WORKSHOP REPORT

NSF WORKSHOP ON SIDE AND COVERT CHANNELS IN COMPUTING SYSTEMS*

Workshop Chairs:

GURU VENKATARAMANI, George Washington University
PATRICK SCHAUMONT, Virginia Tech

Research Area Chairs:

DAVID KAELI, Northeastern University (Computer Architecture)
MILOS PRVULOVIC, Georgia Institute of Technology (Computer Architecture)
SRINIVAS DEVADAS, Massachusetts Institute of Technology (Computer Systems)
DMITRY PONOMAREV, Binghamton University (Computer Systems)
GANG QU, University of Maryland-College Park (Computer Hardware)
YUNSI FEI, Northeastern University (Computer Hardware)

February, 2019

*Sponsored by National Science Foundation, USA

Executive Summary

The leakage of sensitive information is a fast-growing concern among computer users. Side- and covert channels have particularly gained attention recently due to their potential to reveal sensitive data to untrusted parties. Side channels are information leakage channels where an adversary can decipher victim's data through silently monitoring the computing activity via physical effects such as timing, power or electromagnetic analysis. Covert channels, in contrast, work by having a malicious insider, or trojan, who intentionally colludes with the adversary to exfiltrate secrets.

Side and covert channels have become major concerns for the computer industry. In early 2018, the Meltdown and Spectre attacks demonstrated that hardware implementation effects in commercial processor hardware enabled new, previously undiscovered side-channel and covert-channel leakage. These attacks highlight the notoriety of information leakage channels, and they stress the immediate need to address the security risks resulting from them.

Side and covert channels present a multi-layer security challenge in the computing stack as they usually manifest through exploiting multiple aspects of the computing stack. Hence, it is critical to discuss these attacks and the corresponding defense methodologies in the context of application software, middleware and their interactions with computer architecture and hardware layers.

Computer systems encompass user applications and system software layers. Due to the growing complexity of software and other cyber-physical systems, it has now become very important to carefully examine the trusted computing components to avoid side and covert channel-based information leakage. Such analysis should also extend across other system layers, including application libraries, operating systems, hypervisors, firmware, and interactions with hardware.

Computer architecture studies the hardware-software interfaces, as well as micro-architectural and architectural abstractions for better programmability and usability of hardware. In this context, it becomes crucial to understand individual hardware units and their interactions, such that one can estimate side and covert channel-based information leakage during architecture design. Indeed, performance-tuning features inside the microarchitecture implementations, such as speculative instruction execution, are shown to have side-channel vulnerabilities. The time is ripe to reconsider the design and verification methodologies for Instruction Set Architectures (ISA) that govern the interface between the users and the hardware. Furthermore, computer programmers must be aware of the microarchitecture-level variabilities that affect their system security.

At the hardware level, side and covert channel leakage is directly visible as a side effect of physical computing. Emerging hardware technologies and specialized accelerator modules demonstrate the vulnerabilities exposed by the hardware space. As such, the hardware attacker is one who operates beyond the realm of logical boundaries, and who exploits physical effects. Computer hardware designers have yet to fully consider how to address such powerful adversaries.

To address such unique challenges arising from side and covert channels spanning multiple layers of the computing stack, the NSF Workshop on Side and Covert Channels presented a forum for researchers from three different research communities, namely computer systems, computer architecture and hardware. The workshop enabled the broader computing community to discuss and highlight these issues confronting modern computer systems.

Through plenary talks and breakout discussions, researchers from individual areas developed a detailed set of recommendations to address the challenge of side and covert channels. We summarize the findings by the researchers into five key aspects as below:

1. Effective protection against information leakage channels requires hardware-software cooperative solutions. Hardware solutions are usually efficient, but they are inflexible in adapting to threats emerging after deployment. Software solutions have better adaptability but suffer from performance limitations. Therefore, we have to invest in hybrid solutions that span both hardware and software. Research is needed into hardened micro-architectural features and hardware-software protection, in addition to innovation at the ISA level for better auditing and control of hardware resources.
2. Software simulators and test environments are crucial for rapid and rigorous security proofing, especially in emerging computing platforms such as domain-specific computing and quantum computing. Broader community participation efforts can also address these challenges, through sharing timing traces from real application runs. Simulators, testbeds, workloads, and traces relevant to side and covert channel leakage can support research in this field. Virtualized labs, for broader set of researchers to collect leakage-related data on devices, can benefit and create effective solutions for side and covert channels.
3. Objective quantification methods will enable evaluation and comparison of different defense approaches. The effectiveness of mitigation techniques must be quantified and balanced with their relative cost. This requires new methodologies that can evaluate the trade-offs between risk and performance.
4. Cross-layer research will enable the study of side and covert channels across layers of the computing stack. Such studies will offer insight into the information leakage threats across modern computing platforms, such as cloud computing and IoT. The outcome of this research can lead to a better taxonomy for side and covert channel attacks and more effective defense strategies.
5. Side and covert channels are a threat today. Multi-domain investment and support across federal, industry and academia is needed. Federal agencies must support researchers that study side and covert channels from multiple perspectives and across the computing stack. Industry must invest in a hardware vulnerability database, and support the use of security-aware design tools. Academia must participate in prototyping efforts and engage with industry to learn about the real issues that confront computer products today. Publication avenues must increase the visibility and awareness of side and covert channels.

Can side channels be a new frontier in cybersecurity research? The answer to this question is an emphatic *yes* due to the following reasons:

1. There are many different types of side and covert channels in computing systems stemming from various layers of the computing stack, and most of these remain unknown until someone exposes it.
2. Side/covert channels are expensive to remove and impractical to close completely. Instead, we have to understand how to limit the damages caused by them in real-world computing environments.
3. Side channels may be considered beneficial in some cases such as when detecting hardware trojans, detecting anomalous behavior in applications. There is potential benefit in projects that study how to leverage side channels for useful things.

Organization. The workshop attendees comprised 38 participants from academia, 11 participants from industry and 5 participants from Government and industry funding agencies. The workshop was held on March 22-23, 2018 at the Marvin Center in George Washington University's Foggy

Bottom Campus in Washington, DC. The workshop program was chaired by Guru Prasadh Venkataramani (George Washington University) and Patrick Schaumont (Virginia Tech). The technical topics were formulated, and the respective breakout sessions were chaired by experts from three different research areas. For computer architecture area, the chairs were David Kaeli (Northeastern University) and Milos Prvulovic (Georgia Tech). For computer systems area, the chairs were Srini Devadas (MIT) and Dmitry Ponomarev (Binghamton University). For computer hardware area, the chairs were Yunsi Fei (Northeastern University) and Gang Qu (University of Maryland).

The workshop program, presentation materials, talk videos and participants are archived online at https://sites.google.com/view/sccs2018/agenda. Two keynote talks inaugurated the workshop. The first talk, presented by Daniel Genkin (University of Pennsylvania) and Yuval Yarom (University of Adelaide), described the real-world threats due to side- and covert channels. The second talk, presented by Yan Solihin (North Carolina State University), stressed the need to understand side channels, and how they present a new frontier in cybersecurity. These keynote talks formed the catalyst among researchers to capture the threats posed by side and covert channels and discuss the future of cybersecurity research in the context of information leakage attacks.

The area chairs presented an overview of challenges and future research directions in their respective fields. There were six plenary talks in technical sessions, three on threats and upcoming challenges in side and covert channel research, and three on defense strategies. Each plenary talk involved a 30-minute overview by three different researchers from the areas of computer architecture, systems, and hardware areas providing an in-depth coverage of technical challenges and solution roadmap to alleviate them.

Researchers from all of the individual areas participated in breakout sessions that were led by the respective area chairs. Important research challenges and contributions from each area were thoroughly discussed and summarized to shape the future research landscape for side and covert channel areas. This report captures the findings of the researchers during the two-day discussion and outlines the important challenges that should be addressed by this community of researchers to confront side and covert channels in computing systems.

Report Outline. To ensure a comprehensive discussion, the workshop organizers identified three different abstraction levels to discuss side and covert channels. This multi-level formulation considers side and covert channel leakage across the entire computing stack.

- The upper layer of **Computer Systems** encompasses computer systems, networks, middleware and cloud computing platforms.
- Second, **Computer Architecture** covers the detailed architectural design of computers, including the micro-architecture design and its interaction with the system/software layers.
- Third, **Hardware** covers the detailed hardware design expressed at cycle-accurate precision or lower.

An organizational decision taken was to discuss the topic of side and covert channel leakage both as a vulnerability as well as an opportunity for designing mitigation roadmap. The combined discussion from the viewpoint of attack and defense is a common technique in the security community, to apply a common framework to evaluate and reason about potential vulnerabilities. Furthermore, it ensures thorough scrutiny of the proposed mitigation techniques. Because of the complexity of the side and cover channel leakage problem, it is not uncommon that a mitigation to a given vulnerability also creates a new vulnerability against another attack vector.

The structure of this report reflects these organizational decisions. Chapter 1 introduces threats and attacks in side and covert channel leakage, and the vulnerabilities in contemporary computer systems and computer hardware. Chapter 2 offers a roadmap for better mitigation and defense, and summarizes directions for effective solution strategies. The material in these chapters is based on plenary talks by the workshop attendees. Chapter 3 develops a research vision and research recommendations for each of the three abstraction levels considered in the workshop: Computer Systems, Computer Architecture, and Hardware. The material in this chapter is based in part on plenary talks prepared by the area chairs in each of Computer Systems, Computer Architecture, and Hardware. It is also based in part on three breakout sessions, organized during the workshop to stimulate in-depth discussion in each area among the experts. Chapter 4 covers the existing testbeds for side and covert channel research, and stresses the need to grow the infrastructure for future research needs. Chapter 5 covers the educational aspects and workforce training needs to effectively address side and covert channels in computing systems.

Acknowledgments. We sincerely thank National Science Foundation for sponsoring this 2-day workshop under Award no. 1747723 and providing an opportunity for researchers to interact on this important topic. We thank NSF Program Managers Yan Solihin, Nina Amla, Sandip Kundu, Samee Khan, Matt Mutka and CNS Division Director Kenneth Calvert for their support and participation in this workshop. We also thank all of the participants and speakers who participated in this workshop, offered insightful thoughts, contributed to the discussions and for having made it a grand success.

1. Background on Threats due to Side and Covert Channels

The term Instruction Set Architecture (ISA) was first defined on IBM 360 to provide a published definition of hardware services that software programmers could easily target with software [1]. The ISA abstraction enables computer architects to develop hardware optimizations invisible to the software programmer. These optimizations are commonly referred to as micro-architectural features. They enable hardware designers to optimize hardware designs for performance, power, reliability or security, while still adhering to the same hardware/software ISA interface.

While such abstractions are useful for programmability and ease of use for computing architectures, they can also potentially turn into side channels if malicious users were to exploit them. Recently, both architectural ISA-level and micro-architectural features have been demonstrated to provide rich attack surfaces for side and covert channels. As an example, at the ISA level, individual instructions that are used for encryption and decryption can be attacked by analyzing their secret-data-dependent power consumption [2]. At the micro-architectural level, the memory behavior is a common attack surface [3, 4, 5, 6], leaking address information measured through timing analysis. Performance enhancement techniques like speculative execution, branch prediction, multithreading and dynamic frequency scaling have been shown to cause information leakage [7, 8, 9, 10].

Individual instructions can also serve as attack surfaces based on the power dissipation associated with data operand values [11]. Information can be extracted, and encryption keys recovered using Simple Power Analysis [12] or Correlation Power Analysis [13], and even power management units [14]. Electromagnetic analysis have been shown to be a viable side channels [15]. Multi-threading, a commonly used feature in many microprocessors to hide latency [16], can be compromised to produce a covert or side channel [8]. Even single-threaded execution can be attacked, based on memory reference patterns [17, 18], or based on the behavior of program control flow execution [19, 20, 21, 22].

If an attacker can observe execution (using timing, power or electromagnetic emanation) on another computing platform through side channel, or if exfiltration can occur on the same platform through a covert channel, the attacker can leverage these sources of leakage, cracking encryption keys and leaking secrets. Figure 1 shows the vulnerabilities of CPU architecture with various actors and shared hardware and software components that could result in side and covert channel exploits.

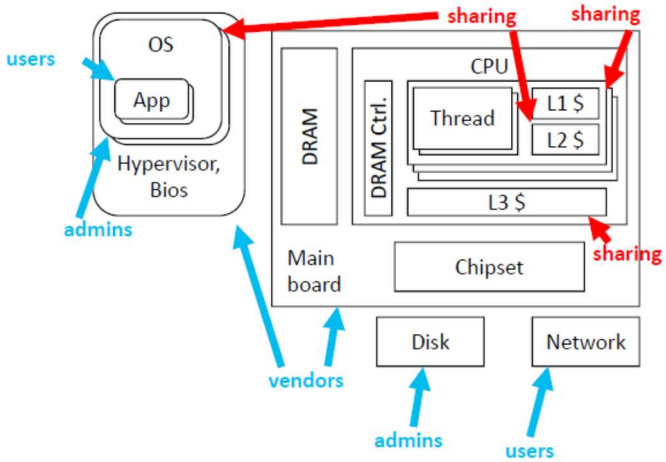

Figure 1. Computer System Architecture with different actors (users, admins and vendors), and shared hardware-software components that lead to side and covert channels.

To illustrate the potential of side and covert channels and their means of exploitation, we describe several broad classes of channels that have been studied by the research community. The attacks are organized based on targeted hardware that is being exploited using *a physical effect such as timing, power or electromagnetic analysis*.

1. Memory subsystems: leakage occurring from memory-related computer operations
2. Control flow: leakage occurring from decision-making in applications
3. Microarchitecture: leakage occurring in microarchitecture and at the boundary of hardware and software
4. Emerging computing platforms: leakage occurring in new and future computing platforms
5. Leakage in IoT: leakage occurring in low-end embedded computers

1.1 Leakage from Memory subsystems

To aid application performance, computer designers introduced the concept of a memory hierarchy [23]. Memories closer to the computing pipeline are faster, yet smaller (in size). Today's CPUs and GPUs include multiple levels of caches. Memory caches have been studied extensively [24], considering design issues like general organization and replacement, coherency protocols [25], and a range of speculation mechanisms [26] to anticipate the future address stream.

Cache memories have two key features that make them susceptible to attacks. First, they can significantly impact performance through cache-hit and cache-miss behavior. Second, they maintain program state. Common cache-based attack techniques use one of the following schemes.

1. The Prime and Probe attack fills the cache with attacker-generated data, and then checks what blocks are evicted after performing an encryption [27].
2. The Evict and Time attack monitors the execution time of the victim process, before and after evicting cache contents [28].
3. The Flush and Reload attack flushes a line from the cache and then measures the time to reload the memory line [29].
4. The Coalesce attack considers the spatial locality associated with an encryption algorithm and correlates this behavior with cache access timing [30].

A recent survey describes side channel attacks on caches, as well as countermeasures [31]. Given that a modern CPU has up to three levels of caching, and that each cache has multiple performance-enhancing features, cache memories are one of the most commonly exploited side-channel vulnerabilities [27, 28, 29, 32]. While most of these enhancements may be invisible to the programmer (i.e., they are implemented in the microarchitecture), little thought is given to the evaluation of their potential as a side-channel attack surface.

In terms of recording program state, memory maintains information that is indexed using an address. When table-based cipher state is loaded in memory, accesses to these tables are revealed using address patterns. If an attacker is able to detect address patterns in the reference stream, he will be able to recover sensitive information and encryption keys through timing information. Execution that utilizes the key values storage in memory can result in power-based [11, 33, 34, 35] or an electromagnetic based [36, 37, 15, 34] side-channels as well.

1.2 Leakage from Application Control Flows

Control-flow in applications includes both conditional and unconditional branches. By discerning the pattern of taken and not-taken branches, the decoding patterns in an encryption algorithm can be identified. Given that conditional branch outcomes are frequently predicted using history-based tables [24], the contents of the table can be monitored by utilizing some of the cache-based attacks described above. Unconditional transfers utilize history-based tables as well [9], and so these transfers suffer from similar vulnerabilities.

During sequential instruction flow, pipeline stalls provide an execution signature (i.e., timing, power or electromagnetic signature). Arithmetic instructions are known side-channel attack surfaces as well, such as the divide instructions on the X86 Intel Core Duo [38].

1.3 Leakage from Microarchitecture

Due to performance optimization features such as speculation, the application leaves traces in microprocessor state even if the processor abandons them in the architectural state as part of mis-speculation recovery. This enables attackers to get sensitive information such as secret keys [39] [40]. Other case studies have shown such attacks on last level caches [41, 8], branch predictors [42], power management unit [14] among others. Other attacks, such as the one demonstrated on Curve25519 cryptography [43], have shown that hardware can still be leaky through other sources such as the EM radiation from the processor chip.

Real-world covert-channel attacks such as Meltdown [40] and Spectre [39] involve undesirable interactions between processor speculative execution and memory protection, and the implications may be still emerging with newer findings about hardware vulnerabilities [44]. With speculative execution, a processor core uses heuristics to guess the next step for execution. Programs execute faster when the guess is correct. When speculation picks an incorrect direction, a core should hide any learned information from user-level software. With these newly disclosed flaws, incorrect outcomes from speculation are properly hidden from the architectural state but can be leaked through timing-based side channels. That is, a devious program can coerce the processor to speculatively access memory and then test the timing of future cache accesses to infer some bits of secret information. These side-channel attacks can be repeated many times to leak information at a rate that depends on the specifics of the attack.

The first bug, dubbed Meltdown, involves a flaw in speculation that lets a user-level program read kernel pages mapped into its page table with escalated privilege. Patches have been designed for most major operating systems. Unfortunately, depending on the frequency of system calls, these patches can have negative performance impacts. This bug is important to most current systems, as leaking the contents is unacceptable.

The second flaw, dubbed Spectre, was reported to affect commercial processors from many vendors. It is rare – and perhaps unprecedented – that a design flaw appears in multiple architectures. The flaw allows a user program to read another user program's memory by accessing side channels involving speculative branches. This class of flaws is most important to computer systems running user-level programs that are potentially hostile to each other, as with infrastructure-as-a-service cloud servers.

As a deeper understanding emerges, it becomes necessary for the computer science technical community to reflect on how to prevent future bugs like these in the cyberinfrastructure and

deliberate the tradeoffs between performance and security. There may be a growing role for formal methods and functional specification to be augmented with security features to identify the risks of micro-architectural side channels.

1.4 Leakage in Emerging Computing Platforms

Newer components in heterogeneous systems may potentially add vulnerabilities. For example, conventional power analysis attacks infer secret information from target device by observing power consumption, and they usually require physical access or proximity to the device. With higher connectivity across devices and faster compute engines, it may be easier to enable or to accelerate power-based side channels. In cloud computing environments, where multiple users can share the FPGA, security problems may arise from sharing [45].

Accelerators and GPUs have become popular targets accelerating challenging general-purpose applications [46, 47]. Over the past decade, accelerator/GPU manufacturers have recognized the potential market for moving encryption to their parallel hardware. Salman et al. [48] presented how to leverage the parallelism of OpenCL running on a Radeon GPU to accelerate bulk encryption. Yamanouchi et al. [49] described how to leverage a NVIDIA GeForce 8 Series GPU to accelerate AES encryption and decryption. The Engine-CUDA is a cryptographic engine for CUDA devices, which is part of ENGINE CUDAMRG for OpenSSL [50]. Public-key ciphers, such as Rivest-Shamir-Adleman (RSA) encryption, have also been ported in both CUDA and OpenCL [51]. There have been a number of research efforts to build a cluster for encryption and decryption that include CPUs, FPGAs and GPUs [52, 53]. As an increasing amount of applications require secure operation or manage sensitive data, there will be a growing reliance on accelerators to encrypt data efficiently.

Side and covert channels can be achieved in these GPU environments by co-locating a spy process next to the application processes, by constructing side-channels through shared hardware units, and by the ability of the adversary to filter side-channel leakage to remove noise. Reverse engineering the scheduling policy is needed for co-location of threads. GPU offers lots of parallelism, where it is possible to create multiple contention across multiple sets of resources. Removal of noise may be performed by exclusive co-location of threads, where possible. This leads to very high bandwidth rate channels [54] [55]. There have been successful timing and power attacks reported on a broad class of accelerators [30, 56]. These attacks are generally targeted at the GPU memory system. They consider cache-related attacks that have been considered in the CPU domain [57, 58, 59], as well as GPU-specific attacks such as the coalescing unit [9]. More recent work has considered computation-based attack surfaces on complex public-key ciphers, such as RSA, on GPUs [2].

In the future, given the growing number and range of accelerators being used in safety-critical applications that require secure execution, hardening GPU designs for better security will become a first-class design objective. This leads to new solutions for obfuscation and secure execution on accelerators [60, 55]. While GPUs have been studied extensively, there will be a large number of new accelerators, especially in the area of machine learning and neural networks [61].

Cloud computing models entail users sending computation and data to external entities. At the same time, users need assurance of security and privacy. It is reasonable to assume strong physical security in the cloud environment. However, with the advent of edge and fog computing where nodes close to the clients perform a big portion of computing, side and covert channels may break this assumption and present newer challenges. System components (services, plugins, schedulers, OS) may be compromised, enabling covert channel leakage. Also, interactions between these

components can be intercepted and observed, leading to side channel leakage. Possible threats to the compute node arise from outsourced (external and unknown) compute or storage components.

1.5 Leakage in IoT and Systems at Scale

Side channels in IoT systems may arise from timing information, sensor data or data traffic rates between devices that are prevalently used in our everyday lives.

Typically, the deployment phase of IoT devices spans several years. Examples include refrigerators, power outlets, attic light bulbs, elevators, window shades, and thermostats, all of which last multiple years. Also, to most users, there is no security notion associated to an everyday object such as an internet-enabled light-bulb. Recent studies have shown control flow-based attacks on medical devices such as a syringe pump [62].

Further complications may arise from the heterogeneity of devices, from low-end battery powered units to high-end CPUs and GPUs. Some of such devices may not have the power budget to implement sophisticated security protocols and software, and they may remain in service for years at a time. There is a growing need to rethink the trusted execution environments (TEE) in such settings.

System-on-Chip (SoC) security also plays a crucial role in protecting assets against sensitive information leakage. Indeed, SoC includes highly sensitive assets that must be protected (such as, mobile devices where personal, financial and other important information are stored). Security assets in SoC include on-device keys, device configuration details and Physically Unclonable Functions (PUF). Also, SoCs contain an increasing amount of sensors that may be exploited by adversaries. It is important to consider a holistic, automated solution design to address these issues associated with SoCs.

System-level leakage aspects such as social network account information and user interface accounts may result in adversely revealing user-related data [63]. Inference based attacks that exploit timing patterns of interrupts and keystrokes remain a real threat [64] [65] [66].

Past and current studies in IoT [62] [67], reveal that information through side-channels is easy to obtain, hard to defend against, difficult to detect and highly profitable. Addressing side and covert channel leakage in IoT is of utmost concern.

2. Roadmap for Effective Defense against Side and Covert Channels

To formulate effective defense strategies against side and covert channels that may physically manifest in several forms (e.g., timing, power, electromagnetic analysis), the workshop attendees explored several paths with an emphasis on formulating a cross-layer solution strategies involving multi-domain research.

The workshop participants actively discussed, and considered three primary avenues to build an effective defense framework for defending against side and covert channels:

1. Design methods that help *quantify* the amount of information leakage and subsequently use them to guide the deployment of defense strategy. Security of a system has been hard to quantify since, unlike performance and power quantification methods, it takes just a single loophole for the attacker to compromise the entire system. Techniques that verify computing systems automatically for information leakage using formal checking tools and automated methods such as machine learning can be immensely helpful.
2. Design holistic security solution frameworks for better system security rather than patch solutions for specific attacks. In other words, the solution framework should be generic enough to mitigate a class of attacks with the *flexibility* to adapt itself to defend against future attack scenarios. Often times, it is impractical for hardware to be completely redesigned (for security), or for software to fully mitigate hardware flaws that lead to side channel exploits. Therefore, techniques that leverage existing hardware and software ecosystems to the best possible extent, while making minimal modifications to them, can be more effective in defending against side and covert channels in computing systems.
3. Design hardened computing paradigms that makes it *difficult* for adversaries to exploit the system. While obfuscation and randomness have long been studied as security boosters, it is necessary for system designers to be aware of performance and power implications of such designs and make sure that they do not inadvertently create additional side channels.

A detailed summary of discussions and findings in the workshop is given below.

2.1 Design Methods for Quantification and Mitigation of Information Leakage

Understanding the leakage model is the first step in side/covert channel countermeasure synthesis. Once this is known, a formal analysis becomes possible in terms of analyzing how information leakage exploits may be constructed. Systems, that are automatically verified using formal methods, can provide a more scalable approach to building systems free of side and covert channels.

Formal methods may also synthesize information leakage-proof hardware and software modules. Such formal techniques are used to check whether critical hardware and software components are designed correctly, whether their implementations are free of runtime errors, and whether they leak sensitive information when running on real devices. Once formally verified, designers can use synthesis to generate hardware/software components from their specifications automatically, and to get more efficient, reliable, and secure solutions than a manually written code.

In addition to formal methods, adopting security features and encrypted memory can reduce the risks of data breaches. For example, random bit-streams may be introduced to sensitive computation to break the statistical dependence between the secret data and side-channel leaks [68, 38]. It is important to note that, unsuspecting hardware features may, on occasion, become targets for side and covert channels. For example, side channels created by a shared memory address bus

and other similar structures are still vulnerable to various attacks such as access pattern leakage. Improved architectural designs such as Oblivious RAM (ORAM) and its optimizations may be useful in guarding against information leakage channels [69] [70] [71].

For more practical defense solutions, classical Machine Learning techniques can detect, quantify (to some extent) and eliminate side channels [72] [73]. While deep learning techniques have started becoming popular lately, they can be expensive and heavyweight solutions in terms of implementation.

In summary, future studies should develop models to help quantify of adversary's strength that can lead to better defenses and potentially enable automation of mitigation frameworks as well.

2.2 Holistic Defense against Information Leakage

While targeted solutions against information leakage can help, an integrated solution is needed to guarantee system-wide information security. Techniques that study how to protect programs from leaking data can provide better defense against side and covert channels. Shielding techniques should protect against data dependent interactions with the system stack, including exceptions, API calls, and hardware resource utilization.

A useful direction to provide holistic defense is to take advantage of Commercial Off-The-Shelf (COTS) hardware that can maximize adoption of the proposed solutions and minimize the security costs. Also, to increase the effectiveness of this defense, solutions should consider threat factors such as program sensitivity, TCB support, and the adversary's capabilities and privileges.

Other ways to improve holistic defense could leverage existing mechanisms such as Record and Replay [74, 75]. This approach involves running the program and recording all the non-deterministic events in a log file. During a replay run, the system injects the recorded events and enforces completely deterministic execution. In case of alarms, the replay unit starts from the most recent checkpoint and performs introspection. COTS hardware, such as cache occupancy monitors and partitioning, have also been leveraged to defend against side and covert channels [76].

In general, holistic approaches need to investigate effective and economically feasible ways to boost system security and prevent information leakage. In addition, if hardware features can be built for better control and usability by trusted users (say, system administrators), it might help them to better audit the system without adversely affecting normal system operation.

2.3 Hardened Computing Platforms to Mitigate Information Leakage

Side and covert channels transcend several layers of the computing stack and frequently share some common solutions. Therefore, new computing paradigms and solutions may also help to address them. A good defense strategy would typically have the following main components (as summarized in Figure 2).

1. Attack detection using performance monitors and program instrumentation;
2. Vulnerability elimination.
    a. Resource partition, either spatially or temporally;
    b. Resource restriction, such as by disabling timers;
    c. Noise injection to obfuscate or to diversify execution;

Figure 2. Attack and Vulnerability Elimination Strategies for Side and Covert Channels [Source: Adapted from Yinqian Zhang's talk at the workshop].

To protect against side/covert channels while maintaining privacy guarantees during program execution, noise injection may be performed during program execution or directly into side channels as well [77, 78, 79]. Stronger systems-level frameworks are needed to eliminate side channels provably.

Primitives such as cryptographic engines in vulnerable hardware structures, such as DRAM, can significantly improve the security guarantees in computing systems. Recent proposals, such as Obfusmem [80], are aimed at reducing the attacks on memory. More research on obfuscation and randomization can further alleviate this problem.

At the architecture-level, current solutions include using special instructions in the ISA that stop speculative results from being available to processes. However, such solutions come with huge performance costs. Therefore, architecture-level innovations are needed to stop information leakage. Attacks that rely on branch poisoning can be defeated through mechanisms that clear the branch predictor, or via separating the predictor entries across applications more carefully. Newer cache designs that limit side channels through partition-based methods are useful as well [57, 81]. DAWG [82] uses secure cache partitioning by strictly isolating both cache hits and misses between application domains Understanding the implications of such hardened solutions, their impact on application performance, and the effectiveness can help boost commercial adoption.

In summary, hardened designs can close certain classes of side and covert channels beyond just cryptography-based solutions. Computer Industry can offer guidelines into relevant design techniques, and academia could offer innovations into developing and using these paradigms beyond just demonstrating proof of concept solutions.

3. Research Challenges: Perspectives from Computer Architecture, Hardware and Systems

Understanding, Detecting and Defending against side and covert channel leakage is indeed a grand challenge for the computing research community. This requires a sound grasp of the underlying causes and actors in the computing stack. That is, in order to build a system free from dangerous impacts of side and covert channels, it is essential to have a close collaboration between three research communities in computer design, namely computer architecture, systems and hardware.

The workshop partitioned the computing stack along the three subdomains. The respective area chairs and participants outlined the following perspectives from their research areas to introduce the notion of side and covert channels, and to define the scope of mitigation techniques against side and convert channel leakage.

3.1 Computer Systems

In computer systems, information leakage channels manifest as a multi-actor threat where it becomes necessary to understand applications, middleware and their interactions with architecture and hardware. A generalized attack schema includes two individual domains: the domain of the victim (holding the secrets), and the domain of the attacker (See Figure 3).

An information leakage channel between victim and attacker typically follows the following sequence: the code belonging to the victim or Trojan is exploited to render a `Data-tap' gadget, that lets the spy access and transmit the secret held by the victim (or Trojan) [82].

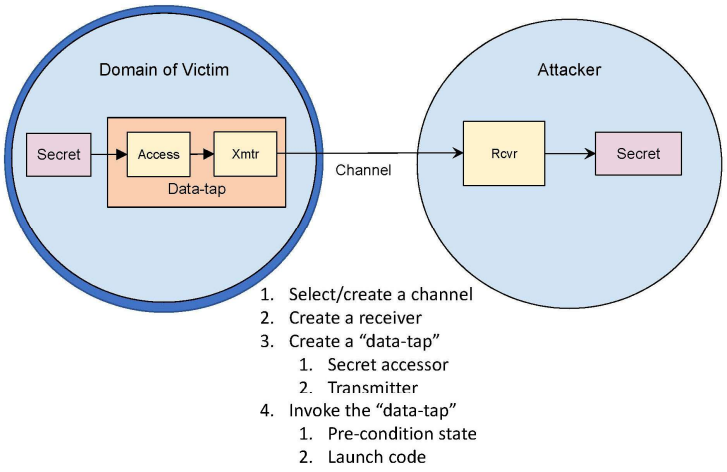

Figure 3. Attack Schema for an adversary gaining secrets from a victim

The granularity of protection to be handled by computer systems researchers can vary widely. Examples include protecting boundaries between User-kernel, User-User, Virtual Machine-Virtual Machine Manager, JavaScript thread–JavaScript Virtual Machine to name a few. Furthermore,

stealthy accesses may be performed via direct access methods, where values are loaded from a memory location, or via indirect methods, such as sequence of routines executed.

At the system level, a thorough and systematic representation of side and covert channels can benefit the design of defenses against them. In large and complex cyber-physical systems, the Trusted Computed Base (TCB) should be tailored to enable the construction of effective protection mechanisms, including a careful examination of the cross-layer computing stack with user applications, libraries, operating system, hypervisors, firmware, and hardware.

To address the threats due to side and covert channels, the grand research challenges in systems research and action items needed from the researcher community include the following:

1. Research is needed to study side- and covert channel exploits in middleware and system layer implementations along with their hardware interactions. Programming languages, compilers, formal verification, Operating Systems and networking are some immediate areas that can fuel further research. Over time, such studies can be useful to understand and spawn broader investigations into other systems areas.
2. Given the rapidly growing number of computing platforms such as cloud computing and IoT, researchers should invest their efforts into developing a taxonomy for side and covert channel attacks. Such studies can lead to more effective solutions and generalized defenses.
3. Tools and platforms are needed to drive systems-level research. Researchers and research sponsors should encourage studies on real system platforms. Often, simulation is a good, first-order study for defense, but it may not provide deep insights until real system implementation and evaluation studies are done.
4. Studies are needed on how to quantify the security threats and methods to characterize the effectiveness of system-level mitigation techniques. Research is needed to determine how much protection is enough without over-designing for security. In other words, studies need to be done on finding the tradeoff between security and performance.

3.2 Computer Architecture

In general, adversaries in both side and covert channels snoop on the activities within individual functional units and their interactions to extract secrets. These activities lead to different forms of implementations that result in successful security attacks.

1. Timing-related side-channels rely on performance variations in units such as caches and branch predictors;
2. Power-related side-channels provide information based on power consumption by micro-architecture;
3. Electro-magnetic (EM) and other acoustic channels rely on physical effects and EM radiations from the microarchitecture.

Information leakage channels are due to data-dependent behavior in applications during their execution, where adversaries may also take advantage of optimizations implemented in hardware. Therefore, measuring and detecting architecture-related side and covert channels boils down to two central ideas. First, find a microarchitecture event that occurs (or that doesn't occur) depending on a secret-data dependency. Second, find a way to determine if an event has occurred, such as by finding a performance counter that counts the events of interest.

Though high-bandwidth attacks are possible in some architecture-level side channels, closing such channels generally requires careful auditing to identify and limit the use of leaking resources. Physical attacks such as Electromagnetic analysis, in contrast, require proximity or even physical contact. They are usually very hard to close or even to limit [11, 33, 34, 35, 36, 37, 15].

To address the threats due to side and covert channels, the grand research challenges in computer architecture research and action items needed from the researcher community include the following:

1. While architecting processor designs, it is beneficial to study the vulnerability of such architectures to information leakage. Computer architecture community have successfully adopted quantitative methods for performance and power analysis. Given the emerging security challenges, it is also necessary to advance research to estimate the capability of the attackers using quantitative and theoretical models.
2. Research is needed to design co-operative solutions that target both software and hardware for effective architecture protection. Hardware solutions usually have no flexibility to adapt to emerging threats after being deployed, and software solutions can sometimes become ineffective due to very high performance overheads. Therefore, it becomes important to invest in hybrid solutions that involve both hardware and software.
3. Architecture design is usually the starting point of hardware development and is often a critical stage in evaluating the robustness of given hardware design. Therefore, it is important to develop simulators and test environments for rigorous security evaluation. Broader research community efforts can also help, such as sharing timing traces from real application runs with other researchers in the field.
4. Thorough investigations are essential hardened micro-architectural features and hardware-software protection, in addition to extending the instruction-set architecture for better auditing and resource control, and obfuscation/randomization-based solutions.
5. Design of objective methods to evaluate and compare different defense approaches are needed. Metrics, simulators, testbeds, workloads, and traces can help improve the objectivity of evaluation and verification. Toward this front, creation of several virtualized lab platforms can allow researchers to collect leakage-related data on many devices.

3.3 Computer Hardware

In hardware, side and covert channel exploits use physical effects. Emerging hardware technologies and new attack surfaces on hardware modules, especially in mobile system environments, are recent examples of vulnerabilities exposed in the hardware space. Opportunities for eliminating side and covert channels at the hardware level lie in the adoption of secure design methods and tools. In addition, cross-layer coordinated development and manufacturing can close these channels.

The attack surface for side and convert channel leakage is varied at the hardware level.

- Passive attacks involve information leakage via power consumption, timing information, electromagnetic emanations, sound, temperature or light. Side-channels leak secrets through the inherent data dependencies or mutual information. In contrast, covert channels modulate the medium with the secret so that the colluding receivers can decode it;
- Active attacks utilize fault injection, such as exploiting the power management unit (dynamic voltage frequency scaling) to overclock the processor – CLKSCREW [83] or frequent accesses to DRAM rows which causes bit flips of adjacent rows, such as Rowhammer [84].

In large-scale systems like Internet of Things (IoT) and cyber-physical systems (CPS), one must identify what side-channel vulnerabilities are realistic threats, since there are an abundant number of channels due to physical effects. Specifically, in the realm of various classes of side and covert channels, research is necessary to further understand the required defenses.

The opportunities for prevention can be enhanced through better coordination between hardware and software. For example, attacks exploiting power managers (CLKSCREW) rely on the ability to access voltage and frequency regulators through special registers, and Rowhammer depends on being able to identify victim data and hammer nearby rows which require system-level manipulation, usually in software. Prevention and detection of these attacks is feasible. For instance, in the case of an attack using CLKSCREW, secure hardware designs can address the issue at the voltage regulator level; in the case of Rowhammer, fundamental countermeasures exist at the hardware level for example by choosing higher refresh rates and better error correction codes.

To effectively protect the hardware against information leakage, individual hardware modules should consider the vulnerabilities at various levels as well. Recent studies have shown a vulnerability of the on-chip interconnect [85] [86] [7], cache structures [27, 8, 29, 87, 8, 29, 87], cache coherence protocols [32], and graphics processing units [30] [88] [55] to name a few. It becomes necessary to systematically evaluate and quantify the leakage.

Further research is also necessary to understand the impact of emerging hardware technologies and computing paradigms. This includes memory technologies, including nano-scale, three-dimensional and nonvolatile memory technologies; computing platforms with specialized hardware such as Graphics Processing Units (GPU), accelerators, Field Programmable Gate Arrays (FPGA), and heterogeneous platform with CPU, GPU and FPGA; and novel computing paradigms including homomorphic computing, quantum computing, and post-quantum cryptography.

To address the threats due to side and covert channels, the grand research challenges in hardware research and action items needed from the researcher community include the following:

1. Research is needed to build root-of-trust at the hardware level to realize secure-by-design computer systems. Also, Trusted Execution Environments in IoT-scale systems are challenging, and needs active research projects. Analog techniques may be leveraged for power or EM side-channel resistance, and for building efficient and low-power security primitives including PUF and random number generators.
2. Rich built-in structures and phenomena on CPUs, SoCs and FPGAs, including performance counters, sensors, cross-talk effects, may become non-invasive side-channel vulnerabilities. New research should investigate such new side channels and their correlation with the traditional physical effects such as power, timing, and fault. Furthermore, new mitigation techniques are needed to avoid abuse of these structures.
3. Academic researchers should closely collaborate with industry to identify unique side channel and covert channel vulnerabilities in different applications, e.g., IoT, industry control systems, medical devices and robotics, and tackle real-world security issues. Industry should invest in a hardware vulnerability database, to support knowledge and document sharing, and to contribute to security-aware design tools.
4. The research community should widen their efforts and participation through providing open-source testbeds and hardware platforms for better understanding of hardware vulnerabilities to side and covert channels.

4. Currently Available Testbeds and Future Needs

Threats and challenges that arise out of side and covert channels are active, and are growing every day. Unfortunately, there does not exist sufficient infrastructure to undertake rigorous research in this important topic. In this section, we outline some of the few well-known testbeds that are currently available [67], and stress on the need for further advanced development efforts.

4.1 Platforms and Testbeds Available for Side Channel Analysis

SAKURA (Side-channel AttacK User Reference Architecture) hardware security project extends SASEBO (Side-channel Attack Standard Evaluation BOard) [89] as an effective analysis hardware platform for side channel evaluation and analysis. In this project, various experimental hardware and software were developed to contribute to research on physical security analysis of cryptographic modules.

Flexible Open-source Board for Side-channel analysis (FOBOS) is an academic, open-source platform for testing side channel attack resistance on FPGA implementations [90]. FOBOS supports multiple FPGA devices and software to analyze differential power analysis attacks**.**

Side-Channel Analysis Resistant Framework (SCARF) is an open-source tool for testing countermeasures for side-channel and fault attacks [91]. SCARF includes a number of custom evaluation boards to test the attack resistance for smart-cards, microprocessors and FPGA as well.

DPA Workstation [92], which was developed by Cryptography Research, may be used in performing side-channel analysis including differential power or electromagnetic analysis on embedded systems. The DPA Workstation includes its own environment and proprietary software that can perform side-channel analysis on all major standard ciphers. InspectorSCA [93], developed by Riscure, can be used for side-channel and fault analysis. ChipWhisperer [94], an open-source tool-chain for embedded hardware security research, may be used for side-channel power analysis.

4.1 Future Needs

While the current tools offer functionality to test and perform side channel analysis on a limited set of scenarios (primarily embedded systems and cryptographic hardware), the workshop participants noted that more efforts are needed from the research community in developing benchmarks and testbeds for a broader range of computing systems (ranging from high-end to low-end platforms).

With the advent of customized hardware accelerators and GPGPUs in computing, it has become even more necessary to be able to understand the side and covert channel exploits on such platforms. Such newer devices bring greater challenges on account of their heterogeneous (vendor-specific) designs, and subsequently, stress the need to address any side channel leakages associated with such hardware. It is vastly important to realize that the landscape of processor and software designs have become inherently heterogeneous by design. Therefore, there exist significant challenges to rigorously test these platforms with the existing infrastructure for side channel analysis. Moreover, recent works have shown timing and electromagnetic analysis channels to be increasingly dangerous in terms of their capability to silently observe program behavior, and causing insurmountable damages to the confidentiality, integrity and privacy of critical data stored in computing systems. Hence, testbeds and benchmarks that help system designers to test their platforms for side channels are extremely critical.

5. Workforce Development and Education

For many years, security has been treated an afterthought of computer system design, and information security has been considered merely one of the possible applications for computer software. Computers were taught to execute as isolated, physically secure entities. This paradigm has been proven incorrect in every major computing domain. For example, cloud-computing faces the threat of side-channel and covert-channel leakage since adversarial processes may share the same computing platform. The Internet-of-Things faces physical adversaries, which can easily access the internals of a computer using low-cost tools. Cyber-physical systems use of cyber components that are sensitive to tamper, even by their own users (as in the case of a smart meter). This workshop used side-channel leakage and covert-channel leakage as a lightning example of how the computer system community can think about the secure design of hardware, systems, and applications in this new era. The workshop presented the potential to look across the traditional barriers of computing systems and discuss novel security-aware design methods for hardware and software, along with the assessment of vulnerabilities in such platforms. The workshop participants stressed the need for having security-aware design as an important skillset of future workforce development and as an educational need in academic courses. While the majority of participants were from academia, industry and government participants amplified this need as well. Industry has benefited increasingly from a deeper understanding of side- and covert channels in creating better hardware and system designs.

The workshop highlighted an immediate need to study security perspectives from a broader vantage point than just individual areas. The workshop was first-of-a-kind engagement between researchers studying side and covert channels from three different domains, namely computer architecture, systems and hardware. The discussions and talks during the workshop recognized the importance of a greater need for increased collaboration between the three communities. There were suggestions to further engage other research communities, such as software, communications, and more broadly, social and behavioral scientists as well.

As part of educational training needs for a better workforce to tackle important computer system security aspects (especially in the context of side and covert channels), the following suggestions were made by the workshop participants:

1. Academic institutions should collaborate and create security competitions that span multiple universities. As an example, championship competitions for graduate and undergraduate students can be created as one day workshops in premier conferences in the respective research areas. This will attract talented students to actively study the side and covert channel topics, and engage in cross-pollinating their ideas with other students. If applicable, such competitions can be held across universities for course projects between collaborating principal investigators as well.

2. Computer security-centric courses and programs must be developed in universities and incorporated into the undergraduate and graduate curricula. Such courses should educate students on how to think of secure computing by design rather than engage in patchy solutions.

3. Publication avenues must increase for security researchers that are actively engaged in studying potential for vulnerabilities in hardware/software and defense methodologies. Priority must be given to security researchers engaged in cross-disciplinary effort spanning multiple research areas.

Workshop participants and speakers

- Nael Abu-Ghazaleh (University of California Riverside)
- Divya Arora (Intel)
- Rajeev Balasubramonian (University of Utah)
- Ro Cammarota (Qualcomm)
- Susan Cheng (George Washington University)
- Srinivas Devadas (Massachusetts Institute of Technology)
- Milos Doroslovacki (George Washington University)
- Dmitry Evtyushkin (College of William and Mary)
- Yunsi Fei (Northeastern University)
- Chris Fletcher (University of Illinois at UC)
- Daniel Genkin (University of Pennsylvania)
- Nahid Ghalaty (Accenture)
- Swaroop Ghosh (Penn State)
- Yier Jin (University of Florida)
- Adwait Jog (College of William and Mary)
- David Kaeli (North Eastern University)
- David Kaplan (AMD)
- Ulya Karpuzcu (University of Minnesota)
- Deniz Karakoyunlu (Analog Devices)
- Samee Khan (National Science Foundation)
- Sandip Kundu (National Science Foundation)
- Scott List (Semiconductor Research Corporation)
- Mark Marson (Rambus)
- Vincent Mooney (Georgia Tech)
- Matt Mutka (National Science Foundation)
- Bhagirath Narahari (George Washington University)
- Richard Newell (Microsemi)
- Dmitry Ponomarev (Binghamton University)
- Lei Poo (Analog Devices)
- Milos Prvulovic (Georgia Tech)
- Gang Qu (University of Maryland)
- Dominic Rizzo (Google)
- Carlos Rozas (Intel)
- Fareena Saqib (University of North Carolina)
- Patrick Schaumont (Virginia Tech)
- Tim Sherwood (University of California at Santa Barbara)

- Weidong Shi (University of Houston)
- Aatmesh Shirvastava (North Eastern University)
- Yan Solihin (North Carolina State University)
- Berk Sunar (Worcester Polytechnic Institute)
- Edward Suh (Cornell University)
- Mark Tehranipoor (University of Florida)
- Mohit Tiwari (University of Texas at Austin)
- Josep Torellas (University of Illinois at UC)
- Akhilesh Tyagi (Iowa State University)
- Marten Van Dijk (University of Connecticut)
- Guru Prasadh Venkataramani (George Washington University)
- Chao Wang (University of South California)
- Ariton Xhafa (Texas Instruments)
- Yuval Yarom (University of Adelaide)
- Jun Yang (University of Pittsburgh)
- Qiaoyan Yu (University of New Hampshire)
- Yinqiang Zhang (Ohio State University)
- Huiyang Zhou (North Carolina State University)

References

[1] G. M. Amdahl, G. A. Blaauw and F. P. Brooks., "Architecture of the IBM System/360," *IBM Journal of Research and Development 8.2,* pp. 87-101, 1964.

[2] C. Luo, Y. Fei and D. Kaeli, "GPU acceleration of RSA is vulnerable to side-channel timing attacks," in *Proceedings of the IEEE International Conference Computer-aided Design*, 2018.

[3] T. M. John, S. K. Haider, H. Omar and M. V. Dijk., "Connecting the dots: Privacy leakage via write-access patterns to the main memory," in *IEEE Transactions on Dependable and Secure Computing*, 2017.

[4] C. W. Fletcher, "Suppressing the oblivious ram timing channel while making information leakage and program efficiency trade-offs.," in *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on.*, 2014.

[5] G. Venkataramani, J. Chen and M. Doroslovacki, "Detecting Hardware Covert Timing Channels," *IEEE Micro,* vol. 36, no. 5, pp. 17-27, 2016.

[6] F. Yao, G. Venkataramani and M. Doroslovački, "Covert Timing Channels Exploiting Non-Uniform Memory Access based Architectures.," in *Proceedings of the Greak Lakes Symposium on VLSI 2017 (GLSVLSI '17)*, 2017.

[7] Y. Wang, A. Ferraiuolo and G. E. Suh, "Timing channel protection for a shared memory controller," in *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on.*, 2014.

[8] F. Liu, Y. Yarom, Q. Ge, G. Heiser and R. B. Lee, "Last-level cache side-channel attacks are practical," in *Security and Privacy (SP)*, 2015.

[9] O. Acıiçmez, Ç. K. Koç and J.-P. Seifert., "Predicting secret keys via branch prediction.," in *Cryptographers' Track at the RSA Conference.*, 2007.

[10] M. Alagappan, J. Rajendran, M. Doroslovački and G. Venkataramani, "DFS covert channels on multi-core platforms," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2017.

[11] P. Kocher, J. Jaffe and B. Jun., "Differential power analysis," in *Annual International Cryptology Conference*, 1999.

[12] S. Mangard, "A simple power-analysis (SPA) attack on implementations of the AES key expansion.," in *International Conference on Information Security and Cryptology*, 2002.

[13] E. Brier, C. Clavier and F. Olivier., "Correlation power analysis with a leakage model," in *International workshop on cryptographic hardware and embedded systems.*, 2004.

[14] M. N. Islam and S. Kundu, "PMU-Trojan: On exploiting power management side channel for information leakage," in *23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018.

[15] M. Vuagnoux and S. Pasini., "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards," in *Proceedings of USENIX Conference on Security Symposium*, 2009.

[16] D. M. Tullsen, S. J. Eggers and H. M. Levy., "Simultaneous multithreading: Maximizing on-chip parallelism," *ACM SIGARCH Computer Architecture News.,* vol. 23, pp. 392-403, 1995.

[17] X. Zhuang, T. Zhang and S. Pande., "Using Branch Correlation to Identify Infeasible Paths for Anomaly Detection," in *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*, Orlando, Florida, 2006.

[18] M. S. Islam, M. Kuzu and M. Kantarcioglu., "Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation.," *Ndss,* vol. 20, p. 12, 2012.

[19] Y. Xu, W. Cui and M. Peinado., "Controlled-channel attacks: Deterministic side channels for untrusted operating systems," in *Security and Privacy (SP)*, Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015.

[20] D. Boneh, R. A. DeMillo and R. J. Lipton., "On the importance of checking cryptographic protocols for faults.," in *International conference on the theory and applications of cryptographic techniques.*, 1997.

[21] E. Biham and A. Shamir., "Differential fault analysis of secret key cryptosystems," in *Annual international cryptology conference*, 1997.

[22] A. Barenghi, G. M. Bertoni, L. Breveglieri, M. Pellicioli and G. Pelosi., "Fault attack on AES with single-bit induced faults.," in *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, 2010.

[23] S. A. Przybylski, Przybylski, Steven A. Cache and memory hierarchy design: a performance-directed approach, 1990.

[24] D. A. Patterson, J. L. Hennessy and D. Goldberg., Computer architecture: a quantitative approach, 1990.

[25] J. Archibald and J.-L. Baer, "Cache coherence protocols: Evaluation using a multiprocessor simulation model.," *ACM Transactions on Computer Systems (TOCS),* vol. 4, pp. 273-298, 1986.

[26] J. H. Mirza and S. W. White, "Cache prefetch and bypass using stride registers". U.S. Patent No. 5,357,618, 18 Oct 1994.

[27] D. J. Bernstein, "Cache-timing attacks on AES," 2005.

[28] D. A. Osvik, A. Shamir and E. Tromer., "Cache attacks and countermeasures: the case of AES.," in *Cryptographers' Track at the RSA Conference.*, 2006.

[29] Y. Yarom and K. Falkner., "FLUSH+ RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack.," in *USENIX Security Symposium.*, 2014.

[30] Z. H. Jiang, Y. Fei and D. Kaeli., "A complete key recovery timing attack on a GPU," in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on.*, 2016.

[31] Q. Ge, Y. Yarom, D. Cock and G. Heiser, "A survey of microarchitectural timing attacks and countermeasures on contemporary hardware.," *Journal of Cryptographic Engineering,,* vol. 8, pp. 1-27, 2018.

[32] F. Yao, M. Doroslovacki and G. Venkataramani., "Are Coherence Protocol States Vulnerable to Information Leakage?," in *High Performance Computer Architecture (HPCA), 2018 IEEE International Symposium on.*, 2018.

[33] A. Moradi, D. Oswald, C. Paar and P. Swierczynski, "Moradi, Amir, et al. "Side-channel attacks on the bitstream encryption mechanism of Altera Stratix II: facilitating black-box analysis using software reverse-engineering.," in *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays. ACM*, 2013.

[34] D. Genkin, I. Pipman and E. Tromer., "Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs.," *Journal of Cryptographic Engineering,* vol. 5.2, pp. 95-112, 2015.

[35] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh and G. Nakibly, "PowerSpy: Location Tracking Using Mobile Device Power Analysis.," in *Proceedings of 24th USENIX Conference on Security Symposium*, 2015.

[36] J.-J. Quisquater and D. Samyde., "Electromagnetic analysis (ema): Measures and counter-measures for smart cards.," in *Proceedings of Smart Card Programming and Security.*, 2001.

[37] C. H. Gebotys, S. Ho and C. C. Tiu., "EM analysis of rijndael and ECC on a wireless java-based PDA," in *International Workshop on Cryptographic Hardware and Embedded Systems.*, 2005.

[38] R. Callan, A. Zajić and M. Prvulovic., "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events.," in *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture.*, 2014.

[39] P. Kocher, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz and Y. Yarom, "Spectre Attacks: Exploiting Speculative Execution," *arXiv:1801.01203 [cs],* 3 1 2018.

[40] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, S. Mangard, P. Kocher, D. Genkin, Y. Yarom and M. Hamburg, "Meltdown," *arXiv:1801.01207 [cs],* 3 1 2018.

[41] D. Gruss, R. Spreitzer and S. Mangard, "Cache Template Attacks: Automating Attacks on Inclusive Last-level Caches," in *Proceedings of the 24th USENIX Conference on Security Symposium*, Berkeley, 2015.

[42] D. Evtyushkin, D. Ponomarev and N. Abu-Ghazaleh, "Understanding and Mitigating Covert Channels Through Branch Predictors," *ACM Transactions on Architecture and Code Optimization (TACO),* vol. 13, no. 1, 2016.

[43] D. Genkin, L. Valenta and Y. Yarom, "May the Fourth Be With You: A Microarchitectural Side Channel Attack on Several Real-World Applications of Curve25519," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2017.

[44] M. D. Hill and K. Fu, "Two Hardware Security Design Flaws Affect Billions of Computers," CCC Blog, January 2018. [Online]. Available: http://www.cccblog.org/2018/01/05/two-hardware-security-design-flaws-affect-billions-of-computers/.

[45] W. Hua, Z. Zhang and G. E. Suh, "Reverse Engineering Convolutional Neural Networks Through Side-channel Information Leaks," in *Proceedings of the 55th Annual Design Automation Conference*, New York, NY, USA, 2018.

[46] D. R. Kaeli and J. Cavazos, Eds., 11th Workshop on General Purpose Processing Using GPUs, GPGPU@PPoPP 2018, February 25, 2018, Vosendorf (Vienna), Austria, ACM, 2018.

[47] W.-m. W. Hwu, GPU Computing Gems Emerald Edition, 1st ed., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[48] S. Haq, J. Masood, A. Majeed and U. Aziz, "Bulk Encryption on AMD GPUs," AMD Developer Forum , 2011.

[49] H. Nguyen, Gpu Gems 3, First ed., Addison-Wesley Professional, 2007.

[50] "Engine CUDA: Engine CUDAmrg for OpenSSL," 2015. [Online]. Available: https://code.google.com/archive/p/engine-cuda/.

[51] S. Mahajan and M. Singh, "Performance Analysis of Efficient RSA Text Encryption Using NVIDIA CUDA-C and OpenCL," in *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*, New York, NY, USA, 2014.

[52] B. Danczul, J. Fuß, S. Gradinger, B. Greslehner, W. Kastl and F. Wex, "Cuteforce Analyzer: A Distributed Bruteforce Attack on PDF Encryption with GPUs and FPGAs," in *2013 International Conference on Availability, Reliability and Security*, 2013.

[53] E. Niewiadomska-Szynkiewicz, M. Marks, J. Jantura and M. Podbielski, "A Hybrid CPU/GPU Cluster for Encryption and Decryption of Large Amounts of Data," *Journal of Telecommunications and Information Technology,* pp. 32-39, 2012.

[54] H. Naghibijouybari, K. N. Khasawneh and N. Abu-Ghazaleh, "Constructing and Characterizing Covert Channels on GPGPUs," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, New York, NY, USA, 2017.

[55] G. Kadam, D. Zhang and A. Jog, "RCoal: Mitigating GPU Timing Attack via Subwarp-Based Randomized Coalescing Techniques," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018.

[56] C. Luo, Y. Fei, L. Zhang, A. A. Ding, P. Luo, S. Mukherjee and D. Kaeli, "Power Analysis Attack of an AES GPU Implementation," *Journal of Hardware and Systems Security,* vol. 2, pp. 69-82, 3 2018.

[57] F. Liu, H. Wu, K. Mai and R. B. Lee, "Newcache: Secure Cache Architecture Thwarting Cache Side-Channel Attacks," *IEEE Micro,* vol. 36, pp. 8-16, 9 2016.

[58] Z. H. Jiang and Y. Fei, "A Novel Cache Bank Timing Attack," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017.

[59] Y. Yarom, D. Genkin and N. Heninger, "CacheBleed: A Timing Attack on OpenSSL Constant-Time RSA," *Journal of Cryptographic Engineering,* vol. 7, pp. 99-112, 6 2017.

[60] L. Domnitser, A. Jaleel, J. Loew, N. Abu-Ghazaleh and D. Ponomarev, "Non-Monopolizable Caches: Low-Complexity Mitigation of Cache Side Channel Attacks," *ACM Trans. Archit. Code Optim.,* vol. 8, pp. 35:1--35:21, 1 2012.

[61] Y. You, A. Buluç and J. Demmel, "Scaling Deep Learning on GPU and Knights Landing Clusters," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, New York, NY, USA, 2017.

[62] Y. Park, Y. Son, H. Shin, D. Kim and Y. Kim, "This Ain't Your Dose: Sensor Spoofing Attack on Medical Infusion Pump," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, Austin, 2016.

[63] W. M. S. Stout and V. E. Urias, "Challenges to securing the Internet of Things," in *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, 2016.

[64] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan and N. Feamster, "Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic," *arXiv:1708.05044 [cs],* 16 8 2017.

[65] L. E. Olson, S. Sethumadhavan and M. D. Hill, "Security Implications of Third-Party Accelerators," *IEEE Comput. Archit. Lett.,* vol. 15, pp. 50-53, 1 2016.

[66] K. Zhang and X. Wang, "Peeping Tom in the Neighborhood: Keystroke Eavesdropping on Multi-user Systems," in *Proceedings of the 18th Conference on USENIX Security Symposium*, Berkeley, 2009.

[67] M. Moukarzel, T. Eisenbarth and B. Sunar, "uLeech: A side-channel evaluation platform for IoT," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017.

[68] J. Demme, R. Martin, A. Waksman and S. Sethumadhavan, "Side-channel Vulnerability Factor: A Metric for Measuring Information Leakage," in *Proceedings of the 39th Annual International Symposium on Computer Architecture*, Washington, 2012.

[69] C. W. Fletcher, L. Ren, A. Kwon, M. v. Dijk, E. Stefanov, D. Serpanos and S. Devadas, "A Low-Latency, Low-Area Hardware Oblivious RAM Controller," in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*, 2015.

[70] R. Wang, Y. Zhang and J. Yang, "D-ORAM: Path-ORAM Delegation for Low Execution Interference on Cloud Servers with Untrusted Memory," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Vienna, Austria, 2018.

[71] A. Shafiee, R. Balasubramonian, M. Tiwari and F. Li, "Secure DIMM: Moving ORAM Primitives Closer to Memory," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Vienna, Austria, 2018.

[72] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan and S. Stolfo, "On the Feasibility of Online Malware Detection with Performance Counters," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, New York, NY, USA, 2013.

[73] M. Chiappetta, E. Savas and C. Yilmaz, "Real Time Detection of Cache-based Side-channel Attacks Using Hardware Performance Counters," *Appl. Soft Comput.,* vol. 49, no. C, pp. 1162-1174, 12 2016.

[74] Y. Shalabi, M. Yan, N. Honarmand, R. B. Lee and J. Torrellas, "Record-Replay Architecture as a General Security Framework," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018.

[75] M. Yan, Y. Shalabi and J. Torrellas, "ReplayConfusion: Detecting cache-based covert channel attacks using record and replay," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016.

[76] F. Yao, H. Fang, M. Doroslovacki and G. Venkataramani, "COTSknight: Practical Defense against Cache Timing Channel Attacks using Cache Monitoring and Partitioning Technologies," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2019.

[77] J. W. Gray, "On introducing noise into the bus-contention channel," in *Proceedings 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, 1993.

[78] W. M. Hu, "Reducing timing channels with fuzzy time," in *Proceedings. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991.

[79] H. Fang, S. S. Dayapule, F. Yao, M. Doroslovački and G. Venkataramani, "Prefetch-guard: Leveraging hardware prefetches to defend against cache timing channels," in *Proceedings of 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018.

[80] A. Awad, Y. Wang, D. Shands and Y. Solihin, "ObfusMem: A Low-Overhead Access Obfuscation for Trusted Memories," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, New York, NY, USA, 2017.

[81] M. Yan, B. Gopireddy, T. Shull and J. Torrellas, "Secure Hierarchy-Aware Cache Replacement Policy (SHARP): Defending Against Cache-Based Side Channel Atacks," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, New York, NY, USA, 2017.

[82] V. Kiriansky, I. Lebedev, S. Amarasinghe, S. Devadas and J. Emer, "DAWG: A Defense Against Cache Timing Attacks in Speculative Execution Processors," *Cryptology ePrint Archive, Report 2018/418,* 2018.

[83] A. Tang, S. Sethumadhavan and S. Stolfo, "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management," in *26th USENIX Security Symposium (USENIX Security 17)*, Vancouver, 2017.

[84] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai and O. Mutlu, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, Piscataway, 2014.

[85] H. M. G. Wassel, Y. Gao, J. K. Oberg, T. Huffmire, R. Kastner, F. T. hong and T. Sherwood, "SurfNoC: A Low Latency and Provably Non-interfering Approach to Secure Networks-on-chip," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, Tel-Aviv, Israel, 2013.

[86] Z. Wu, Z. Xu and H. Wang, "Whispers in the Hyper-space: High-speed Covert Channel Attacks in the Cloud," in *Proceedings of the 21st USENIX Conference on Security Symposium*, Berkeley, 2012.

[87] J. Chen and G. Venkataramani, "CC-Hunter: Uncovering Covert Timing Channels on Shared Processor Hardware," in *Proceedings of the 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 2014.

[88] Z. H. Jiang, Y. Fei and D. Kaeli, "A Novel Side-Channel Timing Attack on GPUs," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*, New York, NY, USA, 2017.

[89] TROCHE-company, "Side-channel Attack Standard Evaluation Board (SASEBO)," [Online]. Available: http://satoh.cs.uec.ac.jp/SAKURA/index.html.

[90] A. Abdulgadir, W. Diehl, R. Velegalati and J. Kaps, "Flexible, Opensource workBench fOr Side-channel analysis (FOBOS)," in *IEEE Hardware Oriented Security and Trust*, 2018.

[91] J. Kim, K. Oh, D. Choi and a. H. Kim, "Scarf: profile-based side channel analysis resistant framework," in *International Conference on Security and Management (SAM)*, 2012.

[92] Rambus, " DPA workstation analysis platform," [Online]. Available: https://www.rambus.com/security/dpa-countermeasures/ dpa-workstation-platform/.

[93] Riscure, "Inspector SCA," [Online]. Available: https://www.riscure.com/ security-tools/inspector-sca/.

[94] C. O'Flynn and Z. D. Chen, "ChipWhisperer: An open-source platform for hardware embedded security research," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2014.

[95] M. Alwani, H. Chen, M. Ferdman and P. Milder, "Fused-layer CNN Accelerators," in *The 49th Annual IEEE/ACM International Symposium on Microarchitecture*, Piscataway, 2016.

[96] M. Moukarzel, T. Eisenbarth and B. Sunar, "μLeech: A side-channel evaluation platform for IoT," in *IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017.

[97] T. Schneider, A. Moradi, F.-X. Standaert and T. Güneysu, "Bridging the Gap: Advanced Tools for Side-Channel Leakage Estimation Beyond Gaussian Templates and Histograms," in *International Conference on Selected Areas in Cryptography*, 2016.