vibration, enabling Air-ViBeR [26] to adjust its internal fan speed through acceleration sensors in smartphones for signal reception. In 2019, Giechaskiel et al. examined the technique of signal injection using mobile phone microphones [27]. In 2022, de Gortari Briseno et al. demonstrated InkFiltration [28], a method employing malicious software to generate specific sound signals during computer printing that can be captured and decoded by nearby acoustic recording devices like smartphones; it achieves a communication distance of up to 4 m with an average bit rate of approximately 0.5 bps.

Optics offers an alternative approach. In 2021, Niclas et al. demonstrated Laser-Shark [29], a technique that utilizes modulated laser aiming devices to operate LEDs on the CPU GPIO interface and capture their flicker, enabling long-distance (25 m), bidirectional air-gap communication channels. In 2022, Guri et al. presented ETHERLED [30], which manipulates the flashing and alternating colors of Network Interface Controller (NIC) LEDs through malicious software to transmit signals. In 2023, Schlauder et al. introduced CD-BLink [31], as a method for extracting data from an air-gap network by modulating the read and write LEDs on an optical drive.

In 2018, Guri et al. demonstrated that PowerHammer [7] effectively regulates system power consumption by deliberately adjusting CPU utilization. The data are modulated, encoded, and transmitted based on current fluctuations, which are then propagated through power lines. In 2023, Guri et al. further presented POWER-SUPPLAY [32], a technique utilizing a computer’s power unit (PSU) to generate sound covertly and transmit it to nearby receivers like smartphones.

We primarily investigated air-gap channels capable of breaching air-gap isolation and presented the channel design, process, analysis, and evaluation. Table 1 enumerates the prevalent existing air-gap communication channels.
<table>
<thead>
<tr>
<th>Air-Gap Channels</th>
<th>Communication Medium</th>
<th>Sender</th>
<th>Receiver</th>
<th>Method Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electromagnetic</td>
<td>Electromagnetic</td>
<td>Memory, hard disk</td>
<td>Smartphone, SDR</td>
<td>BitJabber, AIR-FL, RAMBO</td>
</tr>
<tr>
<td>Sound</td>
<td>Ultrasound, noise, vibration</td>
<td>Loudspeaker, Fan, Power adapter</td>
<td>Mobile phone accelerometer, microphone, gyroscope</td>
<td>AIR-ViBeR, InkFiltration</td>
</tr>
<tr>
<td>Thermal</td>
<td>Radiation of heat air conditioner Infrared sensor</td>
<td>Infrared sensor</td>
<td>HVACKer</td>
<td></td>
</tr>
<tr>
<td>Power sector</td>
<td>Power line</td>
<td>Power adapter</td>
<td>Current transformer, smartphone</td>
<td>PowerHammer, POWER-SUPPLaY</td>
</tr>
</tbody>
</table>

The PowerBridge air-gap channel we propose distinguishes itself from previous research in the following aspects:
Robust universality. Enhanced universality is achieved by leveraging the inherent power measurement function of widely adopted smart plugs, eliminating the need for additional hardware implantation and thereby enhancing the versatility of the PowerBridge air-gap channel.

4. Power Line Communication and Smart Plug

Bidirectional communication. The PowerBridge is a bidirectional channel capable of both exudation and infiltration, surpassing the limitations of existing PowerHammer air-gap channels that are restricted to exudation only. Furthermore, we conducted an evaluation of our approach on a laptop and successfully demonstrated the efficacy of this channel.

4.1. Power Line Communication

Power line communication (PLC) pertains to the utilization of power lines and their transmission and distribution networks as a medium for communication technology and system applications [33], which has witnessed an escalating deployment in recent times. Due to its utilization of existing power grids for communication, it offers a solution without the need for additional wiring in network environments such as hotels and duplex residences that lack wired networks or face routing difficulties (as depicted in Figure 8). Manufacturers of PLC devices report shipping millions of such devices annually with expectations for continued growth in the future [34]; the common topology of the power cat is shown in Figure 9, and the electrical schematic is shown in Figure 10. PLC not only supplements wireless network solutions but also emerges as a formidable competitor; hence, extensive research is being conducted in this field. Images of common electric cats are shown in Figure 11.
4.2. Smart Plug

A smart plug is an Internet-connected device designed to remotely control and automate the use of appliances and devices in homes and businesses, common smart socket usage scenarios, as shown in Figure 12. These small devices are installed on standard power plugs, allowing users to easily control their appliances from anywhere without physical interaction. Smart plugs can also provide energy consumption information, notify users of

Figure 8. Power line WiFi cross-layer coverage.

Figure 9. Power line WiFi schematic diagram.

Figure 10. Power line communication principle diagram.

Figure 11. Images of common electric cats.

4.2. Smart Plug

A smart plug is an Internet-connected device designed to remotely control and automate the use of appliances and devices in homes and businesses, common smart socket usage scenarios, as shown in Figure 12. These small devices are installed on standard power plugs, allowing users to easily control their appliances from anywhere without physical interaction. Smart plugs can also provide energy consumption information, notify users of
excessive electricity usage, and offer timer functions that can be programmed according to specific events or conditions.

Figure 12. Intelligent plug usage scenarios.

With the increasing popularity of smart home technology and the proliferation of Internet-connected devices, it is anticipated that the market for smart plugs will experience significant growth in the coming years.

The global market for smart plugs is projected to reach a value of 9.24 billion US dollars by 2030, exhibiting a compound annual growth rate of 26.1% from 2023 to 2030 [35]. Furthermore, it is anticipated that the shipment volume will attain approximately 104 million units by the year 2030 [36]. In the context of smart homes, these plugs can be directly integrated into wall plugs or connected in series between power lines and appliances through standard power plugs, thereby enabling users to conveniently manage electrical equipment remotely at their convenience.

The structure of smart plugs primarily comprises power conversion modules and energy acquisition modules; their electrical structure is shown in Figure 13. The electric energy acquisition module transmits the collected current information to the WiFi communication module for external device transmission [37]. This enables intelligent measuring plugs to monitor real-time data on the current, electricity, voltage, and power consumption of electrical equipment while also offering early warning capabilities for voltage peaks. Many smart plugs provide detailed reports on energy consumption and usage patterns of appliances or devices, which facilitate informed decision-making regarding energy conservation strategies or upgrades.

Figure 13. Smart plug electrical structure diagram.

In this paper, we propose a novel bidirectional communication channel called PowerBridge for air-gapped computers, which combines a controlled smart plug with Guri et al.’s PowerHammer model.

4.3. Computer Power Management

The current computer system encompasses a wide range of devices, including desktops, mobile platforms, workstations, and servers. These devices primarily utilize the ACPI (Advanced Configuration and Power Interface) for efficient power management.
Developed collaboratively by Intel, Microsoft, and Toshiba in 1997, the ACPI offers comprehensive power management interfaces for operating system application management. The latest version available is 5.0 [38]. The ACPI serves as a powerful specification that allows an operating system to obtain hardware status information such as the source of power (battery or external supply). Consequently, third-party software programs can also determine whether the computer is currently running on battery or connected to a power source based on the provided power management interface offered by the ACPI. The architecture of the ACPI is illustrated in Figure 14 [39].

![ACPI Architecture Diagram](image)

**Figure 14.** The ACPI is the power management architecture.

In practical circuits, the power adapter serves as the central module of the entire system. The AC–DC adapter is commonly employed to convert AC main power into DC power, enabling direct supply to various electronic devices. To enhance power supply efficiency, it becomes imperative to switch between the low-power mode and the normal mode. Figure 15 illustrates common physical components utilized in this process. Initially, the input conditioning circuit rectifies and filters the AC signal, generating the first AC signal that drives the primary winding of the transformer. The secondary winding then converts this into a second AC signal with a desired amplitude. Finally, through regulation and power transmission circuitry, further rectification and filtering of the AC signal occurs, resulting in the generation of DC output voltage, Vout. Figure 16 depicts an electrical structure typical for a standard power adapter [40].

![Power Adapter Image](image)

**Figure 15.** Physical picture of common power adapters.
plans. Notably, one of these commands, '/batteryreport', generates a report detailing the power consumption of the smart plug and transmits it to a remote receiver for decoding, thereby completing the exfiltration process.

For infiltration scenarios, smart plugs serve as intermediary transfer bridges to transmit commands to air-gapped computers through power lines. The Advanced Configuration and Power Interface (ACPI) provides an interface for detecting the external power supply status, enabling the computer operating system itself to offer tools for accessing this interface, such as the `powercfg` command on Windows systems and the `upower` command on Linux systems. These tools can be utilized to manage power plans, utilize available sleep states, control individual device power statuses, and analyze common energy efficiency and battery life issues within the system. The Windows operating system’s `powercfg` command encompasses various functions including creating, initiating, or pausing power plans. Notably, one of these commands, `'/batteryreport'`, generates a report detailing battery usage (as depicted in Figure 17).

There are two methods for communication between the air-gapped computer and the smart plug: the exfiltration and infiltration scenarios.

In the exfiltration scenario, a fluctuation signal of computer power consumption is generated using the CPU, which is then captured by the power measurement module of the smart plug and transmitted to a remote receiver for decoding, thereby completing the exfiltration process.

In practical circuits, the power adapter serves as the central module of the entire system. The ACPI is the power management architecture. For infiltration scenarios, smart plugs serve as intermediary transfer bridges to transmit commands to air-gapped computers through power lines. The ACPI Firmware provides an interface for detecting the external power supply, such as the ‘powercfg’ command on Windows systems and the ‘upower’ command on Linux systems. These tools can be utilized to manage power plans, utilize available sleep states, control individual device power statuses, and analyze common energy efficiency and battery life issues within the system. The Windows operating system’s ‘powercfg’ command encompasses various functions including creating, initiating, or pausing power plans. Notably, one of these commands, `'/batteryreport'`, generates a report detailing battery usage (as depicted in Figure 17).

The graph clearly illustrates the logged records of whether the laptop is powered by an external power supply or a battery. These log records enable analysis of the timing when the external power supply is initiated and disconnected, i.e., when the smart plug switch is turned on and off, providing a viable pathway for infiltration.

5. Channel Signal Generation

In this section, we will elucidate the signal generation methodology employed for air-gap channels.
5.1. Exfiltration (Creating Power Line Current Fluctuations through the CPU)

The computer’s electrical hardware modules primarily consist of the CPU, motherboard, GPU, and hard drive. While CPUs and GPUs possess the highest power consumption, not all computers are equipped with discrete graphics cards [41]. Instead, many utilize internal display modules, which allow for leveraging the CPU to generate current fluctuations in order to enhance compatibility across systems since all computers are equipped with CPUs. By employing a simple dead loop within a single-core CPU, nearly maximum power utilization can be achieved; whereas in multi-core CPUs, engaging multiple threads within the dead loop can fully load all cores of the CPU. Henceforth, this principle offers an effortless means to construct signals exhibiting power fluctuations. Figure 18 illustrates the comparison of low and high CPU loads on the total power consumption of several brands of laptops.

![Figure 18. Comparison chart of power consumption among 5 popular laptop brands.](image)

Universality: The operating system of a computer grants each process the CPU utilization privilege without requiring separate permission requests, thereby minimizing the detection or termination of malicious code responsible for signal transmission by conventional security software.

To assess and evaluate the risks associated with this air-gap channel, we have developed a simulation program that leverages CPU power fluctuations for signal transmission. The program initially reads the relevant data and converts them into the communication encoding format (2ASK), such as the binary format. Subsequently, it is modulated to align with CPU power consumption levels. A low load condition, indicating reduced power consumption over a duration of T, represents 0; whereas a high load condition, indicating increased power consumption over a duration of T, represents 1. At the receiving end (smart plug), the signal can be decoded using a protocol based on received power consumption signals. Algorithm 1 outlines the fundamental process of transmitting signals based on CPU power fluctuations.

**Algorithm 1** CPU Power consumption signal modulation algorithm

1. **Input**: Number of threads, Payload, payloadSize, Cycle
2. binBuff[] = payload
3. i = 0
4. **Do**
5. If (binBuff[i++] == 1) HighCPUload(Cycle);
6. **else**
7. LowCPUload(Cycle)
8. **While** (i < payloadSize)
5.2. Infiltration (Control Power On and Off through Smart Plug)

As laptops are capable of detecting the power supply status, the smart plug switch can serve as a signal to transmit data to the ACPI interface of the laptop and subsequently relay it to an air-gapped computer. To facilitate testing and evaluation, this study presents a simulation system incorporating a modified plug that encodes binary data. The plug comprises MCU chips, WiFi modules, and relay modules. The WiFi module facilitates remote network communication, while the MCU chip receives commands and transmits received exfiltration signals. Additionally, it simultaneously controls the opening and closing of relays for powering on/off purposes. The remote control MCU is responsible for transmitting and receiving binary data based on an encoding scheme.

6. Data Encoding and Decoding

In this section, we will delve into the data encoding and decoding techniques employed in PowerBridge communication. It is worth noting that power line communication technology has been extensively researched in both academic and industrial domains, resulting in a plethora of proposed modulation, demodulation, encoding, and decoding schemes [42]. We present an overview of the modulation and encoding schemes utilized in PowerBridge communication while elucidating their distinctive characteristics.

6.1. Communication Modulation

6.1.1. Modulation of Exfiltration Communication

In the event of exfiltration, we can manipulate the CPU load size to influence the amplitude of the current signal or regulate the frequency of current signal fluctuations by controlling load duration and utilizing inter-load time intervals. Amplitude and frequency represent two pivotal parameters in a signal, enabling diverse modulation techniques such as amplitude-shift keying (ASK), frequency-shift keying (FSK), and phase-shift keying (PSK) through their manipulation. Subsequent sections will elaborate on these three modulation methods.

1. ASK is a digital modulation technique that utilizes the amplitude of the carrier wave to transmit information by varying the digital baseband signal. The program can regulate the power line current’s magnitude (amplitude) through different CPU load sizes. In the Binary Amplitude Key (2ASK), binary digits “0” and “1” correspond to two distinct states of carrier amplitude, and analog modulation and keying methods can generate the 2ASK signal. Although this modulation algorithm is relatively straightforward and intuitive, it suffers from low communication rate issues. To enhance transmission rate, multi-amplitude-shift keying (MASK) can be employed, where M represents the controllable number of CPU cores. MASK exhibits two notable characteristics: Firstly, in the same channel code source modulation scheme, each symbol has the capacity to carry log2M bits of information. Consequently, when the channel frequency band is constrained, it enables an increase in the transmission rate of information and enhances frequency band utilization. However, this advantage comes at the expense of amplified signal power and increased implementation complexity. Secondly, for a given information rate, multi-level base methods exhibit lower channel transmission rates compared to binary methods. As a result, the duration of multi-level signal code sources is wider than that of binary methods. Expanding the symbol width augments energy within signal symbols, while simultaneously mitigating the inter-symbol interference arising from channel characteristics [43]. Although multi-level MASK modulation serves as an efficient transmission technique, its susceptibility to noise and fading renders it more suitable for scenarios with minimal channel interference. Figure 19 shows the waveform of computer power consumption; with 0–8 cores gradually reaching full load on an 8-core CPU, it can display 24-bit binary sequence data.
2. FSK is a digital modulation technique that utilizes the frequency of the carrier wave to transmit information by varying the digital baseband signal. It is well-known that the signal frequency can be manipulated by adjusting the time interval between two CPU loads. Hence, in an exfiltration scenario, binary information “0” and “1” correspond to two distinct frequencies \(f_1\) and \(f_2\) of the carrier signal, respectively. The implementation of 2FSK is relatively straightforward, exhibiting excellent performance in terms of noise and attenuation resistance, making it extensively employed for medium- and low-speed data transmission applications. When employing a multi-base system for modulation, it is referred to as MFSK, which employs multiple carrier frequencies to represent diverse methods of digital information modulation. Figure 20 illustrates both time-domain signals and the corresponding spectra of a synthesized 4-frequency FSK achieved through fluctuations in CPU power consumption at frequencies of 7 Hz, 13 Hz, 17 Hz, and 23 Hz.

3. PSK is a digital modulation scheme that utilizes the phase of the carrier signal to encode input information. Phase-shift keying can be categorized into two types: absolute phase shift and relative phase shift. Absolute phase-shifting refers to phase modulation based on the unmodulated carrier’s initial phase. Taking binary phase modulation as an example, when the symbol is set to “0”, the modulated carrier remains in phase with the unmodulated carrier; whereas, when the symbol is set to “1”, the modulated carrier undergoes a 180° inversion from its original state. The waveform representing the difference in carrier phases between “0” and “1” exhibits a 180° change, and for 2PSK, it follows a modulation pattern of “01001001”, as illustrated in Figure 21.
Figure 21. Waveform during the modulation load of 2PSK. (a) PSK signal modulation; (b) PSK time-domain signal.

6.1.2. Modulation of Infiltration Communication

In the infiltration scenario, it is feasible to exert control over the power state of the smart plug and transmit signals to the laptop. In order to ensure computer and power adapter safety, caution should be exercised regarding the frequency of power switching, as excessively high frequencies may lead to damage to electrical appliances. Consequently, conventional ASK, FSK, and PSK modulation techniques are unsuitable for implementation in this particular scenario. To address this concern, pulse interval modulation (PIM) can be employed for communication purposes by representing data through distinct time durations between two consecutive pulses [44]. A reference time interval denoted as $T$ is defined within the standard document; it represents the temporal width between adjacent pulses with a duration of $T$. The intelligent plug generates a pulse each time it switches on or off, thereby enabling the modulation of signals based on the temporal intervals between these pulses. Figure 22 illustrates the waveform, while Table 2 presents an encoding table associated with PIM technique usage. The receiving software installed on the laptop captures and analyzes these temporal intervals between successive pulses before querying against the encoding table for decoding purposes.

![Waveform during the modulation load of 2PSK.](image)

Figure 22. Pulse interval modulation.

<table>
<thead>
<tr>
<th>Interval (s)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>bits</td>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
</tr>
</tbody>
</table>

6.2. Data Packets

We transmit data in the form of 56-bit frames, which consist of a frame header, payload, and frame tail. The frame header includes an 8-bit preamble, while the payload contains 40 bits. Finally, the frame tail comprises an 8-bit checksum as illustrated in Figure 23.
The frame header, transmitted at the beginning of each packet and comprising eight alternating symbols ("10101010"), facilitates the receiver in determining the channel signal’s starting position. Moreover, it enables synchronization with transmission initiation and the calibration of other parameters, such as the current frame number and device power. The data frame represents the actual binary data to be transmitted for each data frame, with a payload size chosen as 40 digits. To detect and rectify transmission errors, an 8-bit RS error correction code is inserted at the end of the frame. The receiving end calculates this RS error correction code for the received payload, enabling the correction of any errors in data bits during transmission. An 8-bit RS error correction code can correct up to 4 bits of data at any given position [45].

6.3. Signal Reception and Data Decoding

PowerBridge utilizes power lines as a communication medium, resulting in half-duplex communication due to the inability of simultaneous exfiltration and infiltration communication.

In the exfiltration scenario, the software embedded in the air-gapped computer captures power consumption fluctuations from hardware devices to transmit signals through series-connected smart plugs on the power line. Subsequently, these fluctuating signals are forwarded by smart plugs to the receiver and decoded using an algorithm. This approach bears a resemblance to Guri’s PowerHammer technique, which will not be reiterated within this article.

In the infiltration scenario, the current signal is transmitted by remotely controlling the switch of the smart socket to the open and closed positions. The laptop detects the power supply state through the ACPI interface, switches to battery power when disconnected from the external power supply, and logs this information. The computer software utilizes the kernel.dll dynamic library’s API interface GetPowerStatus to detect and record ACOff and ACon as external power status indicators, enabling the determination of signal timing for data decoding. Ultimately, communication content can be decoded based on the agreed-upon communication protocol. Algorithm 2 describes the process of PIM-based signal decoding.

When utilizing the powercfg command to access the system power usage log, it is observed that the logging time interval is significant, resulting in a failure to record power status changes occurring within less than 3 s. This limitation can lead to a reduced data reception rate at the receiving end. Through experimentation, we have discovered that the GetPowerStatus API provided by the Windows operating system serves as an interface function exported from the dynamic library ‘kernel.dll’. This API enables querying of the computer power supply status and related information. Consequently, employing this API for rapid polling effectively resolves the issue of extensive logging time intervals and enhances communication speed.
**7. Evaluation and Analysis**

In this section, we present the evaluation results based on our experiments and analysis, including the maximum bit rates for both communication directions of exfiltration and infiltration; the transmission mode is shown in Table 3. For the exfiltration scenario, we conducted tests using the laptop’s CPU to induce power fluctuations for data exfiltration. We utilized the emitted current signal from the laptop to covertly transmit binary encoded data. In the case of the infiltration scenario, we simulated a smart plug to control power states (on/off) in order to achieve the data infiltration scheme. Additionally, we analyzed how smart plugs can be employed for signaling laptops. To establish two-way communication with an air-gapped computer in exfiltration and infiltration scenarios, we implemented the setup depicted in Figure 24.

<table>
<thead>
<tr>
<th>Transmission Mode</th>
<th>Sender Unit</th>
<th>Receiver Unit</th>
<th>Content of Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exfiltration (leaking data from an air-gapped computer)</td>
<td>Power adapter for your computer</td>
<td>Smart plug power consumption measurement module</td>
<td>Small amounts of data (sensitive files, passwords, etc.)</td>
</tr>
<tr>
<td>Infiltration (sending data to the air-gapped computer)</td>
<td>Smart plug switch module</td>
<td>ACPI interface for the computer</td>
<td>Control parameter</td>
</tr>
</tbody>
</table>

Figure 24. The smart plug is designed for this work.

7.1. Exfiltration

The technology and solutions of power line communication have been extensively applied in commercial and household scenarios, with their maximum bit rate and distance primarily determined by the power and frequency band of the PLC transmitter. Typically, these applications are limited to within the household range, where longer communication distances generally result in lower communication rates. Notably, Guri et al.’s PowerHammer study successfully demonstrated that computers themselves can serve as hardware for signal transmission without requiring additional equipment.

In this section, we exclusively examine the distinctive characteristics of the plug as a receiver. For more comprehensive insights into visible power line communication and IoT security, we recommend interested readers refer to the relevant literature [46]. To conduct testing and an evaluation, we employed the transmitter described in Section 6 for generating power consumption fluctuation signals and performed exfiltration tests on multiple laptop brands.

**Algorithm 2** PIM signal demodulation algorithm

1. **Input**: PowerPulses, EncodeTable
2. **Output**: Transferred data
3. **While** (i < PowerPulses)
4.   Time1 = GetPulseTime(i)
5.   i = i + 1
6.   Time2 = GetPulseTime(i)
7.   Interval = Time2 – Time1
8.   BinCode = SearchEncodeTable(Interval)
9.   Output ← decodedData

---

Table 3. Transmission mode of PowerBridge.
7.1.1. Communication Rate

In the case of exfiltration, the primary factor that restricts the communication rate is the direct correlation between the response time $\Delta T$ from the CPU operation and fluctuations in power consumption. Experimental results obtained from a Gigabyte AERO 15 laptop demonstrated that the minimum $\Delta T$ duration was 33 milliseconds. This limitation may arise due to voltage stabilization effects caused by capacitors present within both the computer itself and the power adapter. It is well-known that power adapters employ high-capacity capacitors to filter out power fluctuations and jitter, thereby stabilizing voltage through peak shaving and valley filling functions [47]. However, since capacitors have the ability to filter out high-frequency waveforms, they impose restrictions on maximum communication frequency and directly impact communication speed. Consequently, it becomes evident that the shortest time interval for symbol signals to avoid overlap determines both the highest effective signal frequency $f = 1/\Delta T$ as well as the maximum communication rate for exfiltration.

The maximum bit rates of exfiltration in the experimental system of this article using two different modulations are recorded in Table 4. Signal transmission in the experimental system is conducted through five mainstream brands of laptops, while reception is achieved using our designed plug. The signal is modulated with 2ASK, and the measured exfiltration rate ranges from 27 to 33 bits/s. Compared to common wired or wireless networks, this communication rate is relatively low, indicating that this channel can only accommodate a limited amount of content. The current signal is received by the smart plug, as shown in Figure 25.

Table 4. Exfiltration rate test.

<table>
<thead>
<tr>
<th>Laptop Brands</th>
<th>Minimum Symbol Interval Time (ms)</th>
<th>Maximum Bit Rate (2ASK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIGABYTE AERO 15</td>
<td>33</td>
<td>33.30 bits/s</td>
</tr>
<tr>
<td>Lenovo X1</td>
<td>31</td>
<td>32.26 bits/s</td>
</tr>
<tr>
<td>Dell Precision5570</td>
<td>37</td>
<td>27.03 bits/s</td>
</tr>
<tr>
<td>HP Spectre X360</td>
<td>34</td>
<td>29.41 bits/s</td>
</tr>
<tr>
<td>Asus Pro Art Q17</td>
<td>31</td>
<td>32.26 bits/s</td>
</tr>
</tbody>
</table>

Figure 25. The current signal received by the smart plug.

7.1.2. Universality

As data modulation is achieved by manipulating CPU power consumption, variations in CPU power consumption levels are reflected in the recorded data obtained from the smart plug power measurement module. When malicious programs excessively occupy and heavily load the CPU or when users meticulously monitor CPU usage through the operating system task manager, abnormal patterns of CPU calls may be observed, potentially exposing air-gap channels. However, detecting such exfiltration of air-gap channels becomes challenging when computers are locked or remain idle for extended periods. Given
that background programs only intermittently utilize 100% of the CPU’s capacity, ordinary computer users face difficulties in detection, while experiencing significant potential harm.

7.2. Infiltration

7.2.1. Bit Rate

Our study utilizes the ACPI interface of a computer to receive switch signals from smart plugs. In laptop power adapters, larger capacitors are commonly employed to effectively filter voltage fluctuations and interference, ensuring the stable operation of the computer. Consequently, the response of a computer to power supply on/off events is influenced by the charging and discharging speed of these capacitors. To determine the minimum response time for laptops during power outages, experimental investigations are conducted. By obtaining this shortest response time, we can calculate the maximum achievable communication rate. The corresponding power outage response times and fastest communication speeds for laptops from different brands are presented in Table 5.

Table 5. Infiltration speed test.

<table>
<thead>
<tr>
<th>Laptop Brands</th>
<th>The Minimum Time Interval for the Power Adapter to Be Disconnected (s)</th>
<th>Maximum Bit Rate (2ASK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIGABYTE AERO 15</td>
<td>0.85 s</td>
<td>1.18 bits/s</td>
</tr>
<tr>
<td>Lenovo X1</td>
<td>0.96 s</td>
<td>1.04 bits/s</td>
</tr>
<tr>
<td>Dell Precision5570</td>
<td>0.81 s</td>
<td>1.24 bits/s</td>
</tr>
<tr>
<td>HP Spectre X360</td>
<td>0.78 s</td>
<td>1.29 bits/s</td>
</tr>
<tr>
<td>Asus Pro Art Q17</td>
<td>0.86 s</td>
<td>1.16 bits/s</td>
</tr>
</tbody>
</table>

7.2.2. Universality

Our study employs the activation and deactivation of intelligent plugs to facilitate signal transmission, resulting in the continuous power cycling of laptop adapters. This research investigates the utilization of pulse interval modulation (PIM) as a modulation technique for communication, wherein data are represented by defining varying time durations between pulse falling edges. Table 6 shows a comparison of the communication capabilities of the disclosed air-gapped covert channels. By employing longer modulation intervals that are challenging for users to detect, this covert infiltration method exhibits significant concealment capabilities and potential risks.

Table 6. Comparison of communication capabilities of air gap channels.

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Air-Gap Channels</th>
<th>Communication Capability</th>
<th>Infiltration Bit Rate (bit/s)</th>
<th>Exfiltration Bit Rate (bit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitJabber</td>
<td>Electromagnetic</td>
<td>Unidirectional</td>
<td>\</td>
<td>300,000</td>
</tr>
<tr>
<td>AIR-FI</td>
<td>\</td>
<td>\</td>
<td>16</td>
<td>\</td>
</tr>
<tr>
<td>RAMBO</td>
<td>\</td>
<td>\</td>
<td>1000</td>
<td>\</td>
</tr>
<tr>
<td>AiR-ViBeR</td>
<td>Sound</td>
<td>Unidirectional</td>
<td>\</td>
<td>0.5</td>
</tr>
<tr>
<td>InkFiltration</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>100</td>
</tr>
<tr>
<td>ETHERLED</td>
<td>Optical</td>
<td>Unidirectional</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td>CD-Blink</td>
<td>\</td>
<td>\</td>
<td>1</td>
<td>\</td>
</tr>
<tr>
<td>HVACKer</td>
<td>Thermal</td>
<td>Bidirectional</td>
<td>0.011</td>
<td>0.011</td>
</tr>
<tr>
<td>PowerHammer</td>
<td>Power sector</td>
<td>Unidirectional</td>
<td>\</td>
<td>1000</td>
</tr>
<tr>
<td>POWER-SUPPLaY</td>
<td>\</td>
<td>\</td>
<td>\</td>
<td>60</td>
</tr>
<tr>
<td>PowerBridge</td>
<td>Bidirectional</td>
<td></td>
<td>1.2</td>
<td>33</td>
</tr>
</tbody>
</table>
8. Discussion

In this section, we will discuss two practical strategies for PowerBridge. Countermeasures can be roughly divided into procedural and technical measures.

8.1. Exfiltration Countermeasures

8.1.1. Detection

Security software is utilized to identify running programs on a computer and uncover the process of establishing air-gap channels. In this approach, security solutions such as antivirus (AV) strive to detect malicious operations by recording and monitoring the logs of software processes interacting with the CPU. When it comes to power line air-gap channels, abnormal workloads are imposed on the CPU, enabling security software to detect them through this characteristic. However, due to the diverse types of software employed in computers and considerable variability in user behavior, both software and user actions can result in unexpected fluctuations in the CPU workload. Consequently, this detection method may yield a significant number of false positives or false negatives.

8.1.2. Signal Interference

The method of signal interference is commonly employed to obstruct electromagnetic [48] and acoustic [49] air-gap channels. Deceptive interference is conducted on the receiving end by generating random signals resembling those of the air-gap channel, or high-power signals are generated to suppress interference on the air-gap channel signal. Conventional WiFi jammers can impede WiFi signal communication within distances of several hundred meters [50]. For the power line air-gap channel discussed in this article, workloads with comparable power to that of the air-gapped computer can be connected in parallel to the power line to generate false or strong signals for interference purposes. However, workloads are energy-intensive devices, which may result in increased user power consumption and could prove impractical in certain environments where the power system cannot support them.

8.2. Infiltration Countermeasures

The most straightforward approach for infiltration scenarios involves prohibiting the connection of computers with air-gap isolation to devices equipped with IoT capabilities, such as smart plugs. This method yields favorable outcomes for computer equipment situated in fixed locations. However, there exists a potential risk of infiltration when critical equipment needs to be introduced into unfamiliar environments (e.g., attending meetings at unfamiliar venues or staying in hotels) and necessitates power supply connectivity. Consequently, users can employ security software to conduct audits on their laptops’ power logs, generate alerts upon detecting suspected attacks, and promptly disconnect hazardous power sources.

9. Conclusions

Electricity serves as an indispensable energy source for contemporary electronic devices, necessitating the utilization of power lines for both power supply and charging purposes, thereby establishing air-gap channels. In this article, we demonstrate the exploitation of plugs to facilitate data exfiltration and infiltration through power lines.

In the exfiltration scenario, the software regulates the CPU workload, constructs the power consumption fluctuations of the device, and generates current fluctuation signals across the air gap. Sensitive data on a computer with air-gap isolation are modulated, encoded, and transmitted using binary-encoded signals from power lines. The remote controller of the plug can receive current signals and decode binary information. The signal in the infiltration scenario is generated by the remote controller through the switch (off or on) of the smart plug.

The ACPI interface on the computer receives the switch signal from the plug, and by continuously polling the power API interface of the operating system, software on the
computer can perceive it in real time. We have described a threat model and analyzed a usage scenario for this power line air-gap channel and have designed a modulation protocol suitable for communication in this scenario.

We conducted a software prototype simulation and performed evaluations on various laptop models, while also discussing countermeasures for the prevention and defense of PowerBridge air-gap channels.

Our evaluation demonstrates that smart plugs enable unidirectional simplex communication with air-gapped computers. Our findings indicate that data can be exfiltrated from the network through power lines at an approximate bit rate of 30 bits per second and transmitted to laptops isolated by air gaps at an approximate bit rate of 1 bit per second.

Author Contributions: Formal analysis, H.S. and L.Q.; Writing—review and editing, Y.L. and Z.L.; Supervision, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References


43. Modulation Mode. 2022. Available online: https://baike.baidu.com/item/%E8%B0%83%E5%88%B6%E6%96%B9%E5%BC%8F/8971326 (accessed on 3 May 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.