IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 March 2023, accepted 22 March 2023, date of publication 24 March 2023, date of current version 31 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261802

SURVEY

A Survey of Trusted Computing Solutions Using FPGAs

PAUL D. ROSERO-MONTALVO[1], **ZSOLT ISTVÁN**[2], **AND WILMAR HERNANDEZ**[3], **(Senior Member, IEEE)**
[1]Computer Science Department, IT University of Copenhagen, 2300 Copenhagen, Denmark
[2]Computer Science Department, Technical University of Darmstadt, 64289 Darmstadt, Germany
[3]Facultad de Ingenieria y Ciencias Aplicadas, Universidad de Las Américas, Quito 170513, Ecuador

Corresponding author: Wilmar Hernandez (wilmar.hernandez@udla.edu.ec)

This work was supported in part by the Novo Nordisk Fonden under Award NNF20OC0064411; in part by the LOEWE Initiative (Hesse, Germany) within the emergenCITY Center; in part by the Corporacion Ecuatoriana para el Desarrollo de la Investigacion y la Academia (CEDIA), Ecuador, under Project CEPRA XII-2018-13; and in part by the Universidad de Las Americas (UDLA), Quito, Ecuador, under Project IEA.WHP.21.02.

ABSTRACT Ensuring the security and privacy of computation and data management in the cloud and edge is an ever-important requirement. There are several working solutions today for trusted computing with general purpose processors, for instance, Intel SGX and ARM TrustZone. However, with the widespread commercial adoption of specialized hardware accelerators in the cloud and at the edge, most importantly FPGAs, two questions emerge: 1) How secure are they against threats? and 2) How could FPGAs be utilized for more efficient trusted computing? In this survey, we investigate these two questions precisely. Even though there have been numerous surveys in the past on the security of FPGAs, we believe it is timely to study the space of related work again, given the large number of data-centric applications aimed at targeting trusted execution environments that have recently appeared. Therefore, in addition to presenting an overview of state of the art, we also highlight some opportunities for FPGAs in the context of providing efficient trusted computation.

INDEX TERMS Cloud computing, security, FPGA, trusted computing.

I. INTRODUCTION

Emerging Big Data systems and analytics require significant computational resources, and general-purpose processors are reaching their limits. As a result, there is an increasing push for adopting hardware acceleration both in cloud environments and at the edge. At the same time, cloud computing introduces many security concerns to which FPGAs and other accelerators are also not immune. Trust in cloud computing is a fundamental concept that drives the relationship between a cloud provider and a user, and even though offering trusted execution using general-purpose hardware today would inevitably introduce performance bottlenecks [1], it is also an opportunity. Cloud providers that offer various trusted computing solutions can establish a stronger position and achieve a better reputation in the eyes of cloud users [2], [3], [4]. This survey focuses on FPGAs that implement a trusted execution environment in the cloud and at the edge [5], [6], [7], potentially in a way that introduces no performance degradation and trades off energy (chip area) for increased trust/security.

Today, FPGAs are becoming commonplace in the cloud, e.g., Amazon EC2 and Huawei FACS provide Xilinx's Virtex UltraScale+ FPGAs, whereas Microsoft Azure uses Intel's Arria-10 GX [8] among other devices. Their quick adoption is thanks to their unique advantages compared to traditional CPU-based or GPU-based cloud computation. However, at the same time, FPGAs also bring unique challenges in terms of security. Whereas on general-purpose processors programs run on top of an Operating System/Hypervisor, in FPGAs there is no such a substrate that could be used to protect against common attacks. On the other hand, it is true that some operating system (OS) features could be mapped to FPGAs [9]. For the moment, it is relevant to understand the types of attacks on FPGAs that run in various places of the cloud or edge architecture. Therefore, the first part of this survey focuses on attacks on FPGAs in general, and highlights mitigations. The second part of this survey focuses on that we expect that even if these mitigations will be part of an FPGA-OS or Shell, it is helpful for FPGA application

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

designers to be aware of the threats and costs of protecting against them.

The use of FPGAs for secure computing can be grouped into two categories: 1) using them as accelerators for cryptographic functions (e.g., AES encryption [10], signature validation [11], implementing homomorphic encryption schemes [12]), and 2) using them to implement both a trusted execution environment (TEE) and the application-specific logic to execute inside said environment [13], [14]. Naturally, to implement a TEE, the FPGA will have to provide features such as encryption and decryption, but the mechanisms to implement these functions most efficiently are outside the scope of this survey. Therefore, for details on the first category, we defer the reader to previous surveys in this space [6], [7]. In this paper, we focus on the second category.

To summarize, the aim of this work is, on the one hand, to provide a better understanding of the most relevant security weaknesses and their mitigations concerning the location of the FPGA. And, on the other hand, to discuss the state-of-the-art in providing trusted execution with FPGAs and the opportunities novel TEE-enabled applications bring to the use of FPGA accelerators for computing. There have been several earlier surveys on the relationship between FPGAs and security [15], and what we provide with this short, timely survey is the additional exploration of the emerging applications of the last five years in the cloud and the edge. Of course, to understand the challenges at hand, we provide an overview of some of the topics already addressed in previous surveys.

The rest of the manuscript is structured as follows. A brief background is provided in Section II. Section III summarises the principal security concerns FPGAs face, organized based on their physical location in the computing architecture. Section IV provides an overview of emerging solutions that implement trusted computing environments using FPGA. In Section V, we discuss several data management and analytics applications designed with TEEs in mind and highlight the opportunities for using FPGAs to make them more efficient. Finally, in Section VI we provide a discussion and closing thoughts.

II. BACKGROUND

This section presents a brief history of FPGAs (Section II-A) and their use in cloud computing (Section II-B). Then, Section II-C gives a summary of earlier surveys covering the intersection of FPGAs and (cloud) security. Finally, Section II-D covers the basics of providing trusted execution environments and some commonly used CPU-based solutions.

A. FIELD PROGRAMMABLE GATE ARRAYS (FPGAS)

FPGA is a programmable semiconductor device with many logic gates, input/output blocks, and programmable routing that connects these logic gates, lookup tables (LUTs), and block-on-chip memory (BRAM). The connection between these blocks is called configurable logic blocks (CLB), or floating-point digital signal processing (DSP) [16]. Therefore, it can be programmed to process different data types according to its configuration. As a result, any logical, combinational, and/or sequential function with a fixed number of inputs and outputs can be implemented using a hardware description language, such as Verilog or VHDL [17], [18]. Consequently, FPGAs avoid the long latency process by repeating fetch and decode operations for every single instruction, which can usually hold no more than two operands [19]. For this reason, their performance speed is high with low power consumption than traditional CPUs. Nowadays, FPGAs are considered a powerful device to deploy different applications using a hardware architecture described as intellectual property (IP), which means processing, memory, control, and communication that can connect multiple IPs to increase scalability. Due to FPGA' functionalities, mainly in processing large workloads efficiently, they are found in applications in the following areas: 5G open Radio Access Network (RAN) [20], prototype design [21], machine learning [22], deep learning [23], reinforcement learning [24], real-time face recognition [16], unmanned aerial vehicle control [25], multi-signal processing [26] and related to this review, trusted execution environments [27].

B. SHORT HISTORY OF FPGAS IN CLOUD COMPUTING

The commercial use of FPGA in cloud applications attracted attention in 2013 when Xilinx offered its CPU+FPGA platform for embedded devices (Zynq SoC) [28]. Later, in 2014 FPGAs were incorporated in a cloud node developed by Microsoft in a project called Catapult [29]. At the end of 2016, Amazon offered the first option to use FPGAs in the cloud. The Amazon AWS F1 had up to eight FPGAs connected to the CPU through PCIe [30]. In 2017, IBM announced cloudFPGAs [31], and Microsoft presented an FPGA-accelerated computing project named Brainwave [32]. In 2019, Xilinx introduced its new device class, Adaptive Compute Acceleration Platform (ACAP), and the first device of this class was named Versal. Versal chips integrate high-performance ARM cores with an improved FPGA fabric and newly introduced vector processors for big data and artificial intelligence applications [33]. Finally, in recent years cloud-based FPGA services have given rise to start-ups, e.g., Accelize [34]. Their business model involves designing custom accelerators for customers and supporting third-party developers that offer their accelerators to the clients of Accelize [35]. It is clear that with more and more FPGAs becoming available in the cloud and facing the users directly, ensuring both the security of the device itself and the security/trustworthiness of the applications running on it becomes of paramount importance. This is the reason we believe that it is time to carry out a survey of the state of the art.

C. FPGAS IN CLOUD COMPUTING AND SECURITY

There are several earlier studies on the state-of-the-art of cloud acceleration, trusted computing, and the security of

FPGAs [2], [3], [4], [36], [37], [38]. We direct the reader to the following surveys on the use of FPGAs for the efficient deployment of hardware accelerators in data centers and clouds: Kachris and Soudris [39], Mohammedali and Agyeman [40], and Rym et al. [18]. In addition, with the increasing interest in FPGA-accelerated applications in the cloud, some earlier surveys focused on the virtualization of FPGAs and presented different approaches to overcoming the challenges of multi-tenant accelerators [41], [42]. Additionally, surveys are explicitly related to the security concerns of using FPGAs in the cloud, focusing, for instance, on authentication and information security questions concerning the different types of attacks that can occur in cloud computing [6], [36], [43]. Furthermore, in recent years cloud infrastructure and operational challenges related to finding the right abstraction for FPGAs have led to security and trust problems and, consequently, works such as [15], [44] make vulnerability and scalability comparisons concerning trusted platform module (TPM) and Trusted Cryptography Module (TCM). Finally, TEE to protect cloud/edge on Internet of Things (IoT) applications can be found in [45].

D. TRUSTED COMPUTING OVERVIEW

Trusted computing is a special technology that allows for the secure execution of user workloads even in untrusted cloud environments. In short, it permits to verify if the behavior of the computer is the expected one. Furthermore, trusted computing implements standardized security policies and strategies by using specific technologies and architectures. In this context, Trusted Platform Module (TPM) is an international standard for a secure crypto processor implemented in hardware [44]. Therefore, a TPM is a reporting agent (witness) that provides a root of trust on which an inquisitor relies for the validation of the current state of a system [46]. The root of trust has the following parts: 1) Root of Trust for Measurement (RTM), 2) Root of Trust for Storage (RTS), and 3) Root of Trust for Reporting (RTR).

The RTM is an independent computing platform with a minimum set of instructions, which are considered to be trusted for measuring the integrity matrix of a system [47]. Commonly, the RTM will be part of the BIOS (Basic Input Output System). The RTS and RTR are based on an independent, self-sufficient, and reliable computing device with pre-defined instructions for the authentication and attestation functionality [48].

The term trusted computing became popular when Microsoft presented a white paper about a foundation of trust in the internet of things (IoT). Then, Jaeger [49] showed the relevance of implementing security tools to share software. Afterward, Windows, Sybari, Giant, and others worked intensively to present the security strategy and road map, announcing new trusted firmware, such as Client Protection and Antigen. As a result, Client Protection and Antigen became Trusted Platform Module [50].

As earlier works have shown [51], FPGAs can be used to implement all six elements required for a TPM [43]. These are as follows: Securely report boot environment, secure storage of data, Secure identification of the user and the system, support for standard security system, support for multiple users, and inexpensive production. In addition, TPM can be implemented within cloud computing for reliable boot and data encryption. For instance, Amazon has implemented the NITRO chip that is a TPM on cloud server motherboards, assisting in securing these servers. However, the security is limited to the host and not to the user VMs and their applications. Nevertheless, there are ways to virtualize the TPMs through the hypervisor and allow the VMs access to their features [52].

More general in its use than a TPM, a Trusted Execution Environment (TEE) is an environment with high levels of trust for executing arbitrary software. This environment protects the applications running within from the (potentially malicious) rest of the device [44]. As we discuss later, TEE can be provided directly by FPGAs [14] or FPGAs can benefit from the presence of a CPU-based TEE in a heterogeneous SoC [53].

An example of the latter is the *ARM Trustzone* technology. It adds a hardware security center inside the ARM processor core and separates the addresses into two "worlds", one secure and one non-secure. The secure world protects code and data, while the non-secure world runs the OS and any other untrusted software. ARM Trust zone is designed to provide hardware isolation for trusted software execution [54], and on heterogeneous devices, their functionality could be extended with specialized hardware, that is, FPGAs.

Perhaps the most widely used method of providing TEEs is through the *Intel Software Guard Extensions* (SGX). This relatively new extension of Intel architecture supports a new set of instructions and a new memory access mechanism, to save code in a container called "enclave" and exclude unprivileged software from the trusted computing base [44], [53]. However, as later explained, SGX incurs significant overheads when trusted domain code accesses I/O or other untrusted resources [1]. Furthermore, SGX is vulnerable to various memory-based side-channel attacks [53].

AMD processors offer *AMD Memory Encryption Technology*. This addition to AMD processors allows for transparent encryption and system memory protection. AMD memory encryption focuses on cloud infrastructure to avoid attacks on system software and protect against physical tampering with servers. Memory Encryption Technology introduces an AES 128 encryption engine inside the processor to encrypt and decrypt the data when the data leaves or enters the device [55]. Based on this technology, AMD provides two primary security features: secure memory encryption (SME) and secure encrypted virtualization (SEV) [56].

III. PRINCIPAL SECURITY CONCERNS OF FPGAS

With the use of FPGAs as both data processing accelerators and components in trusted computing systems becoming increasingly common, it is crucial to have a holistic overview of the main security concerns these devices face in various deployment options [57], [58]. In this survey, we adopt a global approach that maps out the principal security vulnerabilities and points to works that propose solutions for each one specifically. In addition, we differentiate the following deployment models for FPGAs: 1) Close to the local server (''local''), 2) Edge and fog computing (''edge''), 3) Cloud computing (''cloud and datacenters''), and 4) FPGA as a hardware of a service (''virtualization''). Table 1 provides a big picture of the main security concerns of the FPGA, and their possible solutions as provided by reviewed works. The rest of this section explores in more detail the rows and columns of this table.

A. GENERAL ATTACKS IN FPGAS

FPGAs are configured with bitstreams that entirely determine the functionality of the device [70]. This is, of course, an advantage for users looking to create custom compute architectures, but it also introduces a security risk. FPGAs have a **bitstream vulnerability** during the loading process, where a malicious third party could modify the contents, compromising the security of the application running on them. Several solutions have been proposed to protect bitstreams through encryption [7]. The bitstream is decrypted every time the FPGA programming is updated, using a protected decryption key [71]. Unfortunately, bitstream formats are now kept confidential by the vendors, which makes bitstream reversing a laborious, challenging task though not impossible [37]. Even though there are practical solutions for loading encrypted bitstreams on all major vendors, overall, this limits the ability of the open-source research community to explore additional protections.

Specific bitstream-related attacks include:

- Cloning: Cloning is considered the most common security vulnerability of FPGAs [36]. The FPGA configuration bitstream is obtained by hidden listening devices or from the volatile SRAM and then used to configure the FPGA chip [14].
- Hardware Trojan: The increase in demand for semiconductors shortens the time to detect faults that could generate security problems since outsourcing the development of integrated circuits is done to companies that are not verified trustworthy. Hence, logical or electrical attacks can be devised to create conflicts that cause the FPGA to malfunction [72]. Chakraborty et al. [73], demonstrated that a hardware Trojan can be directly inserted by modifying the FPGA configuration bitstream. A hardware trojan works by violating the originally designated functionalities of a circuit.
- JTAG Intercept: JTAG scan chains are effective for debugging but problematic for security, because they provide access to data and functions throughout the FPGA. Activity on a test port, such as a scan chain, may indicate an attack in progress. If a malicious user knows JTAG does not have protection, it can take complete low-level system control. They can even replace the firmware with a rogue version [74].
- Intrinsic Security: Intrinsic security is the ability of a system to protect itself and maintain data confidentiality and integrity as data is sent from the FPGA (PI) out of the device. In [75], it is mentioned that this approach is to ensure that security measures remain effective even in an environment with increased data traffic and increased resource demands. In addition, intrinsic security can also be enhanced through the use of encryption and authentication techniques. This is done to ensure that only authorized parties can access the data and to mitigate the risks of interception and manipulation of data during processing. Unfortunately, FPGA-based systems are vulnerable to attack due to their highly configurable and heterogeneous nature. What has been said above makes these systems very attractive to attackers who, with a malicious IP, can listen to the communication channel and intercept the data [38].

Another general security concern of FPGAs is that of **side channel attacks**. One relevant side channel is power. Such attacks are carried out via electric analyses that collect a series of energy traces. The attacker often uses an oscilloscope connected to the power supply of the FPGA. Then, he records data and can use statistical tools to discover information about the application running inside the FPGA [38]. For instance, there are voltage drops when performing cryptographic calculations, and thus analyzing the power traces collected with an oscilloscope may successfully retrieve the secret key [38], [69]. In SoC devices that different users could share, an FPGA-based power monitor could also be used for the same purpose and, with sufficient time and power resolutions, can be used to carry out traditional power side-channel attacks to learn secrets of other parts of the FPGA [62].

Finally, an additional threat in FPGAs exposed to untrusted parties is the readback or interference with the logic running on the device. Traditionally, IP theft has been a concern, although FPGA vendors provide new tools to avoid these security concerns, such as:

- Logic locking: A locked circuit contains additional inputs, called key inputs, which are controlled by tamper-proof memory on the chip. This locking mechanism commonly uses logic gates such as XOR / XNOR [76].
- Layout camouflaging: Specialized camouflaged cells are employed within the FPGA that are intended to be indistinguishable in various functions. This can be achieved using dummy contacts or intentional voltage variations [77].
- Split manufacturing: Many companies outsource the manufacture of integrated circuits with untrusted parties. Split manufacturing allows the designs to be protected by dividing the circuit into one part, where the transistors are located, and the routing cables in another. Thus, hiding the final manufacture of the hardware [77].

TABLE 1. Trusted computing security concerns overview and principal security criteria through representative papers reviewed. Primary Focus: ★, Secondary focus: ■.

REVIEWED WORKS	SECURITY CONCERNS					SECURITY TECHNIQUES	
	Data confidentiality [68]	Side-channel attacks [6]	Hardware trojan insertion [69]	IP thief [6]	Attestation key [37]	Summary	Common protection
Al-Asli et al. [5]	★	■				Symmetric Re-Encryption Scheme	User's data is protected by a data encryption key only accessible by trusted FPGA devices. This technique works mainly in data confidentiality, attestation key, and it has limited benefits in side-channel attacks
Xu et al. [7]	★			■	■	p Privacy preserving computation in with MapReduce programing model	
Su et al. [12]	★	■			★	Fully homomorphic encryption	
Xu et al. [59]	★				■	AES encryption	
Likhithashree et al. [60]		■	★	■		Physically unclonable function (PUF)	PUF is a widely used for hardware security through ring oscillator (RO) to secure FPGA from side-channel attacks and hardware Trojan insertion
Qin et al. [61]		■	★			Ring Oscillator Network	
Zhao et al. [62]		★	■			Ring Oscillator- PUF	
Gnad et al. [63]		★	■		■	Voltage-Based cover-channels	The covert channel avoids noise generated from other residing tenants' modules in the FPGA
Oh et al. [53]		★	★	★	■	TEE-SGX	TEE is a security solution promising strong and practical security guarantees to allow trusted execution to prevent IP theft and side-channel attacks
Zhao et al. [14]		■		★	■	Shielded Enclaves for Cloud FPGAs framework	
Zeitouni et al. [13]		■	■	★		TEE-Encrypted bitstream	
Oh et al. [27]		★		★	■	TEE specialized and multi-tenancy mechanism	
Kim et al. [8]	■	★		★		Bitstream encryption	Bitstream encryption provides a useful technique to protect against side-channels attacks and attestation key
Kan et al. [64]	■	★			★	Bi-level encryption algorithm based on RSA	
Suo et al. [65]	■	★			★	M4 cryptographic algorithm	
Vinayagamurthy et al. [66]	★					Encrypted database	Encrypted database systems provide a great method for protecting sensitive data in untrusted infrastructures.
Sun et al. [67]	★				★	Enclave storage engines for encrypted databases	

B. THREATS ON THE EDGE

Edge computing is helpful as a way of pre-processing data from sensors/IoT devices before being sent to the cloud. Such an approach is needed because IoT microcontrollers have limited computation resources; therefore, whether the IoT node sends all raw data to the cloud for processing, significant communications bottlenecks will emerge [78]. Instead, by relying on edge computation, the data movement amount can be reduced [79]. In addition, in the context of trusted computation, this also allows for opportunities of applying privacy-preserving computation locally before data is sent to a potentially untrusted cloud provider [80].

Different hardware vulnerabilities are being explored and reported in the context of FPGAs in Edge computing. Moreover, the trustworthiness of hardware devices is drawing significant attention due to issues like trojan insertion, IP cloning, and hardware counterfeits, among others, which are more effective at causing damage with the help of machine learning (ML) techniques [7], [72].

On-chip key storage and generation have been used to authenticate different hardware devices [72]. However, the authentication and identification of FPGAs are challenging due to the lack of nonvolatile memory. To mitigate this limitation, physically unclonable functions (PUFs) were proposed as techniques to generate and store digital keys on FPGA [81]. For a PUF structure, ML techniques can learn its complex input-output mapping from a small number of challenge and response pairs (CRPs) to accurately predict the unknown responses. Nevertheless, the computing complexity of ML algorithms (i.e., support vector machine and logistic regression) increases exponentially with the scale of PUF, and the number of nonlinear logical elements in a PUF [82]. Hybrid attacks use the auxiliary information from the side channel to assist the ML algorithm in modelling the PUF. For example, a hybrid attack combining differential power analysis and logistic regression is proposed in [83].

IP hacking, in particular, is quite multifaceted. An attacker has different avenues to mount such an attack, ranging from an untrustworthy foundry, or an untrusted test facility, to malicious end users. To exploit the IP of the chip, an adversary at the test facility may misuse the test patterns to compromise its security. When employing machine learning techniques, the attack becomes more effective [84]. Finally, there is Semi-Supervised Learning (SSL), which is a powerful derivative for humans to discover hidden knowledge since untrusted unlabeled data leads to many unknown security risks. FPGAs on edge face these new concerns by identifying backdoor threats [85].

C. THREATS IN THE CLOUD AND DATACENTER

In a cloud context, the most concerning issues are authentication, access control, data privacy and security and trust management [86]. Since clouds require efficient and high-performance processing, it is a challenge that the solutions to many of these security concerns can result in performance overhead [38], [87].

For FPGAs deployed in a cloud setting, the following threats are most relevant:

- Data confidentiality: The Cloud Service Provider (CSP) are trusted, and their vendors have a complete monitoring mechanism to prevent the stealing of sensitive information via illegal access to large amounts of private data [8]. However, attackers can execute malicious software to obtain administrative privileges by users and provide them with sensitive information unconsciously [68].
- Black box attacks: A frequent attack on FPGA systems is when an attacker tests all possible input combinations until they gain access to their internal infrastructure and change admin or user credentials. This procedure may be done when the attacker has previous information about the user/admin credentials [13], [88].
- Reprogrammed decryption key: With a protected bitstream file, the decryption key for processing the data remains secure, as it will only be exposed within the FPGA. In contrast, cloud vendors offer multi-tenancy to possible untrusted users applications, and they can find a decryption key to access sensitive information remotely [88].
- Read-back attacks: For debugging, FPGAs often have a read-back feature to allow values to be read from the FPGA through a specialized interface. This functionality should not exist or be physically disabled once the chip has passed production tests. However, third-party sellers may forget to turn off this functionality. An example of this security concern is when the JTAG connector is intercepted to get access to the IO ports of the FPGA [88].

To achieve security goals in cloud computing, FPGA can be part of the cloud service provider. By taking advantage of the unique security features of FPGAs, one can enable such a computing scenario that when privacy-sensitive data is processed on the cloud side using FPGAs (‘‘virtualization’’), the user's data is not disclosed to the service provider who hosts the FPGA cloud [7]. Principal security criteria for FPGAs implementation in the cloud are as follows:

- Persistent storage of keys: For the bitstream protection, [8] proposes a hard-wired logic in FPGAs for the key exchange and authentication using the Public Key Infrastructure (PKI). It implements a fully enclaved processing of application kernels inside the FPGA fabric for data confidentiality, taking and emitting only encrypted data.
- Fully homomorphic encryption: For simple cloud services such as data storage, the user can keep control of outsourced data by encrypting the data before uploading it to the cloud. In order to fully leverage the computation, storage, and communication capacities of the public cloud for more complex tasks, the user usually needs to put full trust in the cloud service providers, as general encryption will make the computation impossible. Fully homomorphic encryption (FHE) supports operations on cypher-texts. A user only needs to send encrypted data to the cloud for computation without the data ever being decrypted for processing. The drawback of FHE based solution is that the existing FHE schemes are very inefficient, and it is not practical to use them for meaningful computation tasks such as a signal or data analysis [7], [12]

Traditional FPGA applications are only accessible to a dedicated user. Although modern commercial FPGAs can support multiple applications through partial reconfiguration, existing FPGA architectures and design flows are not yet optimized for sharing hardware resources among multiple users and applications. Therefore, the multi-tenant computing goal of FPGA virtualization leads to security issues, mainly side-channels attacks [41], [42]. Hence, isolation techniques allow the FPGA virtualization tenant in the system to have shared access to multiple hardware resources, but only their resources at a specific time, and they cannot manipulate information from other tenants. Thus, virtualizing FPGA requires isolation, such as: functional isolation, performance isolation, and fault isolation [89].

IV. PROVIDING TRUSTED EXECUTION ENVIRONMENT WITH FPGAS

There is novel related work on building trusted execution solutions using FPGAs in the cloud, providing similar guarantees to Intel SGX. In this section, we focus on five representative examples, each focusing on different sets of goals that can be achieved with TEEs. While this list of related work is in no way exhaustive, we consider them a representative sample of the directions actively explored today.

A recent work by Zeitouni et al. [13] presents a partial solution to secure the intellectual property (IP) that the user is uploading to the FPGA. The problem lies in the fact that Cloud Services Provider (CSP) needs access to the configuration data representing the circuit of the user to run virus scanner tools. Consequently, the client is forced to reveal the IP of the hardware circuit to the CSP, which may violate the IP protection policy of the companies. The use of encrypted bitstreams does not comply with the requirement of the CSP to check the incoming configurations before they are loaded into the FPGA. Therefore, Zeitouni et al. [13] propose the TruFPGA scheme, which uses a Trusted Execution Environment (TEE) to virus safety check on client bitstreams. The TEE resides either on the client-side or CSP. As a result, CSP has no access to the decrypted bitstream and prevents rogue FPGA configurations.

Similar to the previous work, Zhao et al. [14] explore the weakness of TEEs against direct physical attacks.Reference [14] addresses these concerns with the Shielded Enclaves for Cloud FPGAs (ShEF) framework. ShEF consists of two main components. First, the ShEF boot process centers around a software security kernel that extends the FPGA's hardware root-of-trust. The AES device key is the true root-of-trust, protected by existing mission-critical security mechanisms in current FPGAs. The private device key provides the asymmetric cryptography needed for attestation. The ShEF boot process is directly integrated into the secure boot mechanism provided by FPGA vendors. Second, the ShEF Shield is responsible for communicating with host software and protecting any sensitive data the accelerator uses through a highly customizable and extensible set of soft-logic engines. Users can customize a rich set of parameters, such as encryption logic parallelism, optimizations for memory access patterns, cryptographic primitives, authentication block size, and key size over individual memory regions. Therefore, ShEF shield provides customization as a key feature, enabling users to adapt security mechanisms to match their accelerator's unique bandwidth requirements, memory access characteristics, and threat model at minimum performance and area cost. As a result, ShEF protects any sensitive data of users and is used by the accelerator through a highly customizable and extensible set of soft-logic engines.

Whereas the previous two works focused on a single cloud tenant accessing a dedicated FPGA, there is also work on understanding how mutually distrusting tenants could be allowed to share FPGA resources securely. A recent paper by Hategekimana et al. [90] defines an optimal multi-tenancy of FPGAs in a Cloud environment without performance degradation. However, the FPGA devices have no architectural support to allow isolating mutually distrusted accelerators sharing the same FPGA. Therefore, use insights from domain separation and isolation of guest virtual machines (VMs) execution in MAC-based hypervisors to design and implement a transparent security framework for controlled sharing of hardware modules in CPU+FPGA heterogeneous cloud nodes. The proposed framework goal is to guarantee that in FPGA cloud services, hardware modules execute and reside in the same security context as the "caller" guest VM by propagating to the "caller" modules. Guest VM privilege boundaries are defined at the software level. In the same research line, Mandebi et al. [91] propose an approach for FPGA virtualization in cloud infrastructure that addresses resource pooling and elasticity to allow logically isolated workloads to share a single FPGA. It starts by dividing FPGAs into disjoint regions. The regions are then interfaced to a network-on-chip (NoC) interconnect that extends a task hardware domain.

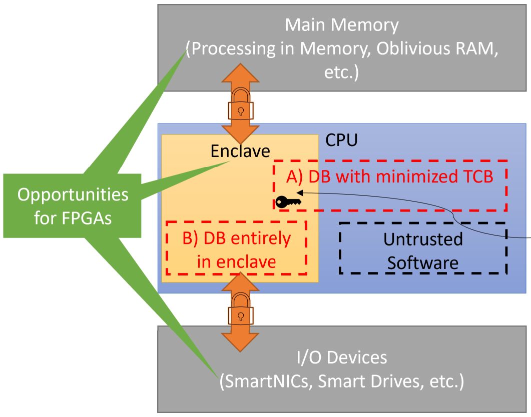
Main Memory
(Processing in Memory, Oblivious RAM, etc.)
CPU
Enclave
A) DB with minimized TCB
Opportunities for FPGAs
B) DB entirely in enclave
Untrusted Software
I/O Devices
(SmartNICs, Smart Drives, etc.)

FIGURE 1. Novel database solutions that incorporate TEEs, do so by prioritizing one of two approaches: either aiming to minimize the size of the trusted code base (TCB) to make verification easier or by aiming to run the entire application inside the enclave to protect against a wider range of attacks and tampering.

Finally, Oh et al. [27], emerging work designs a TEE specialized multi-tenancy mechanism called MeetGo. It enables robust remote computation against insider threats with performance requirements, allowing logically isolated workloads to share a single device. MeetGo initializes by dividing FPGAs into disjoint regions. The regions are then interfaced to a network-on-chip (NoC) interconnect. MeetGo uses CPU-FPGA hybrid architecture to devise a remote attestation mechanism that can verify the integrity of the applications through hardware logic. Then, MeetGo implements an isolation mechanism to block unauthorized access to the applications from the malicious CPU. Lastly, MeetGo has developed a secure communication mechanism to allow secure sensitive data transmissions between the installed applications and remote users.

V. EMERGING DATA MANAGEMENT AND PROCESSING APPLICATIONS ON TEES

Recent years have seen a rapid increase in the number of database and data analytics applications that propose using TEEs to provide trusted and private data management and analysis [66], [67], [92], [93], [94], [95], [96]. On the one hand, this trend is driven by the increasing privacy regulations, such as the General Data Protection Regulation (GDPR) in the EU and California Customer Privacy Act (CCPA) in California. On the other hand, there exist the general availability of TEEs in the cloud, with Intel SGX being a prominent example. In the following, we provide a high level overview of the two general directions adopted by related work in implementing databases on trusted hardware and explain how these open up opportunities for FPGAs in the future. The aforementioned is shown in Fig. 1.

A. APPROACH: MINIMIZE THE TRUSTED COMPUTING BASE

Several proposed systems move their core query processing operations into a TEE and leave the rest of the database (e.g., storage manager, logging, etc.) to run on untrusted hardware [92], [97]. This approach reduces the trusted computing base but requires fine grained interaction among the TEE, the high level database (DB) application, and the operating system. In this model, the data that is being processed is encrypted and appears as a BLOB to the database. The metadata and schema information, however, is typically not encrypted. Secured data can only be decrypted inside the TEE when performing queries. One of the main non-performance-related challenges is the secure management of user encryption keys within the TEE belonging to the database [97]. In terms of performance, there is a strong incentive to encrypt only the most sensitive parts of the data and process the rest in ‘‘plaintext’’, in order to avoid significant performance degradation introduced by the decryption inside the TEEs.

In terms of the type of TEE used, some proposals rely on custom FPGA-based solutions [97], while others on Intel SGX or even Operating System-based software solutions [92]. Overall, however, there is an opportunity for providing TEEs with specialized instructions, e.g., for decryption followed by string manipulation or arithmetic followed by encryption, to improve processing performance. There is a rich related work on FPGA-based SQL acceleration [98] and many of these ideas translate well to the TEE space.

B. APPROACH: FIT ENTIRE APPLICATIONS IN TEE

Many proposals aim to fit entire database applications within the confines of TEEs. Some of these focus on query processing [66], [94], [96], while others look at the question of trustworthy (distributed) storage [53], [67], [99]. Deploying these applications in their entirety inside TEEs has the benefit of not only ensuring trustful query execution but also reducing the attack surface for side-channels. For instance, by using oblivious RAM techniques [93]. Such solutions, however, face significant performance penalties due to, on the one hand, the frequent crossing between I/O and memory in the untrusted domain and the trusted execution environment and, on the other hand, the overhead of providing oblivious RAM behavior.

While there are emerging works that aim to reduce the overhead of dealing with I/O from TEEs [1], it is still an open question how to most efficiently create TEE-based applications that communicate over the network and use large amounts of data. As a related work demonstrates, [1], part of the problem lies in the Operating System and various software-layer crossings, and part of it lies in the hardware design underlying the TEE. These challenges could be addressed with FPGAs. It is possible to increase the trust across the computer architecture, including network interface cards (some of which already contain FPGAs [100]) and flash storage, such as Samsung SmartSSDs [101]. This would allow for staying within the ‘‘trusted domain’’ for most operations, simplifying the role of the Operating System or Hypervisor. Bailleu et al. [102] present a TEE system called Avocado, which has the following properties: (1) introduce a secure in-memory distributed storage system that provides strong security, (2) fault-tolerance, (3) consistency (linearizability), and (4) performance for untrusted cloud environments. These Avocado properties are primarily designed for securing limited physical memory (enclave) within a single-node system. Reference [102] does not aim at protecting against side-channel attacks and access or network pattern attacks. However, its principal contribution is related to avoiding the prominent I/O mechanism employed by TEE frameworks. It is based on asynchronous system calls, which exhibit significant overheads that the system calls present a bottleneck and might sacrifice performance. To overcome this limitation, Avocado opted for a new network stack based on eRPC, a state-of-the-art general-purpose and asynchronous remote procedure call (RPC) library for high-speed networking for lossy Ethernet or lossless fabrics. eRPC uses a polling-based network I/O along with userspace drivers, eliminating interrupts and system call overheads from the datapath.

To conclude, FPGAs could be useful for implementing further protections such as oblivious RAM, especially given the emergence of memory devices with built-in computational elements (as summarized [103]).

VI. DISCUSSION AND CLOSING THOUGHTS

The importance of securing FPGAs against malicious actors needs little additional motivation. The vast body of related work in our community shows the importance of the problem and provides many possible solutions to security threats. The move to the cloud, however, and the use of FPGAs not only as more flexible ASICs but as an extension of the software, requires us to adopt a more holistic view of FPGA security has to be provided ranging from the low-level hardware details to the application behaviour designed by users. The lack of clear connection between low-level and high-level criteria for implementing trusted computation has already been identified in works such as [13], [35]. This gives us a new security point of view on cloud computing and the next steps we should follow: evaluate security proposals with complex applications and data management to determine which part of the cloud can be implemented correctly and securely with FPGAs.

In terms of the usability of security-related tools and results by non-experts, having reviewed a large number of related works, we found that when it comes to trusted computation solutions using FPGAs, they are very distant in their experimental designs. On the one hand, some works focus on security recommendations for hardware attacks and, through this, affect the operation of an FPGA logic but typically do not put this in the perspective of ‘‘real world’’ applications. On the other hand, significant future work remains concerning integration with higher-level software. Clouds are controlled by software frameworks, e.g., for virtualization, and ensuring that these frameworks can manage the FPGA-based security features is of paramount importance.

We believe there is an unique opportunity for FPGAs in the intersection of specialization and trusted computing. They have already demonstrated their usefulness to offload compute-intensive operations from various application domains, and also provide trust in an untrusted environment. They open up novel possibilities. As we discussed in this survey, the database community has a significant push to integrate cloud TEEs in databases and data analytics. Since all these solutions suffer from varying levels of performance degradation due to the use of the TEE, by using FPGAs, we could offer trusted computation without slowing down the applications on top. Finally, deep learning technology may challenge traditional trusted computing, where FPGAs are also involved on the application side. As future work, we consider exploring AI-based security technologies and how they match with the FPGA architecture [104].

REFERENCES

[1] J. Thalheim, H. Unnibhavi, C. Priebe, P. Bhatotia, and P. Pietzuch, "Rkt-IO: A direct I/O stack for shielded execution," in *Proc. 6th Eur. Conf. Comput. Syst.*, 2021, pp. 490–506.

[2] N. Kaja and A. Shaout, "CCTM–cloud computing trust model," in *Proc. Int. Arab Conf. Inf. Technol. (ACIT)*, Nov. 2018, pp. 1–6.

[3] X. Gai, Y. Li, Y. Chen, and C. Shen, "Formal definitions for trust in trusted computing," in *Proc. 7th Int. Conf. Ubiquitous Intell. Comput. 7th Int. Conf. Autonomic Trusted Comput.*, Oct. 2010, pp. 305–310.

[4] K. Eguro and R. Venkatesan, "FPGAs for trusted cloud computing," in *Proc. 22nd Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2012, pp. 63–70.

[5] M. Al-Asli, M. E. S. Elrabaa, and M. Abu-Amara, "FPGA-based symmetric re-encryption scheme to secure data processing for cloud-integrated Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 446–457, Feb. 2019.

[6] C. Jin, V. Gohil, R. Karri, and J. Rajendran, "Security of cloud FPGAs: A survey," 2020, *arXiv:2005.04867*.

[7] L. Xu, W. Shi, and T. Suh, "PFC: Privacy preserving FPGA cloud—A case study of MapReduce," in *Proc. IEEE 7th Int. Conf. Cloud Comput.*, Jun. 2014, pp. 280–287.

[8] H.-Y. Kim, R. Myung, B. Hong, H. Yu, T. Suh, L. Xu, and W. Shi, "SafeDB: Spark acceleration on FPGA clouds with enclaved data processing and bitstream protection," in *Proc. IEEE 12th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2019, pp. 107–114.

[9] D. Korolija, T. Roscoe, and G. Alonso, "Do OS abstractions make sense on FPGAs?" in *Proc. 14th USENIX Symp. Operating Syst. Design Implement. (OSDI)*. Berkeley, CA, USA: USENIX Association, Nov. 2020, pp. 991–1010. [Online]. Available: https://www.usenix.org/conference/osdi20/presentation/roscoe

[10] K. Santhosh, R. Shashidhar, A. Mahalingaswamy, K. Praveen, and M. Roopa, "Design of high speed AES system for efficient data encryption and decryption system using FPGA," in *Proc. Int. Conf. Electr., Electron., Commun., Comput., Optim. Techn. (ICEECCOT)*, Dec. 2018, pp. 1279–1282.

[11] B. Nallathambi and P. Karthigaikumar, "FPGA implementation of hiding information using cryptographic key," in *Proc. Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2014, pp. 1–5.

[12] Y. Su, B. Yang, C. Yang, and L. Tian, "FPGA-based hardware accelerator for leveled ring-lwe fully homomorphic encryption," *IEEE Access*, vol. 8, pp. 168008–168025, 2020.

[13] S. Zeitouni, J. Vliegen, T. Frassetto, D. Koch, A.-R. Sadeghi, and N. Mentens, "Trusted configuration in cloud FPGAs," in *Proc. 29th IEEE Int. Symp. Field-Programmable Custom Comput.*, Mar. 2021, pp. 233–241. [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/125919/

[14] M. Zhao, M. Gao, and C. Kozyrakis, "ShEF: Shielded enclaves for cloud FPGAs," 2021, *arXiv:2103.03500*.

[15] F. Turan and I. Verbauwhede, "Trust in FPGA-accelerated cloud computing," *ACM Comput. Surveys*, vol. 53, no. 6, pp. 1–28, Dec. 2020, doi: 10.1145/3419100.

[16] A. Baobaid, M. Meribout, V. K. Tiwari, and J. P. Pena, "Hardware accelerators for real-time face recognition: A survey," *IEEE Access*, vol. 10, pp. 83723–83739, 2022.

[17] Z. István, K. Kara, and D. Sidler, "FPGA-accelerated analytics: From single nodes to clusters," *Found. Trends Databases*, vol. 9, no. 2, pp. 101–208, 2020, doi: 10.1561/1900000072.

[18] R. Skhiri, V. Fresse, J. P. Jamont, B. Suffran, and J. Malek, "From FPGA to support cloud to cloud of FPGA: State of the art," *Int. J. Reconfigurable Comput.*, vol. 2019, pp. 1–17, Dec. 2019.

[19] U. Farooq, I. Baig, and B. A. Alzahrani, "An efficient inter-FPGA routing exploration environment for multi-FPGA systems," *IEEE Access*, vol. 6, pp. 56301–56310, 2018.

[20] E. A. Papatheofanous, D. Reisis, and K. Nikitopoulos, "LDPC hardware acceleration in 5G open radio access network platforms," *IEEE Access*, vol. 9, pp. 152960–152971, 2021.

[21] J. Gonzalez-Dominguez, L. Wienbrandt, J. C. Kassens, D. Ellinghaus, M. Schimmler, and B. Schmidt, "Parallelizing Epistasis Detection in GWAS on FPGA and GPU-accelerated Computing Systems," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 12, no. 5, pp. 982–994, Sep. 2015.

[22] J. Borrego-Carazo, D. Castells-Rufas, E. Biempica, and J. Carrabina, "Resource-constrained machine learning for ADAS: A systematic review," *IEEE Access*, vol. 8, pp. 40573–40598, 2020.

[23] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019.

[24] M. Rothmann and M. Porrmann, "A survey of domain-specific architectures for reinforcement learning," *IEEE Access*, vol. 10, pp. 13753–13767, 2022.

[25] Y. Zhu, J. Liu, R. Yu, Z. Mu, L. Huang, J. Chen, and J. Chen, "Attitude solving algorithm and FPGA implementation of four-rotor UAV based on improved mahony complementary filter," *Sensors*, vol. 22, no. 17, p. 6411, Aug. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/17/6411

[26] P. Song, Y. Qie, C. Hao, Y. Li, Y. Zhao, Y. Hao, H. Liu, and Y. Qi, "Resource-saving customizable pipeline network architecture for multi-signal processing in edge devices," *Sensors*, vol. 22, no. 15, p. 5720, Jul. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/15/5720

[27] H. Oh, K. Nam, S. Jeon, Y. Cho, and Y. Paek, "Meetgo: A trusted execution environment for remote applications on FPGA," *IEEE Access*, vol. 9, pp. 5124–51313, 2021.

[28] (2013). *Xilinx Announces Full Production of Its Entire ZYNQ-7000 all Programmable SoC Family*. [Online]. Available: http://www.xilinx.com/news/press/2013/xilinx-announces-full-production-of-its-entire-zynq-7000

[29] A. Putnam, "A reconfigurable fabric for accelerating large-scale datacenter services," *IEEE Micro*, vol. 35, no. 3, pp. 10–22, May/Jun. 2015.

[30] (2016). *EC2 Instances (F1) With Programmable Hardware*. [Online]. Available: https://aws.amazon.com/blogs/aws/developer-preview-ec2-instances-f1-with-programmable-hardware

[31] F. Abel, J. Weerasinghe, C. Hagleitner, B. Weiss, and S. Paredes, "An FPGA platform for hyperscalers," in *Proc. IEEE 25th Annu. Symp. High-Performance Interconnects (HOTI)*, Aug. 2017, pp. 29–32.

[32] G. Weisz, "Keynote2: Global-scale FPGA-accelerated deep learning inference with microsoft's project brainwave," in *Proc. Int. Conf. ReConFigurable Comput. FPGAs (ReConFig)*, Cancun, Mexico, 2019, p. 1, doi: 10.1109/ReConFig48160.2019.8994747.

[33] M. K. Pekturk and M. Unal, "Uzaktan algilama uygulamalarinda Gerųek zamanli Büyük veri analizine genel bakiş," in *Proc. 25th Signal Process. Commun. Appl. Conf. (SIU)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Jun. 2017, pp. 1–9.

[34] (2021). *FPGA-Centric Software Acceleration Made Easy*. [Online]. Available: https://www.accelize.com/blog-fpga-centric-software-acceleration-made-easy/

[35] F. Turan, S. S. Roy, and I. Verbauwhede, "HEAWS: An Accelerator for Homomorphic Encryption on the Amazon AWS FPGA," *IEEE Trans. Comput.*, vol. 69, no. 8, pp. 1185–1196, Aug. 2020.

[36] J. Zhang and G. Qu, "A survey on security and trust of FPGA-based systems," in *Proc. Int. Conf. Field-Programmable Technol. (FPT)*, Dec. 2014, pp. 147–152.

[37] R. Druyer, L. Torres, P. Benoit, P. V. Bonzom, and P. Le-Quere, "A survey on security features in modern FPGAs," in *Proc. 10th Int. Symp. Reconfigurable Commun.-Centric Syst. Chip (ReCoSoC)*, Jun. 2015, pp. 1–8.

[38] O. Glamocanin, L. Coulon, F. Regazzoni, and M. Stojilovic, "Are cloud FPGAs really vulnerable to power analysis attacks?" in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 1007–1010.

[39] C. Kachris and D. Soudris, "A survey on reconfigurable accelerators for cloud computing," in *Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2016, pp. 1–10.

[40] N. Mohammedali and M. O. Agyeman, "A study of reconfigurable accelerators for cloud computing," in *Proc. 2nd Int. Symp. Comput. Sci. Intell. Control*, New York, NY, USA: ACM, Sep. 2018, doi: 10.1145/3284557.3284563.

[41] A. Vaishnav, K. D. Pham, and D. Koch, "A survey on FPGA virtualization," in *Proc. 28th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2018, pp. 131–1317.

[42] M. H. Quraishi, E. B. Tavakoli, and F. Ren, "A survey of system architectures and techniques for FPGA virtualization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 9, pp. 2216–2230, Sep. 2021.

[43] J.-A.-M. Mondol, "Cloud security solutions using FPGA," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process.*, Aug. 2011, pp. 747–752.

[44] J. Wang, Y. Shi, G. Peng, H. Zhang, B. Zhao, F. Yan, F. Yu, and L. Zhang, "Survey on key technology development and application in trusted computing," *China Commun.*, vol. 13, no. 11, pp. 70–90, Nov. 2016.

[45] D. C. G. Valadares, N. C. Will, J. Caminha, M. B. Perkusich, A. Perkusich, and K. C. Gorgonio, "Systematic literature review on the use of trusted execution environments to protect cloud/fog-based Internet of Things applications," *IEEE Access*, vol. 9, pp. 80953–80969, 2021.

[46] H. Liang, M. Li, Y. Chen, L. Jiang, Z. Xie, and T. Yang, "Establishing trusted I/O paths for SGX client systems with aurora," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1589–1600, 2020.

[47] Z. Yu, H. Dai, X. Xi, and M. Qiu, "A trust verification architecture with hardware root for secure clouds," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 3, pp. 353–364, Jul. 2020.

[48] R. N. Akram, K. Markantonakis, and K. Mayes, *An Introduction to the Trusted Platform Module and Mobile Trusted Module*. New York, NY, USA: Springer, 2014, pp. 71–93.

[49] L. Jaeger, "Information security awareness: Literature review and integrative framework," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018.

[50] P. England, A. Marochko, D. Mattoon, R. Spiger, S. Thom, and D. Wooten, "RIoT—A foundation for trust in the Internet of Things," Microsoft Res., Tech. Rep. MSR-TR-2016-18, Apr. 2016. [Online]. Available: https://www.microsoft.com/en-us/research/publication/riot-a-foundation-for-trust-in-the-internet-of-things/

[51] W. Arthur, D. Challener, and K. Goldman, *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*, 1st ed. Berkeley, CA, USA: Apress, Jan. 2015. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4302-6584-9, doi: 10.1007/978-1-4302-6584-9.

[52] X. Liang, R. Jiang, and H. Kong, "Secure and reliable VM-vTPM migration in private cloud," in *Proc. 2nd Int. Symp. Instrum. Meas., Sensor Netw. Autom. (IMSNA)*, Dec. 2013, pp. 510–514.

[53] H. Oh, A. Ahmad, S. Park, B. Lee, and Y. Paek, "Trustore: Side-channel resistant storage for SGX using Intel hybrid CPU-FPGA," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: ACM, 2020, pp. 1903–1918, doi: 10.1145/3372297.3417265.

[54] M. Gross, N. Jacob, A. Zankl, and G. Sigl, "Breaking TrustZone memory isolation through malicious hardware on a modern FPGA-SoC," in *Proc. 3rd ACM Workshop Attacks Solutions Hardw. Secur. Workshop*. New York, NY, USA: ACM, Nov. 2019, pp. 3–12, doi: 10.1145/3338508.3359568.

[55] S. Mofrad, F. Zhang, S. Lu, and W. Shi, "A comparison study of Intel SGX and amd memory encryption technology," New York, NY, USA: ACM, 2018, pp. 1–38, doi: 10.1145/3214292.3214301.

[56] D. Kaplan, J. Powell, and T. Wollera, "AMD memory encryption," Adv. Micro Devices, White Paper, Oct. 2021. [Online]. Available: https://developer.amd.com/AMD_Memory_Encryption_Whitepaper_v7Public.pdf, https://www.amd.com/system/files/TechDocs/memory-encryption-white-paper.pdf, and https://www.amd.com/en/developer.html

[57] A. K. Junejo, N. Komninos, M. Sathiyanarayanan, and B. S. Chowdhry, "Trustee: A trust management system for fog-enabled cyber physical systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2030–2041, Oct. 2021.

[58] Y. Xia, Z. Hua, Y. Yu, J. Gu, H. Chen, B. Zang, and H. Guan, "Colony: A privileged trusted execution environment with extensibility," *IEEE Trans. Comput.*, vol. 71, no. 2, pp. 479–492, Feb. 2022.

[59] S. Chen, W. Hu, and Z. Li, "High performance data encryption with AES implementation on FPGA," in *Proc. IEEE IEEE 5th Intl Conf. Big Data Secur. Cloud (BigDataSecurity) Intl Conf. High Perform. Smart Comput., (HPSC) IEEE Intl Conf. Intell. Data Secur. (IDS)*, May 2019, pp. 149–153.

[60] R. Likhithashree and D. Kiran, "Area-efficient physically unclonable functions for FPGA using ring oscillator," in *Proc. 2nd Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA)*, Mar. 2020, pp. 403–408.

[61] Y. Qin and T. Xia, "Sensitivity analysis of ring oscillator based hardware trojan detection," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2017, pp. 1979–1983.

[62] M. Zhao and G. E. Suh, "FPGA-based remote power side-channel attacks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 229–244.

[63] D. R. E. Gnad, C. D. K. Nguyen, S. H. Gillani, and M. B. Tahoori, "Voltage-based covert channels using FPGAs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 26, no. 6, pp. 1–25, Jun. 2021, doi: 10.1145/3460229.

[64] H. Kan, R. Li, D. Su, Y. Wang, Y. Shen, and W. Liu, "Trusted edge cloud computing mechanism based on FPGA cluster," in *Proc. IEEE 8th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Nov. 2020, pp. 146–149.

[65] S. Suo, C. Cui, G. Jian, X. Kuang, Y. Yang, Y. Zhao, and K. Huang, "Implementation of the high-speed SM4 cryptographic algorithm based on random pseudo rounds," in *Proc. IEEE Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA)*, vol. 1, Nov. 2020, pp. 112–117.

[66] D. Vinayagamurthy, A. Gribov, and S. Gorbunov, "StealthDB: A scalable encrypted database with full SQL query support," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 3, pp. 370–388, Jul. 2019.

[67] Y. Sun, S. Wang, H. Li, and F. Li, "Building enclave-native storage engines for practical encrypted databases," *Proc. VLDB Endowment*, vol. 14, no. 6, pp. 1019–1032, Feb. 2021.

[68] M. Okada, T. Suzuki, N. Nishio, H. M. Waidyasooriya, and M. Hariyama, "FPGA-accelerated searchable encrypted database management systems for cloud services," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 1373–1385, Apr. 2022.

[69] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on FPGAs," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 1111–1116.

[70] R. Skhiri, V. Fresse, J.-P. Jamont, and B. Suffran, "Challenges of virtualization FPGA in a cloud context," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environments Meas. Syst. Appl. (CIVEMSA)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Jun. 2017, pp. 171–176.

[71] M. Bahadori and K. Jarvinen, "A programmable SoC-based accelerator for privacy-enhancing technologies and functional encryption," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 10, pp. 2182–2195, Oct. 2020.

[72] X. Xu and J. Zhang, "Rethinking FPGA security in the new era of artificial intelligence," in *Proc. 21st Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2020, pp. 46–51.

[73] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik, "Hardware Trojan insertion by direct modification of FPGA configuration bitstream," *IEEE Design Test*, vol. 30, no. 2, pp. 45–54, Apr. 2013.

[74] S. Trimberger and J. Moore, "FPGA security: From features to capabilities to trusted systems," in *Proc. 51st Annu. Design Autom. Conf.*, Jun. 2014, pp. 1–4.

[75] E. M. Benhani, L. Bossuet, and A. Aubert, "The security of ARM TrustZone in a FPGA-based SoC," *IEEE Trans. Comput.*, vol. 68, no. 8, pp. 1238–1248, Aug. 2019.

[76] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: Attacks, defenses, and challenges," *IEEE Access*, vol. 8, pp. 184013–184035, 2020.

[77] N. Rangarajan, S. Patnaik, J. Knechtel, R. Karri, O. Sinanoglu, and S. Rakheja, "Opening the doors to dynamic camouflaging: Harnessing the power of polymorphic devices," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 137–156, Jan. 2022.

[78] N. Fujii and N. Koike, "IoT remote group experiments in the cyber laboratory: A FPGA-based remote laboratory in the hybrid cloud," in *Proc. Int. Conf. Cyberworlds (CW)*, Sep. 2017, pp. 162–165.

[79] R. Ferdian, R. Aisuwarya, and T. Erlina, "Edge computing for Internet of Things based on FPGA," in *Proc. Int. Conf. Inf. Technol. Syst. Innov. (ICITSI)*, Oct. 2020, pp. 20–23.

[80] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *J. Mach. Learn. Res.*, vol. 17, pp. 492–542, Jan. 2016.

[81] P. Lokhande and A. M. Shah, "Strong authentication and encryption modeling using physical unclonable function based on FPGA," in *Proc. 6th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2021, pp. 192–195.

[82] X. Xu, U. Rührmair, D. E. Holcomb, and W. Burleson, "Security evaluation and enhancement of bistable ring PUFs," in *Radio Frequency Identification*, S. Mangard and P. Schaumont, Eds. Cham, Switzerland: Springer, 2015, pp. 3–16.

[83] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, and W. Burleson, "Efficient power and timing side channels for physical unclonable functions," in *Cryptographic Hardware and Embedded Systems—CHES*, L. Batina and M. Robshaw, Eds. Berlin, Germany: Springer, 2014, pp. 476–492.

[84] N. Rangarajan, S. Patnaik, J. Knechtel, R. Karri, O. Sinanoglu, and S. Rakheja, "Opening the doors to dynamic camouflaging: Harnessing the power of polymorphic devices," 2018, *arXiv:1811.06012*.

[85] Z. Yan, J. Wu, G. Li, S. Li, and M. Guizani, "Deep neural backdoor in semi-supervised learning: Threats and countermeasures," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4827–4842, 2021.

[86] A. Markandey, P. Dhamdhere, and Y. Gajmal, "Data access security in cloud computing: A review," in *Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON)*, Sep. 2018, pp. 633–636.

[87] S. Kaushik and C. Gandhi, "Multi-level trust agreement in cloud environment," in *Proc. 12th Int. Conf. Contemp. Comput. (IC)*, Aug. 2019, pp. 1–5.

[88] M. A. Will and R. K. L. Ko, "Secure FPGA as a service—Towards secure data processing by physicalizing the cloud," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 449–455.

[89] M. Asiatici, N. George, K. Vipin, S. A. Fahmy, and P. Ienne, "Virtualized execution runtime for FPGA accelerators in the cloud," *IEEE Access*, vol. 5, pp. 1900–1910, 2017.

[90] F. Hategekimana, J. M. Mbongue, M. J. H. Pantho, and C. Bobda, "Secure hardware kernels execution in CPU+FPGA heterogeneous cloud," in *Proc. Int. Conf. Field-Programmable Technol. (FPT)*, 2018, pp. 182–189.

[91] J. M. Mbongue, A. M.-I. Shuping, P. Bhowmik, and C. Bobda, "Architecture support for FPGA multi-tenancy in the cloud," in *Proc. IEEE 31st Int. Conf. Application-Specific Syst., Architectures Processors (ASAP)*, Jul. 2020, pp. 125–132.

[92] P. Antonopoulos, A. Arasu, K. D. Singh, K. Eguro, N. Gupta, R. Jain, R. Kaushik, H. Kodavalla, D. Kossmann, N. Ogg, R. Ramamurthy, J. Szymaszek, J. Trimmer, K. Vaswani, R. Venkatesan, and M. Zwilling, "Azure SQL database always encrypted," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2020, pp. 1511–1525.

[93] S. Eskandarian and M. Zaharia, "ObliDB: Oblivious query processing for secure databases," 2017, *arXiv:1710.00458*.

[94] C. Priebe, K. Vaswani, and M. Costa, "EnclaveDB: A secure database using SGX," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 264–278.

[95] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, and I. Stoica, "Opaque: An oblivious and encrypted distributed analytics platform," in *Proc. 14th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2017, pp. 283–298.

[96] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, "VC3: Trustworthy data analytics in the cloud using SGX," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 38–54.

[97] A. Arasu, K. Eguro, M. Joglekar, R. Kaushik, D. Kossmann, and R. Ramamurthy, "Transaction processing on confidential data using cipherbase," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 435–446.

[98] J. Fang, Y. T. B. Mulder, J. Hidders, J. Lee, and H. P. Hofstee, "In-memory database acceleration on FPGAs: A survey," *VLDB J.*, vol. 29, no. 1, pp. 33–59, Jan. 2020.

[99] Z. István, S. Ponnapalli, and V. Chidambaram, "Software-defined data protection: Low overhead policy compliance at the storage layer is within reach!" *Proc. VLDB Endowment*, vol. 14, no. 7, pp. 1167–1174, Mar. 2021, doi: 10.14778/3450980.3450986.

[100] D. Chiou, "The Microsoft catapult project," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Oct. 2017, p. 124.

[101] J. Wang, D. Park, Y.-S. Kee, Y. Papakonstantinou, and S. Swanson, "SSD in-storage computing for list intersection," in *Proc. 12th Int. Workshop Data Manage. New Hardw.*, Jun. 2016, pp. 1–7.

[102] M. Bailleu, D. Giantsidi, V. Gavrielatos, D. L. Quoc, V. Nagarajan, and P. Bhatotia, "Avocado: A secure in-memory distributed storage system," in *Proc. USENIX Annual Tech. Conf. (USENIX ATC)*. Berkeley, CA, USA: USENIX Association, Jul. 2021, pp. 65–79. [Online]. Available: https://www.usenix.org/conference/atc21/presentation/bailleu

[103] S. Ghose, A. Boroumand, J. S. Kim, J. Gómez-Luna, and O. Mutlu, "Processing-in-memory: A workload-driven perspective," *IBM J. Res. Develop.*, vol. 63, no. 6, pp. 1–3, 2019.

[104] G. Li, J. Wu, S. Li, W. Yang, and C. Li, "Multitentacle federated learning over software-defined industrial Internet of Things against adaptive poisoning attacks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1260–1269, Feb. 2023.

PAUL D. ROSERO-MONTALVO received the engineering degree in electronics and telecommunications from Universidad Técnica del Norte (UTN), Ecuador, in 2013, the master's degree in data management systems from Universidad de las Fuerzas Armadas ESPE, Ecuador, in 2018, and the Ph.D. degree from the University of Salamanca, Spain, in November 2020, under the supervision of Prof. Vivian Lopez-Batista. He was a Research Assistant Professor with the Applied Science Department, UTN, for seven years. At the same time, he was a part-time Lecturer with the TI Department, Instituto Superior Tecnológico 17 de Julio, Ecuador. He has been a Postdoctoral Researcher with the IT University of Copenhagen (ITU), Denmark, since June 2021. His research interest includes emerging microcontrollers to run machine learning models in decentralized networks.

ZSOLT ISTVÁN received the Ph.D. degree from the Systems Group, ETH Zürich, Switzerland. He was with the IT University of Copenhagen, Denmark, and the IMDEA Software Institute, Madrid, Spain. He is currently a Full Professor with the Computer Science Department, Technical University of Darmstadt, Germany, where he leads the Distributed and Networked Systems Group.

WILMAR HERNANDEZ (Senior Member, IEEE) received the degree in electronics engineering and the specialist degree in microelectronics from Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), Havana, Cuba, in 1992 and 1994, respectively, and the M.S. degree in signal treatment and the Ph.D. degree in electronic engineering from Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Barcelona, Spain, in 1997 and 1999, respectively. From 1992 to 1995, he was a Lecturer with the Electrical Engineering Faculty, ISPJAE, and a Researcher with the Microelectronics Research Center, ISPJAE. From 1999 to 2003, he was with the Department of Electronics and Instrumentation, University Institute for Automobile Research, Universidad Politecnica de Madrid (UPM), Spain, where he was the Technical Director, from January 2003 to January 2004. From January 2004 to March 2013, he was an Associate Professor of circuits and systems with the Department of Circuits and Systems, EUIT de Telecomunicación, UPM. From September 2014 to September 2015, he was a Researcher with SENESCYT, Ecuador, under the Prometeo Fellowship Program. From December 2015 to November 2017, he was a Professor with Universidad Técnica Particular de Loja, Ecuador. Since January 2018, he has been a Professor with Universidad de Las Américas, Ecuador.

• • •