Journal of Systems Architecture 161 (2025) 103380

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Detection and defence against thermal and timing covert channel attacks in multicore systems

Parisa Rahimi [a,c,*], Amit Kumar Singh [a], Xiaohang Wang [b], Seyedali Pourmoafi [c]

[a] School of Computer Science and Electronic Engineering, University of Essex, United Kingdom
[b] School of Cyber Science and Technology, Zhejiang University, China
[c] School of Physics, Engineering and Computer Science, University of Hertfordshire, United Kingdom

ARTICLE INFO

Keywords:
Multi-covert channel
Thermal covert channel
Timing covert channel
Multicore systems
Countermeasure

ABSTRACT

As interest in multicore systems grows, so does the potential for information leakage through covert channel communication. Covert channel attacks pose severe risks because they can expose confidential information and data. Countering these attacks requires a deep understanding of various covert channel attack types and their characteristics. Thermal covert channel and covert timing channel attacks, which use temperature and timing, respectively to transfer information, are two dominant examples that can compromise sensitive data. In this paper, we propose a methodology for jointly detecting and mitigating these types of attacks, which has been lacking in the literature. Our experiments have demonstrated that the proposed countermeasures can increase the bit error rate (BER) for mitigation while maintaining comparable power consumption to that of the state-of-the-art.

1. Introduction

Sensitive information leakage is increasingly concerning in today's modern computing infrastructures. As technology advances, the risk of side or covert channels providing a secret communication medium between processors has escalated, presenting a significant security risk. It is crucial to differentiate between covert and side channels. Covert channels involve intentionally modifying system behaviour to transmit information in a manner that bypasses normal security mechanisms [1]. This transmission is typically hidden within legitimate system operations (Fig. 1). In contrast, side channels exploit unintentional information leakage through inherent system behaviours, such as timing variations or electromagnetic emissions, without altering the system's normal operation [2]. While both pose significant security risks, our work focuses on covert channels, where the sender actively seeks to hide communication within normal system operations. Among all types of covert channels, thermal and timing channels are notable for their ability to bypass standard security mechanisms and transmit information in hidden ways. These attacks are highly effective in multicore systems, where shared resources are exploited to facilitate covert communication [3].

Thermal Covert Channels: A thermal covert channel is typically established by running a program on the sender core that fluctuates temperature based on the bits to be transmitted. For instance, increasing the temperature could represent a bit "1", while maintaining normal operation could indicate a bit "0". Thermal covert channels exploit the heat generated by a core during processing. For instance, the sender core executes specific instructions to generate heat such as a password, causing a detectable temperature rise and creating a temperature signal, which can be used to transmit data from core A to core B, as Fig. 1 indicates [4].

Timing Covert Channels: Timing channels leverage the execution time of processes to encode information. In a timing covert channel, the sender generates information by manipulating the timing of its execution processes. This is typically done by introducing intentional delays or modifying the timing of specific operations. For instance, the sender may deliberately slow down or speed up certain instructions, causing a detectable variation in execution time. The receiver, monitoring the same resource (such as shared memory or a network channel), measures these timing differences to decode the transmitted information. For example, a longer execution time might represent a bit "1", while a shorter time might represent a bit "0". This subtle manipulation of execution timing allows for covert communication between the sender and receiver without relying on standard communication channels [5].

Thermal and timing covert channel attacks exploit temperature fluctuations and timing discrepancies to stealthily transmit sensitive

* Corresponding author at: School of Physics, Engineering and Computer Science, University of Hertfordshire, United Kingdom.
E-mail addresses: p.rahimi@herts.ac.uk (P. Rahimi), a.k.singh@essex.ac.uk (A.K. Singh), xiaohangwang@zju.edu.cn (X. Wang), s.pourmoafi@herts.ac.uk (S. Pourmoafi).

https://doi.org/10.1016/j.sysarc.2025.103380
Received 24 June 2024; Received in revised form 17 February 2025; Accepted 18 February 2025
Available online 27 February 2025

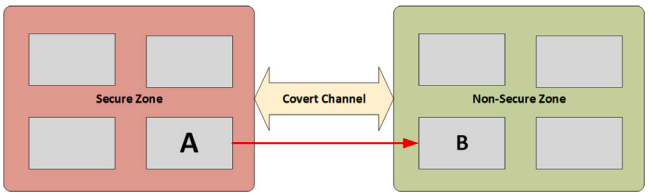

Fig. 1. Covert Channel communication in an eight multicore system. The arrow from core A to core B indicates that the data/information flows from core A to core B.

information. These attacks are particularly effective in embedded systems, where they can be executed without specialized equipment, increasing their accessibility to attackers. Recently, attackers have grown more sophisticated, combining thermal and timing covert channels to create more potent and complex attacks. By manipulating both timing and thermal patterns, they can enhance protection and prevent detection, making these vulnerabilities even more challenging to counter in modern embedded system design.

As these multi-covert channel (thermal and timing) attacks continue to evolve, they pose numerous challenges for detection and prevention [6]. For example, a thermal covert channel presents a unique challenge as it does not rely on shared sources like cache or memory, making it harder to detect. The amalgamation of timing and thermal covert channels leads to an expanded capacity for covert communication, facilitating the transmission of a larger volume of data surreptitiously. Consequently, there is a need to develop new strategies to cope with this heightened security risk [7,8]. Jankowski et al. [9] introduced Inter-Protocol Steganography and presented a new steganographic system called PadSteg, which demonstrates the concept of inter-protocol steganography. Unlike traditional methods that use a single network protocol, PadSteg uses the relationship between two or more protocols to hide information. Specifically, it leverages the Address Resolution Protocol (ARP) and Transmission Control Protocol (TCP) along with the Ether leak vulnerability in Ethernet frame padding to facilitate secret communication within Local Area Networks (LANs).

However, to the best of our knowledge, this is the first time that a multi-covert channel attack has been considered. Like any other communication system, a multi-covert channel includes a pair of transmitters and receivers. In essence, the transmitter creates signals from sensitive data or information, such as a password, and on the receiver side, the receiver reads the information from its sensor. Since a multi-covert channel is committed to transmitting signals over a chip, sensitive information is prone to leak, being disrupted, or being degraded by attackers. In response to this pressing issue, this paper addresses the critical gaps in previous research and makes the following novel contributions:

- A detection strategy for thermal and timing covert channel attacks simultaneously presents a significant challenge in multicore architecture. Taking advantage of signal analysis techniques based on the usage of the power spectrum, our strategy is adapted to the complex dynamics of multi-core system designs.
- A defence strategy for defending against multi-covert channel attacks. Regarding the identified threats, we have developed different defence strategies designed to effectively disrupt multi-covert communications, while maintaining system performance and without increasing power consumption.
- Extensive evaluation and comparison against state-of-the-art, demonstrating the advantages of our strategies that can protect the integrity and confidentiality of our data as covert channel attacks become more pervasive and insidious.

The complexity of modern multicore systems and the evolving nature of attacks have increased. Attackers can use thermal and timing covert channels to transmit sensitive data, combining them to carry out sophisticated attacks that are harder to detect. Furthermore, the current or existing state-of-the-art focuses on individual types of covert channels but lacks comprehensive detection and mitigation strategies for multi-covert channels that continuously leverage thermal and timing channels. Existing detection methods are either too specific or have significant overhead. By understanding the challenges and exploiting both covert channels, attackers can create high-capacity, hard-to-detect communication channels. This paper aims to provide a comprehensive methodology to detect and mitigate these combined attacks effectively.

2. Related work

Timing covert channels transmit data and information based on packet arrival times rather than the actual packet content [10]. Timing covert channel attacks, introduced at CRYPTO'96 in 1996 [11], are fundamental side-channel attack methods. This technique allows attackers to extract secrets, such as cryptographic keys, by analysing variations in computation time. Although primarily used in cryptography, timing attacks can target various systems [12]. Cryptographic algorithms often exhibit non-constant computation times due to performance optimizations, revealing sensitive information through timing variations [13]. Timing attacks exploit the dependency of execution time on encryption keys. By measuring and statistically analysing these times, attackers can infer the secret key [14]. This method relies on the correlation between input data, internal system states, and execution times. Researchers have also explored hardware-based timing channels, such as cache and remote timing attacks [15,16].

Thermal covert channels have gained attention, with Masti et al. [17] improving their design for better noise immunity and higher throughput. Bartolini et al. [18] achieved impressive results with a thermal covert channel, emphasizing the need for countermeasures to protect sensitive information in modern multicore systems.

Detection: Detecting timing covert channels is challenging due to their small time intervals. Statistical analysis and software tools can help identify patterns and anomalies that may indicate covert channel usage [19]. Cabuk et al. [20] developed a methodology for detecting IP covert channels using an epsilon similarity compressibility score. While primarily focused on network-based covert channels, their detection method, which analyses the timing patterns of IP packets is conceptually relevant to our work. It demonstrates how timing analysis can be applied outside traditional network steganography to identify covert channels, providing a foundational technique that we adapt for detecting timing covert channels in multicore systems.

Furthermore, detection methodologies for timing attacks can include change point detection algorithms and frequency scanning techniques [21]. For thermal covert channels, techniques like incorporating multiple tags, analysing clock skews, and employing signal frequency scanning can identify affected cores [22,23]. An alternative method involves utilizing software tools to observe the actions of shared resources like on the CPU or hard drive, identifying irregularities that could suggest the existence of a hidden communication channel. For instance, heightened CPU usage or increased hard drive access by an application might indicate the presence of a covert channel [24]. [25] introduced the Shared Resource Matrix Methodology, a systematic approach for identifying storage and timing channels throughout the software lifecycle. Using a matrix to track shared resources, this method helps determine whether system operations reference or modify these resources, thereby detecting potential covert channels, such as unusual CPU usage or increased hard drive access. [26] extended this concept by developing the Physical Environment Matrix, enhancing Kemmerer's approach to detecting covert channels that exploit physical properties of the environment, such as thermal or acoustic signals. Detection is a crucial step in countermeasure strategies [15].

Defence: Thermal and timing covert channels exhibit behaviours similar to other covert channels. They do not require specific equipment and can bypass system defences by not needing access to shared

resources like cache or memory [27]. For instance, thermal covert channels utilize simple on-off keying techniques to encode bits “0” and “1” [18], while timing covert channels rely on packet arrival times [15]. [16] suggested countering such attacks by randomizing processes, providing good levels of obfuscation. [28] introduced a method that applies the relationship between timing information in encryption processes and unique cache accesses to uncover secret keys.

Timing attacks are not limited to cryptography and can apply to any system exhibiting execution time variations based on input data or internal state. Protecting against timing attacks requires designing algorithms and systems that execute in constant time, independent of input data or internal state. Timing attacks are particularly effective against cryptographic algorithms using conditional statements or loops dependent on the secret key. These can cause execution time variations, which attackers can analyse to infer secret key values. For example, if a cryptographic algorithm uses a conditional statement checking if a certain bit of the secret key is set, the execution time will vary depending on whether the bit is set or not. Attackers measure execution times for different input values and infer the secret key bit values by analysing the statistical variations [29]. Preventing timing attacks involves designing cryptographic algorithms and systems to execute in constant time. This can be achieved using bitwise operations and arithmetic operations independent of the secret key, or by using randomization techniques like blinding or masking [16]. Access control restricts unauthorized access to critical resources, while encryption safeguards sensitive data from covert transfers [23]. Implementing security measures such as isolation and access control can effectively counter thermal covert channels. By limiting access to shared resources and monitoring usage, unauthorized individuals are prevented from using thermal covert channels for information transfer. Additionally, introducing random timing variations through techniques like random delays or jitter complicates the establishment of covert channels [19]. Intrusion detection systems (IDS) and traffic shaping provide further protection by detecting and blocking covert communications over timing channels [30].

Jiang et al. [28] suggested a countermeasure strategy based on rapid frequency scanning to detect possible attacks. When a thermal covert channel is detected and its transmission frequency identified, strong noise is applied to the transmission frequency band for targeted jamming. Dynamic Voltage Frequency Scaling (DVFS) is another mitigation strategy aimed at combating thermal covert channel attacks [4]. Recognized for its role in power consumption, DVFS dynamically adjusts the voltage and frequency of a CPU [21]. While it offers protection against thermal covert channels, it is not effective against timing covert channels. This limitation arises because timing attacks exploit precise variations in task execution times, and DVFS does not randomize execution times sufficiently to disrupt these timing patterns. In some cases, DVFS adjustments may inadvertently preserve the timing structure attackers rely on. Experiments conducted using the Sniper-v7.2 simulator confirmed that while DVFS reduced the thermal signal, it failed to eliminate the risk posed by newer attack methods [22].

Noise-based countermeasures, including selective noise control (SNC), provide an additional line of defence against covert channels. SNC disrupts covert communications by injecting controlled noise into the system, typically in the form of random delays or temperature fluctuations. These disruptions make it more difficult for attackers to decode covert signals [22]. While this method is effective against thermal covert channels, its impact on timing covert channels is limited. Timing-based attacks rely on precisely controlled execution-time variations, and the random noise introduced by SNC often results in only minor timing delays. These small disruptions are insufficient to completely break the timing patterns exploited by attackers. Moreover, while SNC effectively disrupts covert communication, it can introduce performance overhead and increase power consumption, particularly if applied randomly.

To address the limitation of SNC, fan speed control (FSC) has been introduced [31]. FSC is a technique used to mitigate thermal covert channels. By adjusting the system’s fan speed, the cooling process becomes more variable, making it harder for thermal covert channels to rely on consistent temperature changes for data transmission [31]. However, while FSC can disrupt thermal covert channels, it is ineffective against timing covert channels.

Timing-based attacks leverage slight variations in task execution times, which are not influenced by changes in system temperature. Thus, while these countermeasures reduce the risk of thermal covert channels, they do not address the core vulnerabilities that timing-based attacks exploit. To improve its effectiveness against timing and thermal covert channels, targeted modifications are required. Our observations indicate that there is a lack of research considering multiple covert channel attacks jointly and handling them efficiently, which is the focus of this paper.

3. Methodology

3.1. System and threat mode

In our model, the sender and receiver are located in different security zones within the multicore system. Due to strict security policies and isolation, they are unable to communicate through standard communication channels. This restriction motivates the creation of covert channels, where the sender and receiver exploit shared resources like thermal and timing variations to secretly transmit information [32].

Therefore, attackers can exploit software interfaces, such as Intel’s processor Model Specific Registers (MSR), to read local sensor data. This unauthorized access allows them to degrade the integrity of the data, posing a significant risk to the system’s reliability and security [33]. For instance, in a multicore system where each core can perform separate tasks or work collaboratively on shared tasks, security is a serious concern, especially in cases involving sensitive data processing or critical system operations [34]. The system’s architecture should be designed in a way that defends against attackers who aim to exploit software interfaces to access or manipulate data [35]. To protect against such vulnerabilities, our system incorporates a protective mechanism known as the global manager core. This entity runs the operations across all cores, enabling it to monitor the system’s fundamental quantities. It utilizes detection and protection programs, applying them to each core to prevent unauthorized access and data tampering. This proactive approach ensures the integrity and security of the multicore system, safeguarding it from potential risks [24].

In our model, assume that there is a covert channel operating between two different cores (cores A and B). Core A sits in a secured zone in which sensitive data or information is not allowed to be shared with the cores outside this zone, and Core B is in a non-secured zone [4]. The transmitter dispatches sensitive information through a covert channel. Meanwhile, the receiver monitors the temperature fluctuations or delay of the signal by accessing its sensor and then interprets these signals to accurately reconstruct the transmitted data.

3.2. Communication process

To initiate a transmission session, the transmitter and the receiver must first agree on a transmission frequency for the multi-covert channel. In this covert communication setup, the sender begins by modulating its behaviour to generate a detectable pattern in a shared resource, such as creating specific temperature fluctuations or introducing timing delays. The receiver, which has access to the same resource or can monitor its state, detects these patterns and reconstructs the transmitted information. This technique enables the sender and receiver to establish a covert channel without direct communication. The process starts with the sender transmitting a request packet through these modulated signals. Upon detecting and interpreting this covert request, the receiver

sends an acknowledgement back using a similar modulation method, confirming the establishment of the covert channel. This indirect communication method ensures that the covert channel remains concealed from standard monitoring mechanisms [33]. To complete the communication process, the sender encodes the sensitive information into packets, including a special preamble field (e.g., 010001) to indicate the start of a packet and an Error Correcting Code (ECC) field for data integrity. On the receiver's end, the receiver captures the signal using its local sensor and converts the signal into binary bit streams [36].

3.3. Finding a suitable band for covert channel

This study considers the power spectrum of each core in the multicore system when a typical application like Black–Scholes from PARSEC is running. The power spectrum is then divided into three different frequency bands [32]. The first band, Band A, covers frequencies from DC to 50 Hz (highly likely to be disrupted by the signal generated by a typical application), while Band B spans from 50 to 400 Hz, and frequencies above 400 Hz are considered the cut-off frequency falling into Band C, which is not ideal for data transmission. The timing and thermal signals can easily be affected by other signals generated by normal applications [37].

3.4. Proposed detection and defence strategy for separate covert channels (Double covert channel)

This section focuses on strategies for detecting and defending against multi-covert channels, that leverage timing and thermal patterns across multicore systems.

Detection: The core of our detection strategy involves the continuous monitoring and analysis of CPU workload traces from each core, focusing on their frequency domain characteristics. By applying a Fast Fourier Transform (FFT) to these traces [38], we break down the signals into their constituent frequencies, thereby creating a power spectrum that reveals the intensity of each frequency component. This spectrum provides a comprehensive view of the system's operational profile, allowing us to detect patterns that are the result of covert channel activity. Notably, the detection module was implemented as a software module that resides in the main memory. While Wang et al. [32] introduced the concept of using power spectrum analysis to detect thermal covert channels and Rahimi et al. [22] expanded this to include selective noise pattern detection, our approach extends these methodologies significantly. Unlike previous work that focused on single-channel detection, our work's unique contribution lies in the simultaneous detection of multi-covert channels by concurrently analysing both thermal and timing variations. This simultaneous detection capability is crucial for identifying complex attack vectors that leverage multiple covert channels.

We have defined distinct frequency bands for the detection of thermal and timing covert channels. Thermal covert channels typically take place in the 100 Hz to 300 Hz range, where temperature fluctuations induced by data transmission can be discerned. Timing covert channels, on the other hand, are often detectable within the 200 Hz to 400 Hz range, exploiting the subtle timing variations in processor operations. By setting precise bandpass filters for these ranges, we can isolate and analyse the signal components unique to each channel type.

To improve detection accuracy, an adaptive threshold mechanism dynamically adjusts according to the system's baseline power spectrum. This allows for the dynamic identification of covert channels even under varying workload conditions. By continuously adjusting the threshold, our system can effectively differentiate between normal operational fluctuations and covert channel signals. Beyond individual frequency analysis, our method incorporates cross-correlation analysis between thermal and timing signals. This enables the detection of coordinated covert channel activities, where the attacker may attempt to synchronize timing and thermal variations to prevent detection. The cross-correlation process allows us to identify patterns of joint activity, further enhancing the robustness of our detection framework.

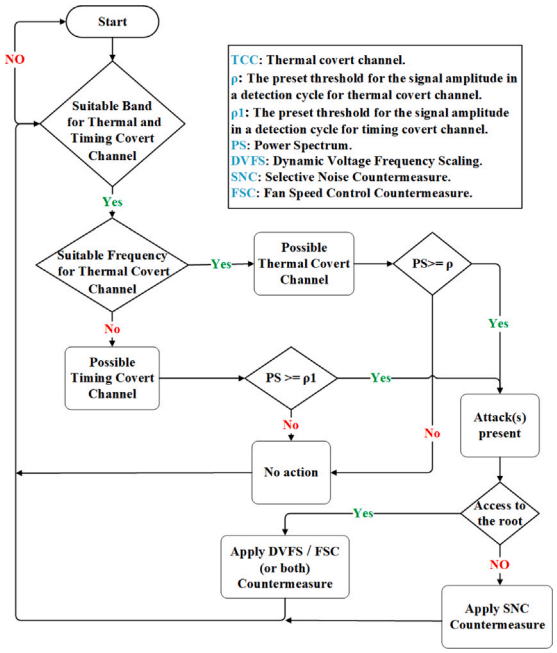

Fig. 2. Diagram of process of detection and defence of Thermal and Timing Covert Channel Attacks in double covert channels (process1).

The methodology for detecting and defending the double covert channels is illustrated in Fig. 2. During the first step, a detection scheme is applied to a multicore system to check for any multi-covert channel attacks. In the second step, the threads involved in the thermal and timing covert channels are determined. In the third step, for all the found thermal and timing covert channel threads identified as engaging in this channel from the previous step, a proper countermeasure is applied to the cores where those threads run to shut down any possible signal transmission at a minimum cost to the system performance. This detection method to mitigate multi-covert channel attacks demonstrates a significant extension of our previous work, which addresses both types of attacks (thermal and timing covert channel) [31]. A summary is as follows.

Through experimental methods, we established the threshold parameters for our detection strategies. For the first technique, which addressed a dual covert channel scenario, a timing covert channel attack is identified if the threshold ρ is in the interval of $20\,\text{dB} < \rho < 40\,\text{dB}$. In the case of a thermal covert channel attack, it is detected when the distinct threshold ρ_1 satisfies $40\,\text{dB} < \rho_1 < 50\,\text{dB}$. To implement the suggested detection and countermeasure approach for thresholds that range from the lower ρ_l to the high ρ_h, to measure the accuracy of detection and determine the high BER.

Countermeasure: If the system realizes that an attack(s) is present, then based on our access to the system root, an appropriate countermeasure will be applied, (as shown in Fig. 2).

3.4.1. Proposed countermeasures

If we have root access, we can apply a DVFS-based or fan speed control-based countermeasure.

Enhanced DVFS for Thermal and Timing Covert Channels: The advent of modern computing technologies has seen widespread adoption of Dynamic Voltage and Frequency Scaling (DVFS) across central processing units (CPUs). DVFS endows systems with the flexibility to dynamically adjust processor voltage and frequency in response to varying workloads and operational demands, a capability thoroughly

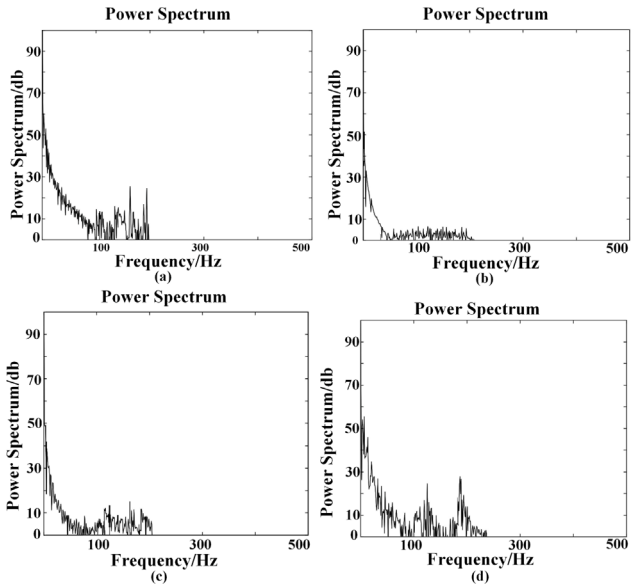

Fig. 3. System behaviour with double covert channels, (a) system without countermeasure, (b) system with DVFS countermeasure, (c) system with Selective Noise countermeasure and (d) system with Fan Speed Control countermeasure.

documented in the study by [30]. We extend the conventional application of DVFS to simultaneously mitigate both thermal and timing covert channels. In our approach, DVFS is leveraged not only to modulate temperature but also to introduce controlled variability in execution times, thereby obstructing timing-based communication. When covert activity is detected, our system dynamically adjusts the frequency and voltage of the implicated core, introducing unpredictable timing delays and temperature fluctuations to disrupt both covert communication channels.

Our proposed DVFS-based countermeasure regulates each core effectively by managing its voltage and frequency settings. This precise adjustment allows us to modulate processor signal frequencies, thereby controlling the transmission of signals through multi-covert channels. The countermeasure can slow down, completely block, or prevent signal transmission, disrupting covert communication as necessary. By adjusting core operations and eliminating unnecessary signal transmissions, our approach mitigates both timing and thermal covert channel attacks, providing a secure and efficient system environment.

Once a core is detected as participating in potentially malicious activity, immediate countermeasures are enacted to halt any further transmission. Our strategy assumes that the timing and thermal manipulation threads remain on the same logical core for the duration of the communication session, or potentially longer. By strategically lowering the frequency of the targeted core using DVFS, we induce significant changes in the transmission characteristics, which substantially increases the bit error rate (BER) experienced by the receiver. As a result, the integrity of any covert communication is severely compromised. As shown in Fig. 3(b), this approach disrupts unauthorized use of the system's communication channels without impacting the overall system performance.

As shown in Table 4, a significant increase in the bit error rate (BER) renders the transmitted signal unintelligible to the receiver. This effectively thwarts covert communication attempts.

Enhanced Fan Speed Control for Thermal Covert Channels: By dynamically increasing fan speed, the system can cool processor cores more rapidly, thereby disrupting the thermal signals used to transmit covert data. The rapid temperature changes hinder the sender's ability to modulate temperature predictably, forcing the covert channel to become unstable and increasing the likelihood of transmission errors.

Table 1
Different statues of fan speed.

Range of temperature (°C)	Fan's speed (RPM)
Tc < 65	1000
65 < Tc < 68	1500
68 < Tc < 71	1700
71 < Tc < 74	1900
74 < Tc < 77	2100
77 < Tc < 80	2300
80 < Tc < 83	2600
Tc ≥ 83	Max

Although fan speed control is primarily designed for temperature management, it also indirectly impacts timing covert channels. Dynamic adjustments to fan speed create cooling and heating cycles, which introduce slight variations in system performance. These thermal shifts can alter execution times, particularly in systems where performance is sensitive to thermal thresholds. By preventing the system from reaching thermal equilibrium, fan speed control introduces sufficient variability in execution times to disrupt the timing patterns necessary for covert channel communication to function reliably.

As part of our multi-covert channel defence methodology, we leverage fan speed adjustments to counteract potential attacks, utilizing strategic thermal management to safeguard multicore systems. As detailed in Table 1, this method modulates fan speeds to manage thermal effects across processor cores by considering the intricate relationship between heat generation, execution time, frequency, and voltage [18]. The dual objectives of this strategy are to obstruct covert communication channels while maintaining system performance.

Sensitive data transactions, especially those vulnerable to timing and thermal covert channels, may be exploited by attackers who can deduce information from thermal and timing variations. However, by actively managing fan speeds, we introduce a layer of unpredictability into the system, complicating attackers' attempts to decode transmitted signals and effectively masking the transmission of sensitive data (Fig. 3(d)). This fan speed control-based countermeasure represents a proactive and nuanced approach to mitigating multi-covert channel attacks. It builds on the system's existing thermal management capabilities, creating an environment where thermal and timing patterns are continuously adjusted, significantly reducing the likelihood of successful data interception.

Through the strategic disruption of signals and obfuscation of sensitive information, this countermeasure strengthens the security of multicore systems against sophisticated multi-covert channel exploitation. Despite the effectiveness of fan speed control countermeasures, it is essential to acknowledge the potential challenges and explore further opportunities for enhancing this approach.

Thermal Management Complexity: Adjusting fan speeds to manage thermal effects is a complex process that requires precise control and monitoring. The relationship between heat generation, execution time, frequency, and voltage must be carefully monitored by reading the local thermal sensor and continuously tracking the temperature of each processor core and other critical components to ensure that the thermal adjustments effectively disrupt covert channels without adversely affecting the system's performance. **System Performance Impact:** While the primary goal is to secure the system from covert channel attacks, altering fan speeds can also impact the overall system performance. Increased fan speeds may cool cores more rapidly but could lead to unnecessary power consumption. By setting the fan speed based on the temperature effectively, we can avoid this type of issue. **Monitoring and Response:** Effective implementation of this countermeasure requires continuous monitoring of system temperatures and performance metrics to ensure timely and appropriate adjustments in fan speed.

If we do not have access to the root, we can add a selective noise-based countermeasure.

Adaptive Selective Noise Injection: Our enhanced approach to selective noise injection dynamically adjusts the level of noise introduced based on monitoring of both thermal and timing covert channels. By targeting specific intervals where covert activity is detected, the system minimizes the performance impact while maximizing the disruption of covert signals. When a timing covert channel is detected, random delays are introduced during memory accesses, while additional heat generation is induced to mask thermal covert signals. Our enhanced approach to selective noise injection dynamically adjusts the level of noise introduced based on real-time monitoring of both thermal and timing covert channels. By targeting specific intervals where covert activity is detected, the system minimizes the performance impact while maximizing the disruption of covert signals. When a timing covert channel is detected, random delays are introduced during memory accesses, while additional heat generation is induced to mask thermal covert signals.

However, the primary challenge of this approach lies in how many threads can be efficiently added to enhance the effectiveness of defence against multi-covert channel attacks without increasing power consumption. While it may effectively disrupt certain channels, such as thermal covert channels, its applicability and ability to act against timing covert channels require further investigation and further development. In the next stages of our defensive strategy against multi-covert channel attacks, we employ another technique known as a selective noise-based countermeasure, complemented by an iterative cycle process, to ensure comprehensive protection. As each bit is prepared for transmission, the system evaluates the current frequency or temperature against a previously established baseline. A drop below this baseline, indicative of a potential “0” bit transmission, triggers the injection of additional noise. This strategic addition is designed to preserve a state suggestive of a “1” bit by maintaining elevated temperature or frequency levels, thereby confusing any attempt to utilize covert channels for unauthorized data transmission. This step not only secures the data but also introduces deliberate timing delays, disrupting the finely tuned timing protocols that covert communications rely on. This methodology includes the following steps:

Step 1-Bit Monitoring: The system continuously monitors the transmitted signal and records the value of each bit.

Step 2-Selective Noise Injection at Bit “1”: Upon recognizing a “1” bit, the system immediately injects a calculated dose of random noise, intentionally perturbing the signal’s established pattern.

Step 3-Temperature or Frequency Tracking: Following selective noise injection, the system records the current frequency or temperature of the signal. This serves as a baseline for comparison in the next step.

Step 4-Adaptive Noise Modulation: During the next bit transmission, the system compares the current frequency or temperature with the previously recorded value. If the current value is lower than the baseline (indicating a potential transition to bit “0”), additional noise is added to maintain the high temperature or frequency state, further disorienting any covert channel attempts. This introduces intentional delays that disrupt the timing requirements of covert communication protocols.

Step 5-Iterative Cycle: The monitoring, selective noise injection, and comparison steps continue for each subsequent bit until the entire transmission is processed.

This multi-faceted approach offers several advantages over the previous single-channel method: **Targeted Defence**: By focusing on bit “1” instead of just temperature, the countermeasure specifically disrupts channels relying on high-frequency or high-temperature bits. **Adaptive Noise Modulation**: The selective noise injection based on signal analysis effectively counters covert transmissions, even when attackers attempt to adapt their encoding pattern. **Delay Introduction**: The intentional delays caused by selective noise injection further disrupt the timing-sensitive nature of covert channels, making successful information extraction challenging.

Our proposed selective noise-based countermeasure, shown in Fig. 3(c), presents a robust and adaptable solution for mitigating multi-channel thermal covert channel attacks, offering enhanced protection for sensitive systems and data. This expanded version provides a more comprehensive explanation of the process, highlights the key benefits of the improved approach, and emphasizes its effectiveness against multi-channel covert channels.

The findings presented in Table 4 demonstrate, that by using two different covert channels for transmitting data, the system will face overheating as well as high power consumption. This is mainly because running two covert channels simultaneously requires additional processing resources, including CPU and memory usage. This increased load can cause the system’s components to work hard and generate more heat and power consumption to cool the system down.

3.5. Proposed strategy for a single covert channel for transmitting both timing and thermal data

Addressing the dual concerns of overheating and excessive power usage, we introduce an alternative strategy (single covert channel). As Fig. 4 illustrates, this strategy utilizes a singular covert channel to transmit both timing and thermal data simultaneously, offering an efficient and robust solution to the challenges at hand. Rather than operating a double channel for each type of data, our method combines timing and thermal information, allowing for concurrent encoding and transmission. Combining thermal and timing covert channels presents unique challenges due to their different operating principles. A thermal covert channel uses temperature variations, while a timing covert channel leverages timing differences. Therefore, we have used packet transmission rates, to detect any possible attacks. To implement this covert channel effectively, both the sender and receiver must initially agree on specific channel parameters, such as frequency and the schedule for sending packets at predetermined intervals. A critical aspect of this technique is the ability to measure and analyse the packet transmission rate, which serves as a potential indicator of covert channel usage or an ongoing attack. This method relies on the establishment of predefined thresholds for packet transmission rates, monitoring for any activity that deviates significantly from these norms, which could signal the presence of unauthorized data transmission or malicious activity. Moreover, in the development of this approach, we considered a signal-to-noise ratio $\rho > 50\,\text{dB}$, ensuring that covert communication remains detectable under strict monitoring criteria. This approach not only aids in the effective transmission of data via a unified channel but also plays a crucial role in the early detection and prevention of covert channel exploitation.

Fig. 5 compares the effects of employing a single covert channel and different defence strategies for timing and thermal data transmissions instead of using separate channels for each data type. As mentioned above, this approach for transmitting both timing and thermal data offers several benefits. This strategy not only optimizes operational performance but also significantly enhances the resilience and reliability of the communication system.

3.6. Combination methodology with DVFS and fan speed control-based countermeasure

In this strategy, we combined DVFS and fan speed control-based countermeasures. Fig. 6 indicates the different steps of this methodology. It is important to note that incorporating selective noise as a countermeasure involves manipulating the system’s operational characteristics to obscure or alter signal patterns that could be exploited by attackers. While selective noise can be an effective strategy in certain contexts, the focus here on hardware optimization (i.e., DVFS and fan speed control) is aimed at achieving a high level of security without compromising on system performance or power efficiency. This

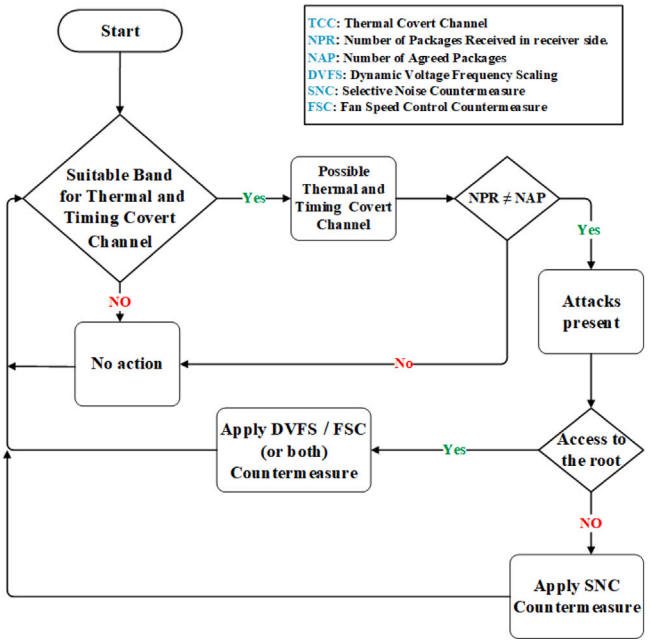

Fig. 4. Diagram of process of detection and defence of Thermal and Timing Covert Channel Attacks in single covert channels (process1).

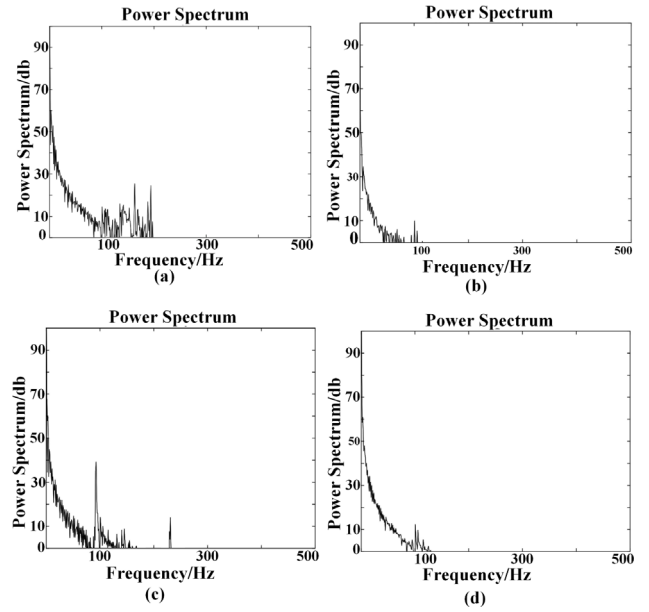

Fig. 5. Thermal and Timing covert channel attacks are using a single channel. (a) Multi attacks without countermeasure, (b) Multi attacks with DVFS countermeasure, (c) Multi attacks with selective noise countermeasure, (d) Multi attacks with Fan speed control counter.

approach leverages direct control over the physical hardware components to mitigate risks, rather than adding complexity or potential performance overhead associated with the software-based introduction of selective noise.

As shown in Tables 4 and 5, integrating a DVFS-based countermeasure with a fan speed control-based one dramatically boosts BER to 98% while significantly reducing power consumption. This remarkable improvement stems from the combined effect of optimizing fan operation and dynamically adjusting processor voltage and frequency. Furthermore, by dynamically adjusting fan speed based on the system's cooling needs, we can avoid unnecessary full-speed operation,

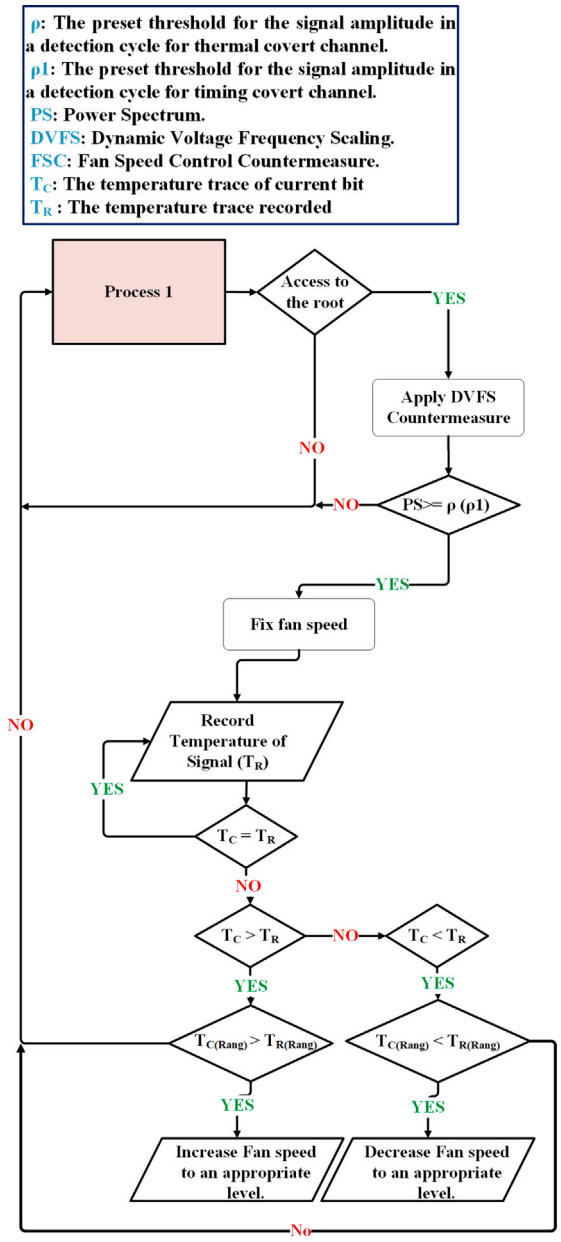

Fig. 6. Diagram of the process combined defence strategy in multi-covert channels.

leading to substantial power savings. Additionally, DVFS allows us to dynamically adjust processor voltage and frequency based on real-time workload requirements, further minimizing power consumption without compromising performance. This synergistic combination of countermeasures effectively thwarts multi-covert channel attacks while ensuring optimal power efficiency for our system (Figs. 6 and 7). Furthermore, dynamically scaling processor voltage and frequency further amplifies the power reduction. Adjusting these parameters in the transmission process based on workload demands allows the system to operate at the minimum power required for its current task, effectively squeezing out every ounce of efficiency. This targeted approach ensures adequate cooling while minimizing energy expenditure. Overall, the combined effect of these two countermeasures creates a powerful synergy, delivering a dramatic boost in BER while simultaneously slashing power consumption. This technique underlines the potential of employing a multifaceted approach to combating covert channel attacks while keeping power efficiency in mind (see Fig. 8).

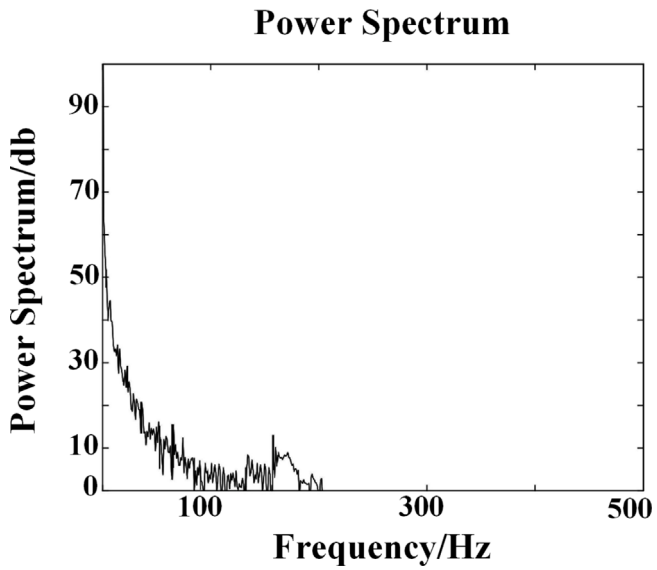

Fig. 7. Double covert channel with DVF and fan speed countermeasure.

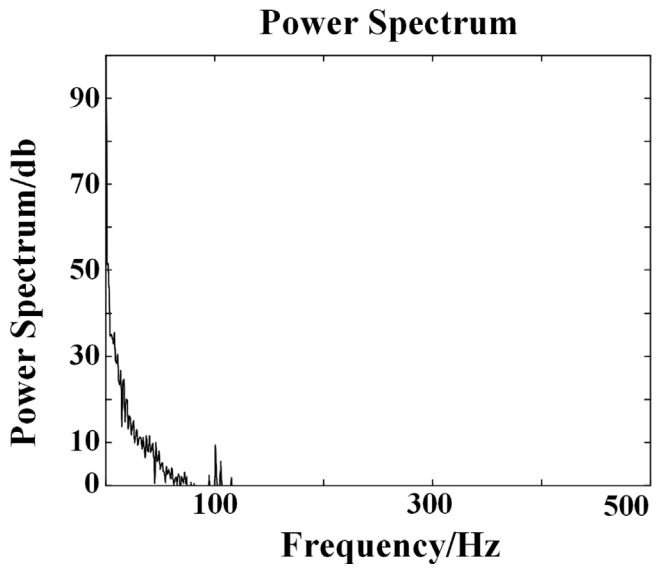

Fig. 8. Single covert channel with DVF and fan speed countermeasure.

4. Experimental results

4.1. Experimental setup

Our research involved conducting two distinct sets of experiments to explore the effectiveness of different covert channels for data transmission within a multicore computing environment. Initially, we examined the implications of operating two separate covert channels dedicated to thermal and timing data transmissions, respectively. Subsequently, we shifted our focus to a unified approach where a single covert channel was employed for both thermal and timing transmissions. These experiments were primarily conducted using the Sniper-v7.2 multicore simulator, renowned for its speed and accuracy in simulating multicore systems. The power consumption analysis was supported by McPAT-v1.0, a comprehensive framework designed for modelling the power, area, and timing aspects of multithreaded and multicore architectures, as highlighted in the work of [39].

This study aims to devise effective countermeasures against covert timing and thermal channel attacks targeting our device. These attacks typically aim to surreptitiously transfer sensitive information from a covert channel. To simulate these thermal covert channel attacks, we utilized the HotSpot-v6.0 thermal model to accurately generate and measure temperature fluctuations across all cores. For benchmarking purposes, we selected a subset of programs from the PARSEC [40] and SPLASH-2 suites [36], running these benchmarks on cores not designated as transmitters or receivers to mimic a realistic multicore processing environment.

Our experimental framework integrated the Sniper and HotSpot simulation tools, alongside tailored thermal covert channel programs, to create a robust simulation setup. The configurations and specific details of our simulation environment are tabulated in Table 2. This comprehensive approach not only facilitates a deeper understanding of the vulnerabilities posed by covert channel attacks but also aids in the development of sophisticated strategies to protect sensitive data within multicore systems.

Table 2
Simulation configuration.

Instruction set architecture	×86-64
Operating system	Ubuntu 16.04.5 LTS
Number of cores	4×4, 8×8
Frequency of CPUs (MHz)	2000
Transmission frequency	10 Hz, 20 Hz, 50 Hz, 80 Hz, 100 Hz, 150 Hz, etc.
Packet size in bits	64
Benchmarks of PARSEC	Black–Scholes, Fluidanimate, Canneal, X-264, Streamcluster, Freqmine, Swaptions, Dedup.
Benchmarks of SPLASH-2	Raytrace, Barnes
Number of SMT threads per core	2
Chip thickness	0.15 mm
Heat sink side	0.06 m
Heat sink thickness	6.9 mm
Heat sink thermal conductivity	400 W/(m K)
Specific heat capacity of heat sink	3.55×10^6 J/(m^3 K)
Silicon thermal conductivity	100 W/(m K)
Silicon specific heat capacity	1.75×10^6 J/(m^3 K)

Table 3
Detection accuracy of P_{acc}.

Type of multi-covert channel	System Sizes	P_{acc}
Double multi-covert channel	8 × 8	97%
Single multi-covert channel	8 × 8	95.5%

4.2. Selecting the parameters for detection

A key aspect of our investigation involved measuring the positioning accuracy (P_{acc}), which is crucial for accurately identifying and tracking the locations of transmitters and receivers involved in thermal covert channel attacks.

The positioning accuracy is defined as follows:

$$P_{acc} = \begin{cases} 100\% & \text{for } P_{detected} = P_{transmitter/receiver} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where, $P_{transmitter}$ is the position (core id) of the detected thermal covert channel transmitter/receiver cores. $P_{receiver}$ is the actual position of the transmitter/receiver cores.

Table 3 presents the average accuracy of positioning transceivers in different multicore systems. Another remarkable fact is that our detection methodologies had no detrimental effects on the average power consumption of the systems. We calculated the average power consumption as follows:

$$P_{avg} = \frac{\sum_{c=1}^{N_c} \left(\sum_{u=1}^{N_c} P_{cu} \right)}{N_c} \text{ [W]} \quad (2)$$

Table 4
Experimental results of double covert channel for timing and thermal.

Countermeasures	BER (Thermal CC)	BER (Timing CC)	Avg power consumption (W)
DVFS	92%	96%	27.77
Selective noise	94%	95%	33.50
Fan speed controlling	95%	93%	25.45
Double covert channel with DVFS and Fan speed	97%	98%	26.86

Table 5
Experimental results of single covert channel for timing and thermal.

Countermeasures	BER	Avg power consumption (W)
DVFS	96%	24.16
Selective noise	93%	29.53
Fan speed controlling	95%	23.38
Single covert channel with DVFS and Fan Speed	98%	21.45

where, N_c: The number of cores. c: representing an individual core (ranges from 1 to N_c). u: representing utilization (ranges from 1 to N_c). P_{cu}: representing the power consumption associated with the utilization of core u by core c. This power consumption metric is derived from our simulation framework, where we model the power usage based on the workload traces of each core. By analysing the workload's intensity and duration on each core, we can estimate the power consumed by u when performing tasks related to c.

The dynamic changes in core activity levels are monitored by tracking various parameters such as CPU frequency, utilization, and thermal outputs. These parameters are captured in real-time through system profiling tools that observe switching activities, voltage variations, and workload traces. These monitored variables are then fed into a power model, which computes the power consumption based on core utilization patterns. Specifically, for each core, the model calculates power consumption as a function of its current frequency and workload intensity. The integration of these dynamic changes into the power model follows established methodologies, such as using McPAT (an architectural power, area, and timing modelling framework), which can simulate how energy is consumed by different components based on operational metrics. This model allows us to estimate the power consumed by each core under varying workloads, ensuring that the power consumption estimation is accurate and reflective of real-time system operations.

4.3. Experimental results

Our experiments, illustrated in Figs. 2 and 4, reveal distinct behaviours across different countermeasure systems. Specifically, the application of Dynamic Voltage and Frequency Scaling aims to disrupt covert communications through thermal and timing channels, as shown in Figs. 2 and 4(b). Furthermore, selective noise-based countermeasure introduces significant fluctuations in the power spectrum due to the strategic injection of noise into the system. Such noise addition distorts the original signal, manifesting as additional peaks and troughs in the power spectrum analysis. Conversely, the strategy of adjusting fan speeds, illustrated in Figs. 2 and 4(d), targets a reduction in system temperature. This thermal management technique directly impacts the system's operational frequency, demonstrating a different method of mitigating covert channel attacks. Moreover, the selective application of noise-based countermeasures, referenced in Figs. 2 and 4(c), further exemplifies the diverse tactics available to combat these security vulnerabilities. By directly applying these countermeasures to the processor cores running the targeted threads, we effectively neutralize potential avenues for timing, thermal, and multi-covert channel leaks. The effectiveness of these strategies is quantitatively summarized in Table 4, offering a detailed comparison of their impact on system security. Through this comprehensive study, we illustrate not only the varied responses of our system to each countermeasure but also underscore the nuanced understanding required to effectively safeguard against sophisticated covert channel attacks.

Table 4 presents the outcomes of an investigation into the thermal effects of using dual covert channels for data transmission within a communication system. The findings highlight a considerable elevation in system temperature attributable to the transmission process, underscoring a potential risk to system integrity. The escalation in temperature prompts the exploration of alternative methods to mitigate overheating risks. Solutions might include the integration of advanced cooling mechanisms or the adoption of more efficient data transmission protocols aimed at minimizing thermal output. In response to these challenges, we propose a novel strategy outlined in Table 5, which consolidates timing and thermal data transmissions through a singular covert channel. This streamlined approach not only aims to reduce heat production but also offers several additional benefits. By employing a unified channel, the complexity of the communication system's architecture is reduced, enhancing its manageability and potentially lowering operational costs. Moreover, this strategy inherently bolsters system security, complicating the efforts of adversaries to exploit covert channels for the unauthorized transfer of sensitive information. The difficulty in disguising and conducting covert transmissions via a single channel significantly elevates the system's resistance to such attacks, reinforcing its overall security posture.

As tabulated in Table 5, while using a single covert channel for both timing and thermal covert channel transmissions can help, the second approach may result in other issues, such as a loss of packets during transmission. This is because using a single channel for both types of covert channel transmissions can increase the likelihood of interference or signal overlap, which can result in lost or corrupted data packets. In addition, the use of a single channel may also limit the amount of data that can be transmitted, which can result in a slower overall transmission rate. To address these issues, it may be necessary to optimize the communication system's design and configuration. This could involve adjusting the transmission rate, optimizing the signal processing algorithms, or implementing error correction techniques to reduce the risk of lost or corrupted data packets.

4.3.1. Impact of detection and countermeasures on system performance

In this section, we present a detailed results analysis to evaluate the impact of our detection mechanisms and countermeasures on system performance. The analysis focuses on key metrics such as processing latency, energy efficiency, and overall system performance. Processing latency was measured by evaluating the time required to complete a set of benchmark tasks under different system configurations. We conducted this analysis across three different situations:

1. Baseline: System running without any covert channel detection or countermeasures.

2. With Detection Only: System with the covert channel detection mechanism present but without countermeasures.

3. With Detection and Countermeasures: System with both detection and countermeasures, including enhanced DVFS, selective noise injection, and fan speed control.

The tasks were selected from the PARSEC and SPLASH-2 benchmark suites to simulate a range of typical multicore workloads. The latency was recorded under multiple system loads to determine the impact on both light and heavy usage models.

The processing latency without any countermeasures was 50 ms on average. When the detection mechanism was enabled, the latency slightly increased to 52 ms. With detection and all countermeasures in place, the latency increased to 58 ms. These values indicate that our proposed countermeasures have a moderate impact on processing time.

Energy efficiency was measured in terms of power consumption relative to the system's workload. We used energy per operation as

Table 6
Detailed power consumption for double covert channel.

Countermeasures	Avg power consumption (W)	Improvement (%)
Baseline (No Countermeasures)	30	–
DVFS	27.77	7.4
Fan speed controlling	25.45	15.2
Selective noise	33.50	−11.7
Double covert channel with DVFS and Fan Speed	26.86	10.5

Table 7
Detailed power consumption for single covert channel.

Countermeasures	Avg power consumption (W)	Improvement (%)
Baseline (No Countermeasures)	30	–
DVFS	24.16	19.5
Fan speed controlling	23.38	22.1
Selective noise	29.53	−1.8
Single covert channel with DVFS and Fan speed	221.45	28.5

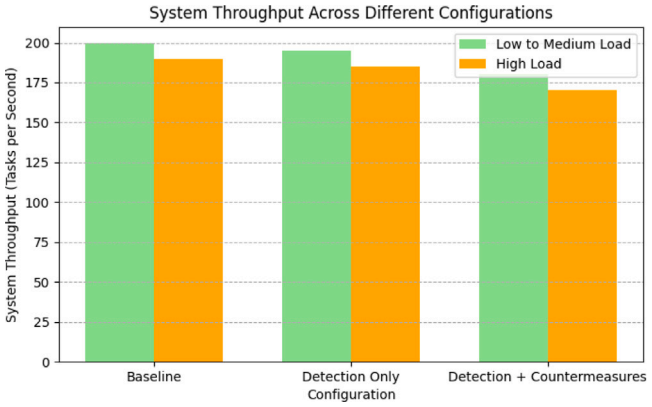

Fig. 9. The processing latency comparison.

the primary metric, analysing how our countermeasures influenced power consumption. As the results illustrate, the enhanced DVFS and fan speed control modules resulted in a moderate decrease in energy consumption, as they optimized core voltage and frequency dynamically. However, the selective noise injection introduced a slight increase in power usage due to the additional processing required for noise generation (see Tables 6 and 7).

These findings reveal that integrating DVFS with fan speed control as countermeasures effectively disrupts multi-covert channel attacks. Additionally, this approach significantly enhances power consumption efficiency, especially when addressing single covert channel scenarios. This combined strategy offers a new solution for securing multicore systems against covert channel vulnerabilities while ensuring energy efficiency.

To evaluate performance comprehensively, each configuration was tested under different load conditions: low, medium, and high, representing different operational scenarios. A balanced workload comprising both CPU-intensive and memory-bound tasks was utilized to provide a detailed view of the system's performance.

As illustrated in Fig. 9, the baseline system without countermeasures maintains an average throughput of 200 tasks per second under low to medium load conditions. However, under high load, throughput experienced a slight decrease to 190 tasks per second. When the detection-only system was enabled, throughput showed a slight reduction to 195 tasks per second under low to medium load and 185 tasks per second under high load, reflecting the minor overhead introduced by continuous monitoring. With both detection and countermeasures activated, the system's throughput averaged 180 tasks per second under low to medium load and decreased further to 170 tasks per second under high load.

The most notable reduction resulted from the selective noise injection, which increased processing overhead, and DVFS and fan speed adjustments, which lowered core frequencies to mitigate multi-covert channels. These findings demonstrate that the proposed methods effectively secure multicore systems while maintaining a manageable impact on overall performance.

Additionally, our proposed work offers several advantages over a single-channel approach, summarized below:

Enhanced Detection and Mitigation: Multi-covert channels allow for the simultaneous handling of different types of covert channels, such as thermal and timing channels. High BER illustrates (Tables 4 and 5) that this facilitates a more comprehensive defence strategy that can effectively address multiple possible risks at the same time.

Improved Security Measures: By combining methods to counteract both thermal and timing channels, the overall security of the system is enhanced (98%). Each type of attack has its characteristics, and exploiting both can create a more robust defence mechanism. For instance, adjustments in fan speeds can affect thermal covert channels, whereas timing channels may be countered by modifying operational frequencies through techniques like DVFS.

Highly Effective Countermeasure Strategies: As experimental results indicate, employing multiple covert channels allows for the use of varied countermeasures that adapt to each channel's specific needs, potentially increasing the effectiveness of the overall security strategy. For example, the use of selective noise-based countermeasures can disrupt the signal patterns specific to each type of covert channel, and we can get a high bit error rate (94%).

Table 8 provides a comparative analysis between our proposed methods and several existing works. As highlighted in the table and discussed previously, the majority of current research tends to concentrate on a single type of covert channel attack. Additionally, in the context of timing covert channel attacks, most researchers did not consider power consumption in line with countermeasures. However, our study reveals that while accounting for two types of covert channel attacks may lead to increased power consumption, the high bit error rates observed suggest that our detection and defence strategies are significantly more efficient.

Despite all the benefits this work has, it also introduces additional complexity and challenges.

Increased Complexity: Designing and implementing a covert channel that combines multiple channels is more complex than a single-channel approach. For instance, combining thermal and timing channels involves specific design considerations to ensure that they do not interfere with each other while still effectively transmitting data. This requires a higher level of precision in system design and implementation compared to managing a single covert channel.

Communication Requirements: Both the sender and receiver need to be set up to effectively utilize both channels for data transmission. Potential Interference: The timing and thermal channels may interfere with each other, making it difficult to distinguish between the two channels.

5. Conclusion

This paper delves into the novel approach of utilizing thermal and covert timing channels to establish a multi-covert channel for data transmission. It outlines the steps in this technique, sheds light on our detection mechanisms, and presents a comprehensive evaluation of the results. The study reveals that employing two channels enhances accuracy but incurs higher power consumption while utilizing a single channel lowers power consumption but may lead to data loss. These

Table 8
Comparison of our approaches by existing works.

Cite	Attack	BER	Avg power consumption (W)
[4] with DVFS	Thermal covert channel	92%	22.31
[31] with Selective noise	Thermal covert channel	94%	22.81
[22] with DVFS and Fan speed controlling	Thermal covert channel	95%	19.65
[10] with Bitrate modulation	Timing covert channel	92%	×
[41] with Signal processing techniques	Timing covert channel	84%	×
[42] with Multi-Stage Verification	Timing covert channel	80%	×
Proposed work: Double covert channel with DVFS	Thermal and Timing covert channel	92%	27.77
Proposed work: Double covert channel with Selective noise	Thermal and Timing covert channel	94%	33.50
Proposed work: Double covert channel with DVFS and Fan speed	Thermal and Timing covert channel	97%	26.86
Proposed work: Single covert channel with DVFS	Thermal and Timing covert channel	96%	24.16
Proposed work: Single covert channel with Selective Noise	Thermal and Timing covert channel	93%	29.53
Proposed work: Single covert channel with DVFS and Fan speed	Thermal and Timing covert channel	98%	21.45

inherent trade-offs underscore the crucial need for meticulous design and careful consideration of accuracy and power consumption factors. Additional research is warranted in this domain to further refine and optimize these techniques.

CRediT authorship contribution statement

Parisa Rahimi: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Amit Kumar Singh:** Writing – review & editing, Supervision. **Xiaohang Wang:** Writing – review & editing. **Seyedali Pourmoafi:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] K.S. Das, TPPD: Targeted pseudo partitioning based defence for cross-core covert channel attacks, J. Syst. Archit. 135 (2023) 102805.
[2] Xiao, et al., Exploiting the microarchitectural leakage of prefetching activities for side-channel attacks, J. Syst. Archit. 139 (2023) 102877.
[3] Rahimi, et al., Trends and challenges in ensuring security for low-power and high-performance embedded SoCs, in: 14th MCSoC, IEEE, 2021.
[4] Huang, et al., Detection of and countermeasure against thermal covert channel in many-core systems, IEEE Trans. (2021).
[5] J. Chen, G. Venkataramani, An algorithm for detecting contention-based covert timing channels on shared hardware, in: Proceedings of the Third Workshop on Hardware and Architectural Support for Security and Privacy, 2014, pp. 1–8.
[6] L. Caviglione, Trends and challenges in network covert channels countermeasures, Appl. Sci. (2021).
[7] G. Gómez, et al., Smart detection of obfuscated thermal covert channel attacks in many-core processors, in: 2023 60th ACM/IEEE Design Automation Conference, DAC, 2023.
[8] Wu, et al., Defending against thermal covert channel attacks by task migration in many-core system, in: IEEE 3rd International Conference on Circuits and Systems, ICCS, 2021.
[9] B. Jankowski, et al., PadSteg: introducing inter-protocol steganography, Telecommun. Syst. 52 (2) (2013) 1101–1111.
[10] S. Soderi, R.D. Nicola, CONNECTION: Covert channel network attack through bitrate modulation, in: Information Security and Applications, Springer, 2023.
[11] P.C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems, Springer, 1996.
[12] A. Onur, C.K. Ko, Microarchitectural Attacks and Countermeasures, Springer, 2009.
[13] Biswas, et al., A survey of timing channels and countermeasures, ACM Comput. Surv. (2017).
[14] D. Bernstein, Cache-timing attacks on AES, in: Department of Mathematics, Statistics, and Computer Science, M/C 249, The University of Illinois at Chicago, 2005.
[15] Cotroneo, et al., Timing covert channel analysis of the VxWorks MILS embedded hypervisor under the common criteria security certification, Comput. Secur. (2021).
[16] Kadam, et al., Rcoal, mitigating GPU timing attack via subwarp-based randomized coalescing techniques, in: IEEE International Symposium on High-Performance Computer Architecture, IEEE, 2018.
[17] Masti, et al., Thermal covert channels on multi-core platforms, in: 24th USENIX Security Symposium, 2015.
[18] Bartolini, et al., On the capacity of thermal covert channels in multi-cores, EuroSys (2016).
[19] Shirvani, et al., A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: taxonomy and challenges, J. King Saud Univ.- Comput. Inf. Sci. (2020).
[20] S. Cabuk, et al., IP covert channel detection, ACM Transm. Inf. Syst. Secur. 12 (4) (2009) 22:1–22:29.
[21] Huang, et al., On countermeasures against the thermal covert channel attacks targeting many-core systems, in: 57th ACM/IEEE Design Automation Conference, DAC, IEEE, 2020.
[22] Rahimi, et al., Fan speed control based defence for thermal covert channel attacks in multi-core systems, in: 29th ICECS, IEEE, 2022.
[23] J. Szefer, Survey of microarchitectural side and covert channels, attacks, and defenses, J. Hardw. Syst. Secur. (2019).
[24] Alcaraz, et al., Covert channels-based stealth attacks in industry 4.0, IEEE Syst. J. 13 (4) (2019).
[25] R.A. Kemmerer, Shared resource matrix methodology: An approach to identifying storage and timing channels, ACM Trans. Comput. Syst. 1 (3) (1983) 256–277.
[26] M. Hanspach, J. Keller, On the implications, the identification and the mitigation of covert physical channels, in: 9th Future Security Conference, 2014.
[27] Ge, et al., A survey of microarchitectural timing attacks and countermeasures on contemporary hardware, J. Cryptogr. Eng. (2018).
[28] Jiang, et al., A Complete Key Recovery Timing Attack on a GPU, HPCA, IEEE, 2016.
[29] Dhananjay, et al., High bandwidth thermal covert channel in 3-D-integrated multicore processors, IEEE Trans. Very Large- Scale Integr. (VLSI) Syst. 30 (11) (2022).
[30] Al-Eidi, et al., Covert timing channel analysis either as cyber attacks or confidential applications, Sensors 20 (8) (2020).
[31] Rahimi, et al., Selective noise-based power-efficient and effective countermeasure against thermal covert channel attacks in multi-core systems, J. Low Power Electron. Appl. (2022).
[32] Wang, et al., Combating enhanced thermal covert channel in multi-/many-core systems with channel-aware jamming, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 39 (11) (2020).

[33] Alagappan, et al., DFS covert channels on multi-core platforms, in: 2017 IFIP/IEEE International Conference on Very Large Scale Integration, VLSI-SoC, IEEE, 2017.
[34] Y. Lu, L.D. Xu, Internet of things (IoT) cybersecurity research: A review of current research topics, IEEE Internet Things J. 6.2 (2018).
[35] Sadhu, et al., Internet of things: Security and solutions survey, Sensors 22 (2022) 19.
[36] Singh, et al., PLASH: Stanford parallel applications for shared-memory, ACM SIGARCH Comput. Archit. News 20 (1) (1992) 5–44.
[37] Sghaier, et al., Fast constant-time modular inversion over fp resistant to simple power analysis attacks for IoT applications, Sensors 22 (7) (2022).
[38] I.R. Palupi, W. Raharjo, The utilization of signal analysis by using short time fast Fourier transform, in: RSF Conference Series: Engineering and Technology 1.1, 2021.
[39] Wang, et al., Modeling and analysis of thermal covert channel attacks, IEEE Trans. Comput. 72 (2) (2022).
[40] C. Bienia, K. Li, Fidelity and scaling of the PARSEC benchmark inputs, in: International Symposium on Workload Characterization, IEEE, 2010.
[41] Edwards, et al., Using covert timing channels for attack detection in MANETs, in: IEEE Military Communications Conference, 2012.
[42] C. Liang, et al., Building covert timing channel of the IoT-enabled MTS based on multi-stage verification, IEEE Trans. Intell. Transp. Syst. (2021).