Covert-channels in FPGA-enabled SmartSSDs

THEODOROS TROCHATOS, Yale University, USA
ANTHONY ETIM, Yale University, USA
JAKUB SZEFER, Yale University, USA

Cloud computing providers today offer access to a variety of devices, which users can rent and access remotely in a shared setting. Among these devices are SmartSSDs, which are solid-state disks (SSD) augmented with an FPGA, enabling users to instantiate custom circuits within the FPGA, including potentially malicious circuits for power and temperature measurement. Normally, cloud users have no remote access to power and temperature data, but with SmartSSDs they could abuse the FPGA component to instantiate circuits to learn this information. Additionally, custom power waster circuits can be instantiated within the FPGA. This paper shows for the first time that by leveraging ring oscillator sensors and power wasters, numerous covert-channels in FPGA-enabled SmartSSDs could be used to transmit information. This work presents two channels in single-tenant setting (SmartSSD is used by one user at a time) and two channels in multi-tenant setting (FPGA and SSD inside SmartSSD are shared by different users). The presented covert channels can reach close to 100% accuracy. Meanwhile, bandwidth of the channels can be easily scaled by cloud users renting more SmartSSDs as the bandwidth of the covert channels is proportional to the number of SmartSSD used.

CCS Concepts: • **Security and privacy → Hardware attacks and countermeasures**; **Embedded systems security**; • **Hardware → Reconfigurable logic and FPGAs**.

Additional Key Words and Phrases: Ring Oscillators, SmartSSDs, Information Leakage, Covert Channels

ACM Reference Format:
Theodoros Trochatos, Anthony Etim, and Jakub Szefer. 2025. Covert-channels in FPGA-enabled SmartSSDs. *ACM Trans. Reconfig. Technol. Syst.* 37, 4, Article 111 (August 2025), 23 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

FPGAs are increasingly being used in cloud computing infrastructures to allow users to accelerate their computation through use of custom hardware logic. Many public cloud providers now provide FPGA-enabled services, including Amazon's EC2 F1 instances [1] or various FPGA-enabled virtual machine instances from VMAccel [6]. Outside general-purpose public cloud providers, there is also Microsoft's Azure, which uses Catapult [3] servers with FPGAs for artificial intelligence computation acceleration. For academic-only use, FPGA-enabled servers are available for remote sharing by academics through deployments such as at the Texas Advanced Computing Center (TACC) [5], for example.

The business model of cloud computing focuses on temporal sharing of the hardware between users. When one user is not using the hardware, it can be assigned to other users. Cloud providers such as Amazon now charge by the minute or even by the second for certain virtual machine instance types [2]. In addition to temporal sharing, there is also the possibility of spatial sharing. A particular piece of hardware (such as CPU or FPGA)

Authors' addresses: Theodoros Trochatos, Yale University, New Haven, CT, USA, theodoros.trochatos@yale.edu; Anthony Etim, Yale University, New Haven, CT, USA, anthony.etim@yale.edu; Jakub Szefer, Yale University, New Haven, CT, USA, jakub.szefer@yale.edu.

1936-7406/2025/8-ART111 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

can be assigned to multiple users at the same time. This is common for CPUs, but also has been explored for cloud-based FPGAs [27].

Cloud computing has originally focused on using CPUs. Later GPUs were added and FPGAs that nowadays can be rented for remote access from the cloud providers. Until recently, the main FPGA-enabled devices available from the various cloud providers were FPGA accelerator cards, consisting of an FPGA chip and a few dedicated DRAM modules on each FPGA accelerator card. Now, a new offering has been introduced: the SmartSSD [4]. SmartSSD is a solid-state disk (SSD) augmented with an FPGA. The disk and FPGA share a PCIe connection to the host computer and are enclosed in a single package. The purpose of the FPGA is to enable computation on the data stored on the disk, without use of the main host computer. Through public cloud providers such as VMAccel [6] it is now possible to rent SmartSSDs on-demand. Following the recent introduction of the SmartSSD into the cloud computing environment, this paper is the first paper to explore the security of SmartSSDs in the cloud in the context of covert channels.

When considering system security, an attacker only needs to find one vulnerability in order to attack the system or leak information. Thermal covert channels could be one way in which an attacker can leak information. If other channels or means of communication are protected, but SmartSSD-based channels are left unprotected, attackers will naturally try to exploit them. Further, understanding covert channels is important as existence of covert channels may indicate that similar side channels could be also created. This work focuses on covert channels and side channels can be explored as future work. In the setting of covert channels, the objective is to leak information in a stealthy way, were high bandwidth is not necessarily required. Usually few tens of bits, e.g. an AES encryption key, are leaked via a covert channel such as one evaluated in this work.

1.1 Thermal Channels Explored in this Work

In particular, this work mainly focuses on a new thermal covert communication channels that leverages the thermal state of the SSD and FPGA components of the SmartSSD. The thermal channel is shown to be extremely easy to establish. It can be from FPGA to SSD or from SSD to FPGA; when sender and receiver share the same SmartSSD concurrently. Or it can be between two users who gain sequential access to the same SmartSSD. In addition to the thermal channels, a channel through shared power of two separate SmartSSDs can be achieved when two users access to separate SmartSSDs concurrently on the same server.

The covert channels analyzed and discovered in this project were tested both in the lab setting and on a public cloud provider who offers SmartSSD enabled virtual machines. The evaluation considers the effects of data center cooling system, which constantly cools the servers and the disks and it cannot be controlled by the attacker as the SmartSSD disks are accessed remotely by the cloud users. Furthermore, due to the abundance of cloud computing resources, multiple SmartSSDs can be easily rented in parallel to increase the bandwidth of the covert transmission in proportion to the number of SmartSSDs used. A high bandwidth channel per SmartSSD is not necessary, as multiple SmartSSDs can be easily rented.

1.2 New Single-tenant Covert Channels

In case of SmartSSD being fully dedicated to one user at a time, there are two possible covert channels. Between users who access the same SmartSSD sequentially, and between users who access separate SmartSSDs concurrently.

1.2.1 Single-tenant, SmartSSD to SmartSSD Sequential Channel. For sequential access to the same SmartSSD, because of the thermal state retention, there can be temporal covert channel between sender and receiver who use the same SmartSSD sequentially. One user can raise the SmartSSD temperature. This work shows that without endangering the public cloud provider's SmartSSDs (as first tested on the university server), large disk activity can significantly raise the temperature of the disk and the associated FPGA. Extensive SSD activity is able to sufficiently raise the temperature of the FPGA within the same SmartSSD enclosure, and that change

can be observed up to a few minutes later, even after the SSD activity is done. Alternatively, a power waster instantiated in the FPGA can be used to consume large amounts of power and raise the FPGA temperature. The subsequent user can, even after a few minutes, measure the SmartSSD temperature by, for example, instantiating ring oscillator sensors within the FPGA component of the SmartSSD.

1.2.2 Single-tenant, Cross-SmartSSD Channel. There can also be cross-SmartSSD channel, between two SmartSSDs in parallel. In this setting, users access separate SmartSSDs within the server. One SmartSSD is the transmitter and the other SmartSSD is the receiver. The sender can use FPGA (in one SmartSSD) to instantiate power wasters, while the receiver can instantiate ring oscillator sensors (in the other SmartSSD). In particular, ring oscillators are not only sensitive to thermal, but also power changes. We observe power waster activity on one SmartSSD can be detected on the second SmartSSD by measuring ring oscillator changes. Large SSD activity on the sending SmartSSD could also be used, although this work focuses on FPGA part and thus the power wasters as the possible source of the information transmission.

1.3 New Multi-tenant Covert Channels

In the case of SmartSSD being shared between different users, e.g. where one can access the FPGA component and the other the SSD component. There are two possible covert channels: from SSD to FPGA and from FPGA to SSD.

1.3.1 Multi-tenant, SSD to FPGA Channel within SmartSSD. For SSD to FPGA channel, to transmit information, the sender can either stress the SSD (which is part of the SmartSSD) by accessing large amounts of data (to generate heat and send 1) or do nothing (to keep the temperature low and send 0). Meanwhile, the receiver can use the FPGA (which is part of the same SmartSSD) to measure the thermal changes by instantiating a ring oscillator. Because the SSD and FPGA are in the same enclosure, the thermal changes due to activity of the SSD affect the temperature of the FPGA chip. No special privileges are required since the user can freely create ring oscillator circuits to measure temperature or power of the FPGA even if the cloud provider does not provide this information.

1.3.2 Multi-tenant, FPGA to SSD Channel within SmartSSD. For FPGA to SSD channel, if the receiver has access to the SSD's thermal information, through software tools such as `nvme` utility, then he or she can observe thermal changes of the SSD (which is part of the SmartSSD). Meanwhile, the sender can instantiate power wasters, also leveraging ring oscillators, on the FPGA (which is part of the same SmartSSD). The power wasters can modulate the thermal state of the whole SmartSSD enclosure, and be detected on the SSD. If access to SSD thermal sensors is not possible, SSD performance could be used as proxy for thermal measurements, although this is left as future work. For the power wasters, no special privileges are required, as users can freely instantiate power wasting circuits in the FPGA part of SmartSSD.

1.4 Contributions

This paper makes a number of new contributions:

- We introduce the first analysis of possible covert channels in SmartSSDs in a cloud setting. We leverage ring oscillators (ROs) to create thermal and power sensors on the FPGA component for covert data receiving, and RO-based power wasters on the FPGA component for covert data transmission.
- We evaluate the thermal behavior and properties of the FPGA (and SSD) within the SmartSSD on a university server and on a public cloud provider.
- In the single-tenant setting of today, we present first covert channels between sequential and parallel SmartSSD users. The accuracy of the channels is as high as 100%.

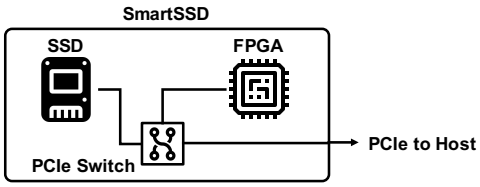


Fig. 1. Block diagram of an FPGA-enabled SmartSSD: it consists of an SSD disk and FPGA, both of which are connected via a PCIe switch to the host computer.

- In the multi-tenant setting of possible future cloud computing, we present the first covert communication channels between FPGA and SSD components within SmartSSDs, approaching 100% accuracy.

2 BACKGROUND

Cloud computing is now an established computing paradigm. However, there are constantly new devices being made available for remote access. CPUs and GPUs have been available for many years, but more recently FPGAs (Field Programmable Gate Arrays) are also available, and now SmartSSDs. One of the first public cloud providers offering FPGA-accelerated virtual machine instances to users, since around 2016, was Amazon Web Services (AWS) [7]. One of the newest public cloud provider offering different types of devices is VMAccel [6]. VMAccel specializes in providing FPGA as a Service (FaaS), where users can easily deploy existing FPGA code or develop new bitstreams in their pre-configured development environments, for example. In addition to numerous FPGAs, VMAccel enables users to access the SmartSSDs.

2.1 FPGA-enabled SmartSSDs

A block diagram of a SmartSSD is shown in Figure 1. The SmartSSD developed by Samsung [4] contains an SSD for data storage, as well as a Xilinx Kintex™ Ultrascale+ KU15P FPGA for data processing. The two components have access to the PCIe bus, which is also used to connect to the host computer. Importantly, the two devices are contained in a same package and share the PCIe and power supply (from the PCIe). By the nature of the packaging, as we show, they are also mutually affected by thermal changes - that is to say, if the SSD is heated up, this affects the temperature of the FPGA.

2.2 Cloud-based Access to SmartSSDs with FPGAs

Similar to other cloud-based computing resources, SmartSSDs are now offered as a cloud-based service, where users can get pay-as-you-go access to SmartSSDs. A typical cloud computing model is a “single-tenant” model where a user gets access to the whole device and when they are done the device is allocated to another user. In a future “multi-tenant” model, multiple users may be assigned to the same device at the same time. While not yet available in the context of SmartSSDs, multi-tenant setting could include one user accessing the SSD, while another accesses the FPGA. Multi-tenant FPGAs have been actively explored in academia [15]. In our work, we consider both single- and multi-tenant settings, and covert channels in both.

2.3 Thermal Measurements with ROs on FPGA

Ring Oscillators (ROs) are circuits which can be instantiated inside FPGAs and can be used to measure temperature or voltage changes [9]. By using ROs, malicious users can bypass security protections that may try to limit access to thermal or voltage data. Because SmartSSDs contain an FPGA, as this work shows for the first time, ROs can be instantiated inside the FPGA of the SmartSSD to measure thermal changes.

A ring oscillator thermal sensor in an FPGA [9] can be built by using an odd number of inverters which are connected in a loop. To bypass any Design Rule Checks (DRCs) imposed by cloud providers, an additional Flip-Flop or Latch can be inserted in the loop for more obfuscation [17]. In our design, we used ROs with LUTs. The RO sensor works by counting the number of oscillations of the loop, compared to a reference counter generated by a crystal oscillator. The delay through the inverters and wires of the RO depends on the temperature, while the crystal oscillator used for the reference counter is not significantly affected by temperature [42].

The RO sensors can be realized as an RTL kernel inside an HLS (High-Level Synthesis) based design, following one of the RTL kernel tutorials [1]. LUT-based ring oscillators with 3 stages were used for the sensor [17]. The directive `ALLOW_COMBINATORIAL_LOOPS = "TRUE"` was used to ensure combinatorial loops were allowed. Because of the directives, the XDC configuration files did not have to be modified. The tools did not block this type of ring oscillator, but other ring oscillators based on latches [17] or flip-flops [15] could be used if the LUT based are blocked.

2.4 Thermal Measurements of the SSD

Detailed analysis of the thermal behavior can be performed using the `nvme` utility to get the ground truth information about the temperature of the different components. We use this method to validate our result and compare to results obtained with ring oscillator thermal measurement circuits realized in the FPGA. Our work's contribution is that if access to `nvme` utility is restricted or disabled, a user can always synthesize an RO into the FPGA fabric of the SmartSSD to do very accurate measurements.

2.5 Security of Cloud-based FPGAs

To the best of our knowledge, all existing security research on cloud-based FPGAs has focused on the dedicated FPGA accelerator cards and has not considered other FPGA-enabled types of devices such as the SmartSSD. Many researchers have focused on exploring spatial side and covert channels in FPGAs. For example, researchers have explored cross-talk based channels, e.g., [46], [16], [21], [14], [20]. Only one major paper [47] has considered thermal temporal channels, similar to the last covert channel we have analyzed.

3 THREAT MODEL

We consider both single-tenant and multi-tenant setting with corresponding threat models.

3.1 Single-Tenant Threat Model

This work assumes a typical cloud-computing setting where users are allocated to hardware they pay for and when a user is done using the hardware, it is allocated to another user. The user is able to directly program the FPGA component of the SmartSSD. They can use any of the existing ideas [17] to bypass design rule checks when instantiating ring oscillator sensors or power wasters. For evaluation, we have access to the `nvme` utility to obtain ground truth information about SSD and FPGA components' temperature. For real security attacks, we assume this is blocked. We assume in this cloud-computing setting that the sender and receiver are able to be allocated to the same SmartSSD and that sender and receiver can reliably be scheduled one after the other on the same SmartSSD – for covert channel using the same SmartSSD. Since SmartSSDs contain an FPGA component, existing research on cloud-based FPGA fingerprinting can be used to identify an FPGA (and thus a SmartSSD). The fingerprints can be used by sender and receiver to establish whether they have found a common SmartSSD. For a covert channel when two SmartSSDs are used concurrently, we assume that the sender and receiver are able to be allocated on the same server (so that their rented SmartSSDs share the PCIe and power infrastructure of the server). Existing work on PCIe contention in FPGA-accelerated clouds could be combined with FPGA fingerprinting to identify which SmartSSDs share the same physical server.

[1] https://github.com/Xilinx/Vitis_Accel_Examples

Table 1. Possible covert channels using multiple measurements methods for both single and multi tenant setting. Note that nvme and xbutil only report temperature, while ROs can be used to measure both temperature and voltage changes.

Tenant Type	**Covert Channel Using** nvme **Utility**	**Scenario**	**Covert Channel Using** xbutil **Utility**	**Scenario**	**Covert Channel using ROs**	**Scenario**
Single Tenant (Sequential)	SSD to SSD FPGA to SSD	– –	SSD to FPGA FPGA to FPGA	– –	SSD to FPGA FPGA to FPGA	**Scen. 1** –
Single Tenant (Cross-SmartSSD)	SSD to SSD FPGA to SSD	– –	SSD to FPGA FPGA to FPGA	– –	SSD to FPGA FPGA to FPGA	– **Scen. 2**
Multi Tenant (Within SmartSSD)	SSD to SSD FPGA to SSD	– **Scen. 4**	SSD to FPGA FPGA to FPGA	– –	SSD to FPGA FPGA to FPGA	**Scen. 3** –

3.2 Multi-Tenant Threat Model

This work also assumes a possible future setting where the SSD and FPGA components of the SmartSSD are assigned to different users. Especially, they are separate devices that happen to share a PCIe switch within the SmartSSD enclosure. We assume some cloud providers may split up the resources, to offer SSD storage to one user, and FPGA fabric to another. As for single-tenant threat model, we assume ring oscillator sensor and power wasters can be instantiated by the sender and receiver. If nvme utility access is blocked, then FPGA ring oscillators can easily measure temperature. Meanwhile, for the covert channel with SSD as receiver, if thermal sensors are not available, SSD performance can be used as proxy for temperature. Thermal throttling and effect of temperature on SSDs is explored in literature [32].

4 COVERT CHANNEL DESIGN

In this section, we discuss the four different types of covert channels presented in this work. All possible covert channels are listed in Table 1. In our work we focus on evaluating a representative subset. Scenarios 1 to 3 each cover each of the three tenant types, one each. Scenarios 1 to 3 leverage ROs for temperature measurements. We further add scenario 4 to demonstrate the covert channel is possible when attacker has access to nvme utility. This of course is a more privileged attacker compared to scenarios 1 to 3 where no privileges are needed, only ablity to program the FPGA with (a malicious bitstream) containing the RO sensors.

4.1 Scenario 1: Single-tenant, Sequential, SSD to FPGA Channel

Figure 2 shows the design of the covert channel between two users aiming to covertly communicate information via the same SmartSSD. First, Alice starts her virtual machine (VM), obtains and heats up the SSD component of the SmartSSD by running the `Flexible IO (FIO)` SSD stress test, discussed later. Next, Alice terminates her instance. Following that, Bob starts up a new instance with access to the same SmartSSD. After the VM instance is started, the FPGA bitstream with the RO ring oscillator sensors is loaded onto the FPGA component of the SmartSSD. Finally, ring oscillator measurements are taken to learn the thermal state of the SSD. To transmit information, SSD is heated up (to transmit bit 1) or left idle (to transmit bit 0). Multiple SmartSSDs can be rented so that Alice can transmit multiple data bits (one bit per SmartSSD). Alternatively, Alice and Bob can keep alternating access to the SmartSSD to transmit one bit each time the SmartSSD access is switched.

In the evaluation, we consider that there may be extra waiting time, as it is shown in the figure, to account for different delays between Alice and Bob and how long it takes to switch between users to access the SmartSSD in

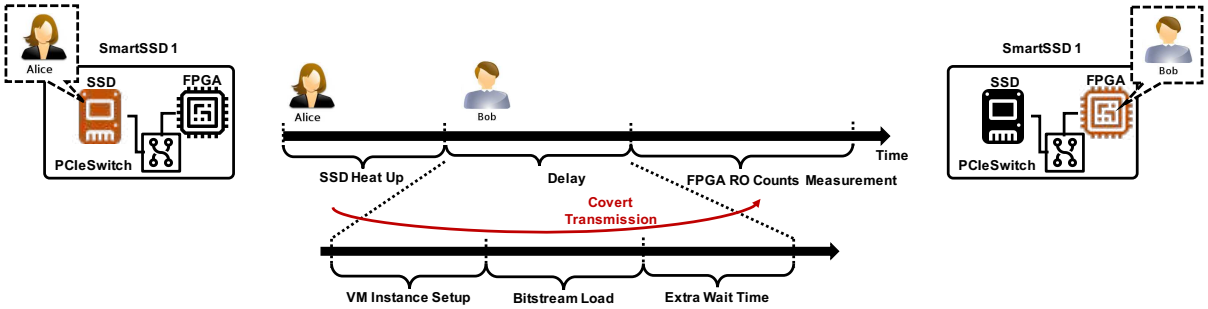


Fig. 2. Scenario 1: Single-tenant, SSD to FPGA Sequential Channel. Timeline demonstrating one of the new covert channel between Alice (sender) and Bob (receiver), who share access to the same cloud-based SmartSSD sequentially. For simplicity, we include the time required for the users to switch access to the SmartSSD in the VM Instance Setup Time. The time in the timeline is not shown to scale. The extra wait time is added for evaluation of different additional delays in time needed to switch access to the SmartSSD.

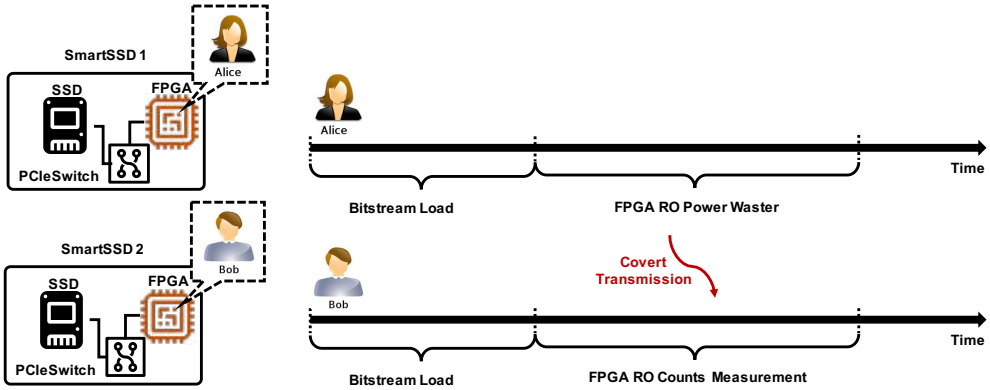


Fig. 3. Scenario 2: Single-tenant, Cross-SmartSSD FPGA to FPGA Channel. Timeline demonstrating one of the new covert channel between Alice (sender) and Bob (receiver), which uses two SmartSSDs in parallel. The time in the timeline is not shown to scale.

the cloud. We evaluated this on a real cloud provider where Alice and Bob are running different VMs (requiring VM termination and VM startup times) and the switch time is realistic.

4.2 Scenario 2: Single-tenant, Cross-SmartSSD, FPGA to FPGA Channel

Figure 3 shows the design of the covert channel between two users, aiming to covertly communicate information via separate SmartSSDs in parallel. In this scenario, Alice instantiates power wasters on the FPGA component of one of the SmartSSDs to generate thermal and power changes, which can be detectable from the ring oscillators sensors on the FPGA component of another SmartSSD that Bob instantiates concurrently on the same server. Using power wasters, 1 is transmitted by turning on power wasters, which disturbs the thermal and power states and can be observed by changes in the ring oscillator counts by Bob. A 0 is transmitted by keeping FPGA component and power wasters idle. Bob can use a simple threshold method to determine if RO counts correspond to transmission or no transmission. The baseline RO counts can be measured by Bob before the transmission starts.

The two users can agree offline on the transmission start time, to synchronize transmission, which is the approach used in this work. This way, Bob can also establish RO baseline counts before transmission. Alternately, borrowing from PCIe contention detection work [46], Alice and Bob can each take turns transmitting an agreed sequence of 1 and 0. Only when they detect the right sequence from each other, they start actual data transmission.

This covert channel requires sharing the same server and SmartSSDs on the same server. By leveraging existing research on cloud-based FPGA fingerprinting, it can be possible to identify SmartSSDs based on their FPGA fingerprints [48]. This information can be used by both the sender and receiver to establish which SmartSSD they are accessing. Additionally, combining FPGA fingerprinting with research on PCIe contention [46] allows for the identification of SmartSSDs that share the same physical server by mapping the PCIe contention. Combining both can allow Alice and Bob to find SmartSSDs that share the same server, and thus which can be used in the covert channel transmission.

4.3 Scenario 3: Multi-tenant, SSD to FPGA Channel within SmartSSD

Figure 4 shows the design of the covert channel for two users aiming to covertly communicate information in a multi-tenant setting, where Alice and Bob have access to the same SmartSSD. In this setting, Alice (sender) stresses the SSD component and Bob (receiver) observes any thermal changes happened in the FPGA component. To achieve this, Bob, instantiates ring oscillators to detect the thermal effect from the SSD, while Alice generate heat by stressing the SSD.

Similar to single-tenant scenario 1, Alice heats up the SSD component of the SmartSSD by running the `Flexible IO (FIO)` SSD stress test. Concurrently, Bob gathers the RO count measurements. Again, to transmit information, SSD is heated up (to transmit bit 1) or left idle (to transmit bit 0).

4.4 Scenario 4: Multi-tenant, FPGA to SSD Channel within SmartSSD

Figure 5 shows the design of the covert channel for two users aiming to covertly communicate information in a multi-tenant setting. In this scenario, the transmission is in the opposite direction from scenario 3: Alice (sender) stress the FPGA component of the SmartSSD and Bob (receiver) is able to observe any thermal changes of the SSD. By deploying the power wasters, Alice, can generate heat in the FPGA. The receiver, Bob, has access to the SSD's thermal information and can observe the thermal changes of the SSD.

The thermal information can be obtained from the `nvme` utility. If `nvme` utility access is blocked and the SSD component's thermal sensors are not available, SSD performance can be used as proxy for temperature. Thermal throttling and effect of temperature on SSDs is explored in literature [32]. This is approach is left as future work and (only for this scenario 4) we assume access to the SSD thermal data.

5 EXPERIMENTAL SETUP

This work evaluated SmartSSDs both on a university server and on a public cloud computing platform from which access to SmartSSDs can be rented.[2]

5.1 University Remote Server

Our university server setup consists of a Linux Ubuntu server equipped with one SmartSSD disk attached to the PCIe port. The server was located in a shared server rack, emulating a simple server room or data center setup. Xilinx Vitis tools version 2021.1 was used to compile the FPGA designs loaded onto the FPGA located inside the SmartSSD. Xilinx XRT version 2.11.634 and shell version `xilinx_u2_gen3x4_xdma_gc_base_2` were used.

[2]The name of the public cloud provider used is withheld from the paper.

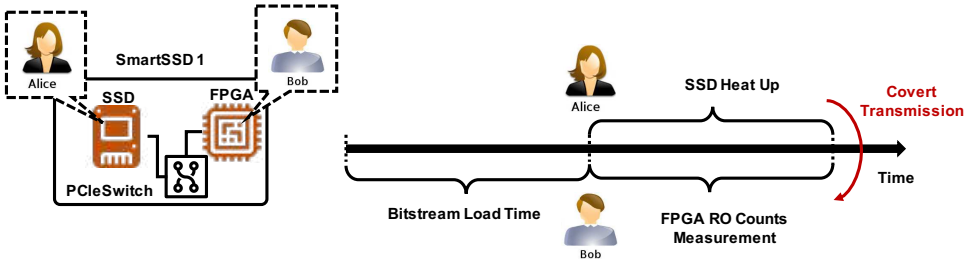


Fig. 4. Scenario 3: Multi-tenant, SSD to FPGA Channel within SmartSSD. Timeline demonstrating one of the new covert channel between Alice (sender) and Bob (receiver) for SSD to FPGA Multi-tenant channel. The time in the timeline is not shown to scale.

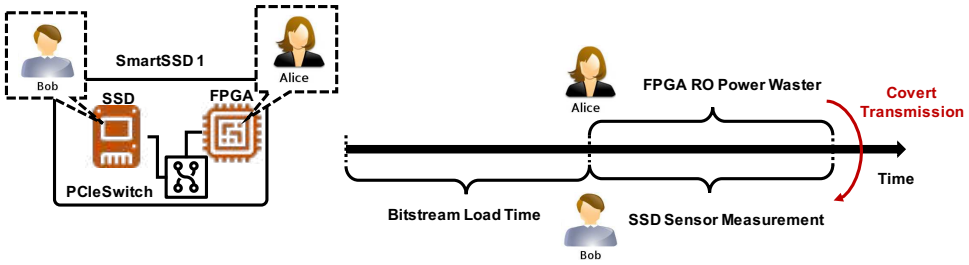


Fig. 5. Scenario 4: Multi-tenant, FPGA to SSD Channel within SmartSSD. Timeline demonstrating one of the new covert channel between Alice (sender) and Bob (receiver) for FPGA to SSD Multi-tenant channel. The time in the timeline is not shown to scale.

5.2 Public Cloud Server

For the public cloud setup, we rented access to a virtual machine enabled with either one or two SmartSSDs. The main difference between the public cloud server and remote university server is the more professional cooling infrastructure, which causes SmartSSDs to operate at lower temperatures than in our remote university server. Further, the cloud server uses virtual machines (VMs), while the local server does not. The cloud server is thus a true cloud setting where users access the SmartSSDs via VMs they rent. There is extra overhead of switching access to the SmartSSDs due to use of VMs that need to be started and terminated. Local server does not have these overheads.

5.3 Thermal Manipulation Methods Used

The covert channels use FPGA component and SSD component. Thus, methods to increase temperature of each are needed and they are different. In order to increase the FPGA component's temperature, we leveraged RO based power wasters. Power wasters are circuits designed to disrupt the supply voltage of the FPGA to induce faults and generally disrupt the normal operation of the FPGA. Power wasters have been demonstrated to bypass design rule checks imposed by FPGA cloud providers [39]. We used 5 power wasters each with 2000 ROs. More power wasters with fewer ROs, or fewer power wasters with more ROs should give similar results.

In order to increase the SSD component's temperature, we used the `Flexible IO (FIO)` tester as the stress test. The variable parameters of the stress test are: `numjobs, size, runtime, ioengine, rw, bs, iodepth,`

Table 2. Parameters of the FIO stress tests.

Parameter	Values Tested
numjobs	1, 2, 4, 8, 16, 32, 64
size (GB)	1, 2, 4, 8, 16, 32, 64, 128
runtime (secs)	60, 70, 80, 120, 240, 300
ioengine	posixaio
rw	randwrite

Parameter	Values Tested
bs (KB)	64
iodepth	16
time_based	true
end_fsync	true

time_based and end_fsync[3]. Table 2 shows the different parameter values that were used for stressing the disk. Typically, we executed the stress tests on the SmartSSD disk on the public cloud provider for 60, 120, 240 and 300 seconds. For our university server, we opted for lower runtimes to prevent disk damage, since the baseline temperature for our disk is already high, compared to the cloud provider's. We set runtime for 60, 70 and 80 seconds for the university server.

5.4 Thermal Measurements Methods Used

To measure the temperature, three methods were used:

(1) The nvme utility was used to read the SSD temperature – this may be available to an attacker with system administrator privileges, but is used by us mainly to get ground truth information about SSD temperature. We note this is used in scenario 4, but not other scenarios.
(2) The xbutil utility also reports the FPGA temperature from a single on-chip thermal diode – this may also be available to an attacker with system administrator privileges, but is used by us mainly to get ground truth information about FPGA temperature. This is used only for evaluation of the thermal changes between the FPGA and SSD component of the SmartSSD and not used in any of the scenarios.
(3) We developed an FPGA module that used the RTL kernel to instantiate an RO. This module can be used to estimate the temperature without need for access to any of the thermal diodes on the SSD or FPGA chips. This is used in all scenarios where the FPGA component is the receiver of the covert information and where RO counts are used to estimate the thermal and power changes of the FPGA (and the SmartSSD that it is contained within).

For method (1) above, we can obtain three temperature measurements, from three different sensors within the SSD component, while for method (2) we get one temperature measurement for the FPGA component. Due to (unknown) placement of the SSD sensors, some SSD sensors are closer to the disk itself, while others may be physically closer to the case of the SSD or to the FPGA chip adjacent to the SSD within the case.

6 SMARTSSD CHARACTERIZATION

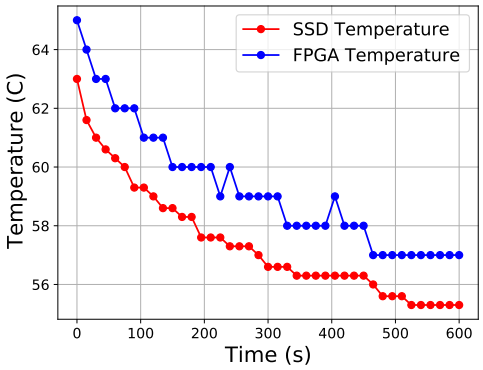
In this section, we present our experimental results and evaluation of the different behaviors of the SmartSSDs.

6.1 Finding Optimal SSD Heating Parameters

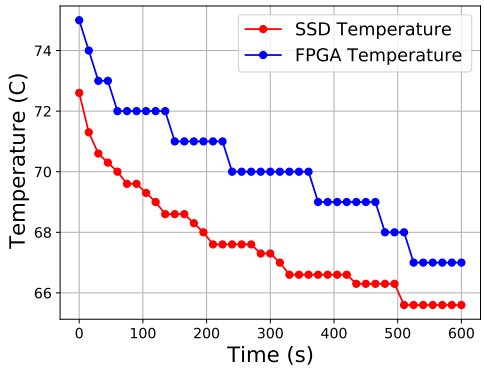
For transmission that leverages SSD heating up, there is the need to analyze the best way to heat up the SSD. We tested different values for the size and numjobs to understand which configuration of the FIO stress test increased the SSD temperature the most. The goal is to help us understand how a malicious user could best raise the temperature of the SSD.

We observed that numjobs ≈ 4 and size ≈ 8 (GB) cause the disk to increase the most in temperature when the runtime = 60 seconds. Having selected the numjobs = 4, we evaluated how the duration of the stress test affects

[3]FIO's Documentation can be found here: https://fio.readthedocs.io/en/latest/



(a) Temperature as a function of time after the stress test has stopped for the public cloud server, maximum measured time was 10 minutes. The SSD was heated for 300s.



(b) Temperature as a function of time after the stress test has stopped for the university remote server, maximum measured time was 10 minutes. The SSD was heated for 80s.

Fig. 6. SSD thermal state retention evaluation figures. The figures show that temperature decrease over time after the SSD stress test has concluded for both the public cloud and the university server.

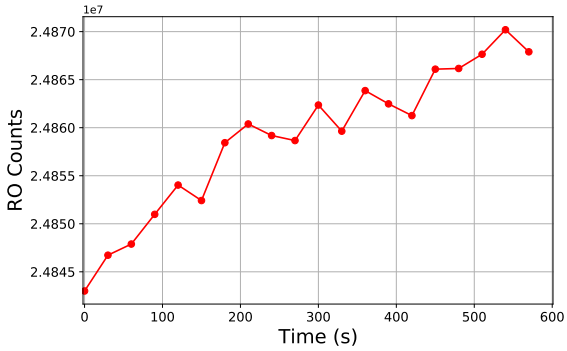
the temperature for different runtimes and sizes. Our analysis showed that we achieve the highest temperature for `size` $\approx$ 8 (GB) again for both SSD and FPGA on public cloud and university remote servers. As the `runtime` increases, both SSD and FPGA temperatures increase. Thus, adjusting runtime can be used to raise SSD to different temperatures.

It is clearly seen that the university remote server has higher baseline temperature than the public cloud server. The reason for this is that the public cloud server likely has a more capable cooling system than we have in our university remote server. We observed that university remote server achieves almost the same relative temperature increase with only about half of the runtime that public cloud server needs. Thus, the period of heating the SSD is faster on university server.

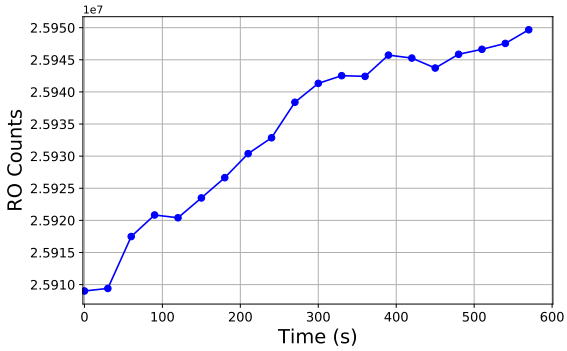
6.2 Duration of SSD Heating Effect

For covert communication where users access the same SmartSSD sequentially, scenario 1, it is necessary to know how long thermal state is maintained by the SSD and FPGA components of the SmartSSD. For different initial SmartSSD temperatures, we measured the temperature at different times after the stress test has concluded. We stressed the disk to reach some initial T_i temperature by running several `FIO` tests sequentially. After this, we performed the measurements at 15 second intervals, while the SSD temperature cools down. We used the `nvme` and `xbutil` utilities to measure the SSD and FPGA temperatures, respectively. The objective of the tests is to analyze how long the SmartSSD (that means both the SSD and FPGA inside it) can retain thermal state, which later is used for the covert communication.

Figure 6a and Figure 6b show how the SSD and FPGA temperatures change after performing the stress tests over a total time of 10 minutes on the public cloud server and the university server, respectively. It can be observed that on the public cloud server, both SSD and FPGA temperatures need at least 10 minutes to return to the baseline temperatures, if they are sufficiently heated. The same effect can be observed on the university server, despite different cooling and different baseline temperatures.



(a) University remote server RO counts after stress tests, showing RO count increase as the SSD temperature cools off back to baseline temperature. The SSD was heated for 80s.



(b) Public cloud server RO counts after stress tests, showing RO count increase as the SSD temperature cools off back to baseline temperature. The SSD was heated for 300s.

Fig. 7. Figures show RO counts after heating of the SSD, due to stress test, has concluded. RO counts increase as a function of time, as expected, while the temperature decreases for both the public cloud and the university remote server.

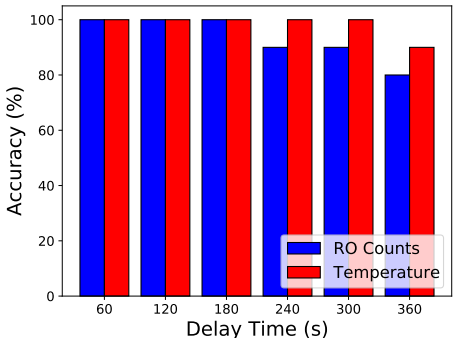
This 10 minute time-window can give potential malicious users sufficient time to perform the new thermal temporal covert communication, since the thermal state of the SmartSSD persists for some time. After SmartSSD is heated up, there is sufficient time for another user to read its thermal state.

6.3 Measuring SSD Heating with ROs

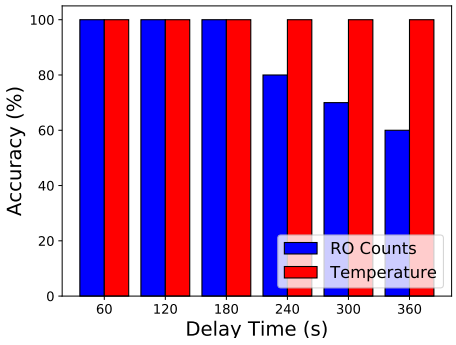
The key to the covert channels is that users can measure temperature even if access to thermal sensors is disabled. To analyze this, we confirm that RO counts correlated to thermal changes as expected. In Figures 7a and 7b we demonstrate that an attacker could measure the SmartSSD temperature using ROs, instead of doing temperature measurements using the SSD disk and FPGA utilities, which could be easily blocked by the cloud provider. To demonstrate this, we run the RO measurements on the FPGA part of the SmartSSD, while the disk cools down, after the SSD has been heated using the `FIO` stress test. We stressed the disk to reach target maximum T_{max} temperature and then we perform the measurements at 30 seconds intervals, while the disk temperature cools down.

Figure 7a shows the RO measurements for 10 minutes after the SSD was heated up using stress test on our university remote server. At each interval, 150 RO counts measurements were taken. Initially, the SSD disk was heated to a temperature of $73C$, which is almost $10C$ higher than the baseline SSD temperature of our university remote server. We expect for higher SSD temperature to observe lower RO counts. That is, as the disk cools down, RO counts should increase since the temperature drops, as it is clearly shown in Figure 7a. As can be seen from the figures, the RO counts follow the correct correlation that as temperature decreases, the RO counts increase back towards the baseline RO counts corresponding to no heating.

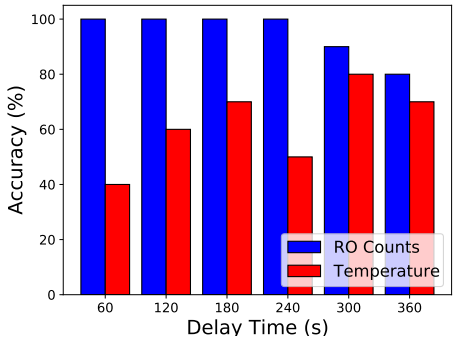
We observe similar behavior for the public cloud server, as it is shown in Figure 7b. With lower baseline temperature, the RO counts tend to be higher on the public cloud server, but also with better cooling system, more stress tests are needed to raise the temperature to $10C$ above baseline. With longer stress test, however, similar patterns in temperature are observed and the disk needs a few minutes to return to baseline temperature.



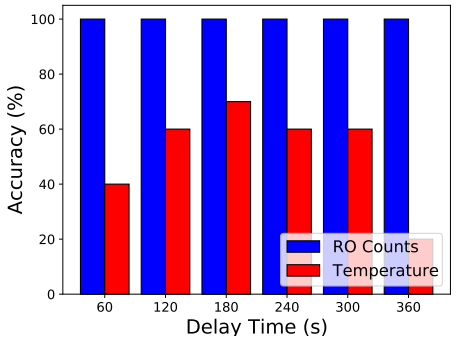
(a) University remote server covert channel transmission test accuracy with different delay times. The SSD was heated for 80s.



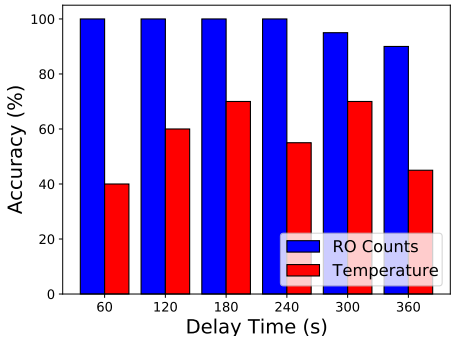
(b) Public cloud server covert channel transmission test accuracy with different delay times. The SSD was heated for 300s.



(c) Public cloud server covert channel parallel transmission test accuracy with different delay times using SSD 0. The SSD was heated for 300s.



(d) Public cloud server covert channel parallel transmission test accuracy with different delay times using SSD 1. The SSD was heated for 300s.



(e) Public cloud server covert channel transmission test accuracy combined with different delay times using 2 SmartSSDs in parallel. The SSDs were heated each for 300s.

Fig. 8. Analysis of single-tenant, sequential channel for different delays between heating and measurement. This analysis is used to estimate how much time is available between users switching access to the same SmartSSD. We also consider the case of two SSDs used in parallel, where interference between SSD affects the accuracy, on top of accuracy drop due to delay between heating and measurement.

7 COVERT CHANNEL RESULTS

In this section, we present the evaluation results of the different covert channels. In all the channels we use simple on-off keying scheme where high temperature, i.e. low RO counts, corresponds to a 1 and low temperature, i.e. high RO counts, corresponds to a 0. There could be other modulations used for data transmission as well, for example Manchester Encoding, for example. However, with Manchester Encoding there should be a decrease in the bandwidth of data transmission. We selected on-off keying as it is simple and works well in our evaluated setting, while future work can explore other types of modulation. The baseline RO counts are assumed to be obtained by the receiver before transmission. We also assume the sender and receiver are synchronized, e.g., using an external clock.

7.1 Scenario 1: Single-tenant, SSD to FPGA Sequential Channel

We first tested the single-tenant sequential covert channel between two users. During testing, we experimented with different delays in the scenario to test how many delays between users switching VMs affects the accuracy. In this case, we added more delay, on top of the time required for the second user to set up the newly obtained VM instance and load the bitstream that is always required.

Our results for the university server and the public cloud server are shown in Figure 8a and Figure 8b, respectively. It can be clearly seen that we achieve the highest accuracy within the first 4 to 5 minutes when using both the SmartSSD temperature and RO counts data. Therefore, within these 4 to 5 minutes the heat generated by one user can be observed by another user who later uses the same SmartSSD and as a result a transfer of data through a covert channel takes place. As the delay time increases, it should be noted that the accuracy drops. This is consistent as it becomes difficult to differentiate between a 1 or 0 for longer periods of time as the SmartSSD returns to the baseline temperature. It can be observed that after the first 3 minutes, the accuracy on the public cloud server is significantly lower than the accuracy on the university server. The main reason for this is the more capable cooling system that the public cloud server is equipped with. Given that, on the public cloud server it is more difficult to differentiate between a 1 and a 0 as time passes, although the accuracy is maintained in high standards even after 6 minutes after stressing. However, even though the accuracy drops for delays longer than 5 minutes, transmission is possible and error-correction codes could be used.

For comparison, the accuracy of the covert channel using the SSD thermal sensor is also shown. It can be seen that RO count based covert channel has only about 10% lower accuracy. Thus, even if there is no access to the SSD thermal sensors, the attackers can always use the RO sensor based covert communication with high accuracy. Further, Manchester encoding could be used for even better accuracy and thus our evaluation gives conservative results for the accuracy of the novel thermal temporal channel.

Having analyzed accuracy for different delays between sender heating and receiver measuring, we now evaluate the covert channel in cloud setting, where we consider actual delays between switching VMs. Figure 9 shows the covert channel transmission accuracy between two users in a public cloud server. It can be clearly seen that as we increase the extra wait time, the accuracy in both RO counts and temperature drops. As mentioned in Section 4, we can refer to Figure 2 to understand more clearly each delay time and how the process goes, as the timeline in switching between two users is shown. The instance is initially dedicated to Alice, where she heats up the SSD. Then, Alice releases the SmartSSD and Bob is able to occupy it. There is some amount of time that Bob requires to wait each time before he is able to measure and get the RO counts. In our experiments, the time required between two users to switch has been measured to be 35 seconds on average. After the instance is dedicated to Bob, Bob needs some amount of time to set up the instance. Secondly, Bob spends time to load the bitstream. In our experiments, this time has been measured to be 5 seconds on average, this is included in the delay when two users are switching VM. In the end, we tested the accuracy of the proposed covert channel by adding extra wait time after loading the bitstream and before taking the measurements. This time corresponds to

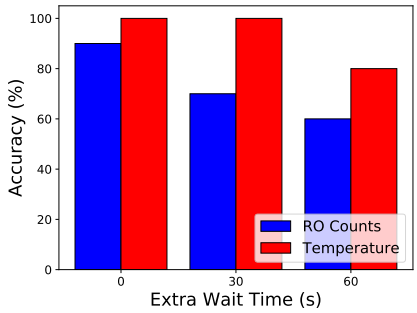


Fig. 9. Covert channel accuracy analysis for Scenario 1: Single-tenant, SSD to FPGA Sequential Channel. The extra wait time is an additional time after loading the bitstream and before taking the measurements, the extra wait time is used to evaluate how the channel behaves if the attacker is not able to immediately take the measurements.

Table 3. Scenario 2: Single-Tenant, Cross-SmartSSD, FPGA to FPGA covert channel accuracy analysis for different RO stressor sizes and bit transmission times.

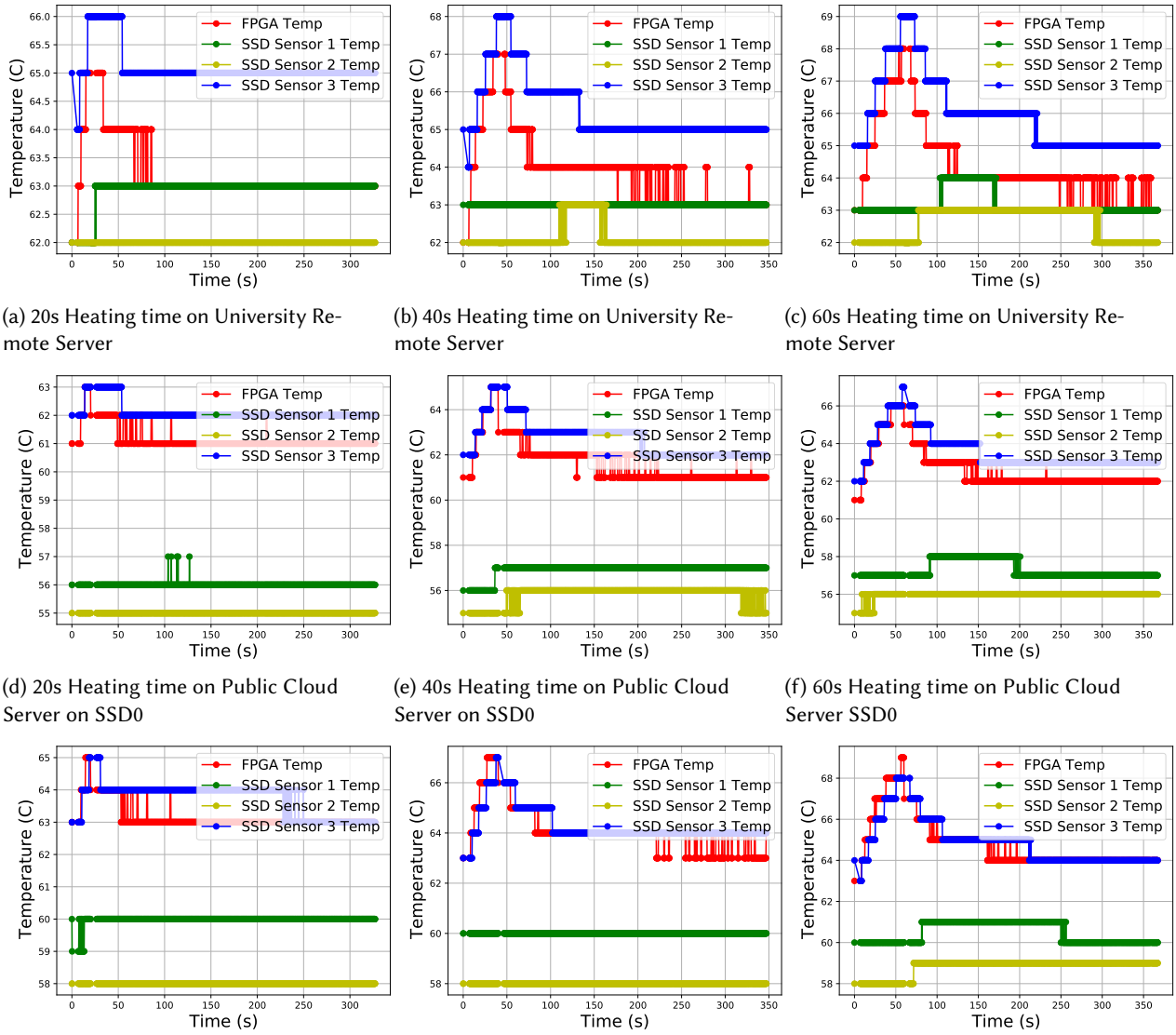
RO Stressor Size	**Accuracy (%)**		
	Bit Trans. Time (60s)	**Bit Trans. Time (30s)**	**Bit Trans. Time (15s)**
2000	53.13	53.13	53.13
4000	59.38	59.38	59.38
6000	71.88	65.63	75.00
10000	43.75	43.75	46.88

30 and 60 seconds of extra wait time, as we show in Figure 9. The 0 extra wait time is the case of no added delays and is the best case used for bandwidth evaluation. In this case, the SSD heat up time is 300s, the delay when two users are switching the VM is 100s, and the FPGA RO counts measurement time is 60s. The total time is 460s for transmission of 1 bit. The accuracy of this channel reaches 90% when using RO counts vs. 100% when using the thermal sensors from `nvme` utility.

7.2 Scenario 2: Single-tenant, Cross-SmartSSD, FPGA to FPGA Channel

We next tested single-tenant, Cross-SmartSSD channel. Here the two users are running on two separate SmartSSDs in parallel. For the sender, we tested various number of RO stressor sizes, shown in Table 3. We also tested different transmission rates, as shown in the table as well. We observed that with longer RO measurement times, the accuracy is generally less. This may indicate that too long RO measurement accumulates error. We found that transmission time of 15s per bit is the best with 6000 ROs in the stressor. Our results for the public cloud provider server for SSD 0 and SSD 1 are shown in Figure 8c and Figure 8d, respectively. Note that each SmartSSD is used to transmit one bit of data, and with two parallel transmissions, each SmartSSD is independent and can have different error rates. We also present the average error rate of the covert channel transmission across the two SmartSSDs in Figure 8e.

We note that while Figure 3 mentioned bitstream load time, once the two users have their FPGAs configured, they can keep them and transmit multiple bits and thus the transmission is limited by the time taken to run the RO stressors and corresponding time for RO sensor measurement (labeled as bit transmission time in the Table 3). In this case, time to transmit 1 bit is 15s, excluding error correction which will have to be applied given the high error rate of this covert channel.



(a) 20s Heating time on University Remote Server

(b) 40s Heating time on University Remote Server

(c) 60s Heating time on University Remote Server

(d) 20s Heating time on Public Cloud Server on SSD0

(e) 40s Heating time on Public Cloud Server on SSD0

(f) 60s Heating time on Public Cloud Server SSD0

(g) 20s Heating time on Public Cloud Server on SSD1

(h) 40s Heating time on Public Cloud Server on SSD1

(i) 60s Heating time on Public Cloud Server SSD1

Fig. 10. FPGA and SSD Temperatures for different FPGA Power Waster Times on both University and public cloud server showing Scenario 4: Multi-tenant, FPGA to SSD Channel within SmartSSD.

7.3 Scenario 3: Multi-tenant, SSD to FPGA Channel within SmartSSD

We next evaluated the scenario 3 of multi-tenant channel, between SSD and FPGA. In the setting, the transmitter has to heat up the SSD, while the receiver measures the temperature using the RO sensors on the FPGA. Although the two are working in parallel, it takes time for the SSD to heat up. This is similar to single-tenant, sequential

Table 4. Scenario 3: Multi-tenant, SSD to FPGA covert channel within SmartSSD accuracy for 300s SSD heating time. Two SSD system means two SmartSSDs are used in parallel to transmit data, while one SSD system means there is only one SmartSSD. Two SmartSSD system has twice the bandwidth as two bits, one per SmartSSD, can be covertly transmitted in parallel.

Two SSD System	
SSD	**Accuracy (%)**
1	100
2	100

One SSD System	
SSD	**Accuracy (%)**
1	100

access to the same SmartSSD, but now the two users have access to the same SmartSSD at the same time due to multi-tenancy. Thus, there is no delay due to switching of the VMs. The receiver can measure as soon as the sender finishes heating up the SSD.

Since there is no delay, the measurement can be done as soon as sender finished heating the SSD. Also, SSD does not have to be heated as much. We evaluated heating time of 60s from single-tenant, sequential access to same SmartSSD case, in which 1 bit can be transmitted every 60s with accuracy of 100%.

Further, in Table 4 we show accuracy when there is just one SmartSSD used, and also when two SmartSSDs are used in parallel to transmit two bits. We observe very limited inter-SmartSSD interference, meaning that multiple SmartSSDs can be easily used in parallel to scale up the bandwidth.

7.4 Scenario 4: Multi-tenant, FPGA to SSD Channel within SmartSSD

We next evaluated the multi-tenant covert channel, with transmission from FPGA to SSD. Here, the sender stresses the RO stressors and measures the SSD temperature using the `nvme` utility. This is only attack where the measurement is done on SSD using `nvme` utility, not using RO sensors on FPGA. We use this to show that the covert transmission can work from not just SSD to FPGA within the SmartSSD, but also from FPGA to SSD within SmartSSD. As mentioned before, without access to `nvme` utility, SSD performance could be used as proxy for estimating temperature. This is left as future work, and would likely result in lower bandwidth for the channel.

We first analyzed different heating times by running the RO stressors for 20, 40, and 60 seconds. The results of the temperature changes of the FPGA and SSD are shown in the subfigures in Figure 10. We can see clearly FPGA temperature is correlated with SSD sensor 3 temperature. This means if RO stressors heat up the FPGA, this temperature increase can be observed from SSD sensor 3. Also, the middle 40s heating time gives large temperature increase and heating vs. no heating can be detected with a simple threshold setting.

For the covert channel, in the end, we selected the heating time of 40s based on the above evaluation. When using SSD thermal sensor, and when we set the FPGA RO stressors to heat the FPGA for 40s, we can achieve 100% transmission accuracy.

In Table 5 we show accuracy when there is just one SmartSSD used, and also when two SmartSSDs are used in parallel to transmit two bits. We observe very limited inter-SmartSSD interference, meaning that multiple SmartSSDs can be easily used in parallel to scale up the bandwidth. The interference should be the same for SSD to FPGA and FPGA to SSD cases. We assume the small drop in accuracy when using two SmartSSDs in parallel may be due to external noise during the experiment.

7.5 Overall Bandwidth Analysis

In Table 6 we present the bandwidth for the four different types of covert channels. For scenario 1, Single-tenant, SmartSSD to SmartSSD Sequential Channel, we observe 0.002174 bit/s as maximum value for bandwidth with nearly 10% channel error, while for scenario 2, Single-tenant, Cross-SmartSSD Channel, we observe 0.066 bit/s as maximum value for bandwidth with nearly 25% channel error. For scenarios 3 and 4, regarding covert channel for

Table 5. Scenario 4: Multi-tenant, FPGA to SSD covert channel within SmartSSD accuracy for 40s RO stressor heating time. Two SSD system means two SmartSSDs are used in parallel to transmit data, while one SSD system means there is only one SmartSSD. Two SmartSSD system has twice the bandwidth as two bits, one per SmartSSD, can be covertly transmitted in parallel.

Two SSD System	
SSD	**Accuracy (%)**
1	96.75
2	98.38

One SSD System	
SSD	**Accuracy (%)**
1	100

Table 6. Bandwidth for the four different types of covert channels. In each case, only one SmartSSD (or a pair of SmartSSDs for cross-SmartSSD channel) is used. The bandwidth is proportional to the number of SmartSSDs used, i.e. doubling the number of SmartSSDs used, automatically doubles the bandwidth.

Scenario 1: Single-tenant Sequential (bit/s)	**Scenario 2: Single-tenant Cross-SmartSSD (bit/s)**	**Scenario 3: Multi-tenant SSD to FPGA (bit/s)**	**Scenario 4: Multi-tenant FPGA to SSD (bit/s)**
≤ 0.002174 (channel error $\sim 10\%$)	≤ 0.066 (channel error $\sim 25\%$)	≤ 0.0033 (channel error $\sim 0\%$)	≤ 0.025 (channel error $\sim 0\%$)

two users in a multi-tenant setting, our analysis shows that maximum values of 0.0033 bit/s and 0.025 bit/s for bandwidth with nearly 0% are achieved for SSD to FPGA and FPGA to SSD covert channels, respectively.

8 THERMAL FINGERPRINTING OF DATA CENTER

In addition to the covert channels, we developed a novel, remote thermal fingerprinting method for data centers. Since virtual machines can be rented for long periods of time, a malicious user could use the RO counts or thermal sensors to monitor behavior of the data center.

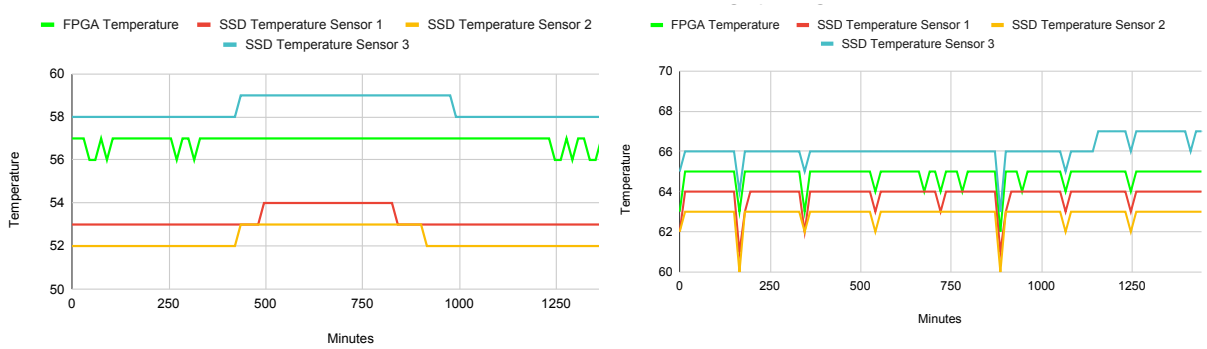
We perform this thermal fingerprinting on both the university remote server and the public server. To remotely monitor temperature, we keep the SSD and FPGA in the idle state. SSD and FPGA temperatures are measured every 15 minutes by using `nvme` and `xbutil` utilities, respectively, over a 24 hours period. Figure 11a and Figure 11b show temperature over time for a 24-hour period on public cloud server and university remote server, respectively.

It can be seen from the Figure 11b that the university remote server has rather constant temperature, which matches with our expectation as the server is only used for this project and generally idle during the experiment time. Meanwhile, for the public cloud server, we can observe an increase of temperature around 12pm PT, which lasts for about 8 hours, as shown in Figure 11a.

Interestingly, it is observable that sensor 3 has the biggest temperature changes, this could indicate the sensor 3 is located closest to the outside of the SmartSSD package. Meanwhile, the FPGA temperature is rather constant. Based on the analysis, thermal fingerprinting of the data center can work with the SSD thermal sensors, but will not work with FPGA thermal sensor or RO based sensor inside the FPGA.

9 DEFENESES

This section outlines a number of possible defenses against the new thermal channels. We suggest that a mixture of the defenses be used to ensure the best protection.



(a) Thermal fingerprinting of public cloud server for 24 hours.

(b) Thermal fingerprinting of university remote server for 24 hours.

Fig. 11. Thermal fingerprinting on both the public cloud and the university remote server.

9.1 Removal of SSD Thermal Sensor Data

The first line of defense is to remove the SSD thermal data. The `nvme` and `xbutil` utilities could be augmented not to show the temperature when used from within a guest virtual machine. While the thermal data is critical to the data center operator, it is not clear that users need access to this data. With the thermal sensor data, not even the FPGA is needed for the covert channel, so that even users unfamiliar with FPGAs could be launching the attack by using only SSD sensor data. Removing the sensor data does not prevent the attack, but is an important step.

9.2 Prevention of Ring Oscillator Sensors

Once the SSD thermal data is removed or become inaccessible to the users, the next step for the attacker is to use the FPGA. The clear solution would be to prevent users from instantiating ring oscillators. This requires two steps. First, require the final bitstream to be compiled by the cloud provider, similar to how Amazon F1 instances work [1]. Second, during compilation, any instances of ring oscillators should be caught and prevented. Unfortunately, a large amount of work has shown that this is very difficult, with researchers coming up with many different [17] and unique [15] ring oscillator designs that bypass protections.

9.3 Allow Disk to Cool Off

Since it is unlikely that ring oscillators can be fully prevented, the next best step is to remove the information contained in the thermal state. To achieve this, letting the disk stay idle for 15 or more minutes should be sufficient. Of course, during the idle time, the disk cannot be used, and the cloud provider looses potential revenue.

9.4 Other Defenses

Other possible defenses include running SSD stress test to always heat up the SSD after user is finished. The stress test will generally require less time than the cool-off time, so this is faster. However, SSD may wear out more quickly due to the heating.

10 RELATED WORK

Since the introduction of FPGA-accelerated public cloud computing about six years ago, various researchers have been exploring different security aspects of FPGAs in the cloud. The main feature differentiating such research from prior work on FPGA security is the threat model, which assumes the FPGAs are located in remote

data centers, and the potential attacks are also remote attackers without physical access to or modifications of the FPGA boards. This section summarizes recent security work that is applicable to the cloud setting, leaving traditional FPGA security topics to existing books [25] or surveys [12, 26, 36, 53].

PCIe, Peripheral Component Interconnect Express, standard provides a high-bandwidth, point-to-point, full-duplex interface for connecting peripherals within servers. Existing work has shown that PCIe switches can cause bottlenecks in multi-GPU systems [8, 11, 13, 43, 44], leading to severe stalls due to their scheduling policy [31]. In terms of PCIe contention in FPGA-accelerated cloud environments, prior work has shown that different driver implementations result in different overheads [51], and that changes in PCIe bandwidth can be used to co-locate different instances on the same server [46]. In parallel to this work, PCIe contention was used for side-channel attacks, which can recover the workload of GPUs and NICs via changes in the delay of PCIe responses [45]. Although the SmartSSD uses PCIe to communicate with the host, our newly uncovered security threat does not depend on PCIe, but on the thermal properties of the SmartSSD and the associated FPGA.

Although SmartSSD is unique in that the FPGA is bundled with the SSD disk, the setting may be somewhat similar to FPGA boards which have DRAM memory: the FPGA is bundled with its associated DRAM. Recent work has shown that direct control of the DRAM connected to the FPGA boards can be used to fingerprint them [48]. This can be combined with existing work [46] to build a map of the cloud data centers where FPGAs are used. Such fingerprinting does not by itself, however, realize a covert channel. Also, no work has been able to show that thermal changes to the DRAM can be detected by the FPGA. By contrast, our SmartSSD work shows that the packaging of the SmartSSD and FPGA in one enclosure leads to new thermal side channels.

Our work is the first work on security analysis of SmartSSDs. There is, however, relevant existing work, mostly concerning FPGAs. It is now well-known that it is possible to implement temperature sensors suitable for thermal monitoring on FPGAs using ring oscillators [9], whose frequency drifts in response to temperature variations [33, 34, 50, 52]. An array of ring oscillators can also be used as a heater or a power waster. Using a ring oscillator, a receiver FPGA could observe the ambient temperature of a data center or the FPGA. For example, existing work [47] has explored a type of temporal thermal attack where heat generated on an FPGA by one ring oscillator heater circuit can be later observed by a ring oscillator sensor circuit that is loaded onto the same FPGA. Our work meanwhile explores how to heat up SSD, the heat retention of the SSD, and how ROs can be used to measure thermal state of the SSD in the same SmartSSD package as the FPGA instantiated with the RO sensors. This type of attack is able to leak information between different users of an FPGA who are assigned to the same FPGA over time. Our work on SmartSSD follows a similar idea, but is unique to the SSD disk setting. In particular, the FPGA temperature is not raised by heating the FPGA with a ring oscillator array, but it is the disk that is heated through data access to the disk. The associated FPGA gets naturally heated up, and the receiver user can sense the temperature by reading SSD thermal sensors, or by implementing a ring oscillator sensor on the FPGA if access to SSD thermal sensors is disabled.

Our work focuses on the single-tenant cloud-based FPGA setting, where each user gets full access to the FPGA, and thus reflects the current environment offered by cloud providers. However, there is also a large body of security research in the multi-tenant context, where a single FPGA is shared by multiple, logically (and potentially physically) isolated users. For example, several researchers have shown how to recover information about the structure [49, 54] or inputs [37] of machine learning models or cause timing faults to reduce their accuracy [10, 41]. Other work in this area has shown that crosstalk due to routing wires [17] and logic elements [19] inside the FPGA chips can be used to leak static signals, while voltage drops due to dynamic signals can lead to covert-channel [18], side-channel [22, 24], and fault [38] attacks. Several works have also tried to address such issues to enable multi-tenant applications, proposing static checks [28, 30], voltage monitors [23, 35, 40], or a combination of the two [29]. Our work on SmartSSDs is orthogonal to such work, but is directly applicable to current cloud FPGA deployments. Especially, if multi-tenancy is enabled for SmartSSDs, even more threats will be possible.

For example, if one user is accessing the disk, another user with access to the FPGA could directly observe the disk activity.

11 CONCLUSION AND FUTURE WORK

This work explored security threats to FPGA-enabled SmartSSDs. A SmartSSD is a solid-state disk augmented with an FPGA. The disk and FPGA share a PCIe connection to the host computer and are enclosed in a single package. The purpose of the FPGA is to enable computation on the data stored on the disk, without use of the main host computer. Through public cloud providers, it is now possible to rent SmartSSDs on-demand. The SmartSSDs can be shared by different users, where one user accesses the disk at a time, and then the disk is allocated to another user. The sharing can enable better utilization of the disks, but also leads to new security attacks. This paper in particular showed that the heat generated by a cloud user accessing the SSD component of the SmartSSD and the resulting temperature increase, can be measured by a different cloud user accessing the FPGA component of the same SmartSSD by using the ring oscillators circuits to measure temperature. Conversely, heating up of the FPGA component by one cloud user leveraging ring oscillators can be observed by another user who can access the SSD component's thermal sensors. Both thermal states of FPGA and SSD remain elevated for a few minutes after the SSD is heated up and can be measured by a subsequent user for up to a few minutes after the heating is done. Based on the evaluation of the thermal state retention, novel thermal communication channels were demonstrated for the first time both from SSD to FPGA, and from FPGA to SSD. This work presented in particular two channels in single-tenant setting (SmartSSD is used by one user at a time) and two channels in multi-tenant setting (FPGA and SSD inside SmartSSD is shared by different users). The presented covert channels can reach close to 100% accuracy. Meanwhile, bandwidth of the channels can be easily scaled by cloud users renting more SmartSSDs as the bandwidth of the covert channels is proportional to number of SmartSSD used.

ACKNOWLEDGEMENT

This work was supported in part through NSF grants 2245344 and 1901901.

REFERENCES

[1] [n. d.]. Amazon EC2 F1 Instances. https://aws.amazon.com/ec2/instance-types/f1/.
[2] [n. d.]. AWS News Blog. https://aws.amazon.com/blogs/aws/.
[3] [n. d.]. Project Catapult Microsoft. https://www.microsoft.com/en-us/research/project/project-catapult/.
[4] [n. d.]. Samsung SmartSSD. https://www.xilinx.com/applications/data-center/computational-storage/smartssd.html.
[5] [n. d.]. TACC. https://www.tacc.utexas.edu/.
[6] [n. d.]. VMAccel. https://www.vmaccel.com/.
[7] Amazon Web Services. 2016. Developer Preview—EC2 Instances (F1) with Programmable Hardware. https://aws.amazon.com/blogs/aws/developer-preview-ec2-instances-f1-with-programmable-hardware/. Accessed: 2022-01-15.
[8] Gavin Baker and Chris Lupo. 2017. TARUC: A Topology-Aware Resource Usability and Contention Benchmark. In *ACM/SPEC International Conference on Performance Engineering (ICPE)*.
[9] Eduardo I. Boemo and Sergio López-Buedo. 1997. Thermal Monitoring on FPGAs Using Ring-Oscillators. In *Proceedings of the 7th International Workshop on Field-Programmable Logic and Applications*. Berlin, Heidelberg.
[10] Andrew Boutros, Matthew Hall, Nicolas Papernot, and Vaughn Betz. 2020. Neighbors From Hell: Voltage Attacks Against Deep Learning Accelerators on Multi-Tenant FPGAs. In *International Conference on Field-Programmable Technology (FPT)*.
[11] Anthony Danalis, Gabriel Marin, Collin McCurdy, Jeremy S. Meredith, Philip C. Roth, Kyle Spafford, Vinod Tipparaju, and Jeffrey S. Vetter. 2010. The Scalable Heterogeneous Computing (SHOC) Benchmark Suite. In *Workshop on General-Purpose Processing on Graphics Processing Units (GPGPU)*.
[12] Shijin Duan, Wenhao Wang, Yukui Luo, and Xiaolin Xu. 2021. A Survey of Recent Attacks and Mitigation on FPGA Systems. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*.
[13] Iman Faraji, Seyed H. Mirsadeghi, and Ahmad Afsahi. 2016. Topology-Aware GPU Selection on Multi-GPU Nodes. In *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*.
[14] Ilias Giechaskiel, Kasper Rasmussen, and Ken Eguro. 2022. Long-Wire Leakage: The Threat of Crosstalk. *IEEE Design & Test* (2022).

[15] Ilias Giechaskiel, Kasper Rasmussen, and Jakub Szefer. 2019. Reading Between the Dies: Cross-SLR Covert Channels on Multi-Tenant Cloud FPGAs. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*.

[16] Ilias Giechaskiel, Kasper Rasmussen, and Jakub Szefer. 2021. CAPSULe: Cross-FPGA Covert-Channel Attacks through Power Supply Unit Leakage. *IEEE Symposium on Security and Privacy* (2021).

[17] Ilias Giechaskiel, Kasper Bonne Rasmussen, and Jakub Szefer. 2019. Measuring long wire leakage with ring oscillators in cloud FPGAs. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE.

[18] Ilias Giechaskiel, Kasper B. Rasmussen, and Jakub Szefer. 2019. Reading Between the Dies: Cross-SLR Covert Channels on Multi-Tenant Cloud FPGAs. In *IEEE International Conference on Computer Design (ICCD)*.

[19] Ilias Giechaskiel and Jakub Szefer. 2020. Information Leakage from FPGA Routing and Logic Elements. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*.

[20] Ilias Giechaskiel, Shanquan Tian, and Jakub Szefer. 2021. Cross-VM Information Leaks in FPGA-Accelerated Cloud Environments. In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*.

[21] Ilias Giechaskiel, Shanquan Tian, and Jakub Szefer. 2022. Cross-VM Covert- and Side-Channel Attacks in Cloud FPGAs. *ACM Transactions on Reconfigurable Technology and Systems* (2022).

[22] Ognjen Glamočanin, Louis Coulon, Francesco Regazzoni, and Mirjana Stojilović. 2020. Are Cloud FPGAs Really Vulnerable to Power Analysis Attacks?. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

[23] Ognjen Glamočanin, Dina G. Mahmoud, Francesco Regazzoni, and Mirjana Stojilović. 2021. Shared FPGAs and the Holy Grail: Protections against Side-Channel and Fault Attacks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

[24] Mustafa Gobulukoglu, Colin Drewes, William Hunter, Ryan Kastner, and Dustin Richmond. 2021. Classifying Computations on Multi-Tenant FPGAs. In *Design Automation Conference (DAC)*.

[25] Ted Huffmire, Cynthia Irvine, Thuy D. Nguyen, Timothy Levin, Ryan Kastner, and Timothy Sherwood. 2010. *Handbook of FPGA Design Security*. Springer.

[26] Chenglu Jin, Vasudev Gohil, Ramesh Karri, and Jeyavijayan Rajendran. 2020. Security of Cloud FPGAs: A Survey.

[27] Ahmed Khawaja, Joshua Landgraf, Rohith Prakash, Michael Wei, Eric Schkufza, and Christopher J Rossbach. 2018. Sharing, Protection, and Compatibility for Reconfigurable Fabric with {AmorphOS}. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*.

[28] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. 2019. Mitigating Electrical-level Attacks towards Secure Multi-Tenant FPGAs in the Cloud. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* (2019).

[29] Tuan La, Khoa Pham, Joseph Powell, and Dirk Koch. 2021. Denial-of-Service on FPGA-based Cloud Infrastructures – Attack and Defense. *Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* (2021).

[30] Tuan Minh La, Kaspar Matas, Nikola Grunchevski, Khoa Dang Pham, and Dirk Koch. 2020. FPGADefender: Malicious Self-oscillator Scanning for Xilinx UltraScale+ FPGAs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* (2020).

[31] Chen Li, Yifan Sun, Lingling Jin, Lingjie Xu, Zheng Cao, Pengfei Fan, David Kaeli, Sheng Ma, Yang Guo, and Jun Yang. 2019. Priority-Based PCIe Scheduling for Multi-Tenant Multi-GPU Systems. *IEEE Computer Architecture Letters (LCA)* (2019).

[32] Yangming Liu, Ning Ye, Ernold Thompson, Dmitry Vaysman, In-Soo Yoon, and Hem Takiar. 2018. Fast prediction of thermal throttling design in M. 2 solid state drive. In *2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. IEEE.

[33] Sergio López-Buedo, Javier Garrido, and Eduardo Boemo. 2000. Thermal Testing on Reconfigurable Computers. *IEEE Design & Test of Computers (D&T)* (2000).

[34] Sergio López-Buedo, Javier Garrido, and Eduardo Boemo. 2002. Dynamically Inserting, Operating, and Eliminating Thermal Sensors of FPGA-based Systems. *IEEE Transactions on Components and Packaging Technologies (TCAPT)* (2002).

[35] Yukui Luo and Xiaolin Xu. 2020. A Quantitative Defense Framework against Power Attacks on Multi-tenant FPGA. In *International Conference on Computer-Aided Design (ICCAD)*.

[36] Seyedeh Sharareh Mirzargar and Mirjana Stojilović. 2019. Physical Side-Channel Attacks and Covert Communication on FPGAs: A Survey. In *International Conference on Field Programmable Logic and Applications (FPL)*.

[37] Shayan Moini, Shanquan Tian, Daniel Holcomb, Jakub Szefer, and Russell Tessier. 2021. Remote Power Side-Channel Attacks on BNN Accelerators in FPGAs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

[38] George Provelengios, Daniel Holcomb, and Russell Tessier. 2020. Power Distribution Attacks in Multi-Tenant FPGAs. *IEEE Transactions on Very Large Scale Integartion (VLSI) Systems (TVLSI)* (2020).

[39] George Provelengios, Daniel Holcomb, and Russell Tessier. 2020. Power Wasting Circuits for Cloud FPGA Attacks. In *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*.

[40] George Provelengios, Daniel Holcomb, and Russell Tessier. 2021. Mitigating Voltage Attacks in Multi-Tenant FPGAs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* (2021).

[41] Adnan Siraj Rakin, Yukui Luo, Xiaolin Xu, and Deliang Fan. 2021. Deep-Dup: An Adversarial Weight Duplication Attack Framework to Crush Deep Neural Network in Multi-Tenant FPGA. In *USENIX Security Symposium*.

[42] Behzad Razavi. 2017. The Crystal Oscillator [A Circuit for All Seasons]. *IEEE Solid-State Circuits Magazine* (2017).
[43] Dana Schaa and David Kaeli. 2009. Exploring the Multiple-GPU Design Space. In *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*.
[44] Kyle Spafford, Jeremy S. Meredith, and Jeffrey S. Vetter. 2011. Quantifying NUMA and Contention Effects in Multi-GPU Systems. In *Workshop on General-Purpose Processing on Graphics Processing Units (GPGPU)*.
[45] Mingtian Tan, Junpeng Wan, Zhe Zho, and Zhou Li. 2021. Invisible Probe: Timing Attacks with PCIe Congestion Side-channel. In *IEEE Symposium on Security and Privacy (S&P)*.
[46] Shanquan Tian, Ilias Giechaskiel, Wenjie Xiong, and Jakub Szefer. 2021. Cloud FPGA Cartography using PCIe Contention. In *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*.
[47] Shanquan Tian and Jakub Szefer. 2019. Temporal Thermal Covert Channels in Cloud FPGAs. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (Seaside, CA, USA) *(FPGA '19)*.
[48] Shanquan Tian, Wenjie Xiong, Ilias Giechaskiel, Kasper B. Rasmussen, and Jakub Szefer. 2020. Fingerprinting Cloud FPGA Infrastructures. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*.
[49] Shanquan Tian, Wenjie Xiong, Ilias Giechaskiel, and Jakub Szefer. 2021. Remote Power Attacks on the Versatile Tensor Accelerator in Multi-Tenant FPGAs. In *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*.
[50] Boyan Valtchanov, Alain Aubert, Florent Bernard, and Viktor Fischer. 2008. Modeling and Observing the Jitter in Ring Oscillators Implemented in FPGAs. In *IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*.
[51] Xiuxiu Wang, Yipei Niu, Fangming Liu, and Zichen Xu. 2020. When FPGA Meets Cloud: A First Look at Performance. *IEEE Transactions on Cloud Computing (TCC)* (2020).
[52] Chi-En Yin and Gang Qu. 2009. Temperature-aware Cooperative Ring Oscillator PUF. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*.
[53] Jiliang Zhang and Gang Qu. 2019. Recent Attacks and Defenses on FPGA-Based Systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* (2019).
[54] Yicheng Zhang, Rozhin Yasaei, Hao Chen, Zhou Li, and Mohammad Abdullah Al Faruque. 2021. Stealing Neural Network Structure Through Remote FPGA Side-Channel Analysis. *IEEE Transactions on Information Forensics and Security (TIFS)* (2021).